Design and Analysis of Random Access Protocols for Massive Machine-Type Communications

A Thesis

Submitted for the Degree of

Doctor of Philosophy

in the **Faculty of Engineering**

by

Chirag Ramesh

under the Guidance of

Prof. Chandra R. Murthy

Robert Bosch Centre for Cyber-Physical Systems
Indian Institute of Science
Bangalore – 560 012, INDIA

October 2023

TO

My Family

Acknowledgments

I express my sincere gratitude to all the people who have helped me on the long journey to finish my Ph.D. thesis. This includes several family members who have always been supportive of my dreams, countless teachers who have helped me forge my own path, and many friends with whom I have had some amazing life experiences. Without their guidance, I would not be where I am right now.

First of all, I would like to thank my mother Janaki, my father Ramesh, and my sister Chethana. I have been blessed with a very loving family. I thank them for motivating me to achieve higher goals in life. I thank my mother for all her efforts in preparing me for school, and then later for college, and finally for IISc. I thank my father for working hard so that I can achieve my dreams. I thank my sister who has been my best friend and offering me entertainment forever. I thank them for accepting me working every weeknight and every weekend during my Ph.D. I thank them for their patience and comfort during the covid pandemic. Most of all, I thank them for having unconditionally supported me through every avenue I have taken. I cannot list all the things they have done for me, else that could be a thesis-length document on its own! I dedicate this thesis to my family.

I am grateful to my advisor Prof. Chandra R. Murthy for his patience, support, and guidance during my Ph.D. He has been highly instrumental in shaping me as a researcher, and also honing my paper writing skills and my presentation skills. His pursuit of excellence has inspired me to be a better researcher and person. I am thankful for the many opportunities I have been presented being a part of his lab. In the weekly meetings with him, even a few words from him would help me sustain my research for weeks. In the weekly group meetings, I have learnt how to improve my presentation skills and paper

reading skills, as well how to ask questions and what kind of questions to ask. I would always learn something new in every meeting I had with him, if not, then a new thought that could potentially be a research idea. I am grateful for his impeccable inputs for the monthly meetings during the period when we won the Qualcomm Innovation Fellowship. I thank him for all the online meetings and discussions at odd hours, and paper reviews (even across continents). In short, my experience with him has been invaluable, and it has been a real honour to have worked with him.

I am indebted to all the innumerable teachers and mentors I have had throughout my life. I thank Dr. Abhay Sharma for his support during my initial Ph.D. years. A special mention to my comprehensive exam committee members, Prof. Neelesh Mehta, Prof. Rajesh Sundaresan, Prof. Chandramani Singh, and Prof. Bharadwaj Amrutur, for their kind support throughout my Ph.D. As a student, I have had the privilege of enjoying several courses at IISc, taught by a few of the best teachers in the world. I thank Profs. Aditya Gopalan, Chandra Murthy, Chiranjib Bhattacharya, KVS Hari, Neelesh Mehta, Parimal Parag, PS Sastry, P Vijay Kumar, Raghu Krishnapuram, and Utpal Mukherji for their excellent teaching. From my undergraduate school, I thank Profs. Sanjeev G, Vamsi KT, and AN Nagamani.

I want to express my gratitude to my closest friends Vidyadhar, Sriharsha, and Sachin whose love and support has helped me for around twelve years now. I thank my numerous cousins, aunts, uncles, grandparents for their continuous support, and my nephews and nieces for keeping me entertained. A few family members who deserve a special mention are Deepa, Nidhi, Nitya, Harsha, Pavan, Sameera, Arjun, and Vinay. They have provided a lot of comfort and help, many from even before my school days, and many during my Ph.D. as well. I thank my undergraduate friends Nikhil A, Manish, Nihal, Desik, Varshini, Amrutha, and Nikhil R for their constant support throughout my formative years.

As an interdisciplinary student, I have the honour of calling two departments my home, the Robert Bosch Centre for Cyber-Physical Systems (RBCCPS) and the Electrical Communication Engineering Department (ECE). My first home at IISc is my department, RBCCPS. I feel incredibly grateful to my RBCCPS batchmates without whom I could not

have completed my Ph.D. These are the friends who have made my Ph.D. journey smooth sailing and very entertaining. Be it Siddarth with his horrible jokes or Tarun (and his several identical clones) as an adjacent hostelite or Rooji with her disgusted expressions towards Siddarth's puns or Akshatha with her wide choices of places to eat at or Chaitanya always losing the rummy game-nights or Srikrishna with his insider information or Vaishali with her dazed looks at Tarun's dad jokes or Shayal with her insane energy, I have never been short of happiness when I am with them. We have been to so many pizza meetings, movies, and restaurants (and once to a cricket stadium as well!), and the ensuing cab rides have been very fun. A special mention to Siddarth for the lunch retreats to Nesara, Prakruti, and Sarvam, and also for being my weekly source of entertainment. We have been tagged together for so many volunteering events, symposiums, conferences, lunches, presentations, and also the Qualcomm trip to Delhi! He has been a key staple of my Ph.D. and has made my experience unforgettable.

My second home at IISc is the Signal Processing for Communications (SPC) Lab at the ECE department. I am grateful to my close friends from the SPC lab who have made my Ph.D. a very enjoyable journey for me (with weekly Prakruti/Sarvam visits, daily bakery visits, and more): Anubhab, Dheeraj, Pradip, Sai, Sameera, Soumendu, and Yashvanth. I thank several of my other SPC labmates for their lab interactions, weekly meeting interactions, and lab lunches which have helped me grow in a nurturing research environment: Arunkumar, Chandrasekhar, Geethu, Lekshmi, Nandan, Niladri, Rupam, and Rubin. I sat in a joint space comprising of the SPC Lab and the Next Gen Lab, for which I thank Prof. Neelesh Mehta and my Next Gen Lab colleagues Bala, Ram, Samaresh, Sayan, Sayantan, Sarvendranath, and Sruthy. A special mention to Anubhab for being available $24 \times 7 \times 365$! Be it for research discussions, biryani, lunches, paper revisions, academic section issues, getting a form signed, philosophy discussions, being an adjacent hostelite or just simply rebooting my system, I cannot thank him enough for his help.

My special thanks go to a few folks who have been close friends to me throughout my Ph.D., who were also TAs for courses at IISc. I thank Karthik PN and Rakshit Ramesh for

being outstanding TAs for the Random Processes and Autonomous Navigation courses, respectively. I thank Nagabhushan, Sai, and Anubhab for being excellent co-TAs for the Matrix Theory course. Without them, I could not have been an effective TA myself.

I thank the workers in A-mess for the tasty food they used to serve during my stay at IISc. I thank Tarun, Rooji, Vaishali, Nagabhushan, Dheeraj, Sai, Anubhab, and Yashvanth for the fun mess interactions. I thank all the doctors, frontline workers, and healthcare workers for their work during the covid pandemic and their outstanding mental strength. I appreciate the cleaning staff at NBH, A-mess, RBCCPS, and ECE who have always kept our environment clean for us. I appreciate the financial support from RBCCPS during my initial years. I thank Qualcomm for providing me research funding via the Qualcomm Innovation Fellowship, 2019-2020. I appreciate the financial support from MHRD, Govt. of India, for pursuing my Ph.D. and conference travels to Finland in 2022. I appreciate the financial support from IEEE ComSoc (via the Three-Minute-Thesis competition), for the conference travels to Italy in 2023. I am indebted to IISc for offering me this wonderful opportunity to pursue my Ph.D.

Lastly I would like to thank the department staff who always used to help me, especially to navigate the bureaucratic maze. At RBCCPS, the office staff have never told us no and have helped us for the miniscule things as well: Sreedevi, Lavanya, Rajeshwari, Shruthi, and Subramani. At ECE, Rajani and Srinivasamurthy have been very helpful.

Chirag Ramesh

Statement of Originality

I hereby declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of higher education.

I certify that to the best of my knowledge, the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Chirag Ramesh

Abstract

Massive machine-type communications (mMTC) is a 5G and beyond use case, where the network is expected to serve millions of devices per square kilometre. Typical mMTC devices include smart energy meters, pressure sensors, temperature indicators, smart factory equipment, etc. These devices sporadically transmit short packets, i.e., they transmit a short burst of data once in a while and then largely remain inactive. In order to serve mMTC scenarios, we need to use grant-free random access (GFRA) protocols since they have the advantage of a low control and signalling overhead as well as non-orthogonal use of the channel. GFRA for mMTC is a relatively new research topic and has received immense interest in the recent past. In this thesis, we analyze several practical aspects of irregular repetition slotted aloha (IRSA), which is a GFRA protocol for mMTC.

IRSA is a distributed GFRA protocol where users transmit multiple replicas of their packets in randomly selected resource blocks within a frame to a base station (BS). The BS recovers the packets using successive interference cancellation (SIC). Existing studies have analyzed IRSA with idealized assumptions, i.e., neglecting fading, path-loss, channel estimation errors, pilot contamination, multi-cell interference, etc. These non-idealities can greatly reduce the performance of the system and must be accounted for in the design and analysis of any mMTC system.

In this thesis, we first analyze channel estimation in IRSA, exploiting the sparsity structure of IRSA transmissions, when non-orthogonal pilots are employed across users to facilitate channel estimation at the BS. Allowing for the use of non-orthogonal pilots is important, as the length of orthogonal pilots scales linearly with the total number of devices, leading to prohibitive overhead as the number of devices increases. Next, we present

a novel analysis of the throughput of IRSA under practical channel estimation errors, and with the use of multiple antennas at the BS. Finally, we theoretically characterize the asymptotic throughput of IRSA using a density evolution based analysis. Simulation results underline the importance of accounting for channel estimation errors in analyzing IRSA, which can even lead to 70% loss in performance in severely interference-limited regimes. We also provide novel insights on the effect of parameters such as pilot length, SNR, number of antennas at the BS, etc, on the system throughput.

Next, we develop a novel Bayesian user activity detection (UAD) algorithm for IRSA, which exploits both the sparsity in user activity as well as the underlying structure of IRSA transmissions. We then derive the Cramér-Rao bound (CRB) on the mean squared error in channel estimation. We empirically show that the channel estimates obtained as a by-product of the proposed UAD algorithm achieves the CRB. Then, we analyze the signal to interference plus noise ratio achieved by the users, accounting for UAD, channel estimation errors, and pilot contamination. Finally, we illustrate the impact of these non-idealities on the throughput of IRSA via Monte Carlo simulations. For example, in a system with 1500 users and 10% of the users being active per frame, a pilot length of as low as 20 symbols is sufficient for accurate user activity detection. In contrast, using classical compressed sensing approaches for UAD would require a pilot length of about 346 symbols. Our results reveal crucial insights into dependence of UAD errors and throughput on parameters such as the length of the pilot sequence, the number of antennas at the BS, the number of users, and the SNR.

Then, we develop an enhanced version of IRSA that can be operated at the peak performance even at high system loads. IRSA can be used to serve a large number of users in mMTC while achieving a near-zero packet loss rate (PLR). However, in overloaded mMTC scenarios, the system is interference-limited, and the PLR is close to one. We develop a variant of IRSA in the interference limited-regime, namely Censored-IRSA (C-IRSA), in which users with poor channel states self-censor, i.e., they refrain from transmitting their packets. This censoring depends on a censor threshold that can be varied depending on the number of users in the system. Firstly, we empirically and

theoretically analyze the performance of C-IRSA. Next, we derive the optimal choice of the censor threshold via a semi-analytic approach and a PLR-optimal algorithmic approach. This choice of the threshold maximizes the throughput while achieving zero PLR among uncensored users. Through extensive numerical simulations, we show that C-IRSA operates at full system throughput at high system loads compared to vanilla IRSA which has near-zero throughput.

After this, we analyze IRSA in the multi-cell (MC) and cell-free (CF) setups, accounting for pilot contamination, channel estimation errors, and multi-user interference. Via extensive simulations, we illustrate that, in practical settings, MC IRSA can have a drastic loss of throughput, up to 70%, compared to SC IRSA. Further, MC IRSA requires a significantly higher training length, in order to support the same user density and achieve the same throughput: for example, MC IRSA may need about $4-5\times$ compared to SC IRSA. We provide insights into the effect of system parameters such as number of antennas, pilot length, and SNR on the throughput of MC IRSA and CF IRSA. With the proposed CF architectures, we show that we can achieve more than $14\times$ improvement in the throughput of CF IRSA compared to a massive MIMO SC setup. We also study the densification trends in MC IRSA, where we observe an inverse behaviour in the throughput compared to CF IRSA.

Finally, we optimize the repetition distributions in IRSA with the throughput and the energy efficiency objectives. Via extensive numerical simulations, we study the effect of various system parameters such as the maximum repetition factor, the average repetition factor, the number of antennas, and the pilot length, on the repetition distributions, the inflection load, and the peak energy efficiency. Compared to the best existing distributions, we show that our optimized distributions can achieve up to 58% increase in the inflection load and up to 49% increase in the peak energy efficiency.

Overall, this thesis analyzes and designs the IRSA protocol under several practical non-idealities. The developed algorithms vastly outperform state-of-the-art and can efficiently serve mMTC applications.

Contents

List of Figures

List of Tables

Glossary

3GPP	Third Generation Partnership Project
5G	Fifth Generation
6G	Sixth Generation
AP	Access Point
APM	Access Pattern Matrix
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
BS	Base Station
CCDF	Complementary Cumulative Distribution Function
CCF	Centralized Cell Free
CF	Cell Free
C-IRSA	Censored Irregular Repetition Slotted Aloha
CRA	Coded Random Access
CRC	Cyclic Redundancy Check
CRDSA	Contention Resolution Diversity Slotted Aloha
CRLB	Cramér-Rao Lower Bound
CS	Compressed Sensing
CSA	Coded Slotted Aloha
CSI	Channel State Information
DAA	Distributed Antenna Array

DE	Density Evolution
DEA	Differential Evolution Algorithm
DRRSA	D-Regular Repetition Slotted Aloha
EE	Energy Efficiency
EM	Expectation Maximization
eMBB	Enhanced Mobile Broadband
FIM	Fisher Information Matrix
FNR	False Negative Rate
FPR	False Positive Rate
GA	Genetic Algorithm
GBRA	Grant-Based Random Access
GFRA	Grant-Free Random Access
HCF	Hybrid Cell Free
ICI	Inter-Cell Interference
i.i.d.	Independent and Identically Distributed
InCI	Intra-Cell Interference
InPC	Intra-cell Pilot Contamination
IoT	Internet-of-Things
IPC	Inter-cell Pilot Contamination
IRSA	Irregular Repetition Slotted Aloha
ITU	International Telecommunication Union
LCF	Local Cell Free
LCMMSE	Low Complexity Minimum Mean Squared Error
LDPC	Low Density Parity Check
LMMSE	Linear Minimum Mean Squared Error
LS	Least-Squares
LTE	Long-Term Evolution

M2M	Machine to Machine
MAC	Medium Access Control
MAP	Maximum A-Posteriori
MC	Multi Cell
MDG	Macro-Diversity Gain
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
mMIMO	Massive Multiple Input Multiple Output
mMTC	Massive Machine Type-Communications
MMV	Multiple Measurement Vector
MRA	Massive Random Access
MRC	Maximal Ratio Combining
MSBL	Multiple Sparse Bayesian Learning
MSE	Mean Squared Error
MUI	Multi-User Interference
NB-IoT	Narrow-Band Internet-of-Things
NMSE	Normalized Mean Squared Error
NNLS	Non Negative Least Squares
NOMA	Non-Orthogonal Multiple Access
OMA	Orthogonal Multiple Access
PC	Pilot Contamination
PCSI	Perfect Channel State Information
PDMA	Pattern Division Multiple Access
PLI	Path Loss Inversion
PLR	Packet Loss Rate
QAM	Quadrature Amplitude Modulation

QPSK	Quadrature Phase Shift Keying
QoS	Quality of Service
RA	Random Access
RB	Resource Block
RE	Resource Element
RZF	Regularized Zero Forcing
SBL	Sparse Bayesian Learning
SC	Single Cell
SE	Spectral Efficiency
SIC	Successive Interference Cancellation
SINR	Signal-to-Interference-plus-Noise Ratio
SISO	Single Input Single Output
SMV	Single Measurement Vector
SNR	Signal-to-Noise Ratio
UAD	User Activity Detection
UL	Up Link
URA	Unsourced Random Access
URLLC	Ultra Reliable Low Latency Communications
ZF	Zero Forcing

Notation

Sets

$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
$\mathbb{R}^+$	The set of positive real numbers
$\mathbb{R}^{N\times M}$	The set of real matrices of size $N \times M$
$\mathbb{C}^{N\times M}$	The set of complex matrices of size $N \times M$
$[M]$	The set $\{1, \ldots, M\}$
$\lvert\mathcal{S}\rvert$	Cardinality of the set $\mathcal{S}$

Random Variables

i.i.d.	Independent and identically distributed
$\Pr(\mathcal{E})$	Probability of an event $\mathcal{E}$
$\Pr(\mathcal{E}\lvert\mathcal{A})$	Probability of an event $\mathcal{E}$ conditioned on an event $\mathcal{A}$
$\mathbb{E}[X]$	Expectation of a random variable X
$\mathbb{E}_{\mathcal{A}}[X]$	Expectation of a random variable X conditioned on an event $\mathcal{A}$
$\mathcal{N}(\mathbf{a}, \mathbf{A})$	Complex Gaussian distribution with mean $\mathbf{a}$ and covariance $\mathbf{A}$
$\mathcal{CN}(\mathbf{a}, \mathbf{A})$	Real Gaussian distribution with mean $\mathbf{a}$ and covariance $\mathbf{A}$

Miscellaneous

$\mathcal{Q}(a)$	Gaussian-Q function evaluated at a
$\Gamma(s)$	Ordinary gamma function with $\Gamma(s) = \int_0^\infty t^{s-1}\, e^{-t}\, \mathrm{d}t$
$\Gamma_{\text{inc}}(s, x)$	Upper incomplete gamma function with $\Gamma_{\text{inc}}(s, x) = \int_x^\infty t^{s-1}\, e^{-t}\, \mathrm{d}t$
$\gamma_{\text{inc}}(s, x)$	Lower incomplete gamma function with $\gamma_{\text{inc}}(s, x) = \int_0^x t^{s-1}\, e^{-t}\, \mathrm{d}t$

Vectors and matrices

a	A scalar
$\mathbf{a}$	A vector
$\mathbf{A}$	A matrix
$\lvert a\rvert$	Magnitude of the scalar a
$[\mathbf{A}]_{i,:}$	The ith row of a matrix $\mathbf{A}$
$[\mathbf{A}]_{:,j}$	The jth column of a matrix $\mathbf{A}$
$[\mathbf{a}]_{\mathcal{S}}$	The elements of $\mathbf{a}$ indexed by the set $\mathcal{S}$
$[\mathbf{A}]_{:,\mathcal{S}}$	The columns of $\mathbf{A}$ indexed by the set $\mathcal{S}$
$\mathbf{0}_N$	All zero vector of length N
$\mathbf{1}_N$	All ones vector of length N
$\mathbf{I}_N$	Identity matrix of size $N \times N$
$\mathbf{A}^T$	Transpose of a matrix $\mathbf{A}$
$\bar{\mathbf{A}}$	Complex conjugate of a matrix $\mathbf{A}$
$\mathbf{A}^H$	Conjugate transpose (or Hermitian) of a matrix $\mathbf{A}$
$\lvert\mathbf{A}\rvert$	Determinant of a matrix $\mathbf{A}$
$\mathbf{A} \succeq \mathbf{B}$	$\mathbf{A} - \mathbf{B}$ is positive semi-definite
$\lVert\mathbf{x}\rVert_2$	ℓ_2 norm of a vector $\mathbf{x} \in \mathbb{C}^N$
$\lVert\mathbf{A}\rVert_F$	Frobenius norm of a matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$
$\mathrm{Tr}(\mathbf{A})$	Trace of a matrix $\mathbf{A}$
$\mathbf{A}^\dagger$	Moore-Penrose pseudoinverse of a matrix $\mathbf{A}$
$\mathrm{vec}(\mathbf{A})$	Vectorized version of a matrix $\mathbf{A}$
$\mathrm{diag}(\mathbf{a})$	A diagonal matrix with the entries of vector $\mathbf{a}$ on the diagonal
$\mathrm{diag}(\mathbf{A})$	A vector with its entries as the diagonal elements of a matrix $\mathbf{A}$
$\mathrm{blkdiag}(\mathbf{A}, \mathbf{B})$	Block diagonal matrix with $\mathbf{A}$ and $\mathbf{B}$ as the diagonal blocks
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$

Chapter 1

Introduction

Massive machine-type communications (mMTC) is an evolving next generation use-case, expected to serve around 10^6 devices per square kilometer [1]. The users in mMTC applications are sporadically active and transmit short packets to a central base station (BS) [2]. Typical mMTC devices include smart energy meters, pressure sensors, temperature indicators, smart factory equipment, etc. mMTC applications require random access protocols that serve large numbers of users [3, 4]. Distributed grant-free random access (GFRA) protocols are appropriate for mMTC, since they incur a low control and signaling overhead [5], as well as non-orthogonal use of the channel [6]. GFRA for mMTC is a relatively new research topic and has received immense interest in the recent past.

Irregular repetition slotted aloha (IRSA) is a high performing distributed GFRA protocol suitable for mMTC applications. In IRSA, users transmit multiple replicas of their packets to a base station (BS) in randomly selected resource blocks (RBs) [7]. The access of the RBs is represented by the access pattern matrix (APM). The BS recovers the packets using successive interference cancellation (SIC). Existing studies have analyzed IRSA with idealized assumptions, i.e., neglecting fading, path-loss, channel estimation errors, pilot contamination, multi-cell interference, etc. These non-idealities can greatly reduce the performance of the system and must be accounted in the design and analysis of any mMTC system [8]. In this thesis, we analyze IRSA under several of these practical aspects.

1.1 Outline of this Thesis

In this section, we briefly describe the problems addressed in this thesis. The organization of the thesis is represented as a flow diagram in Fig. 1.1. The main problem addressed in each chapter is highlighted in solid rectangles, whereas the main analytical technique used in each chapter is represented in dotted rectangles.

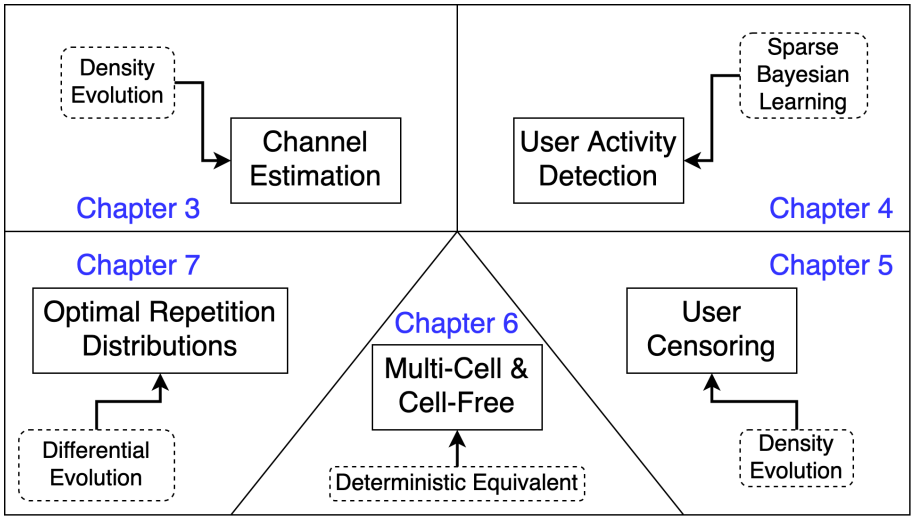


Figure 1.1: Organization of the Thesis.

1.1.1 Motivation: Irregular Repetition Slotted Aloha for Massive Machine-Type Communications

In the second chapter, we motivate the central problem of this thesis. We first present an overview of challenges and standards for mMTC. We then present the working of the IRSA protocol, and give an overview of the existing works related to IRSA and mMTC. We conclude this chapter by presenting density evolution, which is an analysis technique used to find the theoretical performance of IRSA.

1.1.2 Channel Estimation and Data Decoding in IRSA

In the third chapter, we consider the problem of channel estimation in IRSA. We derive channel estimates in IRSA, exploiting the sparsity structure of IRSA transmissions,

when non-orthogonal pilots are employed across users to facilitate channel estimation at the BS. Assigning mutually orthogonal pilots to users avoids pilot contamination, but is prohibitively expensive in mMTC, since the pilot overhead would be proportional to the total number of users [9]. Thus, *pilot contamination (PC)*, which reduces the accuracy of channel estimation and makes the estimates correlated [10], is unavoidable in mMTC, and significantly degrades the throughput of IRSA. Thus, pilot contamination has to be accounted for while analyzing the performance of GFRA protocols for mMTC.

The contents of this chapter are published in [11,12]. We list our contributions in this chapter below:

1. We derive channel estimates for IRSA under three schemes: the first one exploits the sparsity in the APM to estimate the channels of the users, and the other two assume knowledge of the APM and output minimum mean square error (MMSE) estimates.

2. We present a novel analysis of the signal-to-interference-plus-noise-ratio (SINR) in IRSA accounting for channel estimation errors, where estimates are acquired via non-orthogonal pilots under the three estimation schemes. We account for multiple antennas at the BS, fading, path loss, and pilot contamination.

3. We theoretically analyze the throughput of IRSA via density evolution (DE), when users perform path loss inversion based power control. The analysis reveals the asymptotic performance of the protocols as the number of users and RBs get large.

Through extensive simulations, we show that channel estimation errors lead to a significant loss of throughput compared to the ideal scenario with perfect channel state information (CSI) at the BS, even resulting in up to 70% loss in severely interference-limited regimes. Our analysis also reveals an inflection load, beyond which the system becomes interference-limited, resulting in a dramatic reduction of the throughput.

1.1.3 User Activity Detection in IRSA

In the fourth chapter, we consider the problem of user activity detection in IRSA. Since only a subset of users are active in any frame in mMTC [2], it is essential for the BS to detect the set of users that are active, before proceeding to perform channel estimation and data decoding. This process is termed *user activity detection (UAD)*. Knowing the subset of active users not only saves computational resources by helping the BS decide which users it needs to decode, it is also important for channel estimation [13, 14]. We develop a novel Bayesian UAD algorithm to detect the subset of active users in IRSA, which exploits both the sparsity in user activity as well as the underlying structure of IRSA transmissions. Errors arising from the UAD process, namely, false positives and false negatives, deteriorate the channel estimates computed at the BS, which in turn affects the data decoding. Hence, it is crucial to account for these errors while analyzing the performance of GFRA protocols.

The contents of this chapter are published in [15]. Our main contributions are:

1. We develop a novel Bayesian algorithm to detect the set of active users in IRSA. UAD in IRSA is a joint-sparse signal recovery problem with an important twist: different and unknown subsets of the row indices of the joint-sparse matrix participate in different measurements. Our algorithm is an enhancement to the multiple sparse Bayesian learning (MSBL) algorithm [16] to cater to this scenario.

2. We derive the channel estimates at the BS for users in all RBs in IRSA, acquired via non-orthogonal pilots. We also derive the Cramér-Rao bound (CRB) on the mean squared error (MSE) of the channels estimated by our proposed UAD algorithm. We show that a genie-aided MMSE estimator (that has knowledge of the second-order channel statistics and the user activities) achieves the CRB. We also empirically show that the MSE of the channel estimates output by the proposed UAD algorithm meets the CRB.

3. We analyze the SINR achieved by all the users in all RBs, accounting for UAD errors, channel estimation errors, and pilot contamination. The SINR expression allows us

to determine the throughput of IRSA, accounting for all these non-idealities.

Our numerical experiments show that there is at least a 4-fold reduction in the number of pilot symbols required to achieve a similar UAD performance as that of existing approaches. Our results reveal crucial insights into dependence of UAD errors and throughput on parameters such as the length of the pilot sequence, the number of antennas at the BS, the number of users, and the signal to noise ratio. The loss in performance due to UAD errors can be recuperated by judiciously choosing the system parameters such as pilot length, number of antennas, and SNR. For example, in a system with 1500 users and 10% of the users being active per frame, a pilot length of as low as 20 symbols is sufficient for accurate user activity detection. In contrast, using classical compressed sensing approaches for UAD would require a pilot length of about 346 symbols.

1.1.4 Censored-IRSA for Interference-Limited mMTC

In the fifth chapter, we develop an enhanced version of IRSA in the interference limited regime, namely Censored-IRSA (C-IRSA), wherein users with poor channel self-censor, i.e., they refrain from transmitting their packets. Typically, IRSA can be used to serve a large number of users while achieving a packet loss rate (PLR) close to zero [7]. However, in overloaded mMTC applications, the number of users is too high, then the system is interference limited and the PLR is close to one [17]. The censoring in C-IRSA depends on a censor threshold that can be varied depending on the number of users in the system. C-IRSA maintains the distributed nature of IRSA.

The contents of this chapter are published in part in a conference paper in [18], and a journal paper is under preparation [19]. The contributions of this chapter are as follows:

1. We propose C-IRSA to tackle the interference-limitation of IRSA at high system loads. This involves self-censoring of users, wherein users with poor CSI refrain themselves from transmitting, which decreases the effective system load and ensures that all the uncensored users are successfully decoded.

2. We empirically analyze the performance of C-IRSA accounting for path loss, channel

estimation errors, MIMO, fading, and pilot contamination.

3. Using DE, we theoretically analyze the performance of C-IRSA when the users perform power control based on the path loss between themselves and the BS.

4. We derive the optimal censor threshold in the general case: accounting for path loss, MIMO, fading, channel estimation, and pilot contamination. We present two approaches: the first is a semi-analytic approach, whereas the second is an algorithmic approach that is packet loss rate (PLR) optimal. Using these choices of the censor threshold, the PLR of uncensored users can be driven close to zero at all system loads, while maintaining the throughput of the system at its highest value.

Using extensive numerical simulations, we show that, C-IRSA operates at the full throughput at all loads, in contrast to vanilla IRSA which has near-zero throughput as the load is increased. In particular, at high loads, C-IRSA offers a $10\times$ throughput improvement over IRSA without user censoring.

1.1.5 Analysis of IRSA in Multi-Cell and Cell-Free Systems

In the sixth chapter, we analyze the performance of IRSA in the multi-cell (MC) and cell-free (CF) setups, accounting for pilot contamination and multi-user interference. Existing studies have analyzed IRSA in the single-cell (SC) setup, which does not extend to the more practically relevant multi-cell (MC) setup due to the inter-cell interference. Further, SC processing neglecting inter-cell interference is highly suboptimal when applied to MC systems. Cell-free (CF) architectures have been proposed for expanding the coverage of communication systems [20]. In a conventional CF system, instead of a conventional BS at the center of a cell, several access points (APs) are used to jointly serve the users [21]. The APs are connected to a central processing unit (CPU) which is responsible for data aggregation and network coordination [22]. mMTC has the goal of increased connectivity and packet success rates. This is especially challenging to achieve when there are several cell-edge users who may not be decoded in mMTC due to high path losses. Further, these devices are expected to consume low power and have long battery lives, because of which

they cannot transmit at high powers to compensate for the path loss. CF architectures naturally overcome this issue due to the *macro-diversity gain* (MDG), which helps decode these edge users [23]. Thus, mMTC scenarios are a natural application for using CF architectures, and studying IRSA for mMTC in a CF setup is very relevant.

The contents of this chapter are published in [24] and a journal paper is under preparation [25]. We list our contributions in this chapter below:

1. We study IRSA in the MC setup. We derive the channel estimates and the SINR in MC IRSA accounting for path loss, MIMO fading, intra-cell pilot contamination (PC), and inter-cell PC.

2. We analyze IRSA in the CF setup, with all of the above non-idealities. Specifically, we study three CF architectures for IRSA: one with local processing at each AP, termed local-cell-free (LCF); next with fully centralized processing at the CPU, termed centralized-cell-free (CCF); and finally, with hybrid processing at both the APs and the CPU, termed hybrid-cell-free (HCF).

3. We provide insights into the effect of system parameters such as number of antennas, number of APs (or BSs), pilot length, and SNR on the throughput of MC IRSA and CF IRSA.

4. We study the effect of BS and AP densification in MC and LCF IRSA, respectively, where we observe an inverse behaviour in the throughput compared to CCF IRSA.

To the best of our knowledge, no existing work has analyzed the effect of MC interference or CF processing in IRSA. Through numerical simulations, we show that inter-cell PC and inter-cell interference result in up to 70% loss in the MC throughput compared to the SC setup. Further, MC IRSA requires a significantly higher training length (about $4-5\times$ compared to SC IRSA), in order to support the same user density and achieve the same throughput. Under the CF architecture, we can achieve more than $14\times$ improvement in the throughput of CCF IRSA compared to a massive MIMO SC setup at high loads. We also study the densification trends in MC IRSA and CF IRSA: for CCF IRSA and

HCF IRSA, densification always improves the performance; for LCF IRSA and MC IRSA, densification does not help at loads near the inflection loads, i.e., it is better not to densify and to operate with a massive MIMO SC setup.

1.1.6 Optimal Repetition Distributions in IRSA

In the seventh chapter, we optimize the repetition distributions in IRSA using the differential evolution algorithm (DEA) [26], under the throughput and energy efficiency objectives. Energy efficiency is a fundamental aspect of both cellular [27] and machine-type communications [28]. Energy efficiency becomes even more important in mMTC since the low-power devices deployed in mMTC scenarios are expected to be IoT devices which last for several years [29]. These devices are expected to consume as low power as possible while maintaining high throughput [30]. Thus, energy efficiency is a fundamental metric of mMTC systems, and in particular, in the IRSA protocol as well. The obtained optimal distributions can be used to operate mMTC at the peak throughputs as well as the peak energy efficiencies.

Our contributions in this chapter are as follows:

1. Firstly, we optimize the repetition distributions of IRSA with the throughput and the energy efficiency objectives.

2. Next, we study the effect of the optimal repetition distributions under three cases: first case with the K-collision channel; second case with perfect CSI, MIMO, and MRC; and third case with channel estimation errors, MIMO, pilot contamination, and MMSE combining.

3. Via extensive numerical simulations, we study the effect of various system parameters such as the maximum repetition factor, the average repetition factor, the number of antennas, the pilot length on the repetition distributions, the inflection load, and the peak energy efficiency.

We demonstrate that the 2-regular distribution is the most energy efficient distribution for IRSA at high number of antennas and high pilot lengths. The 2-regular distribution

is also nearly throughput-optimal at high number of antennas. Compared to the best existing distributions, we show that our optimized distributions can achieve up to 58% increase in the inflection load and up to 49% increase in the peak energy efficiency.

1.1.7 Conclusion

In the last chapter, we conclude this thesis. We summarize the designs and analyses presented in this thesis. Overall, this thesis analyzes the IRSA protocol under several practical non-idealities. The developed algorithms vastly outperform state-of-the-art and can efficiently serve mMTC applications. We outline some future research directions for IRSA and mMTC.

1.2 Publications

1.2.1 Journal Papers

1. **C. R. Srivatsa** and C. R. Murthy, "User Activity Detection for Irregular Repetition Slotted Aloha based MMTC," in *IEEE Transactions on Signal Processing*, vol. 70, pp. 3616-3631, 2022.

2. **C. R. Srivatsa** and C. R. Murthy, "On the Impact of Channel Estimation on the Design and Analysis of IRSA based Systems," in *IEEE Transactions on Signal Processing*, vol. 70, pp. 4186-4200, 2022.

3. **C. R. Srivatsa** and C. R. Murthy, "Censored-IRSA for Massive Random Access in Interference-Limited mMTC Scenarios," Manuscript under preparation.

4. **C. R. Srivatsa** and C. R. Murthy, "IRSA for Massive Random Access in Multi-Cell and Cell-Free Massive MIMO Systems," Manuscript under preparation.

1.2.2 Conference Papers

1. **C. R. Srivatsa** and C. R. Murthy, "Throughput Analysis of PDMA/IRSA under Practical Channel Estimation," *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Cannes, France, 2019, pp. 1-5.

2. **C. R. Srivatsa** and C. R. Murthy, "Performance Analysis of Irregular Repetition Slotted Aloha with Multi-Cell Interference," *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, Oulu, Finland, 2022, pp. 1-5.

3. **C. R. Srivatsa** and C. R. Murthy, "Channel State Information Based User Censoring in Irregular Repetition Slotted Aloha," *2023 IEEE International Conference on Communications (ICC)*, Rome, Italy, 2023, pp. 1-6.

Chapter 2

Irregular Repetition Slotted Aloha

In this chapter, we first motivate the central problem of this thesis, i.e., massive machine type-communications (mMTC). Firstly, we explain the challenges faced in mMTC, and the protocols used in mMTC applications. Secondly, we discuss the working of the irregular repetition slotted aloha (IRSA) protocol, including the structure of the access pattern matrix, the successive interference cancellation process, and the decoding process in IRSA. Then, we survey existing works on IRSA, and also other papers that are relevant to this thesis. Finally, we describe the process of density evolution, which is an iterative recipe that can be used to compute the theoretical performance of IRSA.

2.1 Massive Machine-Type Communications

The internet-of-things (IoT) is a network of physical devices that communicate and exchange information with each other over the internet [31]. Each IoT device contains the embedded systems, processors, software, and hardware that can enable the automatic working of the related IoT ecosystem, where all the devices are internet-enabled [32]. The goal could be to have an understanding of the environment using the vast amounts of data produced by these sensors, perhaps, via a gateway to which all the IoT devices send data to [33]. Typical IoT applications include smart factories, smart cities, environmental sensing applications, and healthcare [34]. One major application is in a smart factory

setting, where there are thousands of such static IoT devices (e.g., temperature sensors, smart energy meters, pressure indicators) which collectively operate over the internet to ensure the automated smooth running of the smart factory [35]. It is expected that the scale of such devices will grow in the near future to perhaps millions of devices per square kilometre. According to the expected requirements [36], in 2025, the total number of connected devices in the world will be in billions. Communication amongst the IoT devices is a challenge, especially when the number of devices is huge [37]. This application is known as massive machine-type communications (mMTC), and this application is the main focus of this thesis.

The three pillars of next-generation communication systems, especially 6G and beyond-5G communication systems, are expected to be mMTC, ultra-reliable low latency communications (URLLC), and enhanced mobile broadband (eMBB), according to the third generation partnership project (3GPP) and the international telecommunication union (ITU) [38]. mMTC is a use case expected to serve millions of IoT-type devices per square kilometre. mMTC devices transmit short packets sporadically, i.e., they transmit data once in a while and then largely remain inactive [37]. URLLC applications require protocols that serve users with very low latencies (up to 1ms) and very high reliability (i.e., with a low loss rate up to 10^{-5}) [39]. eMBB aims at serving as many cellular users as possible with higher data rates compared to existing cellular systems [40]. The above three applications have widely varied quality of service (QoS), and thus, different protocols and algorithms are needed to serve each application [41]. The focus of this thesis is on the design and analysis of access protocols, specifically for mMTC applications.

2.1.1 Challenges in mMTC

One of the main challenges in mMTC is allocation of orthogonal resources to the devices or the users for communications [37]. Conventionally, orthogonal multiple access protocols (OMA) have been used for cellular communications, wherein users are allocated orthogonal resource blocks (RBs) for communicating with the BS. This is made possible by assigning orthogonal resources to users in specific time resources (as in time division

multiple access) or in specific frequency resources (as with frequency division multiple access) [9]. Devices typically contend for resources prior to data transmission, and after receiving an acknowledgement (ACK) of an allocated resource by the BS, they transmit data on that resource. This is the process used for random access of the channel in current standards [37]. However, using such protocols for mMTC is prohibitive since orthogonal RBs cannot be preassigned for such a massive number of users [38]. Millions of users contending and requesting for RBs lead to excessive delays in acquiring the RBs, and consequently, high control and signalling overheads [36]. Thus, non-orthogonal use of the channel is unavoidable in mMTC. For example, using non-orthogonal pilots results in the phenomenon termed *pilot contamination*, where the channel estimates of users become correlated since the pilots of users contaminate the channel estimates of the users [10]. In particular, in mMTC applications, since it is not possible to assign orthogonal pilots to all users, the resulting *pilot contamination* can significantly degrade the performance of IRSA [42].

The next main challenge in mMTC is the sporadic activities of the users. Each user has a packet to transmit only once in a while and they largely remain inactive. Hence, it is unknown which user would be active a priori and as a consequence, unlike OMA, orthogonal RBs or pilot sequences cannot be assigned [38]. This challenge has led to the development of many sparse signal recovery techniques for user activity detection in random access applications. Compressed sensing (CS) techniques can be used to detect devices only when a very small percent of them are active [5].

The most famous non-orthogonal access protocol is perhaps the non-orthogonal multiple access (NOMA) protocol. NOMA is a promising solution for mMTC since without allocating any extra RBs, multiple users can share and use the same resources [43]. Power-domain NOMA is a *centralized* protocol in which the BS pairs two users with disparate received powers on the same resource block. *Centralized* implies that there is a central entity, namely the BS, that coordinates the access of the channels [44]. The user with the stronger received signal at the BS is decoded first, with the assumption that

the other user's signal is additive noise [45]. Once the first user is decoded, the BS performs successive interference cancellation (SIC) from the received signal using the decoded data symbols. Then, the BS attempts to decode the user with the lower received power from the residual received signal [46]. Other variants of NOMA that are also *centralized* are code-domain NOMA, delay-domain NOMA, scrambling/spreading based NOMA, and interleaving-NOMA. These all involve a non-orthogonal collision between users in different domains, and are all centralized [47]. *Distributed* protocols do not have the overhead seen in centralized protocols, and the devices act on their own will by not requesting the BS for a grant of resources [44]. Hence, we need to use distributed access protocols for mMTC.

In existing cellular communications, the traditional goal is to ensure all users or devices are connected to each other with maximal data rates, with irregular traffic from the users [48]. Existing works in NOMA are applicable only for cellular communications and not for mMTC, since they all only analyze NOMA when very few users are connected to the BS [44]. Typically, these works explore the detection and decoding of users with an objective to maximize the information theoretic capacity of the system [49]. For mMTC, the motivation is to ensure connectivity for sporadically transmitting devices [37], whereas for URLLC, we need to support low latencies and high reliabilities [39], which are both different from the existing capacity maximization techniques. Further, in mMTC, the devices transmit only a few packets, infrequently, and unpredictably [42]. Consequently, we need to develop fresh designs and medium access control (MAC) algorithms specifically accounting for the challenges in mMTC.

2.1.2 Standards for mMTC

Historically, IoT communications have been handled with a plethora of machine to machine (M2M) communication protocols [34]. Some famous M2M technologies are wireless fidelity (Wi-Fi), bluetooth, zigbee, low-power wireless personal area networks (LPWAN), and long range wide area network (LoRaWAN). IoT protocols have evolved from these

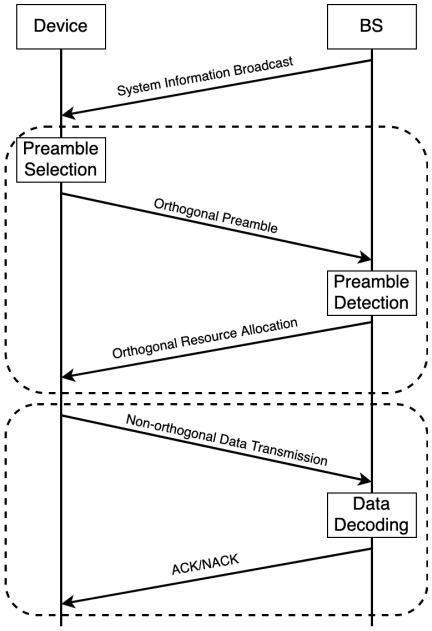
M2M technologies. These technologies, such as LoRaWAN, can be used for mMTC specifically when there are low throughput long-range applications, but not universally for all mMTC applications [32]. Further, end-to-end establishment of a dedicated access network is a disadvantage for such applications [33], because it would incur a large control and signalling overhead.

3GPP has suggested adapting existing cellular protocols for the mMTC requirements, especially for applications with high rates and coverage [31]. Long term evolution for mMTC, which is similar to LPWAN is a standard that can serve low power devices [50]. One other popular protocol is narrow band IoT (NB-IoT), which is a LPWAN standard applicable for low power devices with high connection density [36]. The development of these standards for mMTC is towards improving energy efficiency and connectivity [37]. These standards use either traditional multiple access techniques or modern random access procotols [51]. We now explain these modern random access protocols in detail in the next section.

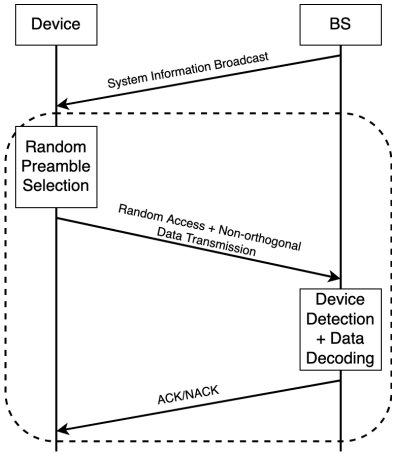
2.1.3 Grant-Based vs Grant-Free Protocols for mMTC

There are several protocols that have been proposed and analyzed for mMTC applications. They are typically divided into *grant-based* and *grant-free* protocols [44]. For grant-based protocols, the users typically request the BS for a grant of resources as depicted in Fig. 2.1(a). The BS periodically transmits a system information broadcast after which users select a preamble, perhaps orthogonal, and transmit it in the first phase [36]. The BS detects the preambles and allocates orthogonal resources to subsets of paired users, similar to the NOMA protocol. Users who have been allocated the same resource block transmit at the same time and are received with collisions at the BS [45]. The BS decodes the data, with perhaps multi-user detection, and then sends back an acknowledgement (ACK) or a negative-ACK (NACK) based on whether the user was decoded correctly or not, respectively. This is verified typically using a forward error correction code, or a cyclic redundancy check code [47]. Conventional OMA is also grant-based.

In grant-free random access (GFRA) protocols, the initial preamble stage is completely



(a) Grant-Based Random Access.



(b) Grant-Free Random Access.

Figure 2.1: Grant-Based vs Grant-Free Random Access.

skipped [52]. Users randomly select a preamble (perhaps pre-assigned) and transmit data in a random access non-orthogonal fashion. The BS performs both device detection and data decoding (perhaps jointly or separately) and then sends an ACK or NACK, as required. Thus, GFRA do not have the overhead of the additional phase at the beginning, and devices can transmit as and when required, making GFRA protocols inherently *distributed* [53]. That is, devices act on their own will and transmit as necessary, and there is no central entity that coordinates this transmission. These characteristics of GFRA protocols make them a promising solution to serve mMTC devices. In particular, GFRA protocols have the advantage of a low control and signalling overhead, which helps in transmission of short packets [5]. The non-orthogonal use of the channel in GFRA helps achieve the high scale of the number of users in mMTC.

Random access (RA) for mMTC is a relatively new research topic and has received immense interest in the recent past. We now list some commonly used RA protocols for mMTC.

1. *Coded Random Access (CRA)*: The CRA family of protocols is a popular set of high-performing protocols in which users' packets are repeated multiple times across different resource blocks [54]. In contention resolution diversity slotted aloha (CRDSA), users transmit their packet replicas in two randomly chosen slots. As an extension, in D-regular repetition slotted aloha (DRRSA), users transmit their replicas in D slots each. Next, in Irregular Repetition Slotted Aloha (IRSA), each user transmits a randomly chosen number of packet replicas in randomly chosen slots [7]. In coded slotted aloha (CSA), each user encodes their packets into multiple packets and then transmit them in randomly chosen slots. CRA has been shown to perform well at high system loads with very low packet loss rates.

2. *Sparse code multiple access (SCMA)*: The main design feature of SCMA is its codebook design [55]. The raw data bits are mapped to coded bits using a channel encoder. Then, the SCMA encoder maps the coded bits to multi-dimensional codewords. The codewords are sparse in nature and are designed under different criteria to mitigate the multi-user interference. Similar to CRA, the decoding can be performed with message passing. SCMA also has been shown to perform well when the system is overloaded.

3. *Unsourced random access (URA)*: URA is a GFRA protocol where a massive number of users transmit their messages to the BS, without a mechanism to identify the transmitting users [56]. In URA applications, the unique identifiers of the active users are not important and the receiver is only interested in the message content itself. URA has been massively studied in recent times [57–59].

4. *Pattern division multiple access (PDMA)*: The devices in PDMA map symbols to resource elements according to a binary codebook [60]. The receiver decodes the users with message passing and successive interference cancellation. The codebooks are constructed to maximize the constellation constrained capacity and also the Hamming distance. PDMA has been shown to work well when there are a lot of users in the system [61]. A difference of CRA with PDMA is that the access

matrix in PDMA is designed in a centralized manner to maximize the so-called *constellation-constrained capacity,* which is not scalable to a massive number of users in mMTC.

5. *Others*: There are several other GFRA protocols such as non-orthogonal coded access, group orthogonal coded access, random phase multiple access, resource spread multiple access and multi-user shared access (MUSA) [45]. The device sequences in MUSA are low cross-correlation spreading sequences, whereas the first two use Grassmanian sequences [62]. MUSA is similar to code division multiple access, where low length code sequences are used [47]. The error-optimal receiver for these protocols is a minimum mean squared error successive interference cancellation (MMSE-SIC) receiver. Enhanced versions of these include the interleave grid multiple access and interleave division multiple access protocols, in which bit-level and symbol-level interleavers are additionally used, respectively [43].

From the above set of protocols, we now focus on the CRA family of protocols.

2.1.4 Coded Random Access Family of Protocols

The coded random access (CRA) family of protocols include contention resolution diversity slotted aloha (CRDSA), D-regular repetition slotted aloha (DRRSA), irregular repetition slotted aloha (IRSA), and coded slotted aloha (CSA) [63]. These protocols evolved as variants of the original slotted aloha protocol, in which users transmit their packet in a randomly chosen time slot. In CRDSA, each user transmits exactly two packet replicas in randomly chosen two slots, whereas in DRRSA, each user transmits exactly D packet replicas in randomly chosen D slots. In IRSA the ith user samples a repetition factor $d_i \in \{2, 3, \ldots, d_{\max}\}$ and transmits packet replicas in d_i randomly chosen slots [7]. Thus, CRDSA and DRRSA are special cases of IRSA in which each user uses the same deterministic d_i. CSA is a generalized version of IRSA, where the data payload of any user is split into multiple encoded packets via linear block codes, which are preassigned. That is, each user encodes its d packet segments across r slots [54], instead of being simply

repeated as in IRSA. The block code rate is d/r, and the decoding of users exploits the knowledge of the encoding code.

The CRA family of protocols, and in particular, the IRSA protocol, is the main focus of this thesis. The common idea across the CRA family of protocols is the transmission of multiple replicas of users' packets over shared RBs or slots [17]. The goal is to ensure that the BS retrieves at least one of these packets and then, using physical layer techniques (such as successive interference cancellation, optimal combining across antennas, etc.), the interference is cancelled across other RBs [7]. IRSA can be seen as an instance of CSA, where users employ repetition codes. The decoding performance of CSA is superior to IRSA, since each user encodes their packets across the multiple transmissions [54]. However, when it comes to the access of RBs, CSA still employs exactly the same strategy as IRSA, and thus, the same analyses can be applied to CSA, CRDSA, DRRSA, and IRSA. Thus, the designs and analyses presented in this thesis are applicable to any of the CRA protocols as well as any other GFRA protocol that accesses the resources in a similar fashion.

2.2 Working of Irregular Repetition Slotted Aloha

Irregular repetition slotted aloha (IRSA) is a distributed GFRA protocol that was proposed as a variant of the slotted aloha protocol in [7]. Typical setup in an IRSA is an uplink system with M users contending to transmit packets over a frame consisting of T RBs of equal sizes. The system load of IRSA is $L = M/T$. We focus on any single frame in IRSA and elucidate the working in the single frame. Users in IRSA transmit replicas of their packets on a randomly selected subset of the available RBs in any frame. The access pattern matrix (APM) is made up by the indices of the RBs in which users transmit in. The BS performs decoding across the RBs using successive interference cancellation. In this section, we first discuss the structure of the APM, then we discuss the interference cancellation process, and finally, we discuss the decoding process in IRSA.

Note: We use the term "resource block" (RB) and "slot" interchangeably in this thesis; both refer to a time frequency resource which can accommodate a whole data packet.

2.2.1 Access Pattern Matrix

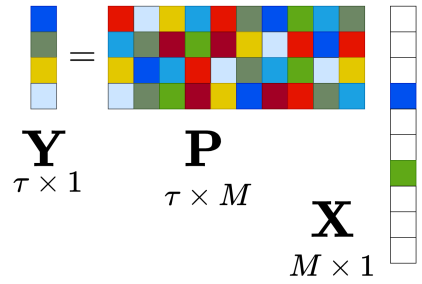
In IRSA, the mth user samples its repetition factor d_m from a predefined distribution, independently of other users. Then it chooses d_m slots from a total of T slots uniformly at random, and transmits replicas of its packet in these d_m slots. The access of slots in a given frame by all the users is represented by a binary APM, which is formed as $[\mathbf{G}]_{tm} = g_{tm}, t \in [T], m \in [M]$, where $g_{tm} = 1$ if the mth user has chosen to transmit in the tth slot, and $g_{tm} = 0$ otherwise. Thus, the APMs in IRSA are random matrices, which are dependent on the distributions used to generate the repetition factors. With such a distribution-based pattern generation, users can independently sample their repetition factors, and by extension, the APM. Since all users generate their patterns independently, this process is scalable to a massive number of users. Further, this process is completely distributed in nature, and is thus appropriate for mMTC. In practice, the random subset of slots is generated using a pseudo-random number generator, whose seed completely determines the sequence [7]. This seed can be pre-programmed at each user, and made available to the BS.

In this thesis, we consider both the case where the APM is known at the BS, and the case where the APM is unknown and needs to be estimated prior to decoding the users' packets (see Chapter 3).

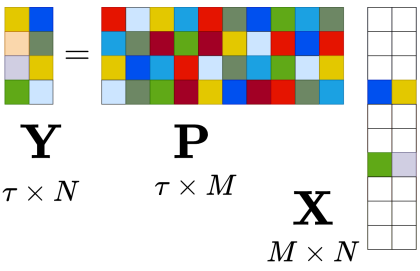
2.2.1.1 Impact of APM on User Activity Detection

The user activity detection (UAD) problem in massive random access usually leverage techniques from compressed sensing (CS). CS is a set of signal processing techniques that are used to efficiently acquire and reconstruct a signal, by finding solutions to underdetermined linear systems [5]. Applications of CS include massive random access, image processing, group testing, photography, and magnetic resonance imaging [64]. CS can be efficiently solved with greedy algorithms, optimization methods, and Bayesian learning techniques [65]. A typical problem in CS is presented in Fig. 2.2(a), where we need to recover a *sparse* vector $\mathbf{X}$ from a system of underdetermined linear equations $\mathbf{Y} = \mathbf{PX}$. This is known as a single measurement vector (SMV) recovery problem, since a single

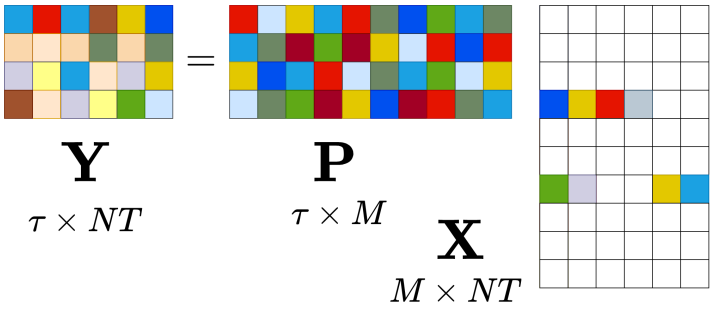
vector needs to be recovered. Once the support of $\mathbf{X}$ is recovered, a least-squares (LS) problem is used to recover the entries of $\mathbf{X}$. Alternately, the support and the entries of $\mathbf{X}$ can be recovered jointly as well.



(a) A typical SMV recovery problem.



(b) A typical MMV recovery problem.



(c) Row-chunk sparsity structure in IRSA.

Figure 2.2: Structures in typical compressed sensing vs Structure in IRSA.

For the UAD problem in conventional single-antenna random access, the entries of $\mathbf{X}$ contain the product of the channels of the users and the corresponding activity coefficients [66]. The entries of $\mathbf{X}$ are zero if the corresponding user is inactive and non-zero if the user is active [67]. Here, we use τ to denote the length of the pilot sequences used by the users, and M to denote the number of users (with $\tau < M$). Further, $\mathbf{P} \in \mathbb{C}^{\tau \times M}$ contains the pilot sequences of all the users, and $\mathbf{Y}$ represents the received pilot signal at the BS. Majority of the entries of $\mathbf{X}$ are zero since the number of active devices in mMTC applications are less than 5%-10% of the total number of devices [47]. Thus, $\mathbf{X}$ is said to be a *sparse vector*. For the single-antenna setup, the sparse Bayesian learning (SBL) algorithm can be used to recover the users' channels and their corresponding activity

coefficients [68, 69].

When multiple antennas are used at the BS for decoding for conventional random access, the structure is a multiple measurement vector (MMV) recovery problem as seen in Fig. 2.2(b). Here, N represents the number of measurement vectors that need to be recovered, and it also corresponds to the number of antennas at the BS. If any user is inactive, the entire row corresponding to that user is all zero, whereas if any user is active, the entire row corresponding to that user is all non-zero. Thus, $\mathbf{X} \in \mathbb{C}^{M \times N}$ is a matrix with *jointly sparse columns*, i.e., a common support across all the columns. For the multiple-antenna setup with jointly sparse columns, the multiple sparse Bayesian learning (MSBL) algorithm can be used to detect which users are active and estimate the active users' channel vectors.

Due to the APM structure in IRSA, the UAD problem is not MMV as seen in Fig. 2.2(c). This structure accounts for the repetitions across the T slots in a frame. If a user is inactive, the corresponding row of $\mathbf{X} \in \mathbb{C}^{M \times NT}$ is still all zero. However, if a user is active, the corresponding row of $\mathbf{X}$ is not all non-zero. It is in fact non-zero only in chunks, where the non-zero chunk indices are indicated by the slots in which that user has transmitted a packet replica. This structure is termed as "Row-chunk sparsity" [70]. Thus, there is a double sparsity structure induced by both the activity sparsity of the users and the sparsity of the columns of the APM. Application of an existing MMV algorithm to the row-chunk sparsity problem in IRSA will yield suboptimal performance since majority of the chunks in any row are zero. In Chapter 4, we develop a UAD algorithm for IRSA, specifically to solving the row-chunk sparsity CS problem induced by the APM.

Note: The UAD algorithm proposed in this thesis is also applicable to CRDSA, DRRSA, and CSA. With CRDSA, each column of the APM, which corresponds to different users, have only two ones at randomly chosen locations and all other entries are zero. With DRRSA, each column has D ones at randomly chosen locations. Similarly, in CSA, each column of the APM has r ones and $T - r$ zeroes, whereas in IRSA, the ith column has d_i ones and $T - d_i$ zeroes. Thus, using any of these protocols only affects the specific instance of the APM, and not the UAD algorithm. While the performance

of our algorithm applied to these protocols can be different based on the instantaneous APM, the algorithm itself can still be directly used. This is because the proposed UAD algorithm is independent of how the APM is generated. The specific instances of the APM could change individually, but the UAD algorithm, design and analysis does not change. Thus, our UAD algorithm can be applied to all the procotols in the CRA family.

2.2.2 Successive Interference Cancellation

The decoding in IRSA is an iterative process involving successive interference cancellation (SIC) [71], where the users are decoded via a combination of inter-RB and intra-RB SIC [8]. *Inter-RB SIC* refers to the removal of packet replicas from a different RB than the one the packet was decoded in, while *intra-RB SIC* refers to the removal of a packet from the same RB in which the packet was decoded, in order to facilitate decoding additional packets that may have been transmitted in that RB. Early works in IRSA used only inter-RB SIC to decode users and assumed a collision model, wherein only singleton RBs can be decoded [7]. Here, singleton RBs are RBs in which only a single user has transmitted, and since there are no collisions in such RBs, users can be decoded with high probability. The decoding proceeds in iterations, and stops when there is no singleton RB available. Since no packets can be decoded in RBs where collisions occur, the maximum possible throughput is one packet per RB, the same as the throughput with perfectly coordinated multiple access. This maximum can be achieved asymptotically as the number of users and RBs go to infinity, when the soliton distribution is used to generate the repetition factors of the users [72].

When the BS is equipped with multiple antennas, it can potentially decode multiple packets in a single RB [73], and thus singleton RBs are not necessary for decoding. Multiple packets can be decoded if the signal to interference plus noise ratios (SINRs) of the packets are sufficiently high [49]. Thus, using an *SINR threshold* model has also been considered for IRSA, where users can be decoded if and only if their SINR is higher than a predetermined threshold [17]. Any user with a sufficiently high instantaneous signal to interference plus noise ratio (SINR) can first be decoded, and the contribution of that

user can be removed from the same RB. After the user with the highest SINR is decoded, other users could potentially be decoded as well. After decoding users with sufficiently high SINRs, with a combination of intra-RB and inter-RB SIC, the packet replicas of the decoded users can be removed from all the RBs in which they have transmitted packets. Then, all the RBs can be revisited to see if further users can be decoded from the residual signal. This procedure is continued iteratively until no further packets can be decoded. This yields a higher throughput compared to the collision model, and can potentially achieve a throughput greater than one packet per RB.

2.2.3 Decoding Process in IRSA

The decoding in IRSA is an iterative process involving SIC [71], where the users are decoded via a combination of inter-RB and intra-RB SIC [8]. SIC-based decoding can be viewed as message passing on a bipartite graph [7], and thus IRSA, which uses SIC decoding, can be decoded on graphs. A typical IRSA frame can be represented as a bipartite graph, which is made up of M user nodes (one node for each user), T slot nodes (one node for each slot), and the edges between them. An edge connects a user node to an slot node if and only if that user has transmitted a packet in that corresponding slot. For example, in Fig. 2.3, there will be an edge between user node u_1 and slot node s_1 if and only if user u_1 has transmitted a packet replica in slot s_1.

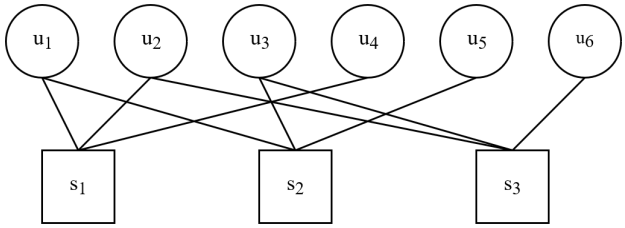


Figure 2.3: IRSA represented as a bipartite graph.

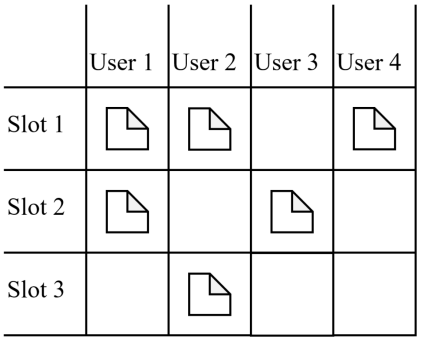
During decoding, edges that connect to users whose SINR is above a threshold are removed from each slot. Each decoding iteration consists of several intra-slot SIC and inter-slot SIC steps. Once an SIC step is performed, the corresponding edge in the bipartite graph is removed. Thus, the edge between user node u_1 and slot node s_1 is

removed if the user u_1 is decoded in any of the slots in which the user has transmitted a packet. Decoding is successful if, at the end of the SIC process, all edges in the graph get removed. A decoding failure is declared if not all edges have been removed or no new edge is removed from the graph in two consecutive iterations.

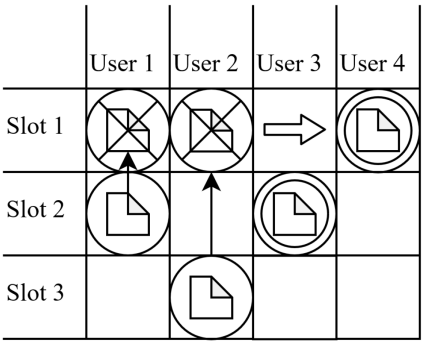
We now illustrate the process of decoding in IRSA via an example. In Fig. 2.4(a), we consider an IRSA frame with 3 slots and 4 users. A packet is depicted at the intersection of a user and a slot if that user has transmitted in that corresponding slot. Note that this is a toy example and a realistic frame would have hundreds of slots and thousands of users. User 1 has chosen to transmit in slots 1 and 2, and would transmit packet replicas in the corresponding slots. Each packet replica contains the pilot symbols, the (coded) data symbols, and the packet decoding error detection symbols, such as a cyclic redundancy check (CRC). The equivalent bipartite graph and the ensuing decoding is presented in Fig. 2.4(b) and Fig. 2.4(c).

During decoding, user 2 is first decoded in slot 3, and the contribution of interference of user 2 is removed from slot 1. Slot 3 is a singleton slot, which is a slot wherein a single user's packet is received without collision. The corresponding edge is removed from the bipartite graph in two stages: one edge from slot node 3 to user node 2, and then the other edge(s) connected to user node 2, since user 2 is already decoded. Next, if capture occurs in slot 2, i.e., if user 1's signal is strong enough compared to user 3, then user 1 can be decoded assuming user 3's signal is noise (see Fig. 2.8). The corresponding edge from slot node 2 to user node 1 is removed. Next, the contribution of interference from all the other slot(s) in which user 1 has transmitted in is removed. This corresponds to edge removal of all edges user node 1 is connected to. Finally, users 3 and 4 are singleton users in slots 2 and 3, respectively. They get decoded and the corresponding edges from the graph are removed. This results in a graph in which all the edges are removed and thus, all the users are decoded.

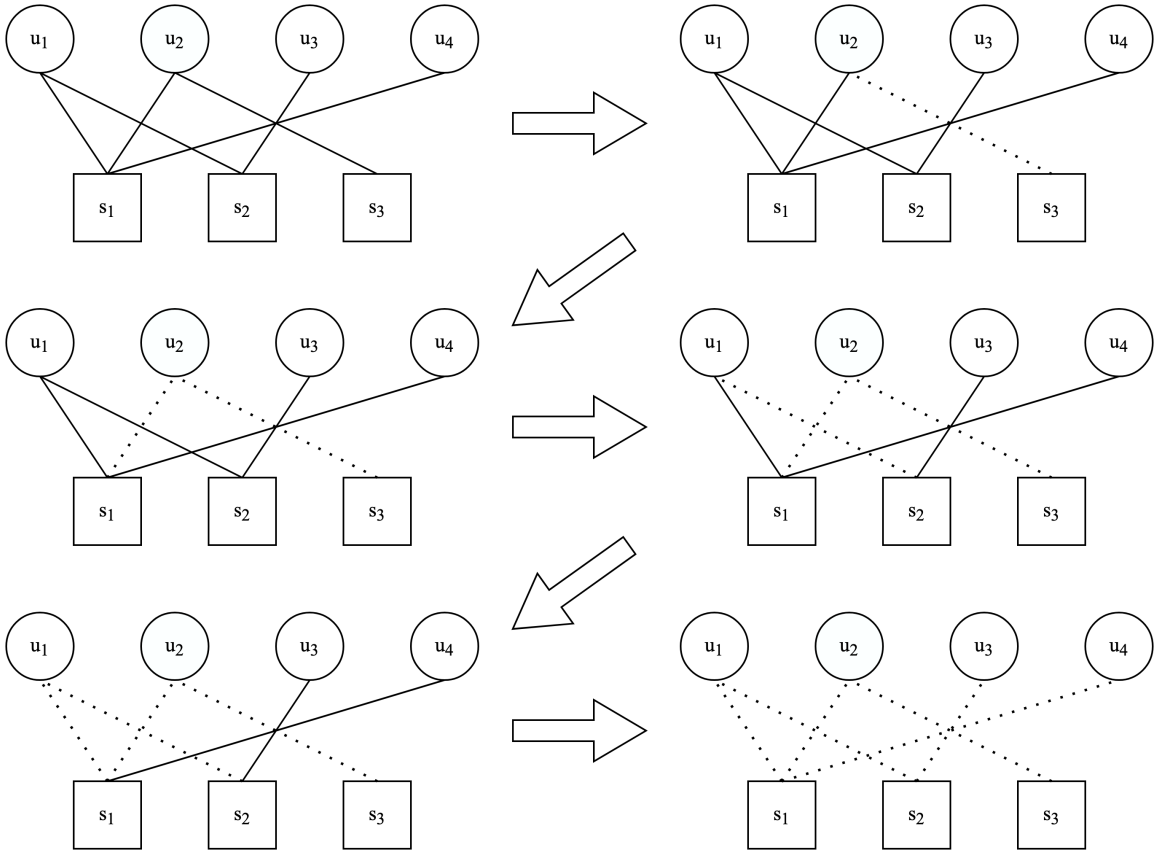
The decoding process as explained above exploits capture effect, and is not dependent on singleton decoding. We now explain the decoding process isolated to the cases accounting and not accounting for capture effect.



(a) A typical frame in IRSA.



(b) Typical decoding process in IRSA.



(c) Graph based decoding in IRSA.

Figure 2.4: SIC-based decoding in IRSA.

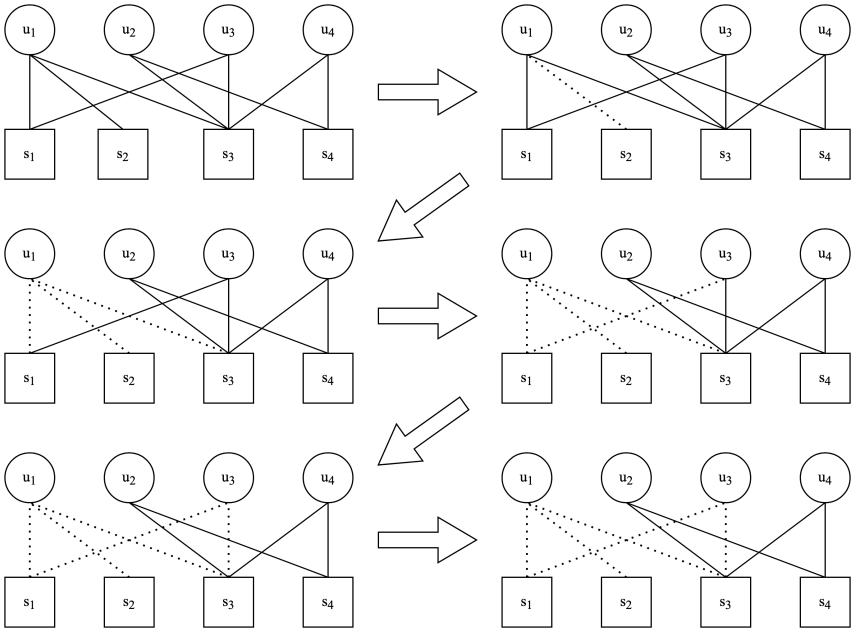


Figure 2.5: Decoding in IRSA – Without capture effect.

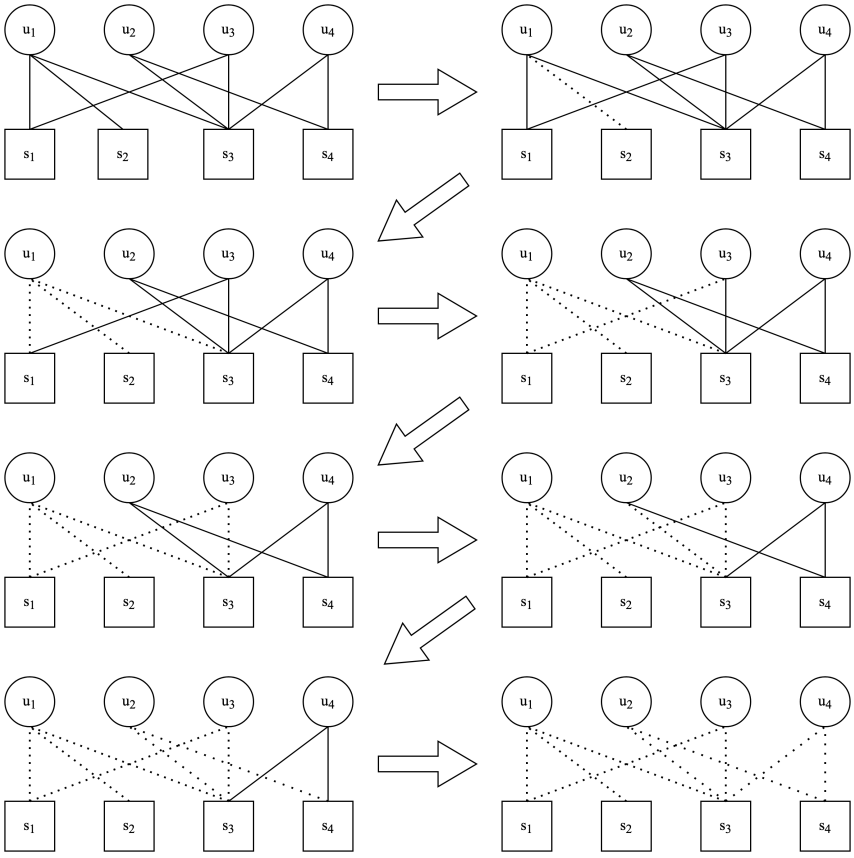


Figure 2.6: Decoding in IRSA – With capture effect.

Without capture effect, the decoding process can result in a residual graph with many undecoded edges. This is depicted in Fig. 2.5. Here, there are four users transmitting in four slots. User 1 gets decoded in slot 2 and then user 3 gets decoded in slot 1. The residual graph contains two edges connected to each slot node. Without capture effect, the decoding process stops here. The residual graph is known as a *stopping set*. In Fig. 2.6, the decoding in IRSA is depicted with the introduction of capture effect (due to multiple antennas, path loss, multi-packet reception capability at the receiver). The residual graph could have user 2 having a higher SINR in slot 3 compared to user 4 in slot 3 (if not, the decoding ends here). Then, the edge between user node 2 and slot node 3 gets removed. Finally users 2 and 4 are decoded, and all the edges in the graph are removed.

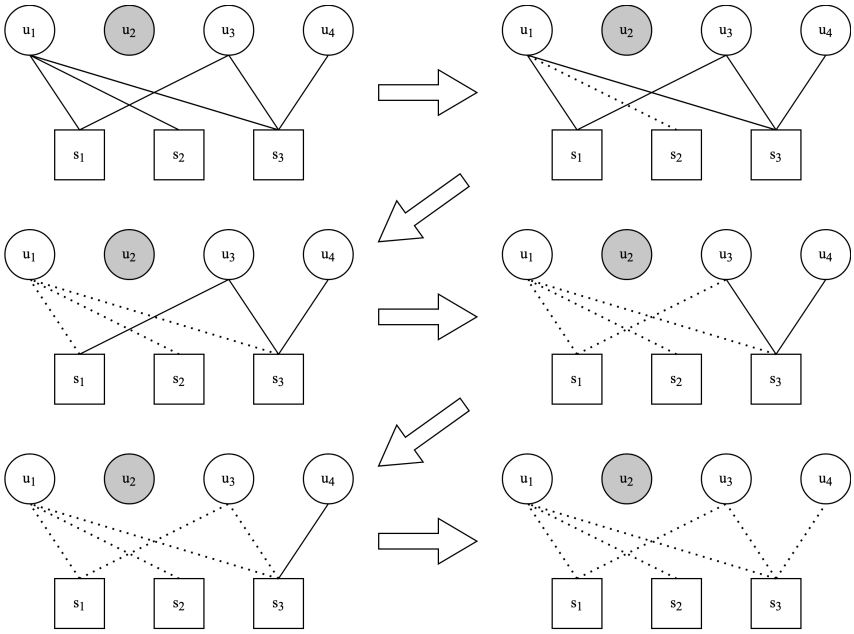


Figure 2.7: Decoding in IRSA – Singleton decoding.

In Fig. 2.7, we depict the decoding of IRSA with singleton decoding. Singleton decoding refers to the decoding of a user's packet in a slot, if that user were the only user transmitting in that slot. This was typically assumed in the early works on IRSA [7], in the *collision channel*. Here, decoding happens only if a slot node has one edge connected to it. Note that this is not the same as in Fig. 2.5, since the decoding could still be using a multi-user decoder that does not exploit capture effect (e.g., the 2-collision

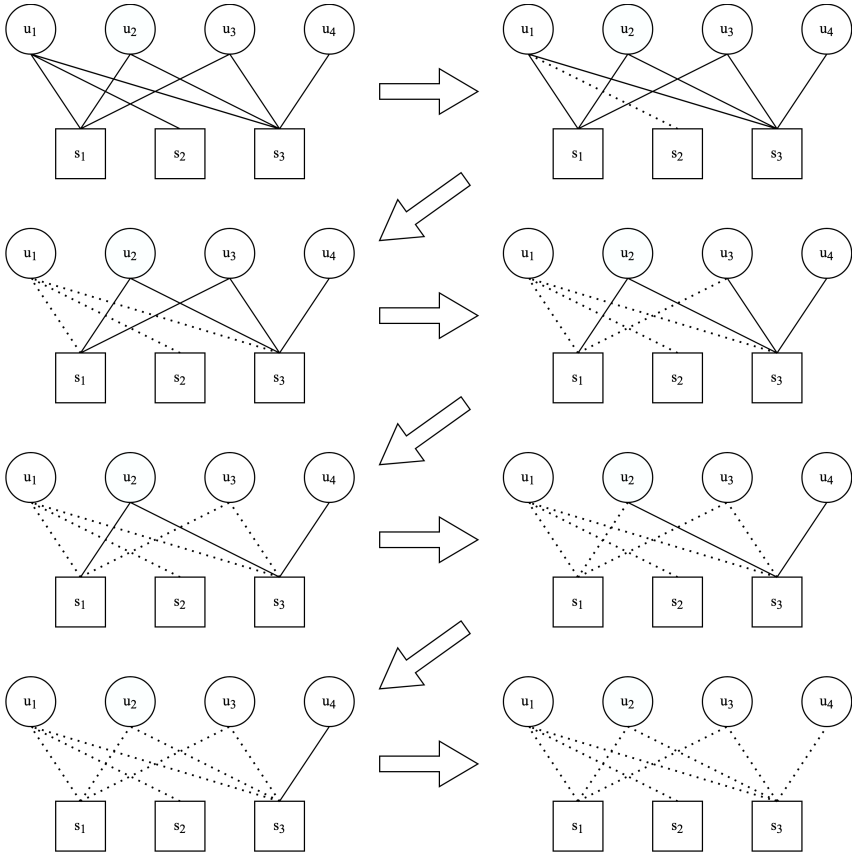


Figure 2.8: Capture effect can yield a throughput greater than 1.

channel without capture effect). This decoding process continues iteratively and it ends with 3 users decoded in 3 slots, with an inactive user node 2. Since no packets can be decoded in slots where collisions occur, the maximum possible throughput is one packet per slot, the same as the throughput with perfectly coordinated multiple access. In fact, for collision channels, the optimal throughput can be made arbitrarily close to 1 packet per RB asymptotically as the number of users and slots go to infinity, and the repetition factors are generated according to the *truncated soliton distribution* [72]. This maximum can be achieved asymptotically as the number of users and slots go to infinity.

When the BS is equipped with multiple antennas, it can potentially decode multiple packets in a single slot, due to capture effect. In Fig. 2.8, user node 1 is first decoded in slot node 2 and its edges are removed. Next, due to capture effect, user node 3 is decoded in slot node 1, and its edges are removed. Then, user node 2 is decoded in slot node 1 and finally user node 4 is decoded in slot node 3. This depicts how IRSA can achieve throughputs greater than 1 due to capture effect as seen in Fig. 2.4.

2.3 Literature Review

Existing literature in IRSA typically assume some idealizations in the physical layer. The advantage of this is to analyze each imperfection (such as fading, path loss, pilot contamination, modulation and coding scheme, and perfect SIC) and capture their individual effects on the performance of IRSA. They can also be compared to other existing works and can function as benchmarks when realistic assumptions are analyzed. Further, they help us evaluate if analyzing with non-idealities is even required, and if so, how much gain or loss we get from doing so. Very few works have analyzed IRSA with non-idealizations. One of the main focuses of this thesis is to analyze IRSA under such non-idealities.

2.3.1 Early Works in IRSA

The IRSA protocol was initially proposed in [7], where it was studied for the collision channel, wherein packets could only be decoded in singleton RBs. Singleton slots refer

to RBs in which only a single packet has been received. This work also connected the decoding in IRSA to the graph-based decoding of LDPC codes, and analyzed IRSA using graph-based decoding. The maximum throughput of IRSA, with a collision channel, was shown to be one packet per RB, when the Soliton distribution is used to generate the repetition factors [72], which is achievable asymptotically when the number of RBs and users go to infinity. Note that the slotted aloha protocol has a peak throughput of $1/e$ packets per RB.

IRSA was extended to CSA in [54], where error correcting codes are used on the replicas of the packets and the encoded packets are transmitted instead of the packet replicas. The authors in [17] first studied IRSA with the Rayleigh fading channel (albeit only a single antenna at the BS), and the authors in [8] studied IRSA with path loss. When the BS is equipped with multiple antennas, it can potentially decode multiple packets in a single slot, if the SINRs of the packets are sufficiently high, which can result in throughputs greater than 1. Thus, using an *SINR threshold* model has been considered for IRSA, where users can be decoded if and only if their SINR is higher than a predetermined threshold [8, 74]. Further, a variety of physical layer abstractions are studied in [74].

The throughput of the IRSA family of multiple access protocols is analyzed using the density evolution (DE) approach, wherein two probability densities are obtained as functions of each other [7]. This iterative recipe provides the asymptotic performance of the system. We explain the working of DE in Section 2.4. The asymptotic throughput has been obtained for IRSA via DE for the MAC channel [72], accounting for path loss [8], for the scalar Rayleigh fading channel [17], with multiuser detectors [75], for the polarized MIMO channel in satellite networks [76], and other variants of IRSA [77, 78].

All of the above works in IRSA assume the availability of perfect channel state information (CSI) at the BS, which is difficult to achieve, especially when non-orthogonal pilots are employed, which is inevitable in mMTC. Channel estimation errors and pilot contamination due to non-orthogonal pilots can erase much of the gains promised by IRSA protocols. Thus, in Chapter 3, we analyze impact of estimated CSI on the performance of IRSA when non-orthogonal pilots are used.

2.3.2 Recent Works in IRSA

The age-of-information metric has recently gained interest in IRSA [51, 79]. IRSA has been examined with energy harvesting [80, 81], and analyzed in an information theoretic setting [82]. Several variants of aloha have been proposed like polar slotted aloha [83], and K-repetition [84, 85]. Authors in [86] and [87] separately propose feedback mechanisms for IRSA based on which decoded users cease transmissions. CSA was recently analyzed with imperfect SIC [88]. The authors in [63, 89] studied CSA with an acknowledgement mechanism between frames. IRSA was analyzed with an SIC limit, i.e., a limit on the maximum number of packets that can be recovered in each RB, in [86]. IRSA has been analyzed for LEO satellite channels [90]. NOMA based CSA has been proposed for mMTC [91]. Finally, a joint PHY and MAC design for CRA has been proposed [92].

2.3.3 UAD in IRSA

To the best of our knowledge, the problem of UAD in IRSA has not yet been considered in the literature. An initial study into estimating the number of active users in IRSA was conducted in [93], which does not identify the subset of active users. UAD has been studied for massive random access outside the context of IRSA [13, 14]. The activity matrix to be estimated has jointly-sparse columns, i.e., columns that have the same sparse support [94]. Typical UAD solutions involve compressed sensing-based solutions [53] or a maximum aposteriori probability (MAP) detection [95]. The sparse Bayesian learning (SBL) framework has been employed to perform UAD in mMTC [96]. Faster SBL algorithms for UAD in mMTC have also been developed [97]. Other low complexity algorithms for UAD include approximate message passing [98] and orthogonal matching pursuit [99]. These approaches, however, cannot be used in IRSA due to the structure imposed by the APM. A naïve approach would be to perform UAD on an RB-by-RB basis and declare users inactive if they are found to be inactive in all the RBs. As we will show in Chapter 4, this approach is inefficient and results in large error rates, especially when non-orthogonal pilots are used.

2.3.4 Interference Limitation in IRSA

The performance of IRSA crucially depends on the *load* of the system, which is the ratio of the number of users participating in a frame to the number of slots in the frame [7]. At low loads, the system is not interference-limited, and the packet loss rate (PLR) is near-zero [8]. Existing works in IRSA talk about an *inflection load*, which is the load beyond which the system becomes overloaded or interference-limited, resulting in a dramatic reduction of the throughput of IRSA [86]. Beyond the *inflection load*, the system is overloaded, IRSA is MUI-limited, and the PLR rapidly goes to one [8].

To the best of our knowledge, no existing work has addressed the interference limitation in IRSA. Thus, in Chapter 5, we improve the performance of IRSA in the interference-limited regime, by developing an enhanced version of IRSA termed as censored-IRSA (C-IRSA).

2.3.5 Multi-Cell and Cell-Free IRSA

In practice, multiple BSs are deployed to cover a large region, and thus, inter-cell interference is inevitable [100]. Both intra-cell interference and inter-cell interference significantly affect the decodability of users [101]. Pilot contamination is caused by both within-cell and out-of-cell users, termed intra-cell pilot contamination (InPC) and inter-cell pilot contamination (IPC), respectively. Furthermore, MC processing (e.g., MC MMSE combining of signals) schemes can achieve better performance compared to SC processing, since it accounts for inter-cell interference [49].

Cell-free (CF) architectures have been proposed for expanding the coverage of communication systems [20]. In a typical CF system, instead of conventional BSs deployed at the centers of cells and serving only the users within the cell, several small access points (APs) are used to jointly and cooperatively serve the users [21]. These APs are spread across the entire region of interest where users have to be served. The APs are connected to a central processing unit (CPU) which is responsible for data aggregation and network coordination [22]. mMTC has the goal of increased connectivity and packet success rates.

This is especially important when there are several cell-edge users who may not be decoded in mMTC due to high path losses. Further, these devices are expected to consume low power and have long battery lives, because of which they cannot transmit at high powers to compensate for the high path loss. CF architectures naturally overcome this issue due to the *macro-diversity gain* (MDG), which helps decode these edge users [23]. Thus, mMTC scenarios are a natural application for using CF architectures, and studying IRSA for mMTC in a CF setup is important.

To the best of our knowledge, no existing work has analyzed the effect of multi-cell (MC) interference on IRSA, nor analyzed IRSA in a cell-free (CF) setup. Thus, in Chapter 6, we account for intra-cell interference and inter-cell interference to analyze IRSA in both the MC and CF setups. We improve upon the peak performance of IRSA in these setups.

2.3.6 Repetition Distributions

In IRSA, the mth user samples their repetition factor d_m from a predefined probability distribution. Existing works have used many distributions for generating the repetition factor. The first paper on IRSA [7] has optimized the repetition distributions for IRSA under the collision channel and proposed a few distributions which have a peak inflection load of 0.965. The *truncated soliton* distribution [72] has been shown to be optimal for IRSA under a collision channel, wherein users can be decoded only if they are received collision-free at the BS, and can push the corresponding throughput close to unity. The truncated soliton distribution is defined by

$$\phi_d = \begin{cases} \dfrac{1-a_s}{2z}, & d = 2, \\ \dfrac{1}{d(d-1)z}, & 3 \leq d \leq k_s, \\ 0 & \text{otherwise}, \end{cases} \tag{2.1}$$

where ϕ_d is the probability that a user has a repetition factor d, k_s is the maximum value that the repetition factor can take, $a_s \in (0, 1)$ is a convergence parameter, and

$z = 1 - a_s/2 - 1/k_s$ is a normalization constant. However, under different physical layer assumptions and with more efficient techniques (both of these are integral aspects of this thesis), we require better repetition distributions, so that the peak performance of IRSA can be improved.

Typically, *numerically optimized* distributions have been obtained for IRSA (e.g., for IRSA with the collision channel [7], and for IRSA with the pure fading channel [17]). The *differential evolution algorithm* can be used to numerically obtain the throughput-optimal distributions in IRSA [102]. The authors in [17] have optimized the repetition distributions for IRSA with pure fading channels in the SISO case and have demonstrated that the achievable peak inflection loads with the optimized distributions exceed unity. The authors in [8] have shown that the soliton distribution is nearly optimal for IRSA with path-loss-only-channels, but they also show better performing distributions. The authors in [74] have claimed that CRDSA, i.e., a 2-regular distribution is the most energy efficient distribution for IRSA.

All of the above papers have optimized the repetition distributions of IRSA under different assumptions. There is no guarantee that those distributions will be optimal for IRSA in the general case, i.e., with pilot contamination, channel estimation errors, multiple antennas, etc. Thus, in Chapter 7, we find repetition distributions that perform better than the soliton distribution or other existing distributions in the general case.

2.4 Density Evolution

Density Evolution (DE) analysis has been applied to characterize the asymptotic performance of message passing-based decoding on graphs for low density parity check (LDPC) codes [103] and IRSA [7]. DE is an iterative recipe to compute the asymptotic throughput. A typical IRSA frame can be represented as a bipartite graph, which is made up of M user nodes (one node for each user), T RB nodes (one node for each RB), and the edges between them. An edge connects a user node to an RB node if and only if that user has transmitted a packet in that corresponding RB. DE is applicable as M and $T \to \infty$ with $L = M/T$ kept fixed [7]. Detailed discussion of the DE technique is found in Chapter 3.

We present only a summary of the technique now.

2.4.1 Overview of DE

The user nodes and the RB nodes exchange failure messages along the edges in any iteration if there is a decoding failure in that iteration. DE involves the characterization of two decoding failure probabilities, one each from users to RBs and from RBs to users. The probability that an edge carries a failure message from an RB node to a user node in the ith iteration is denoted by p_i, and the probability that an edge carries a failure message from a user node to an RB node in the ith iteration is denoted by q_i.

We now discuss the degree distributions in the bipartite graph, which we later use to characterize the failure probabilities. The total number of packets transmitted by a user in any frame in IRSA is referred to as the repetition factor of that user. Each user can have different repetition factors in IRSA. The *node-perspective user degree distribution* is defined as the set of probabilities $\{\phi_d\}_{d=2}^{d_{\max}}$, where ϕ_d represents the probability that a user has a repetition factor d, with $d_{\max}$ being the maximum number of RBs in which any user is allowed to transmit. The total number of packets received in an RB is referred to as the collision factor of that RB. The *node-perspective RB degree distribution* is defined as the set of probabilities $\{\psi_c\}_{c=0}^{M}$, where ψ_c represents the probability that an RB has a collision factor c. The polynomial representations of the node-perspective user and RB degree distributions are

$$\phi(x) = \sum\nolimits_{d=2}^{d_{\max}} \phi_d x^d, \quad \psi(x) = \sum\nolimits_{c=0}^{M} \psi_c x^c, \tag{2.2}$$

respectively. The corresponding *edge-perspective user and RB degree distributions* are defined as $\lambda(x) = \sum_{d=2}^{d_{\max}} \lambda_d x^{d-1} = \phi'(x)/\phi'(1)$; $\xi(x) = \sum_{c=1}^{M} \xi_c x^{c-1} = \psi'(x)/\psi'(1)$, respectively, where $\lambda_d = d\phi_d/\phi'(1)$ represents the probability that an edge is connected to a user with repetition factor d and $\xi_c = c\psi_c/\psi'(1)$ represents the probability that an edge is connected to an RB with collision factor c. The four degree distributions defined above are all probability generating functions. The *input load* L of the system is defined as the

ratio of the number of users to the number of RBs, $L \triangleq M/T$. The average repetition factor is $\bar{d} = \phi'(1) = \sum_d d\phi_d$ and the average collision factor is $\bar{c} = \psi'(1) = \sum_c c\psi_c$, making the load $L = M/T = \bar{c}/\bar{d}$. Thus, fixing the load L and the repetition distribution $\phi(x)$ fixes the other three distributions as well.

The failure probability q_i is calculated using the edge-perspective user degree distribution as

$$q_i = \sum\nolimits_{d=2}^{d_{\max}} \lambda_d q_i^{(d)} = \sum\nolimits_{d=2}^{d_{\max}} \lambda_d p_{i-1}^{d-1} = \lambda(p_{i-1}). \tag{2.3}$$

The failure probability p_i is calculated using the edge-perspective RB degree distribution as

$$p_i = 1 - e^{-L\bar{d}q_i} \sum_{r=1}^{\infty} \theta_r \frac{(L\bar{d}q_i)^{r-1}}{(r-1)!} \triangleq f(q_i), \tag{2.4}$$

where θ_r denotes the probability that the reference packet gets decoded in the current decoding iteration starting from degree r using only intra-RB SIC [8].

Thus, $q_i = \lambda(p_{i-1})$ and $p_i = f(q_i)$ are calculated alternately as functions of each other as seen in (2.3) and (2.4). The procedure can be initialized with either $q_0 = 1$ or $p_0 = f(1)$. The failure probability at the end of decoding is $p_\infty = \lim_{i\to\infty} p_i$ and $(p_\infty)^d$ is the probability that a packet transmitted from a user with repetition factor d does not get decoded at the receiver. Therefore, the asymptotic packet loss rate (PLR), which is the fraction of packets that are not decoded at the BS, is calculated as

$$\mathsf{PLR} = \phi(p_\infty) = \sum\nolimits_{d=2}^{d_{\max}} \phi_d (p_\infty)^d. \tag{2.5}$$

The asymptotic throughput of the system can now be obtained from the asymptotic PLR as $\mathcal{T} = L(1 - \mathsf{PLR})$. The iterations $p_i = f(\lambda(p_{i-1}))$ converge asymptotically to $p_\infty = 0$ if the system load $L < L^*$ [7]. Here, L^* is called the *inflection load* of the system: for any $L \geq L^*$, the system becomes interference limited and the PLR does not converge to 0 as L increases. Thus, for $L < L^*$, $p_\infty = 0$ and therefore the asymptotic PLR $= 0$, and the

throughput equals L. For $L \geq L^*$, the throughput decreases monotonically with L.

2.4.2 Application of DE

We illustrate the DE process by applying it to earlier variants of IRSA.

1. We first analyze the performance of the conventional **slotted aloha** protocol. This is the simplest version of IRSA where users transmit their packet randomly in any one slot (without any repetitions). For slotted aloha, we have the *singleton decoding model*, where we have that a packet can be decoded in a slot if it was received without any collisions. Thus, we have that $\theta_r = \mathbb{1}\{r = 1\}$ and the degree distributions become $\phi(x) = x, \lambda(x) = 1$, and $\psi(x) = \exp(-L(1-x))$. Then, the failure probabilities become $q_i = 1, p_i = 1 - \exp(-L) = p_\infty$, the PLR becomes $\mathsf{PLR} = \phi(p_\infty) = 1 - \exp(-L)$ and thus, the throughput is $\mathcal{T} = Le^{-L}$. The throughput can be maximized analytically, and thus, the peak throughput is $\mathcal{T}^* = 1/e \simeq 0.37$, which occurs at $L^* = 1$.

2. We now analyze the performance of the **contention resolution diversity slotted aloha** protocol, which is the next simplest version of IRSA where users transmit replicas of their packets randomly in any two slots. We still assume the *singleton decoding model*, and thus, $\theta_r = \mathbb{1}\{r = 1\}$. The user degree distributions are $\phi(x) = x^2, \lambda(x) = x$, and $\psi(x) = \exp(-2L(1-x))$, and thus, the failure probabilities are $q_i = p_{i-1}, p_i = 1 - \exp(-2Lq_i)$. The maximum throughput can be numerically found to be $\mathcal{T}^* = 0.55$, but not in closed form, using DE. In **D-regular repetition slotted aloha**, $\phi(x) = x^D, \lambda(x) = x^{D-1}$, and $\psi(x) = \exp(-DL(1-x))$, and thus, the failure probabilities are $q_i = p_{i-1}, p_i = 1 - \exp(-DLq_i)$. Again, the maximum throughput has to be numerically found.

3. We next analyze the performance of **IRSA with singleton decoding** [7], for which we have that $\theta_r = \mathbb{1}\{r = 1\}$. With the degree distribution $\phi(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$ [7], we have that $\overline{d} = 3.6, \lambda(x) = 0.28x + 0.23x^2 + 0.49x^7$, and $\psi(x) = \exp(-3.6L(1-x)) = \rho(x)$. Thus, the failure probabilities are $q_i = \lambda(p_{i-1})$,

and $p_i = 1-\exp(-3.6Lq_i)$. Here, the PLR and the throughput can be only calculated numerically as $\mathsf{PLR} = \phi(p_\infty)$ and $\mathcal{T} = L(1 - \mathsf{PLR})$, and not in closed form. The above distribution has shown to achieve a maximum throughput of $\mathcal{T}^* = 0.938$. Further, the soliton distribution [72] has been proven to be an optimal distribution that pushes $\mathcal{T}^* \to 1$ for $L \to 1$.

4. We now analyze the performance of **IRSA with multi-packet decoding**, for which we have that $\theta_r = 1$, $1 \leq r \leq K$. Under multi-packet decoding, the receiver can decode K or fewer packets perfectly and this would result in $\mathcal{T}^* > 1$ due to multi-user decoding. This yields

$$p_i = \frac{\gamma_{\text{inc}}(K, L\bar{d}q_i)}{\Gamma(K)}.$$

Once again, the throughput can be only calculated numerically and not in closed form.

We use the DE process in Chapters 3, 5, and 7 to find the asymptotic performance of IRSA. The DE process can also be used with the help of the differential evolution algorithm to optimize the repetition distributions of IRSA, which forms the main focus of Chapter 7.

2.5 Summary

In this chapter, we first introduced the requirements for mMTC and then surveyed GFRA protocols proposed for mMTC. We then introduced the CRA family of protocols, and expanded on the working of the IRSA protocol and its decoding. We surveyed existing works on IRSA and then summzrized the DE recipe to compute the throughput of IRSA. The next chapter focuses on the analysis of IRSA with practical channel estimation, i.e., with non-orthogonal pilots and pilot contamination.

Part 1

Chapter 3

Channel Estimation and Data Decoding in IRSA

3.1 Introduction

Massive machine-type communications (mMTC) is an evolving 5G use-case, expected to serve around 10^6 devices per square kilometer [1]. The users in mMTC applications are sporadically active and transmit short packets to a central base station (BS) [2]. Grant-free random access (GFRA) protocols are appropriate in mMTC applications since they incur a low control and signaling overhead [5, 6]. Typically, in these protocols, users transmit packets (consisting of a header containing pilot symbols followed by the data payload) by randomly accessing resource blocks (RBs).[1] Since the length of orthogonal pilots scales linearly with the number of users, the overhead of assigning orthogonal pilots becomes prohibitively expensive. Thus, pilot contamination is inevitable due to the use of non-orthogonal pilots, and has to be accounted for while analyzing the performance of GFRA protocols for mMTC.

One popular GFRA protocol is irregular repetition slotted aloha (IRSA) [7, 54]. Users in IRSA transmit replicas of their packets on a randomly selected subset of the available

[1]We refer to the time-frequency resource as resource blocks (RBs), since each RB can accommodate a whole data packet.

RBs. The indices of the RBs in which they transmit make up the access pattern matrix (APM). Existing works in IRSA assume availability of perfect channel state information (CSI) at the BS, which is difficult to achieve, especially when non-orthogonal pilots are employed. Channel estimation errors and pilot contamination due to non-orthogonal pilots can erase much of the gains promised by IRSA protocols. Thus, one of the main goals of this chapter is to understand the impact of estimated CSI on the performance of IRSA when non-orthogonal pilots are used.

The contents of this chapter is published in a conference paper in IEEE SPAWC in 2019 [12], for a pilot length $\tau = 1$, and in a full length journal paper in the IEEE Transactions on Signal Processing in 2023 [11], for the general case.

3.1.1 The IRSA protocol

The decoding in IRSA is an iterative process involving successive interference cancellation (SIC) [71], where the users are decoded via a combination of inter-RB and intra-RB SIC [8]. *Inter-RB SIC* refers to the removal of packet replicas from a different RB than the one the packet was decoded in, while *intra-RB SIC* refers to the removal of a packet from the same RB in which the packet was decoded, in order to facilitate decoding additional packets that may have been transmitted in that RB. The conventional version of IRSA used only inter-RB SIC to decode users and assumed a collision model, wherein only singleton RBs can be decoded [7]. Here, a singleton RB refers to an RB where a single user's packet is received without collision. Since no packets can be decoded in RBs where collisions occur, the maximum possible throughput is one packet per RB, the same as the throughput with perfectly coordinated multiple access. This maximum can be achieved asymptotically as the number of users and RBs go to infinity, when the soliton distribution is used to generate the repetition factors of the users [72].

When the BS is equipped with multiple antennas, it can potentially decode multiple packets in a single RB, i.e., if the signal to interference plus noise ratios (SINRs) of the packets are sufficiently high. Thus, using an *SINR threshold* model has also been considered for IRSA, where users can be decoded if and only if their SINR is higher than

a predetermined threshold [17]. After decoding users with sufficiently high SINRs, with a combination of intra-RB and inter-RB SIC, the packet replicas of the decoded users can be removed from all the RBs in which they have transmitted packets. Then, all the RBs can be revisited to see if further users can be decoded from the residual signal. This procedure is continued iteratively until no further packets can be decoded. This yields a higher throughput compared to the collision model, and can potentially achieve a throughput greater than one packet per RB. Thus, a second goal of this chapter is to characterize the performance of IRSA under estimated CSI as a function of system parameters such as the number of antennas at the BS, the pilot length, the SINR threshold, etc.

3.1.2 Related Works

The throughput of the IRSA family of multiple access protocols is analyzed using the density evolution (DE) approach, wherein two probability densities are obtained as functions of each other [7]. This iterative recipe provides the asymptotic performance of the system. The asymptotic throughput has been obtained for IRSA via DE for the MAC [72], accounting for path loss [8], for the scalar Rayleigh fading channel [17], with multiuser detectors [75], for the polarized MIMO channel in satellite networks [76], and other enhanced variants of IRSA [77, 78]. We have proposed an algorithm to detect the subset of active users in IRSA [15], wherein we also study the effect of imperfect SIC on IRSA (See Chapter 4). In contrast, we focus on the effect of channel estimation errors on the performance of IRSA.

Thus, a theoretical analysis of the throughput of the IRSA protocol under pilot contamination, accounting for the effect of channel estimation errors, path loss, fading, and multiple antennas at the BS, is not yet available in the literature, to the best of our knowledge.

3.1.2.1 Pattern Division Multiple Access

A closely related protocol is pattern division multiple access (PDMA) [60], where users replicate their packets across a subset of RBs governed by a binary APM. The non-orthogonal channel access, the repetition of packets across multiple RBs, and the SIC-based decoding are common to both PDMA and IRSA based systems. Hence, the design and analysis of IRSA can be adapted to PDMA and vice versa. A difference with PDMA is that the APM is designed in a centralized manner to maximize the so-called *constellation-constrained capacity* [60]. This method quickly becomes computationally prohibitive as the number of users gets large in mMTC. On the other hand, for IRSA, the truncated soliton distribution has been used both in the absence [72] and presence [8] of the so-called *capture effect*, to obtain near-optimal throughput. Owing to the similarity of PDMA with IRSA, the truncated soliton distribution [72], which offers near-optimal throughput, can be used by users in PDMA, independently of one another.

3.1.3 Contributions

Our main contributions in this chapter are as follows:

1. We derive channel estimates for IRSA under three schemes: the first one exploits the sparsity in the APM to estimate the channels of the users, and the other two assume knowledge of the APM and output minimum mean square error (MMSE) estimates. (See Theorem 3.1 in Sec. 3.3.)

2. We present a novel analysis of the SINR in IRSA accounting for channel estimation errors, where estimates are acquired via non-orthogonal pilots under the three estimation schemes. (See Theorem 3.2 in Sec. 3.4.)

3. We theoretically analyze the throughput of IRSA via DE, when users perform path loss inversion based power control. The analysis reveals the asymptotic performance of the protocols as the number of users and RBs get large. (See Theorem 3.3 in Sec. 3.5.4 and also Sec. 3.5.3.)

Through extensive simulations, we show that channel estimation errors lead to a significant loss of throughput compared to the ideal scenario with perfect CSI at the BS, even resulting in up to 70% loss in severely interference-limited regimes. In particular, in mMTC applications, since it is not possible to assign orthogonal pilots to all users, the resulting pilot contamination can significantly degrade the SINR, leading to poor performance. On the positive side, this loss in performance can be recuperated by optimizing system parameters such as pilot length, number of antennas, frame length, signal to noise ratio, and SINR threshold. In particular, we show that the pilot length required to obtain near-optimal performance is orders of magnitude lower than the pilot length needed to assign orthogonal pilots to all users. For example, a pilot length of $\tau = 12$ is sufficient to obtain optimal performance with $M = 150$ users, whereas the use of orthogonal pilot sequences requires $\tau = 150$ pilot symbols. (See Fig. 3.2). This is possible because only a small fraction of users transmit in a given RB in IRSA; exploiting this sparsity in user access allows one to obtain accurate channel estimates even when the pilots are non-orthogonal. (See Algorithm 1.)

Our analysis also allows us to determine the *inflection load*, beyond which the system becomes interference-limited, resulting in a dramatic reduction of the throughput. The asymptotic throughput obtained via DE serves as an upper bound for the achievable throughput, and facilitates numerical optimization of the throughput with respect to the system parameters.

Notation

The symbols a, $\mathbf{a}$, $\mathbf{A}$, $[\mathbf{A}]_{i,:}$, $[\mathbf{A}]_{:,j}$, $\mathbf{0}_N$, $\mathbf{1}_N$, and $\mathbf{I}_N$ denote a scalar, a vector, a matrix, the ith row of $\mathbf{A}$, the jth column of $\mathbf{A}$, all-zero vector of length N, all ones vector of length N, and an identity matrix of size $N \times N$, respectively. $[\mathbf{a}]_{\mathcal{S}}$ and $[\mathbf{A}]_{:,\mathcal{S}}$ denote the elements of $\mathbf{a}$ and the columns of $\mathbf{A}$ indexed by the set $\mathcal{S}$ respectively. $\mathrm{diag}(\mathbf{a})$ is a diagonal matrix with diagonal entries given by $\mathbf{a}$. The set of real and complex matrices of size $N \times M$ are denoted as $\mathbb{R}^{N\times M}$ and $\mathbb{C}^{N\times M}$. $\mathcal{N}(\mathbf{a}, \mathbf{A})$ and $\mathcal{CN}(\mathbf{a}, \mathbf{A})$ denote the real and complex Gaussian distribution, respectively, with mean $\mathbf{a}$ and covariance $\mathbf{A}$. $[N]$ denotes the set

$\{1, 2, \ldots, N\}$. $|\cdot|$, $\|\cdot\|$, $[\cdot]^T$, $[\cdot]^*$, $[\cdot]^H$, $\mathbb{E}[\cdot]$, and $\mathbb{E}_{\mathbf{a}}[\cdot]$ denote the magnitude (or cardinality of a set), ℓ_2 norm, transpose, conjugate, hermitian, expectation, and the expectation conditioned on $\mathbf{a}$, respectively. The superscript $\mathtt{p}$ is used as a descriptive superscript in association with a symbol that is related to the *pilots*. All the other superscripts (or subscripts) that have not been defined as above are indices. A non-exhaustive list of symbols used in this chapter is presented in Table 3.1.

Table 3.1: Mathematical symbols used in this chapter.

Symbol	Quantity	Symbol	Quantity
L	Load	γ_{pr}	Threshold used to declare support
θ_r	Success probability	P	Data power
τ	Pilot length	γ_{th}	Capture threshold
T	Number of RBs	$P^{\mathtt{p}}$	Pilot power
τ_c	Packet length	$\mathbf{G}$	Access pattern matrix
N	Number of antennas	N_0	Noise variance
$\mathcal{T}$	Throughput	λ	Regularization parameter
M	Number of users	σ^2_{h}	Channel variance

3.2 System Model

An IRSA system is considered with M single-antenna users communicating with a central BS equipped with N antennas. The users are assumed to be arbitrarily located within a cell, with the BS located at the cell center. The fading is modeled as block-fading, quasi-static and Rayleigh distributed. The time-frequency resource is divided into RBs, and T RBs together constitute a *frame*. The RBs can be slots, subcarriers or both. In each frame, the users contend for the channel by randomly selecting a subset of RBs, and they transmit replicas of their packets in the selected RBs. Each packet replica comprises of a header containing pilot symbols and payload containing data and error correction symbols.

The access of RBs in a given frame by all the users can be represented by a binary access

pattern matrix (APM) $\mathbf{G} \in \{0,1\}^{T\times M}$. The entries of $\mathbf{G}$ are denoted by $g_{tm} = [\mathbf{G}]_{tm}$, and $g_{tm} = 1$ if the mth user transmits its packet in the tth RB, and $g_{tm} = 0$ otherwise. The mth user samples their repetition factor d_m from a preset probability distribution. They then choose d_m RBs from a total of T RBs uniformly at random for transmission. We note that, due to the distributed nature of the protocol, the M columns of $\mathbf{G}$ are i.i.d., and $\mathbf{G}$ is independently generated from one frame to the next.

At the BS, the received signal in the tth RB is a superposition of the packets transmitted by the users that are scheduled to transmit in the same RB. In the pilot phase, if $g_{tm} = 1$, the mth user transmits a τ-length pilot $\mathbf{p}_m \in \mathbb{C}^\tau$ in the tth RB, with each pilot symbol transmitted at an average power $P^{\mathtt{p}}$, and thus, $\mathbb{E}[\|\mathbf{p}_m\|^2] = \tau P^{\mathtt{p}}$. The pilot signal $\mathbf{Y}_t^{\mathtt{p}} \in \mathbb{C}^{N\times\tau}$ received at the BS using its N antennas and in the tth RB is given by

$$\mathbf{Y}_t^{\mathtt{p}} = \sum\nolimits_{m=1}^{M} g_{tm}\mathbf{h}_{tm}\mathbf{p}_m^H + \mathbf{N}_t^{\mathtt{p}}, \tag{3.1}$$

where $\mathbf{N}_t^{\mathtt{p}} \in \mathbb{C}^{N\times\tau}$ is the complex additive Gaussian noise at the BS with $[\mathbf{N}_t^{\mathtt{p}}]_{nj} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0)\ \forall\ n \in [N],\ j \in [\tau]$ and $t \in [T]$, and N_0 is the noise variance. Here $\mathbf{h}_{tm} = [h_{tm1}, \ldots, h_{tmN}]^T$ is the uplink channel vector of the mth user in the tth RB, with $h_{tmn} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, \beta_m\sigma_{\mathtt{h}}^2),\ \forall\ t \in [T],\ m \in [M]$ and $n \in [N]$, where β_m is the path loss coefficient and $\sigma_{\mathtt{h}}^2$ is the fading variance.

In the data phase, users transmit their data symbols. Considering one of the data symbols, the mth user transmits a data symbol x_m with $\mathbb{E}[x_m] = 0$ and $\mathbb{E}[|x_m|^2] = P$, i.e., with transmit power P. The corresponding received data signal $\mathbf{y}_t \in \mathbb{C}^N$ at the BS in the tth RB is

$$\mathbf{y}_t = \sum\nolimits_{m=1}^{M} g_{tm}\mathbf{h}_{tm}x_m + \mathbf{n}_t, \tag{3.2}$$

where $\mathbf{n}_t \in \mathbb{C}^N$ is the complex additive white Gaussian noise at the BS with $[\mathbf{n}_t]_n \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0),\ \forall\ n \in [N]$ and $t \in [T]$.

3.2.1 SIC-based Decoding

The received data is processed iteratively at the BS. The BS computes channel estimates for all users in all RBs using the pilot symbols.[2] It uses these channel estimates to combine the received data signal across the BS antennas and attempts to decode the user's data packet, treating interference from other users as noise. If it successfully decodes any user, which can be verified via a cyclic redundancy check, it performs SIC in all RBs in which that user has transmitted, with both inter-RB and intra-RB SIC. The BS proceeds with the next iteration, where the channels are re-estimated for the remaining users, and this decoding process proceeds iteratively.

That is, if the SINR of a packet in a given RB in any decoding iteration exceeds a threshold γ_{th}, then the packet can be decoded correctly [8, 17]. *Packet capture* occurs when a packet can be decoded correctly as per the SINR threshold model, even though it collides with another packet, and is thus considered a good abstraction of the decoding in the physical layer.

We now describe the performance evaluation of IRSA via the SINR threshold model. We first compute channel estimates and SINR achieved by all users in all RBs. If we find a user with SINR $\geq \gamma_{\text{th}}$ in some RB, we mark the data packet as having been decoded successfully and remove the contribution of the user's packet from all RBs that contain a replica of that packet. In the next iteration, the channels are re-estimated from the residual pilot symbols after SIC, the SINRs are recomputed in all RBs, and the decoding of users' packets continues. The decoding process proceeds in iterations and stops when no additional users are decoded in two successive iterations. The system throughput $\mathcal{T}$ is calculated as the number of correctly decoded unique packets divided by the number of RBs.

[2] As we will see, when the BS does not know the APM, the BS first detects which users have transmitted in each RB, and computes the channel estimates for the users detected to be active in each of the RBs.

3.3 Channel Estimation

In this section, the channel estimates for all users are derived under three schemes. The first scheme, termed the sparsity-based estimation scheme, estimates both the APM and the channels of the users. In contrast with this, the other schemes exploit the knowledge of $\mathbf{G}$ and output MMSE estimates. Assuming the knowledge of $\mathbf{G}$ is not a strong assumption and can be made possible by using pseudo-random pattern matrices generated from a seed that is available at the BS and the users.[3]

Channel estimation is performed based on the received pilot signal, which contains the pilots transmitted by all the users who have transmitted in that RB. The estimates are recomputed in every iteration, and hence the signals and channel estimates are indexed by the decoding iteration k. Let the set of users who have not yet been decoded in the first $k-1$ iterations be denoted by $\mathcal{S}_k$, and for some $m \in \mathcal{S}_k$, let $\mathcal{S}_k^m \triangleq \mathcal{S}_k \setminus \{m\}$, with $\mathcal{S}_1 = [M]$. The received pilot signal at the BS, in the tth RB, and during the kth decoding iteration, is given by

$$\mathbf{Y}_t^{\mathrm{p}k} = \sum\nolimits_{i \in \mathcal{S}_k} g_{ti} \mathbf{h}_{ti} \mathbf{p}_i^H + \mathbf{N}_t^{\mathrm{p}}. \tag{3.3}$$

We now discuss three channel estimation schemes for IRSA.

3.3.1 Sparsity-based APM and Channel Estimation

The first scheme is the sparsity-based estimation scheme in which we estimate the APM and the channels in each decoding iteration. We consider the conjugate transpose of the received pilot signal in the tth RB from (3.3) as $\overline{\mathbf{Y}}_t^{\mathrm{p}k} \triangleq \mathbf{Y}_t^{\mathrm{p}kH}$, with $\overline{\mathbf{N}}_t \triangleq \mathbf{N}_t^{\mathrm{p}H}$. Let $\mathbf{P} \in \mathbb{C}^{\tau \times M}$ contain the known pilots of the M users as its columns and $\mathbf{P}^k = [\mathbf{P}]_{:,\mathcal{S}_k}$. The

[3]This assumption is substantiated in Sec. 3.9.5.

signal $\overline{\mathbf{Y}}_t^{\mathtt{p}k}$ can be factorized into the product of two matrices as follows:

$$\underbrace{\overline{\mathbf{Y}}_t^{\mathtt{p}k}}_{\tau\times N} = \underbrace{\left[\mathbf{p}_{i_1},\ldots,\mathbf{p}_{i_{M^k}}\right]}_{\mathbf{P}^k} \underbrace{\begin{bmatrix} g_{ti_1}\mathbf{h}_{ti_1}^H \\ \vdots \\ g_{ti_{M^k}}\mathbf{h}_{ti_{M^k}}^H \end{bmatrix}}_{\mathbf{Z}_t^k} + \underbrace{\overline{\mathbf{N}}_t}_{\tau\times N}, \tag{3.4}$$

where $\mathcal{S}_k = \{i_1, i_2, \ldots, i_{M^k}\}$, with $M^k = |\mathcal{S}_k|$. Here, $\mathbf{Z}_t^k \in \mathbb{C}^{M^k\times N}$ contains the tth row of the unknown APM $\mathbf{G}$, and the unknown channels. The rows of $\mathbf{Z}_t^k$ are either all-zero or all-nonzero depending on whether the corresponding $g_{ti} = 0$ or 1. This results in an under-determined system of equations, where the columns of $\mathbf{Z}_t^k$ share the same support. This structure is called as a multiple measurement vector (MMV) recovery problem in compressed sensing. The estimation of $\mathbf{Z}_t^k$ from (3.4) can be performed using well known MMV recovery algorithms from compressed sensing literature to recover $\{g_{ti}\}$ in the each of the T RBs.

Multiple sparse Bayesian learning[4] (MSBL) [16] is an empirical Bayesian algorithm that can recover $\mathbf{Z}_t^k$ from linear under-determined observations $\overline{\mathbf{Y}}_t^{\mathtt{p}k}$. In MSBL, a Gaussian prior is imposed on the columns of $\mathbf{Z}_t^k$ as

$$p(\mathbf{Z}_t^k;\boldsymbol{\gamma}_{kt}) = \prod_{n=1}^{N} p([\mathbf{Z}_t^k]_{:,n};\boldsymbol{\gamma}_{kt}) = \prod_{n=1}^{N} \mathcal{CN}(\mathbf{0}_{M_t},\boldsymbol{\Gamma}_{kt}), \tag{3.5}$$

where $\boldsymbol{\Gamma}_{kt} = \mathrm{diag}(\boldsymbol{\gamma}_{kt})$ and the columns of $\mathbf{Z}_t^k$ are i.i.d. The elements of $\boldsymbol{\gamma}_{kt} \in \mathbb{R}_+^{M^k}$ are unknown hyperparameters for the undecoded users. Recovering the hyperparameters would yield g_{tm} since $[\boldsymbol{\gamma}_{kt}]_m$ models the variance of the mth user's channel in the tth RB. The hyperparameters are estimated by iteratively maximizing the log-likelihood $\log p(\overline{\mathbf{Y}}_t^{\mathtt{p}k};\boldsymbol{\gamma}_{kt})$, with $p(\overline{\mathbf{Y}}_t^{\mathtt{p}k};\boldsymbol{\gamma}_{kt}) = \prod_{n=1}^{N} p([\overline{\mathbf{Y}}_t^{\mathtt{p}k}]_{:,n};\boldsymbol{\gamma}_{kt})$.

Stated in our notation, the overall estimation procedure is summarized in Algorithm 1.

[4]Any MMV algorithm can be used to recover joint-sparse columns of $\mathbf{Z}_t^k$, but we use MSBL due to its high performance. MSBL also outputs a "plug-in" MMSE channel estimate which can then be used to find a meaningful SINR expression since the estimate is uncorrelated with the estimation error [104].

Algorithm 1: APM and Channel Estimation in tth RB

Input: $\tau, N, N_0, \mathcal{S}_k, \mathbf{P}, \overline{\mathbf{Y}}_t^{\mathrm{p}k}, \gamma_{\mathrm{pr}}, j_{\max}$

Compute: $M^k = |\mathcal{S}_k|$, $\mathbf{P}^k = [\mathbf{P}]_{:,\mathcal{S}_k}$

Initialize: $\boldsymbol{\gamma}_{kt}^0 = \mathbf{1}_{M^k}$

for $j = 0, 1, 2, \ldots, j_{\max}$ **do**

  Compute $\boldsymbol{\Gamma}_{kt}^j = \mathrm{diag}(\boldsymbol{\gamma}_{kt}^j)$

  $\boldsymbol{\Sigma}_{kt}^{j+1} = \boldsymbol{\Gamma}_{kt}^j - \boldsymbol{\Gamma}_{kt}^j \mathbf{P}^{kH} (N_0 \mathbf{I}_\tau + \mathbf{P}^k \boldsymbol{\Gamma}_{kt}^j \mathbf{P}^{kH})^{-1} \mathbf{P}^k \boldsymbol{\Gamma}_{kt}^j$

  $\boldsymbol{\mu}_{ktn}^{j+1} = N_0^{-1} \boldsymbol{\Sigma}_{kt}^{j+1} \mathbf{P}^{kH} [\overline{\mathbf{Y}}_t^{\mathrm{p}k}]_{:,n},\ 1 \le n \le N$

  $[\boldsymbol{\gamma}_{kt}^{j+1}]_i = \frac{1}{N} \sum_{n=1}^{N} ([\boldsymbol{\Sigma}_{kt}^{j+1}]_{i,i} + |[\boldsymbol{\mu}_{ktn}^{j+1}]_i|^2),\ \forall\ i \in [M^k]$

end

Output: $\hat{g}_{tm}^k = \begin{cases} 1, & [\boldsymbol{\gamma}_{kt}^{j_{\max}}]_m \ge \gamma_{\mathrm{pr}} \\ 0, & [\boldsymbol{\gamma}_{kt}^{j_{\max}}]_m < \gamma_{\mathrm{pr}} \end{cases},\ \forall\ m \in [M^k],\ \hat{\mathbf{Z}}_t^k = [\boldsymbol{\mu}_{kt1}^{j_{\max}} \boldsymbol{\mu}_{kt2}^{j_{\max}} \ldots \boldsymbol{\mu}_{ktN}^{j_{\max}}]$

The MSBL algorithm converges to a saddle point or a local maximizer of the overall log-likelihood [16]. Further, the MSBL algorithm has been empirically shown to correctly recover the support of $\mathbf{Z}_t^k$, provided τ and N are large enough [16], if the signal to noise ratio is good enough. The algorithm is run for $j_{\max}$ iterations in each of the T RBs. As the iterations proceed, the hyperparameters corresponding to users with $g_{ti} = 0$ converge to zero, resulting in sparse estimates. At the end of the iterations, the estimated coefficient $\hat{g}_{tm}^k$ for the mth user in the tth RB in the kth decoding iteration is obtained by thresholding $[\boldsymbol{\gamma}_{kt}^{j_{\max}}]_m$ at a value γ_{pr}. This can result in errors in estimating g_{ti}, and the errors in APM estimation can be described by

$$\mathcal{F}_t^k = \{i \in [M^k] \mid \hat{g}_{ti}^k (1 - g_{ti}) = 1\}, \tag{3.6a}$$

$$\mathcal{M}_t^k = \{i \in [M^k] \mid (1 - \hat{g}_{ti}^k) g_{ti} = 1\}, \tag{3.6b}$$

where $\mathcal{F}_t^k$ is the set of false positive users, and $\mathcal{M}_t^k$ is the set of false negative users. These errors affect decoding of other users in two ways: both kinds of users contaminate the channel estimates of other users, and users in $\mathcal{M}_t^k$ interfere with the data decoding of

other users as well. The effect of errors in detection of users is described in detail in [15] (See Chapter 4).

The algorithm also outputs the maximum aposteriori probability estimates of the channels $\hat{\mathbf{Z}}_t^k$ in each of the T RBs. The estimate $\hat{\mathbf{H}}_t^k = \hat{\mathbf{Z}}_t^{kH} \in \mathbb{C}^{N \times M^k}$ of the channels of the M^k users is described in Theorem 1 and can be calculated as $\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k}\mathbf{P}^k\hat{\mathbf{\Gamma}}_{kt}(\mathbf{P}^{kH}\mathbf{P}^k\hat{\mathbf{\Gamma}}_{kt} + N_0\mathbf{I}_{M^k})^{-1}$, where $\hat{\mathbf{\Gamma}}_{kt} = \mathrm{diag}(\boldsymbol{\gamma}_{kt}^{j_{\max}})$. This estimate is a "plug-in" MMSE estimate and it contains estimates for erroneously detected users as well. An added advantage of MSBL is that the path loss coefficient can be calculated by averaging the estimated hyperparameters across RBs as $\hat{\beta}_i^k = (\sum_{t=1}^T \hat{g}_{ti}^k[\boldsymbol{\gamma}_{kt}^{j_{\max}}]_i)/(\sigma_{\mathtt{h}}^2 \sum_{t=1}^T \hat{g}_{ti}^k)$. Thus, Algorithm 1 does not require any prior information about the APM or $\{\beta_i\}_{i=1}^M$ to estimate the channels.

3.3.2 MMSE Channel Estimation with Known APM

We now derive the MMSE channel estimates for all users in each RB, exploiting the knowledge of the APM $\mathbf{G}$ and $\{\beta_i\}_{i=1}^M$. By using a common seed at the BS and the users, the APM can be generated at the BS and thus, we can assume that the BS has knowledge of $\mathbf{G}$.[5] Let $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$ be the set of users who have transmitted in the tth RB. Let $M_t^k = |\mathcal{G}_t \cap \mathcal{S}_k|$ be the number of users who have transmitted in the tth RB and have not been decoded in the first $k-1$ iterations, $\mathbf{H}_t^k \in \mathbb{C}^{N \times M_t^k}$ denote the channel matrix which contains the channels of the M_t^k users, $\mathbf{P}_t^k \in \mathbb{C}^{\tau \times M_t^k}$ denote a matrix that contains the pilot sequences of the M_t^k users and $\mathbf{B}_t^k \triangleq \sigma_{\mathtt{h}}^2 \mathrm{diag}(\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_{M_t^k}})$ be a diagonal matrix containing the path loss coefficients of the M_t^k users, with $\mathcal{G}_t \cap \mathcal{S}_k = \{i_1, i_2, \ldots, i_{M_t^k}\}$. Thus, the received signal from (3.3) can be written as $\mathbf{Y}_t^{\mathtt{p}k} = \mathbf{H}_t^k\mathbf{P}_t^{kH} + \mathbf{N}_t^{\mathtt{p}}$, where $\mathbf{P}_t^k = [\mathbf{P}]_{:,\mathcal{G}_t \cap \mathcal{S}_k}$. The MMSE estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ is presented in Theorem 3.1, and can be written as

$$\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k}(\mathbf{P}_t^k\mathbf{B}_t^k\mathbf{P}_t^{kH} + N_0\mathbf{I}_\tau)^{-1}\mathbf{P}_t^k\mathbf{B}_t^k, \tag{3.7a}$$

$$\overset{(a)}{=} \mathbf{Y}_t^{\mathtt{p}k}\mathbf{P}_t^k\mathbf{B}_t^k(\mathbf{P}_t^{kH}\mathbf{P}_t^k\mathbf{B}_t^k + N_0\mathbf{I}_{M_t^k})^{-1}, \tag{3.7b}$$

[5] This assumption is substantiated in Sec. 3.9.5.

where (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$. Here, the estimate can be calculated via an inverse of either a $\tau \times \tau$ matrix or an $M_t^k \times M_t^k$ matrix as required. The MSBL estimate converges to the MMSE estimate when the hyperparameters are estimated well enough, as will be seen in Sec. 3.6.

3.3.3 Low Complexity MMSE with Known APM

We now describe a low complexity MMSE (LCMMSE) estimate that does not require a matrix inversion computation. For this purpose, the received signal in (3.3) is right-multiplied by the pilot $\mathbf{p}_m$ to obtain

$$\mathbf{y}_{tm}^{\mathrm{p}k} = \mathbf{Y}_t^{\mathrm{p}k}\mathbf{p}_m = \sum\nolimits_{i\in\mathcal{S}_k} g_{ti}\mathbf{h}_{ti}(\mathbf{p}_i^H\mathbf{p}_m) + \mathbf{N}_t^{\mathrm{p}}\mathbf{p}_m, \tag{3.8}$$

which is used to find an MMSE estimate of the channel $\mathbf{h}_{tm}$ of the mth user in the tth RB. The LCMMSE channel estimate $\hat{\mathbf{h}}_{tm}^k$ is described in Theorem 3.1 and is calculated as

$$\hat{\mathbf{h}}_{tm}^k = \frac{g_{tm}\beta_m\|\mathbf{p}_m\|^2\sigma_{\mathrm{h}}^2}{N_0\|\mathbf{p}_m\|^2 + \sum_{i\in\mathcal{S}_k}|\mathbf{p}_i^H\mathbf{p}_m|^2 g_{ti}\beta_i\sigma_{\mathrm{h}}^2}\mathbf{y}_{tm}^{\mathrm{p}k} \triangleq \eta_{tm}^k\mathbf{y}_{tm}^{\mathrm{p}k}. \tag{3.9}$$

Similar to the MMSE estimate, the LCMMSE estimate uses the knowledge of the APM and $\{\beta_i\}_{i=1}^M$. While the MMSE estimator uses the signal $\mathbf{Y}_t^{\mathrm{p}k}$ to compute the estimates, and thus exploits all the information available at the BS, the LCMMSE estimator uses only $\mathbf{y}_{tm}^{\mathrm{p}k}$, i.e., the projection of $\mathbf{Y}_t^{\mathrm{p}k}$ onto $\mathbf{p}_m$, to estimate $\mathbf{h}_{tm}$.

The channel estimates under the three schemes and their error variances are given by the following theorem.

Table 3.2: Channel estimates and error variances under three estimation schemes.

	Sparsity-based estimation with MSBL	MMSE	LCMMSE
$\hat{\mathbf{H}}_t^k$	$\mathbf{Y}_t^{\mathrm{p}k}\mathbf{P}^k\hat{\mathbf{\Gamma}}_{kt}(\mathbf{P}^{kH}\mathbf{P}^k\hat{\mathbf{\Gamma}}_{kt} + N_0\mathbf{I}_{M^k})^{-1}$	$\mathbf{Y}_t^{\mathrm{p}k}\mathbf{P}_t^k\mathbf{B}_t^k(\mathbf{P}_t^{kH}\mathbf{P}_t^k\mathbf{B}_t^k + N_0\mathbf{I}_{M_t^k})^{-1}$	$\mathbf{Y}_t^{\mathrm{p}k}\mathbf{P}_t^k\ \mathrm{diag}(\eta_{tti_1}^k,\ldots,\eta_{tti_{M_t^k}}^k)$
δ_{ti}^k	$\beta_i\sigma_{\mathrm{h}}^2\left(\frac{N_0\|\mathbf{c}_{ti}^k\|^2+\sum_{j\in\mathcal{S}_k^i}\|r_{jti}^k\|^2\hat{g}_{tj}^k g_{tj}\beta_j\sigma_{\mathrm{h}}^2}{N_0\|\mathbf{c}_{ti}^k\|^2+\sum_{j\in\mathcal{S}_k}\|r_{jti}^k\|^2\hat{g}_{tj}^k g_{tj}\beta_j\sigma_{\mathrm{h}}^2}\right)$	$\beta_i\sigma_{\mathrm{h}}^2\left(\frac{N_0\|\mathbf{c}_{ti}^k\|^2+\sum_{j\in\mathcal{S}_k^i}\|r_{jti}^k\|^2 g_{tj}\beta_j\sigma_{\mathrm{h}}^2}{N_0\|\mathbf{c}_{ti}^k\|^2+\sum_{j\in\mathcal{S}_k}\|r_{jti}^k\|^2 g_{tj}\beta_j\sigma_{\mathrm{h}}^2}\right)$	$\beta_i\sigma_{\mathrm{h}}^2\left(\frac{N_0\|\mathbf{p}_i\|^2+\sum_{j\in\mathcal{S}_k^i}\|\mathbf{p}_j^H\mathbf{p}_i\|^2 g_{tj}\beta_j\sigma_{\mathrm{h}}^2}{N_0\|\mathbf{p}_i\|^2+\sum_{j\in\mathcal{S}_k}\|\mathbf{p}_j^H\mathbf{p}_i\|^2 g_{tj}\beta_j\sigma_{\mathrm{h}}^2}\right)$

Theorem 3.1 ▶ Channel Estimation in IRSA.

The channel estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ in the tth RB in the kth decoding iteration, under the three estimation schemes, namely MSBL, MMSE, and LCMMSE, is given in Table 3.2. Specifically, the estimate of the channel $\mathbf{h}_{ti}$ of the ith user is calculated as $\hat{\mathbf{h}}_{ti}^k = [\hat{\mathbf{H}}_t^k]_{:,i}$. Further, the covariance of the estimation error $\tilde{\mathbf{h}}_{ti}^k \triangleq \hat{\mathbf{h}}_{ti}^k - \mathbf{h}_{ti}$ is $\delta_{ti}^k \mathbf{I}_N$, where δ_{ti}^k is listed in Table 3.2, with $\mathbf{c}_{ti}^k = [\mathbf{C}_t^k]_{:,i}$ and $r_{jti}^k \triangleq \mathbf{p}_j^H \mathbf{c}_{ti}^k$. For MSBL, we have $\mathbf{C}_t^k \triangleq \mathbf{P}^k \mathbf{D}_t^k (\mathbf{P}^{kH} \mathbf{P}^k \mathbf{D}_t^k + N_0 \mathbf{I}_{M^k})^{-1}$, where $\mathbf{D}_t^k \triangleq \mathrm{diag}(d_{ti_1}^k, d_{ti_2}^k, \ldots, d_{ti_{M^k}}^k)$ with $d_{ti}^k = \hat{g}_{ti}^k g_{ti} \beta_i \sigma_{\mathtt{h}}^2$. For MMSE, we have $\mathbf{C}_t^k \triangleq \mathbf{P}_t^k \mathbf{B}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{B}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}$.

Proof. See Sec. 3.8.1. □

Remark: The LCMMSE estimate is composed of two components: a scaling coefficient η_{tm}^k and the post-combined received pilot signal $\mathbf{y}_{tm}^{\mathtt{p}k}$. From (3.8), we see that the received pilot signal $\mathbf{y}_{tm}^{\mathtt{p}k}$ contains pilots of other users, if pilot sequences are not orthogonal. With orthogonal pilots, $\mathbf{p}_i^H \mathbf{p}_m = 0, \forall i \neq m$, the LCMMSE estimate is

$$\hat{\mathbf{h}}_{tm}^k = \frac{g_{tm} \beta_m \sigma_{\mathtt{h}}^2}{N_0 + \|\mathbf{p}_m\|^2 g_{tm} \beta_m \sigma_{\mathtt{h}}^2} \left(g_{tm} \mathbf{h}_{tm} \|\mathbf{p}_m\|^2 + \mathbf{N}_t^{\mathtt{p}} \mathbf{p}_m \right), \tag{3.10}$$

and $\delta_{tm}^k = g_{tm} \beta_m \sigma_{\mathtt{h}}^2 N_0 / (N_0 + g_{tm} \beta_m \sigma_{\mathtt{h}}^2 \|\mathbf{p}_m\|^2)$, i.e., there is no pilot contamination, and the LCMMSE estimate coincides with the MMSE estimate. Further, if we use a codebook of orthogonal pilots and reuse it across all users, then the MMSE estimate coincides with the LCMMSE estimate. As we will see in Sec. 3.6 and Sec. 3.9.6, for any choice of the *non-orthogonal* pilots, MMSE estimation outperforms LCMMSE estimation.

Complexity: The MMSE scheme has a complexity of $\mathcal{O}(\tau^2 M_t^k)$ floating point operations (flops) since it involves inverting a $\tau \times \tau$ matrix. The MSBL scheme, with s iterations, has a complexity of $\mathcal{O}(s\tau^2 M^k)$ flops [16]. The LCMMSE scheme has the lowest complexity of $\mathcal{O}(M_t^k)$ flops since it does not need any matrix inversion.

3.4 SINR Analysis

In this section, the SINR of each user in all the RBs where it transmits data is derived, accounting for pilot contamination and channel estimation errors. Let ρ_{tm}^k denote the SINR of the mth user in the tth RB in the kth decoding iteration. Similar to (3.2), the received data signal in the tth RB and kth decoding iteration can be written as

$$\mathbf{y}_t^k = \sum\nolimits_{i \in \mathcal{S}_k} g_{ti} \mathbf{h}_{ti} x_i + \mathbf{n}_t. \tag{3.11}$$

A combining vector $\mathbf{a}_{tm}^k$ is used to decode the mth user in the tth RB and kth decoding iteration, and thus we obtain

$$\tilde{y}_{tm}^k = \mathbf{a}_{tm}^{kH} \mathbf{y}_t^k = \mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k g_{tm} x_m - \mathbf{a}_{tm}^{kH} \tilde{\mathbf{h}}_{tm}^k g_{tm} x_m + \mathbf{a}_{tm}^{kH} \sum\nolimits_{i \in \mathcal{S}_k^m} g_{ti} \mathbf{h}_{ti} x_i + \mathbf{a}_{tm}^{kH} \mathbf{n}_t, \tag{3.12}$$

where $\tilde{\mathbf{h}}_{tm}^k$ is as defined in Theorem 3.1. From the above, we see that the signal used to decode the mth user's data is composed of four terms. The term $T_1 \triangleq \mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k g_{tm} x_m$ is the useful signal component of the mth user; the term $T_2 \triangleq \mathbf{a}_{tm}^{kH} \tilde{\mathbf{h}}_{tm}^k g_{tm} x_m$ is contributed by the channel estimation error $\tilde{\mathbf{h}}_{tm}^k$ of the mth user; the term $T_3 \triangleq \sum_{i \in \mathcal{S}_k^m} \mathbf{a}_{tm}^{kH} \mathbf{h}_{ti} g_{ti} x_i$ captures the inter-user interference from the users who have also transmitted in the tth RB and have not yet been decoded up to the kth decoding iteration; and the last term $T_4 \triangleq \mathbf{a}_{tm}^{kH} \mathbf{n}_t$ is the additive noise component.

In order to compute the SINR, the power in the received signal is calculated conditioned on the knowledge of the estimates [49]. Since MMSE estimates are employed, all three estimates are uncorrelated with the channel estimation error, and thus T_2 is uncorrelated with T_1. The additive noise is uncorrelated with the signal, and since the users' data signals are independent, T_3 is uncorrelated with the other terms. Thus, all four components in the received signal are uncorrelated and the total power is the sum of the powers of the individual components

$$\mathbb{E}_{\mathbf{z}}[|\tilde{y}_{tm}^k|^2] = \sum\nolimits_{i=1}^{4} \mathbb{E}_{\mathbf{z}}[|T_i|^2], \tag{3.13}$$

where $\mathbf{z}$ contains the channel estimates of the users. The SINR for all the users is now presented.

Theorem 3.2 ▶ SINR in Multi-Cell IRSA.

The signal to interference plus noise ratio (SINR) achieved by the mth user in the tth RB in the kth decoding iteration can be written as

$$\rho_{tm}^{k} = \frac{\mathtt{Gain}_{tm}^{k}}{N_0/P + \mathtt{MUI}_{tm}^{k} + \mathtt{Est}_{tm}^{k}}, \ \forall m \in \mathcal{S}_k, \tag{3.14}$$

where $\mathtt{Gain}_{tm}^{k}$ represents the useful signal power of the mth user, $\mathtt{MUI}_{tm}^{k}$ represents the multi-user interference power of other users, and $\mathtt{Est}_{tm}^{k}$ represents the interference power caused due to the channel estimation errors. Under MMSE and LCMMSE channel estimation, these can be expressed as

$$\mathtt{Gain}_{tm}^{k} = g_{tm}\frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}|^2}{\|\mathbf{a}_{tm}^{k}\|^2}, \ \mathtt{MUI}_{tm}^{k} = \sum_{i\in\mathcal{S}_k^m} g_{ti}\frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^{k}|^2}{\|\mathbf{a}_{tm}^{k}\|^2}, \tag{3.15}$$

$$\mathtt{Est}_{tm}^{k} = \sum_{i\in\mathcal{S}_k} g_{ti}\delta_{ti}^{k}. \tag{3.16}$$

With the sparsity-based scheme, the SINR denominator contains an additional term, $\mathtt{FNU}_{tm}^{k}$, which represents the interference power caused due to false negative users. The corresponding terms with MSBL can be expressed as

$$\mathtt{Gain}_{tm}^{k} = \hat{g}_{tm}^{k} g_{tm}\frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}|^2}{\|\mathbf{a}_{tm}^{k}\|^2}, \ \mathtt{MUI}_{tm}^{k} = \sum_{i\in\mathcal{S}_k^m} \hat{g}_{ti}^{k} g_{ti}\frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^{k}|^2}{\|\mathbf{a}_{tm}^{k}\|^2}, \tag{3.17}$$

$$\mathtt{Est}_{tm}^{k} = \sum_{i\in\mathcal{S}_k} \hat{g}_{ti}^{k} g_{ti}\delta_{ti}^{k}, \ \mathtt{FNU}_{tm}^{k} = \sum_{i\in\mathcal{S}_k^m} (1-\hat{g}_{ti}^{k}) g_{ti}\beta_i\sigma_{\mathtt{h}}^{2}. \tag{3.18}$$

Here, the estimates $\hat{\mathbf{h}}_{ti}^{k} = [\hat{\mathbf{H}}_{t}^{k}]_{:,i}$ and the error variances δ_{ti}^{k} are obtained from Theorem 3.1 for all the three schemes.

Proof. See Sec. 3.8.2. □

The SINR expression derived in Theorem 3.2 is applicable to any arbitrary receive

combining scheme given by the matrix $\mathbf{A}_t^k$, with $\mathbf{a}_{tm}^k = [\mathbf{A}_t^k]_{:,m}$. When regularized zero forcing (RZF) combining is used, the combining matrix is

$$\mathbf{A}_t^k = \hat{\mathbf{H}}_t^k (\hat{\mathbf{H}}_t^{kH} \hat{\mathbf{H}}_t^k + \lambda \mathbf{I}_{M_t^k})^{-1}, \tag{3.19}$$

where λ is the regularization parameter. The SINR with RZF can be computed by substituting the columns of the above matrix into (3.14). We now describe two popular combining schemes, which are special cases of RZF, in which simpler expressions for the SINR can be computed.[6] The expressions are written for MMSE/LCMMSE, and can be extended to MSBL as detailed in Theorem 3.2.

3.4.1 Maximal Ratio Combining (MRC)

MRC is obtained from RZF as $\lambda \to \infty$ and the combining matrix becomes $\mathbf{A}_t^k = \hat{\mathbf{H}}_t^k$. Thus $\mathbf{a}_{tm}^k = \hat{\mathbf{h}}_{tm}^k$, and SINR can be computed as

$$\rho_{tm}^k = \frac{P g_{tm} \|\hat{\mathbf{h}}_{tm}^k\|^2}{N_0 + \sum_{i \in \mathcal{S}_k} P g_{ti} \delta_{ti}^k + \sum_{i \in \mathcal{S}_k^m} P g_{ti} \frac{|\hat{\mathbf{h}}_{tm}^{kH} \hat{\mathbf{h}}_{ti}^k|^2}{\|\hat{\mathbf{h}}_{tm}^k\|^2}}. \tag{3.20}$$

3.4.2 Zero Forcing (ZF)

The RZF combiner reduces to the ZF combiner as $\lambda \to 0$. The inverse of the gram-matrix of the channel estimates exists with probability one when $N \geq M_t^k$ and $\hat{\mathbf{H}}_t^k$ has full column rank.[7] Hence, we can compute the combining matrix as $\mathbf{A}_t^k = \hat{\mathbf{H}}_t^k (\hat{\mathbf{H}}_t^{kH} \hat{\mathbf{H}}_t^k)^{-1}$. Using the above, it is easy to show that the SINR expression simplifies as [49]

$$\rho_{tm}^k = \frac{P g_{tm}}{(N_0 + \sum_{i \in \mathcal{S}_k} P g_{ti} \delta_{ti}^k)[(\hat{\mathbf{H}}_t^{kH} \hat{\mathbf{H}}_t^k)^{-1}]_{mm}}. \tag{3.21}$$

[6] In this chapter, we do not consider the MMSE combiner, which is a special case of RZF combining [49].

[7] We note that the condition $N \geq M_t^k$ is not hard to satisfy in IRSA. For example, with $L = 2, 3, 4$, each RB will be occupied by $6, 9, 12$ users on an average, respectively, if the average repetition factor is $\bar{d} = 3$. Thus any N greater than, say, 16 would be sufficient to decode the users in most RBs.

Note that the third term in the denominator of (3.20) has been suppressed with ZF combining. However, due to pilot contamination, the term $[(\hat{\mathbf{H}}_t^{kH}\hat{\mathbf{H}}_t^k)^{-1}]_{mm}$ may contain contributions from the channels of all users. As a consequence, the gram matrix could be ill-conditioned, and the denominator term could be large. Thus, the pilot length, which determines the pilot contamination incurred, is crucial in comparing the performance obtained by the combining schemes. The system throughput can now be calculated from the above SINR expressions via the decoding model described in Sec. 3.2.1, and is described in Algorithm 2 for MMSE/LCMMSE. For MSBL, the initial step in each RB instead consists of finding $M^k = |\mathcal{S}_k|$, and $\mathbf{P}^k = [\mathbf{P}]_{:,\mathcal{S}_k}$. We also estimate $\{g_{ti}\}$ and $\{\mathbf{h}_{ti}\}$ via Algorithm 1 before finding the SINR.

Algorithm 2: Performance Evaluation of IRSA

Input: $\tau, N, T, M, N_0, \mathbf{G}, \mathbf{P}, \{\mathbf{Y}_t^{\mathsf{p}}\}_{t=1}^T, \{\mathbf{y}_t\}_{t=1}^T, k_{\max}$

Initialize: $\mathcal{S}_1 = [M]$, $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$

for $k = 1, 2, \ldots, k_{\max}$ **do**

  for $t = 1, 2, \ldots, T$ **do**

    Find $M_t^k = |\mathcal{G}_t \cap \mathcal{S}_k|$, $\mathbf{P}_t^k = [\mathbf{P}]_{:,\mathcal{G}_t \cap \mathcal{S}_k}$, $\mathbf{Y}_t^{\mathsf{p}k}, \mathbf{y}_t^k$

    Compute $\hat{\mathbf{h}}_{ti}^k,\ \forall i \in \mathcal{S}_k$ via Theorem 3.1

    Evaluate the SINR ρ_{ti}^k via Theorem 3.2

    If $\rho_{ti}^k \geq \gamma_{\text{th}}$, remove user i from $\mathcal{S}_k$ and perform IC in all RBs where $g_{ti} = 1$

  end

end

Output: $\mathcal{T} = (M - |\mathcal{S}_{k_{\max}}|)/T,\ \mathsf{PLR} = |\mathcal{S}_{k_{\max}}|/M$

3.4.3 SINR in the Massive MIMO Regime

Before proceeding with the analysis of the throughput, we briefly discuss the SINR in the massive MIMO regime, which helps us in interpreting the SINR expressions. We note that the results presented in Sec. 3.6 hold true for any N. However, when N is large, a simpler expression for SINR with MRC can be obtained as follows.

Lemma 1. *As the number of antennas N gets large, the SINR with MRC converges almost surely to*

$$\overline{\rho}_{tm}^{k} = \frac{N\mathtt{Sig}_{tm}^{k}}{\epsilon_{tm}^{k}\left(N_0/P + \mathtt{IntNC}_{tm}^{k}\right) + \mathtt{IntC}_{tm}^{k}}, \tag{3.22}$$

where $\mathtt{Sig}_{tm}^{k}$ is the desired signal, $\mathtt{IntNC}_{tm}^{k}$ represents the non-coherent interference, and $\mathtt{IntC}_{tm}^{k}$ represents the coherent interference. Each of these can be found in Table 3.3. Here, δ_{tm}^{k} and $\mathbf{c}_{tm}^{k}$ are obtained from Theorems 3.1 and 3.2, respectively, for the three estimation schemes.

Proof. See Sec. 3.8.3. □

Table 3.3: Deterministic equivalent approximation to the SINR.

	Sparsity-based estimation with MSBL	MMSE	LCMMSE
ϵ_{tm}^{k}	$N_0\|\mathbf{c}_{tm}^{k}\|^2 + \sum_{i\in\mathcal{S}_k} g_{ti}\beta_i\sigma_{\mathtt{h}}^2\|\mathbf{c}_{tm}^{kH}\mathbf{p}_i\|^2$	$N_0\|\mathbf{c}_{tm}^{k}\|^2 + \sum_{i\in\mathcal{S}_k} g_{ti}\beta_i\sigma_{\mathtt{h}}^2\|\mathbf{c}_{tm}^{kH}\mathbf{p}_i\|^2$	$N_0\|\mathbf{p}_m\|^2 + \sum_{i\in\mathcal{S}_k} g_{ti}\beta_i\sigma_{\mathtt{h}}^2\|\mathbf{p}_m^{H}\mathbf{p}_i\|^2$
$\mathtt{Sig}_{tm}^{k}$	$\hat{g}_{tm}^{k} g_{tm}(\epsilon_{tm}^{k})^2$	$g_{tm}(\epsilon_{tm}^{k})^2$	$g_{tm}\beta_m^2\sigma_{\mathtt{h}}^4\|\mathbf{p}_m\|^4$
$\mathtt{IntNC}_{tm}^{k}$	$\hat{g}_{tm}^{k} g_{tm}\delta_{tm}^{k} + \sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i\sigma_{\mathtt{h}}^2$	$g_{tm}\delta_{tm}^{k} + \sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i\sigma_{\mathtt{h}}^2$	$g_{tm}\delta_{tm}^{k} + \sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i\sigma_{\mathtt{h}}^2$
$\mathtt{IntC}_{tm}^{k}$	$N\sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i^2\sigma_{\mathtt{h}}^4\|\mathbf{c}_{tm}^{kH}\mathbf{p}_i\|^2$	$N\sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i^2\sigma_{\mathtt{h}}^4\|\mathbf{c}_{tm}^{kH}\mathbf{p}_i\|^2$	$N\sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i^2\sigma_{\mathtt{h}}^4\|\mathbf{p}_m^{H}\mathbf{p}_i\|^2$

Remark: $\mathtt{IntNC}_{tm}^{k}$ arises due to channel estimation errors and is independent of N, while $\mathtt{IntC}_{tm}^{k}$ is due to pilot contamination and increases linearly with N. Further, since $\overline{\rho}_{tm}^{k}$ is independent of the fading states of each user, it assures successful recovery of packets with high probability if $\overline{\rho}_{tm}^{k} \gg \gamma_{\text{th}}$. Similarly, the packet will not be decodable with probability close to 1 if $\overline{\rho}_{tm}^{k} \ll \gamma_{\text{th}}$. However, it turns out that in order to characterize the throughput of IRSA, it is necessary to capture the statistics of the SINR when $\overline{\rho}_{tm}^{k} \approx \gamma_{\text{th}}$. The small fluctuations in ρ_{tm}^{k} around $\overline{\rho}_{tm}^{k}$ due to fading, and the resulting probability of packet decoding error, need to be calculated accurately. Hence, the calculation of the statistics of the SINR using (3.14) is vital to find the throughput of IRSA. We address this in the next section.

3.5 Theoretical Analysis of Throughput

Density Evolution (DE) analysis has been applied to characterize the asymptotic performance of message passing-based decoding on graphs for low density parity check codes [103] and IRSA [7]. In this section, the representation of IRSA decoding as a bipartite graph is discussed first. Then the graph perspective distributions are defined, the failure probabilities are derived, and finally, the asymptotic throughput of IRSA is characterized. It is assumed that users perform path loss inversion-based power control. We note that a closed form expression for the throughput cannot be derived even for the most basic variant of IRSA due to the underlying graph structure [7].[8] Hence, we need to resort to DE, which provides an iterative recipe to compute the throughput.

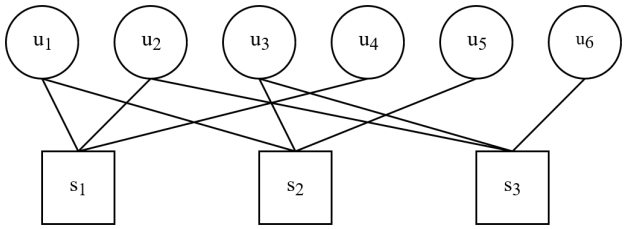


Figure 3.1: IRSA represented as a bipartite graph.

SIC-based decoding can be viewed as message passing on a bipartite graph [7], and thus IRSA, which uses SIC decoding, can be decoded on graphs. A typical IRSA frame can be represented as a bipartite graph, which is made up of M user nodes (one node for each user), T RB nodes (one node for each RB), and the edges between them. An edge connects a user node to an RB node if and only if that user has transmitted a packet in that corresponding RB. For example, in Fig. 3.1, there will be an edge between user node u_1 and RB node s_1 if and only if user u_1 has transmitted a packet replica in RB s_1. During decoding, edges that connect to users whose SINR is above a threshold are removed from each RB. Each decoding iteration consists of several intra-RB SIC and inter-RB SIC steps. Once an SIC step is performed, the corresponding edge in the bipartite graph is removed. Thus, the edge between user node u_1 and RB node s_1 is removed if the user u_1 is decoded

[8]This is elaborated in Sec. 3.9.3.

in any of the RBs in which the user has transmitted a packet. Decoding is successful if, at the end of the SIC process, all edges in the graph get removed. A decoding failure is declared if not all edges have been removed or no new edge is removed from the graph in two consecutive iterations.

3.5.1 Graph Perspective Degree Distributions

The total number of packets transmitted by a user in a given frame is referred to as the repetition factor of that user. It is equal to the degree of the user node at the start of decoding, and is the same as the number of edges connected to that user node in the bipartite graph representation of SIC decoding. The *node-perspective user degree distribution* is defined as the set of probabilities $\{\phi_d\}_{d=2}^{d_{\max}}$, where ϕ_d represents the probability that a user has a repetition factor d with $d_{\max}$ being the maximum number of RBs in which any user is allowed to transmit. Here, ϕ_d is nonzero for $d \geq 2$ since each user transmits at least 2 packets in IRSA.

The total number of packets received in an RB is referred to as the collision factor of that RB. It is equal to the degree of the RB node at the start of decoding, which is the number of edges connected to that RB node. The *node-perspective RB degree distribution* is defined as the set of probabilities $\{\psi_c\}_{c=0}^{M}$, where ψ_c represents the probability that an RB has a collision factor c. The polynomial representations of the node-perspective user and RB degree distributions are

$$\phi(x) = \sum\nolimits_{d=2}^{d_{\max}} \phi_d x^d, \quad \psi(x) = \sum\nolimits_{c=0}^{M} \psi_c x^c, \tag{3.23}$$

respectively. The corresponding *edge-perspective user and RB degree distributions* are defined as $\lambda(x) = \sum_{d=2}^{d_{\max}} \lambda_d x^{d-1} = \phi'(x)/\phi'(1)$; $\xi(x) = \sum_{c=1}^{M} \xi_c x^{c-1} = \psi'(x)/\psi'(1)$, respectively, where $\lambda_d = d\phi_d/\phi'(1)$ represents the probability that an edge is connected to a user with repetition factor d and $\xi_c = c\psi_c/\psi'(1)$ represents the probability that an edge is connected to an RB with collision factor c.

The *input load* L of the system is defined as the ratio of the number of users to the

number of RBs, $L \triangleq M/T$. The average repetition factor is $\bar{d} = \phi'(1) = \sum_d d\phi_d$ and the average collision factor is $\bar{c} = \psi'(1) = \sum_c c\psi_c$, making the load $L = M/T = \bar{c}/\bar{d}$. Since $\bar{c} = L\bar{d}$, fixing the load and the node-perspective user degree distribution fixes the other three degree distributions as well. The probability that a generic user, from a total of M users, transmits within an RB is $\bar{c}/M$. Since the users transmit their packets independently of each other, ψ_c follows a binomial distribution. Thus, the coefficients of the polynomials representing the node and edge-perspective RB degree distributions are respectively given by

$$\psi_c = \binom{M}{c} \left(\frac{\bar{c}}{M}\right)^c \left(1 - \frac{\bar{c}}{M}\right)^{M-c}, \tag{3.24a}$$

$$\text{and} \quad \xi_c = \binom{M-1}{c-1} \left(\frac{\bar{c}}{M}\right)^{c-1} \left(1 - \frac{\bar{c}}{M}\right)^{M-c}. \tag{3.24b}$$

For a fixed $L = M/T$, as $M, T \to \infty$, the node-perspective and edge-perspective RB degree distributions, which are binomial, become Poisson distributed [105]:

$$\psi_c = \frac{(\bar{c})^c \exp(-\bar{c})}{c!} \text{ and } \xi_c = \frac{(\bar{c})^{c-1} \exp(-\bar{c})}{(c-1)!}. \tag{3.25}$$

We now use the degree distributions defined above to find the failure probabilities in the next subsection.

3.5.2 Failure Probabilities

In the case of a decoding failure, failure messages are exchanged along the edges between the user and the RB nodes. The probability that an edge carries a failure message from an RB node to a user node in the ith iteration is denoted by p_i. The probability that an edge carries a failure message from a user node to an RB node in the ith iteration is denoted by q_i.

The failure probability q_i is calculated using the edge-perspective user degree distribution as

$$q_i = \sum\nolimits_{d=2}^{d_{\max}} \lambda_d q_i^{(d)} = \sum\nolimits_{d=2}^{d_{\max}} \lambda_d p_{i-1}^{d-1} = \lambda(p_{i-1}). \tag{3.26}$$

Here, $q_i^{(d)}$ is the probability that an edge carries a failure message in the ith iteration given that it is connected to a user node with repetition factor d. The edges carry a failure message from a user if and only if all the other $d-1$ incoming edges to that user carry failure messages in the previous iteration, i.e., $q_i^{(d)} = p_{i-1}^{d-1}$.

The failure probability p_i is calculated using the edge-perspective RB degree distribution as

$$p_i = \sum\nolimits_{c=1}^{M} \xi_c p_i^{(c)} \xrightarrow{M\to\infty} p_i = \sum\nolimits_{c=1}^{\infty} \xi_c p_i^{(c)}, \tag{3.27}$$

where $p_i^{(c)}$ is the probability that an edge carries a failure message in the ith iteration given that it is connected to an RB node with collision factor c. DE is applicable as M and $T \to \infty$ with $L = M/T$ kept fixed [7]. Hence the above probability is computed as an infinite summation.

In the SINR threshold model, decoding failure happens at an RB node if the SINR of all users who have transmitted in that RB and have not yet been decoded is below the SINR threshold. This constitutes a failure message from the RB node [17]. In order to determine $p_i^{(c)}$, any one of the c packets is considered to be a reference packet, which can get decoded with a combination of intra-RB and inter-RB SIC. Separating the intra-RB and inter-RB SIC, $p_i^{(c)}$ can be evaluated as

$$p_i^{(c)} = 1 - \sum\nolimits_{r=1}^{c} \theta_r \binom{c-1}{r-1} q_i^{r-1} (1-q_i)^{c-r}. \tag{3.28}$$

Here, θ_r denotes the probability that the reference packet gets decoded in the current decoding iteration starting from degree r using only intra-RB SIC, and $\binom{c-1}{r-1} q_i^{r-1}(1-q_i)^{c-r}$ denotes the probability that the collision factor of the RB node reduces from c to r using

only inter-RB SIC [8]. The evaluation of θ_r is discussed in Sec. 3.5.4. Substituting for $p_i^{(c)}$ from (3.28), we obtain p_i as a function of q_i:

$$p_i = 1 - \sum_{c=1}^{\infty} \sum_{r=1}^{c} \xi_c \theta_r \binom{c-1}{r-1} q_i^{r-1} (1-q_i)^{c-r}. \tag{3.29}$$

Thus, we compute the failure probabilities p_i and q_i recursively from each other, as observed in (3.26) and (3.29).

3.5.3 Evaluation of Throughput

We now describe the evaluation of the throughput. Substituting for ξ_c from (3.25), we can simplify (3.29) to

$$p_i = 1 - e^{-\bar{c} q_i} \sum_{r=1}^{\infty} \theta_r \frac{(\bar{c} q_i)^{r-1}}{(r-1)!} \triangleq f(q_i). \tag{3.30}$$

Thus, $q_i = \lambda(p_{i-1})$ and $p_i = f(q_i)$ are calculated alternately as functions of each other as seen in (3.26) and (3.30). The procedure can be initialized with either $q_0 = 1$ or $p_0 = f(1)$.

The failure probability at the end of decoding is $p_\infty = \lim_{i\to\infty} p_i$ and $(p_\infty)^d$ is the probability that a packet transmitted from a user with repetition factor d does not get decoded at the receiver. Therefore, the asymptotic packet loss rate (PLR), which is the fraction of packets that are not decoded at the BS, is calculated as

$$\mathsf{PLR} = \phi(p_\infty) = \sum\nolimits_{d=2}^{d_{\max}} \phi_d (p_\infty)^d. \tag{3.31}$$

The asymptotic throughput of the system can now be obtained from the asymptotic PLR as[9]

$$\mathcal{T} = L(1 - \mathsf{PLR}). \tag{3.32}$$

[9]The DE process yields an iterative recipe to obtain the asymptotic throughput. While DE is an analytical approach, it cannot be used to find a closed-form relationship between the system parameters and the throughput.

The iterations $p_i = f(\lambda(p_{i-1}))$ converge asymptotically to $p_\infty = 0$ if the system load $L < L^*$ [7]. Here, L^* is called the *inflection load* of the system: for any $L \geq L^*$, the system becomes interference limited and the PLR does not converge to 0 as L increases. Thus, for $L < L^*$, $p_\infty = 0$ and therefore the asymptotic PLR $= 0$, and the throughput equals L. For $L \geq L^*$, the throughput decreases monotonically with L.

The crucial step in the evaluation of the throughput lies in the computation of θ_r, which we now describe.

3.5.4 Characterization of θ_r

We now describe a procedure to evaluate the success probability θ_r, which is the probability of decoding the reference packet in an RB with degree r via intra-RB SIC only. There are r users whose packets have not yet been decoded in the RB. The reference packet can get decoded in any of the intra-RB SIC steps. The packets with SINR higher than that of the reference packet get decoded first. Further, the reference packet can only be decoded if decoding has been successful for higher SINR packets, i.e., if they satisfied SINR $\geq \gamma_{\text{th}}$ as well. Thus, θ_r is the joint probability that the reference packet and the packets with higher SINRs all get decoded.

Clazzer et al. [17] evaluate θ_r as the probability "$D(r)$" under a Rayleigh fading SISO channel setup with a perfect CSI assumption. The same method cannot be applied here, since we consider MIMO Rayleigh fading and account for imperfect CSI due to pilot contamination and channel estimation errors. In particular, in a MIMO setup, it is possible that multiple users' SINR simultaneously exceed the decoding threshold. Further, their work is limited to the case where the decoding threshold γ_{th} is such that only one user can be decoded in any decoding iteration, while we make no such assumptions.

Since θ_r is evaluated based on the SINR of multiple users in a *single RB*, we consider only one RB wherein r users have transmitted their packets. These users are decoded via only intra-RB iterations since there is only a single RB under consideration. Let the set of users who have not yet been decoded in the first $k-1$ intra-RB decoding iterations be

denoted by $\mathcal{S}_k$, and $\mathcal{S}_k^m \triangleq \mathcal{S}_k \setminus \{m\}$, with $\mathcal{S}_1 = [r]$.[10] In each intra-RB decoding iteration, a single user with the highest SINR is decoded if their SINR $\geq \gamma_{\text{th}}$.

The SINR of the mth user in the kth intra-RB decoding iteration, ρ_m^k, is calculated as seen before in Theorem 3.2. Specifically, when users are only decoded via intra-RB SIC within one RB, we obtain the SINR as

$$\rho_m^k = \frac{|\mathbf{a}_m^{kH}\hat{\mathbf{h}}_m^k|^2}{\|\mathbf{a}_m^k\|^2(N_0/P + \sum_{i\in\mathcal{S}_k}\delta_i^k) + \sum_{i\in\mathcal{S}_k^m}|\mathbf{a}_m^{kH}\hat{\mathbf{h}}_i^k|^2}. \tag{3.33}$$

Here, δ_i^k is the error variance of the ith user in the kth intra-RB decoding iteration, $\hat{\mathbf{h}}_m^k$ is the channel estimate of the mth user, both obtained from Theorem 3.1, and $\mathbf{a}_m^k$ is the combining vector for the mth user.[11] Let $\rho_{\max}^k$ denote the SINR of the user with the highest SINR in the kth intra-RB decoding iteration, calculated as $\rho_{\max}^k = \max_{m\in\mathcal{S}_k}\ \rho_m^k$. Let s be the index of the intra-RB decoding iteration in which the reference packet is decoded, with $1 \leq s \leq r$. Thus, θ_r is calculated as

$$\theta_r = \Pr(\rho_{\max}^1 \geq \gamma_{\text{th}}, \rho_{\max}^2 \geq \gamma_{\text{th}}, \ldots, \rho_{\max}^s \geq \gamma_{\text{th}}). \tag{3.34}$$

Recall that the reference packet is tagged uniformly at random from the users. With path loss inversion based power control, users have identical channel statistics, and thus, θ_r is independent of which packet is tagged as the reference packet.

The computation of the success probability θ_r is involved because there is no clear relation between the peak SINRs across decoding iterations. Also, the channel estimates of different users are correlated, across both the user index and the decoding iteration index, making it difficult to use order statistics. Further, θ_r is dependent on a large number of random channel vectors, the order statistics of the peak SINRs, and the pilot sequences of all the users. As a consequence, θ_r cannot be found in closed form, and needs to be

[10]The set $\mathcal{S}_k$ as defined here is a slight abuse of notation. In Sec. 3.3, the set $\mathcal{S}_k$ consisted of users being decoded via both intra-RB and inter-RB iterations, whereas here, $\mathcal{S}_k$ consists of users being decoded via only intra-RB iterations.

[11]Since the decoding process with intra-RB SIC involves only the RB in consideration, the RB index and the APM are dropped in this section.

empirically evaluated. However, we present three approximations to θ_r, which are valid when perfect CSI is available at the BS, i.e., there is no pilot contamination or estimation errors. The assumptions are made for analytical tractability. These lead to interpretable expressions for the SINR and θ_r, and provide upper bounds on the throughput with estimated CSI.

Theorem 3.3 ▶ Calculation of the Success Probability θ_r.

When perfect CSI is available at the BS, and MRC is used for decoding, θ_1 is given by

$$\theta_1 = \Gamma_{\text{inc}}(N, \rho_0^{-1}\gamma_{\text{th}})/\Gamma(N), \tag{3.35}$$

where $\rho_0 \triangleq P\sigma_{\text{h}}^2/N_0$, $\Gamma_{\text{inc}}(s,x) = \int_x^\infty t^{s-1}\, e^{-t}\, \mathrm{d}t$ is the upper incomplete gamma function, and $\Gamma(s)$ is the ordinary gamma function. For $r \geq 2$, the SINR with MRC and large N can be computed as $\rho_m^k = N(\rho_0^{-1} + N\sum_{i\in\mathcal{S}_k^m} t_{mi})^{-1}$, where $t_{mi} \triangleq |\mathbf{h}_m^H\mathbf{h}_i|^2/(\|\mathbf{h}_m\|^2\|\mathbf{h}_i\|^2)$. With $t_0 \triangleq \gamma_{\text{th}}^{-1} - N^{-1}\rho_0^{-1}$, θ_2 can be calculated as

$$\theta_2 = \mathbb{1}\{t_0 \geq 1\} + (1-(1-t_0)^N)\mathbb{1}\{0 \leq t_0 \leq 1\}. \tag{3.36}$$

Three approximations to θ_r for $r \geq 3$ and large N are described below. Approximating $\rho_{\max}^1$ as ρ_1^1, and assuming u_m as i.i.d. Gamma distributed with shape $r-1$ and rate N, we obtain the *Gamma* approximation:

$$\textit{Gamma:} \quad \theta_r = 1 - \Gamma_{\text{inc}}(r-1, Nt_0)/\Gamma(r-1). \tag{3.37}$$

Approximating $\rho_{\max}^1 = \rho_1^1$ and $u_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}((r-1)\mu_N, (r-1)\sigma_N^2)$, where $\mu_N \triangleq (N+1)^{-1}$, and $\sigma_N^2 \triangleq N(N+1)^{-2}(N+2)^{-1}$, we obtain the *Normal* approximation:

$$\textit{Normal:} \quad \theta_r = 1 - \mathcal{Q}\left(\frac{t_0 - (r-1)\mu_N}{\sqrt{r-1}\sigma_N}\right), \tag{3.38}$$

where $\mathcal{Q}(\cdot)$ is the standard Normal Q-function. Finally, in the *Deterministic* approximation, the SINR becomes $\rho_m^k = N/(\rho_0^{-1} + r - k)$, and θ_r becomes

$$\textit{Deterministic:} \quad \theta_r = \mathbb{1}\{r \leq \lfloor N/\gamma_{\text{th}} - \rho_0^{-1} + 1 \rfloor\}. \tag{3.39}$$

Proof. See Sec. 3.8.4. □

Remark: The above approximations provide closed form expressions for θ_r and are valid when N is large [106]. The first two approximations have SINRs that are obtained by applying the theory of deterministic equivalents to only the norms of the channels, and yields an SINR that is affected only by the randomness in the multi-user interference components. This is supported by the fact that the interference components converge to their deterministic equivalents slower than the norms converge to their deterministic equivalents [106]. The deterministic approximation follows directly from Lemma 1, where the SINR is a deterministic quantity, and hence θ_r is a binary function of r. With finite number of antennas, due to small scale fading, the SINR of the users vary around this approximate SINR. These variations affect the value of θ_r, and are not captured by the deterministic approximation, even though we obtain simple closed form expressions for it. As a consequence, the throughput computed using the deterministic approximation can be far from the actual throughput in certain regimes and close to the actual throughput in other regimes, as will be seen in Sec. 3.6.1.

3.6 Numerical Results

In this section, the previously derived SINR analysis is used to evaluate the throughput of IRSA with estimated channels via Monte Carlo simulations, and provide insights into the dependence of the system performance on the various system parameters. In each simulation, independent realizations of the user locations, the APM, and the fades experienced by the users are generated. The throughput for each simulation is calculated as described in Sec. 3.2.1, and the effective system throughput $\mathcal{T}$ is calculated by averaging

over the simulations.

The results in this section are for $T = 50$ RBs, $N_s = 10^3$ Monte Carlo runs, $\lambda = 10^{-2}$, $\alpha = 3.76$, $\sigma_{\mathtt{h}}^2 = 1$, SINR threshold $\gamma_{\text{th}} = 10$, MSBL threshold $\gamma_{\text{pr}} = 10^{-6}$, cell radius $r_{\max} = 1000$m, and reference distance $r_0 = 100$m [49]. The number of users contending for the T RBs is computed based on the load L as $M = \lfloor LT \rceil$. The soliton distribution [72] with $d_{\max} = 27$ maximum repetitions is used to generate the repetition factor d_m for the mth user, whose access pattern is formed by uniformly randomly choosing d_m RBs from T RBs [7]. The APM is formed by stacking the pattern vectors of all the users. The location of each user is uniformly sampled from within a cell of radius $r_{\max}$ centered at the BS. The path loss coefficient is calculated as $\beta_m = (r_m/r_0)^{-\alpha}$ where r_m is the radial distance of the mth user from the BS. The signal to noise ratio (SNR) for the mth user is calculated as $P\sigma_{\mathtt{h}}^2\beta_m/N_0$. The received SNR of a user at the edge of the cell at the BS is termed as the *cell edge SNR*, and is denoted by SNR_{edge}. The power levels of all users is chosen such that the signal from a user at a distance $r_{\max}$ from the BS is received at SNR_{edge}. This ensures that all users' signals are received at an SINR that at least SNR_{edge} on average, in *singleton* RBs. If $\text{SNR}_{\text{edge}} \geq \gamma_{\text{th}}$, i.e., it is such that the cell edge user's signal is decodable, then all users' signals are decodable with high probability in singleton RBs. The power levels of users is set to $P = P^{\mathtt{p}} = 20$ dBm [49] and N_0 is chosen such that the cell edge SNR is 10 dB, unless otherwise stated.[12] The pilot sequence for each user is generated as $\mathbf{p}_m \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_\tau, P^{\mathtt{p}}\mathbf{I}_\tau)$. The effect of different pilot sequences is studied in Sec. 3.9.6.

Fig. 3.2 shows the effect of pilot length on the system throughput at different L under the three estimation schemes, with $N = 16$. MMSE scheme performs the best and reaches the optimal throughputs of $\mathcal{T} = L$ for very low pilot lengths. MSBL scheme achieves the optimal throughputs for $L = 1, 2, 3$ at $\tau = 4, 8, 12$ respectively, and beyond that, the performance is the same as that of MMSE. This shows that with a few additional pilot symbols, we can do away with the assumption of knowing the APM and path loss

[12] We consider equal pilot and data power for simplicity. Via simulations, we have observed that pilot power boosting can yield good improvement in the throughput, especially at cell edge SNRs close to 0 dB. This is elaborated in Sec. 3.9.4.

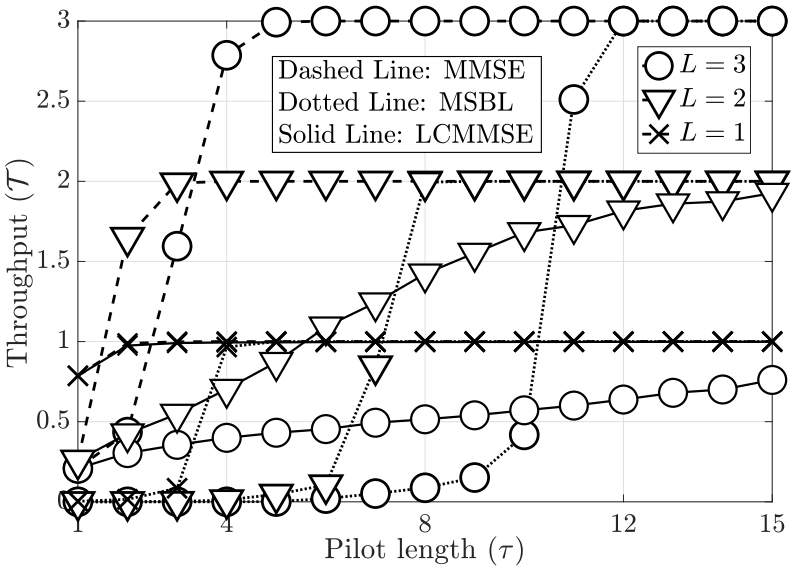


Figure 3.2: Comparison between MMSE, MSBL, and LCMMSE schemes: Effect of τ.

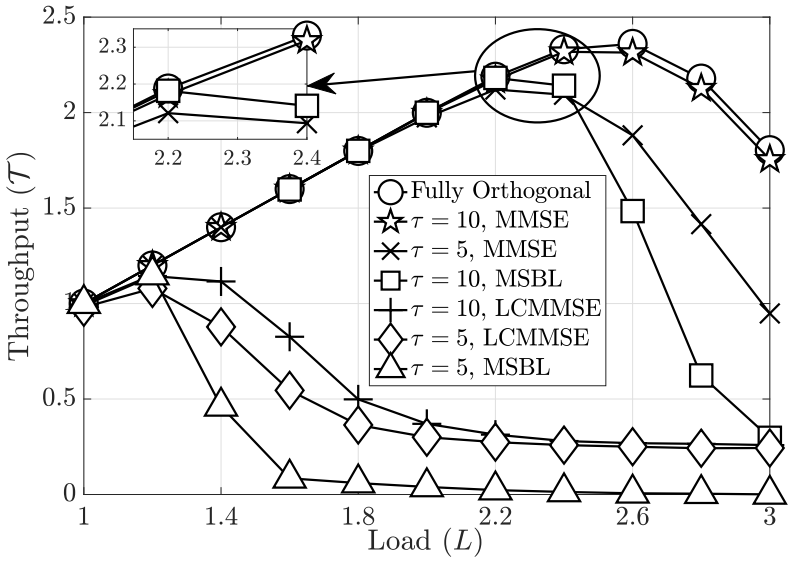


Figure 3.3: Comparison between MMSE, MSBL, and LCMMSE schemes: Effect of L.

coefficients. LCMMSE scheme matches MMSE for $L = 1$ and for higher L, it needs a lot more pilot symbols. This is because of both pilot contamination and low quality channel estimates. In Fig. 3.3, we observe the effect of increase in L on the throughput. At $L = 1, 2, 3$, the number of users in the system are $50, 100, 150$, respectively. Thus, the use of orthogonal pilots would require $\tau \geq 50, 100, 150$ for $L = 1, 2, 3$, respectively. Specifically, at $L = 2$, there are $M = 100$ users in the system, and employing orthogonal pilots would require $\tau \geq 100$. But the optimal throughput of $\mathcal{T} = L = 2$ is achieved under all the three estimation schemes with a very low pilot length of $\tau = 10$, which is a $10\times$ reduction in the pilot length. This is because only a small subset of users transmit in any RB in IRSA. Thus, we can achieve the best performance in IRSA with far fewer pilot symbols than using orthogonal pilots. Finally, under all the three schemes, we can achieve $\mathcal{T} \geq 1$, which is the maximum throughput achievable under perfectly coordinated orthogonal access, i.e., grant-based orthogonal access. This shows the utility of using IRSA as a GFRA protocol for mMTC, especially due to it's high performance at medium to high L. To summarize, the pilot length has a significant impact on the performance of IRSA and yields near-optimal throughputs at significantly lower pilot lengths than that required for orthogonal pilot transmission. The drop in $\mathcal{T}$ at low pilot lengths under estimated channels underscores the importance of accounting for the effect of imperfect CSI in analyzing the performance of IRSA.

We focus on MMSE/LCMMSE hereafter in order to avoid clutter in the plots, since MSBL matches the performance of MMSE with slightly higher τ. In Fig. 3.4, we investigate the effect of L, τ_c and γ_{th} on the achievable rate $\mathcal{R}$ of the system with MMSE, with $N = 16$. Here, the rate is obtained as $\mathcal{R} = (1 - \tau/\tau_c)\mathcal{T}\log_2(1 + \gamma_{\text{th}})$ (bps/Hz), where τ_c is the total length of any user's packet. Firstly, we look at the effect of changing γ_{th} by fixing $\tau_c = 100$. For $L = 2$, $\gamma_{\text{th}} = 20$ offers a higher rate than $\gamma_{\text{th}} = 10$, provided $\tau \geq 3$. Thus, at low loads, increasing γ_{th} (correspondingly, selecting a higher order modulation and coding scheme) leads to better achievable rates. In contrast, when $L = 4$, $\gamma_{\text{th}} = 10$ outperforms $\gamma_{\text{th}} = 20$, because the system is highly interference limited. Next, comparing $L = 2, 3, 4$ for $\tau_c = 100$ and $\gamma_{\text{th}} = 10$, we see that the rate improves with L, provided

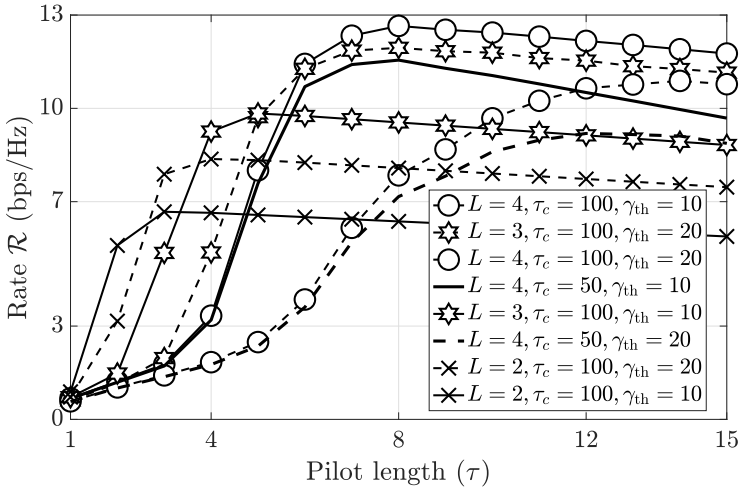


Figure 3.4: Impact of pilot length τ on rate with MMSE.

the pilot length is large enough. Finally, decreasing τ_c reduces the achievable rate, as the relative overhead due to pilots increases. Thus, at high loads, the throughput $\mathcal{T}$ limits the achievable rate, while at low loads, the SINR threshold γ_{th} is the primary factor in determining the achievable rate.

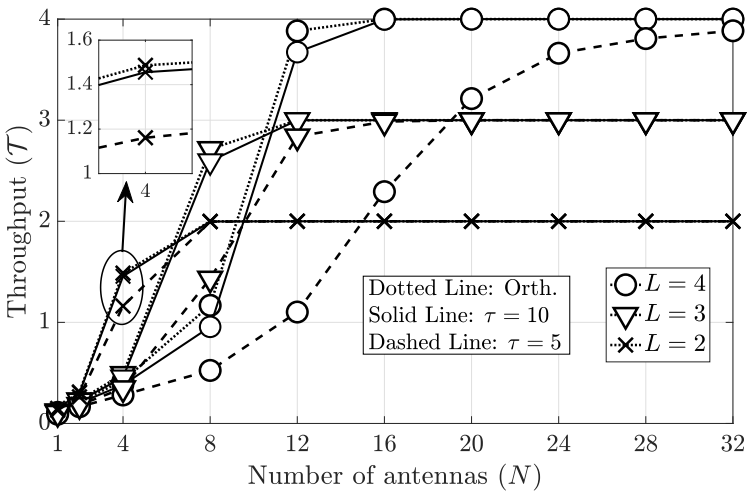


Figure 3.5: Effect of number of antennas N with MMSE.

In Fig. 3.5, we investigate the effect of the number of antennas at the BS, by plotting the throughput with MMSE channel estimation for different L and τ. Intuitively, we expect that, to achieve the optimal throughput of L, we would require slightly more

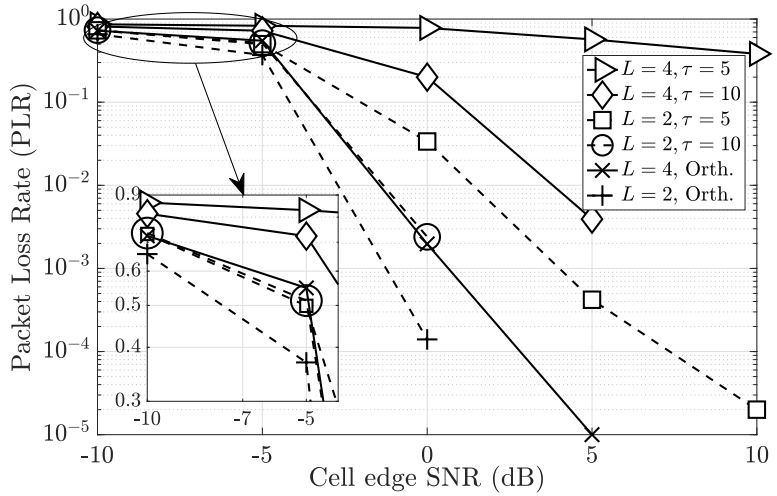


Figure 3.6: Effect of cell edge SNR with MMSE.

than $L\bar{d}$ antennas at the BS, since $L\bar{d}$ users transmit packets per RB on average. The orthogonal pilots curve is obtained by allocating $\tau = M = \lfloor LT \rceil$ for each L. Under all configurations, it is observed that increasing N has a significant impact, and the peak throughput achieved reaches its maximum of $\mathcal{T} = L$. Further, $\tau = 10$ achieves a very similar performance as that of orthogonal pilots, and $\tau = 5$ performs poorly at low N and high L. For $L = 2$, the throughput reaches the peak $\mathcal{T} = 2$ for $N \geq 8$ for all three values of τ. Similarly, for a high load of $L = 3$, the throughput reaches the peak, $\mathcal{T} = 3$, for $N \geq 16$. For $L = 2, 3$, since the average repetition factor $\bar{d} = 3$, each RB is occupied by $6, 9$ users, respectively. Thus, a slightly higher number of antennas is sufficient to recover all the packets, provided accurate channel estimates are available (i.e., τ is large enough). It is observed that at $L = 2,\ N = 4$ and $L = 3,\ N = 8$, improving τ greatly improves the throughput. Increasing the number of antennas increases the array gain and the decoding capability of the regularized zero forcing decoder at the BS, which in turn leads to more users getting decoded. This shows the effectiveness of the number of antennas in improving the throughput. Also, when $N = 12$, the dramatic drop in the throughput of $\mathcal{T} = 3.8$ for $\tau = 200$ (orthogonal pilots) to $\mathcal{T} = 1.2$ for $\tau = 5$, which is around 70% loss in performance, shows that it is crucial to account for estimated CSI while analyzing the performance of IRSA systems.

Fig. 3.6 shows the impact of cell edge SNR on the packet loss rate PLR with MMSE, with $N = 16$. For SNR < -5 dB, the PLR is high, and in the noise-limited regime ($-5 <$ SNR < 0 dB), an increase in cell edge SNR sharply decreases the PLR. For $L = 4$, $\tau = 5$, the system becomes interference-limited, and thus the performance saturates at high SNR. This is because, at low τ, both signal and interference powers get scaled equally, and the SINR remains roughly constant. Increasing τ from 5 to 10 and then to orthogonal pilots, we observe that the PLR falls from 0.5 to $10^{-2.5}$ to 10^{-5}. The higher τ and SNR result in accurate channel estimates, and thus very low PLR is observed. Similarly, at $L = 2$, the drop of PLR from $10^{-1.7}$ to $10^{-2.8}$ to $10^{-3.9}$ for $\tau = 5, 10$ and orthogonal pilots emphasizes the need to account for estimated CSI when analyzing the performance of IRSA. In summary, the overall performance can be improved by increasing the pilot length, number of antennas, or cell edge SNR, but these need to be increased judiciously, keeping the other parameters in mind.

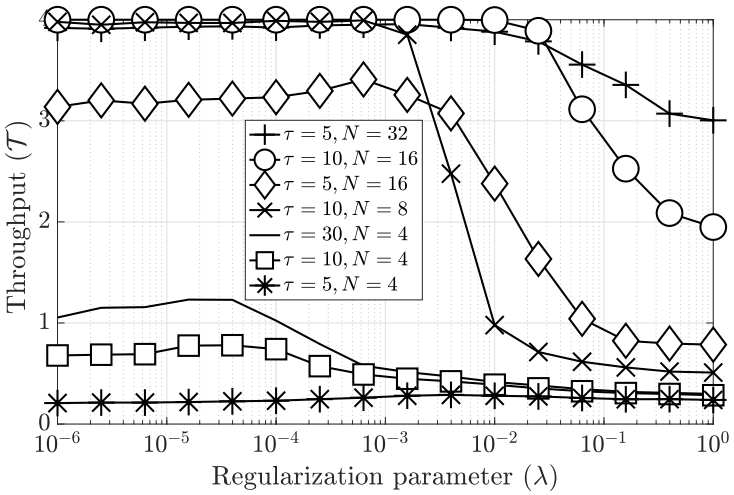


Figure 3.7: Effect of regularization parameter and τ with MMSE.

Fig. 3.7 shows the effect of the regularization parameter, λ, on the throughput of the system when MMSE estimation is employed, with $L = 4$. As λ is varied from 10^{-6} to 1, the curves go from ZF on the left to RZF in the middle and finally to MRC on the right. For $N = 4$, increasing τ from 5 to 10 to 30 only marginally improves the throughput. This is because the system is highly interference limited, and hence channel inversion does not

work well at low N. For $\tau = 5$, increasing N from 4 to 16 to 32 improves the performance due to the interference suppression capability of RZF. Similar observations can be made for $\tau = 10$ as well. MRC does not have the interference suppression capability of RZF, and thus the performance saturates at a low value for all τ. We note that the optimal throughput of $\mathcal{T} = 4$ is obtained over a wide range of λ, and thus precise optimization of λ is not necessary to obtain near-optimal throughputs.

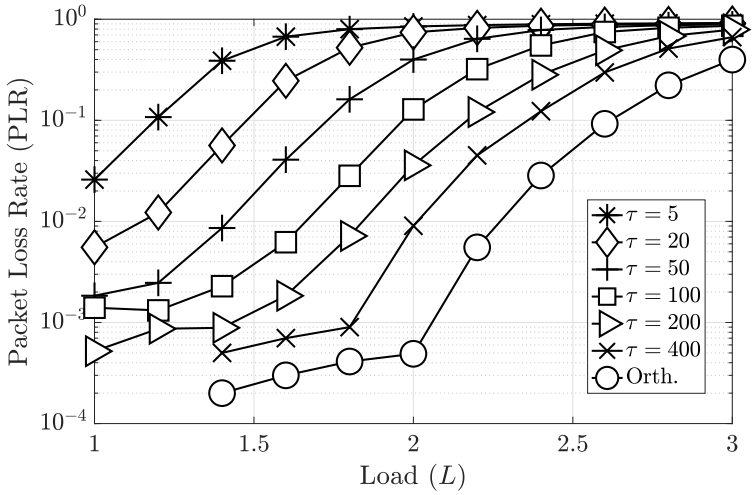


Figure 3.8: Impact of load on PLR with LCMMSE.

Fig. 3.8 studies the impact of L and τ on the system packet loss rate, PLR, evaluated with $N = 16$, $\gamma_{\text{th}} = 16$, and $\lambda = 1$. As the pilot length τ increases, better quality channel estimates are obtained, and the corresponding SINR increases. In particular, the system requires higher pilot lengths due to the use of LCMMSE estimates. The loss rates reduce with increase in τ, and gets closer to the orthogonal loss rate. The PLR of perfectly coordinated orthogonal access is the lowest. Similar to existing works, there is an error floor region where the PLR is very low (up to $L = 2$ for orthogonal pilots) after which the PLR increases rapidly and is called the waterfall region. Here $L = 2$ marks the inflection load, where the system transitions from the error floor to the waterfall region.

In Fig. 3.9, the impact of power control on the throughput with LCMMSE is characterized. For this plot, users transmit at powers that are dependent on their distances from the BS. Specifically, the mth user, who is located at a distance r_m from the BS transmits

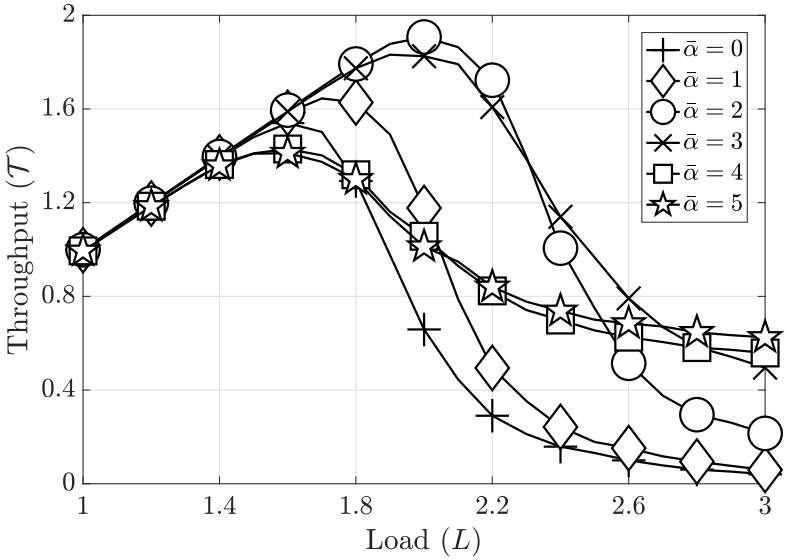


Figure 3.9: Impact of power control on throughput.

at a power $P(r_m/r_0)^{\alpha-\bar{\alpha}}$, making $\bar{\alpha}$ the effective path loss exponent. The cell edge SNR is fixed to 10 dB, and the throughputs are obtained by varying $\bar{\alpha}$ and P. When $\bar{\alpha} = 0$, the signals of the users undergo pure fading, and the system achieves a peak throughput of $\mathcal{T} = 1.52$ at $L = 1.6$. Further, as L is increased, the throughput drops to 0. The throughput of the system increases as $\bar{\alpha}$ increases, until $\bar{\alpha} = 2/3$. The exact $\bar{\alpha}$ that yields the highest throughput is dependent on other system parameters such as SNR, γ_{th}, and N. As $\bar{\alpha}$ is increased, the channel coefficients of the users become more disparate, and thus offer a higher degree of capture effect. Beyond $\bar{\alpha} = 3$, the throughput decreases as the exponent is so high that the received signal power becomes comparable to the noise. For higher $\bar{\alpha}$, the throughput saturates as L is increased since a few users are always decoded due to path loss disparity. The channel fades and the path loss coefficients contribute to the disparity amongst the channel coefficients of the users, and thus such a system has higher throughputs than a system with only path loss [8] or only fading [17]. Thus, it is useful to consider the combined effects of fading and path loss in optimizing the performance.

Note: We have presented only a couple of results for the empirical performance of the LCMMSE scheme above. The detailed empirical results for the LCMMSE scheme is

presented in Sec. 3.9.2.

3.6.1 Theoretical Validation of Throughput

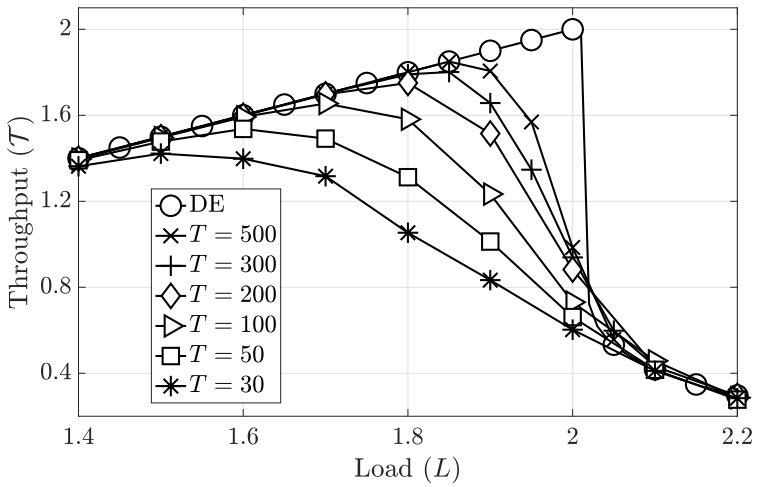


Figure 3.10: Effect of T on the throughput.

The results in this subsection are presented for $\tau = 10$, cell edge SNR = 10 dB, $N = 16$, $\lambda = 10^{-2}$, $\gamma_{\text{th}} = 16$, $d_{\max} = 8$ maximum repetitions and $N_s = 10^3$ Monte Carlo runs. To reduce clutter in the plots, we present the theoretical results for the lowest complexity (LCMMSE) channel estimation scheme. We note that the trends in the theoretical performance of MMSE and MSBL are similar to the trends of LCMMSE.

Fig. 3.10 investigates the effect of increasing the number of RBs on the throughput. The peak throughput increases from $\mathcal{T} = 1.52$ at $L = 1.6$ for $T = 50$ to $\mathcal{T} = 1.85$ at $L = 1.85$ for $T = 500$. Since $\bar{d}$ is fixed, each user has a larger number of RBs to choose from as T is increased. Thus, the interference reduces, and the throughput increases until it reaches a peak and then drops off. The success probability θ_r is evaluated empirically via 10^4 Monte Carlo runs, and this in turn yields the asymptotic theoretical throughput, which is marked as "DE". This can be achieved as $M, T \to \infty$ with a fixed L. It is seen that this asymptotic throughput increases linearly with the load until it hits a maximum at the inflection load of the system, which occurs at $L^* = 2$ in this case. The throughput drops sharply beyond this load. The asymptotic throughput provides an upper bound on

the throughput achievable with finitely many RBs for low to moderate loads. At very low and high loads, the throughput achieved with finitely many RBs exactly matches with the DE asymptotic throughput. A convenient operating point would be to set the system load to, say, 90% of the inflection load, as, in this case, only finitely many RBs would be sufficient to achieve the asymptotic throughput. Finally, it can be observed that the throughput of the system can be increased by increasing T, but only when the system is operated at a load that is lower than the inflection load. Beyond the inflection load, the system is always interference-limited and increasing T does not help.

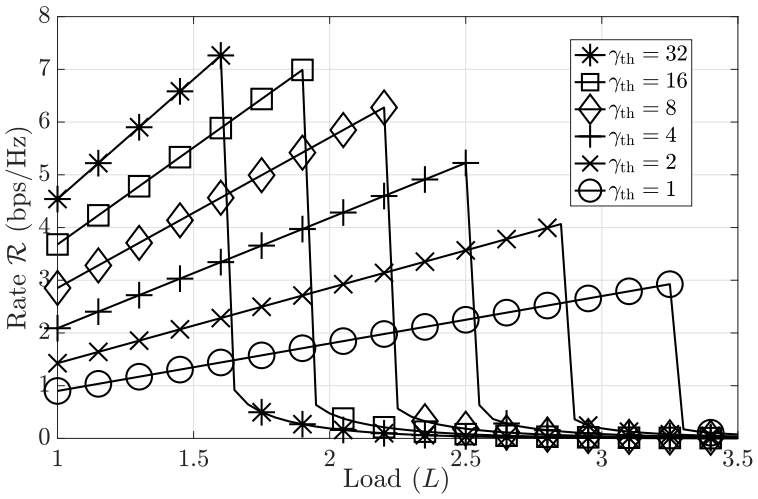


Figure 3.11: Rate for different SINR thresholds.

In Fig. 3.11, the asymptotic rate of the system is plotted versus the system load for different SINR thresholds with $\tau_c = 100$. For a fixed γ_{th}, $\mathcal{R}$ increases until the inflection load and then drops off to zero. It is observed that a high $\mathcal{R}$ can be achieved at lower loads by choosing a high γ_{th}, whereas, at high loads, in order to serve more users, γ_{th} must be kept low. The choice of the threshold γ_{th} decides the rate of transmission, which in turn is related to the modulation and coding scheme to be used. In summary, the SINR threshold γ_{th}, which depends on the modulation and coding scheme employed and determines the data rate, can be chosen based on the system parameters such as the number of antennas, training duration, number of users/RBs, and the transmit power.

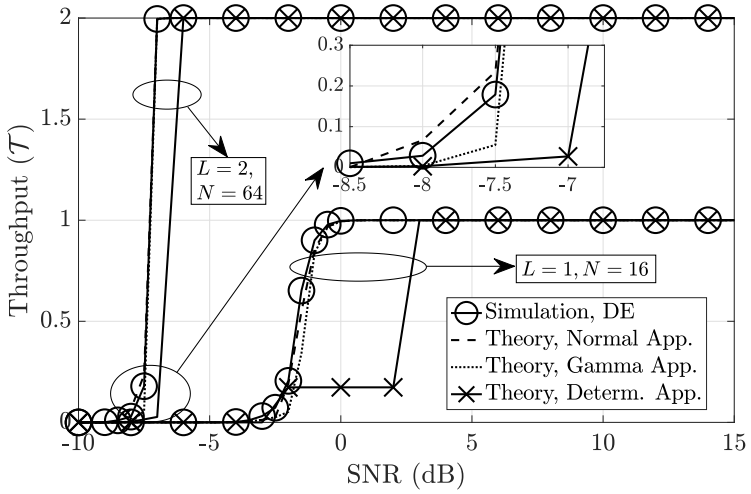


Figure 3.12: Validation of theoretical approximations.

We now validate the approximations derived in Theorem 3.3 with the simulations obtained with MRC, $d_{\max} = 27$ maximum repetitions, and $\gamma_{\text{th}} = 10$. Fig. 3.12, reveals an inflection SNR* of 0 dB and -7 dB for $L = 1, N = 16$ and $L = 2, N = 64$ respectively, which behaves similar to the inflection load L^*. Both the normal and the gamma approximations match well with the asymptotic throughput obtained from the DE process. This is because the deterministic approximation results in an SINR that is completely deterministic and θ_r that is a binary function of r, and consequently does not capture the statistics of the SINRs very well. Further, the deterministic approximation results in a throughput that acts as a step function since θ_r depends binarily on N, γ_{th}, and SNR. As we go from $L = 1, N = 16$ to $L = 2, N = 64$, the approximations become closer, and both the normal and the gamma approximations match perfectly with the asymptotic throughput. In summary, the theoretical curves with the approximations match the simulations when N is increased, as expected.

Fig. 3.13 examines the effect of T on the approximations with $L = 2$ and SNR $= 10$ dB. With finitely many RBs, such as $T = 50, 100, 300$, the throughput achieves the optimal throughput $\mathcal{T} = 2$ for $N = 24, 18, 16$. The asymptotic throughput obtained with DE provide an inflection $N^* = 12$, which matches perfectly with the normal approximation. The gamma approximation does not match as well as the normal approximation. Here,

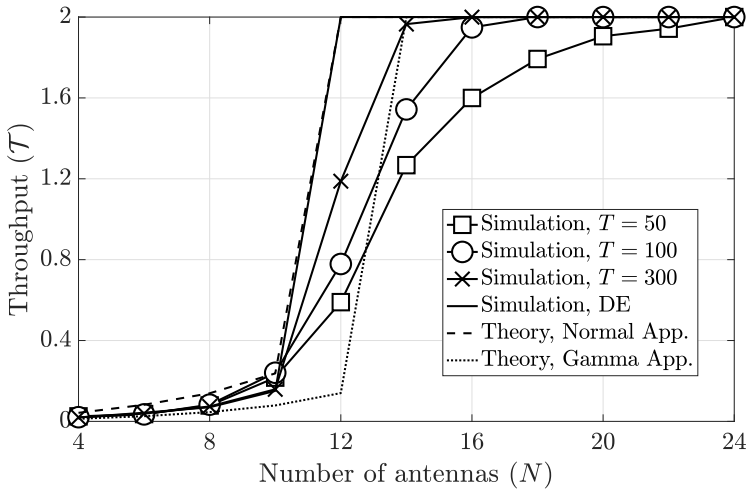


Figure 3.13: Comparison of approximations with simulation.

the curves are with MRC and perfect CSI, and the presented curves are valid upper bounds to the throughputs with estimated CSI. These can be achieved with high enough τ as observed in Fig. 3.2, and thus the derived results provide very good approximations to the asymptotic throughput achievable with estimated CSI.

3.7 Summary

This chapter studied the effect of estimated CSI on the throughput of IRSA, which is a distributed medium access protocol for mMTC involving repetition of packets across different randomly selected RBs. Decoding the users' packets at the BS involves successive interference cancellation. First, the channel estimates were derived under three schemes: a sparsity-based scheme with MSBL, MMSE, and LCMMSE. The corresponding SINR of all the users were obtained under all three schemes accounting for pilot contamination, channel estimation errors, path loss as well as multiple antennas at the BS. It was seen that these errors significantly reduce the peak achievable throughput, even resulting in up to 70% loss in certain regimes. Further, a density evolution based analysis was presented to characterize the asymptotic performance of the protocol when users perform path loss inversion based power control. Here, several approximations to the success probability

θ_r were derived and it was seen that these approximations match well as the number of antennas at the BS becomes large. Finally, several new insights into the design of IRSA-based systems was discussed, namely, the improvement of the system throughput, the evaluation of the operating load beyond which the system becomes interference limited, and the choice of the decoding threshold γ_{th}. The results underscored the importance of accounting for practical channel estimation in studying the throughput offered by the IRSA protocol. Future work could involve using differential evolution techniques [26] to obtain the optimal repetition distribution that maximizes the throughput in the finite frame length regime. This analysis of the optimal distribution is presented in Chapter 7.

3.8 Proofs

3.8.1 Proof of Theorem 3.1: Channel Estimation

3.8.1.1 MMSE

We first vectorize the signal as

$$\overline{\mathbf{y}}_t^k \triangleq \text{vec}(\mathbf{Y}_t^{\mathtt{p}k}) = (\mathbf{P}_t^{k*} \otimes \mathbf{I}_N)\mathbf{h}_t^k + \overline{\mathbf{n}}_t, \tag{3.40}$$

where $\mathbf{h}_t^k \triangleq \text{vec}(\mathbf{H}_t^k)$, $\overline{\mathbf{n}}_t \triangleq \text{vec}(\mathbf{N}_t^{\mathtt{p}})$, and $\otimes$ is the Kronecker product. The MMSE estimator is $\hat{\mathbf{h}}_t^k \triangleq \mathbb{E}_{\mathbf{z}}\left[\mathbf{h}_t^k\right]$, where $\mathbf{z} = \overline{\mathbf{y}}_t^k$. The error $\tilde{\mathbf{h}}_t^k \triangleq \hat{\mathbf{h}}_t^k - \mathbf{h}_t^k$ is uncorrelated with $\mathbf{z}$ and the estimate. The conditional statistics of a Gaussian random vector $\mathbf{x}$ are

$$\mathbb{E}_{\mathbf{z}}\left[\mathbf{x}\right] = \mathbb{E}\left[\mathbf{x}\right] + \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}\left(\mathbf{z} - \mathbb{E}\left[\mathbf{z}\right]\right), \tag{3.41}$$

$$\mathbf{K}_{\mathbf{x}\mathbf{x}|\mathbf{z}} = \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}\mathbf{K}_{\mathbf{zx}}. \tag{3.42}$$

Here, $\mathbf{K}_{\mathbf{xx}}$, $\mathbf{K}_{\mathbf{x}\mathbf{x}|\mathbf{z}}$, and $\mathbf{K}_{\mathbf{xz}}$ are the unconditional covariance of $\mathbf{x}$, the conditional covariance of $\mathbf{x}$ conditioned on $\mathbf{z}$, and the cross-covariance of $\mathbf{x}$ & $\mathbf{z}$ respectively. From (3.41),

the MMSE channel estimate $\hat{\mathbf{h}}_t^k$ can be calculated as

$$\hat{\mathbf{h}}_t^k = \mathbb{E}\left[\mathbf{h}_t^k\right] + \mathbb{E}\left[\mathbf{h}_t^k \overline{\mathbf{y}}_t^{kH}\right] \mathbb{E}[\overline{\mathbf{y}}_t^k \overline{\mathbf{y}}_t^{kH}]^{-1} (\overline{\mathbf{y}}_t^k - \mathbb{E}\left[\overline{\mathbf{y}}_t^k\right]). \tag{3.43}$$

The terms in the above expression can be evaluated as

$$\mathbb{E}\left[\mathbf{h}_t^k \overline{\mathbf{y}}_t^{kH}\right] = \mathbf{B}_t^k \mathbf{P}_t^{kT} \otimes \mathbf{I}_N, \tag{3.44}$$

$$\mathbb{E}[\overline{\mathbf{y}}_t^k \overline{\mathbf{y}}_t^{kH}] = (\mathbf{P}_t^{k*} \mathbf{B}_t^k \mathbf{P}_t^{kT} + N_0 \mathbf{I}_\tau) \otimes \mathbf{I}_N, \tag{3.45}$$

$$\hat{\mathbf{h}}_t^k = (\mathbf{B}_t^k \mathbf{P}_t^{kT} (\mathbf{P}_t^{k*} \mathbf{B}_t^k \mathbf{P}_t^{kT} + N_0 \mathbf{I}_\tau)^{-1} \otimes \mathbf{I}_N) \overline{\mathbf{y}}_t^k, \tag{3.46}$$

and thus, the MMSE estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ is

$$\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{pk} (\mathbf{P}_t^k \mathbf{B}_t^k \mathbf{P}_t^{kH} + N_0 \mathbf{I}_\tau)^{-1} \mathbf{P}_t^k \mathbf{B}_t^k, \tag{3.47}$$

$$\stackrel{(a)}{=} \mathbf{Y}_t^{pk} \mathbf{P}_t^k \mathbf{B}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{B}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}, \tag{3.48}$$

where (a) follows from $(\mathbf{AB} + \mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA} + \mathbf{I})^{-1}$.

3.8.1.2 LCMMSE

The LCMMSE estimator is $\hat{\mathbf{h}}_{tm}^k \triangleq \mathbb{E}_{\mathbf{z}}\left[\mathbf{h}_{tm}\right]$, where $\mathbf{z} = \mathbf{y}_{tm}^{\mathtt{p}k}$ is the received pilot signal. The error $\tilde{\mathbf{h}}_{tm}^k \triangleq \hat{\mathbf{h}}_{tm}^k - \mathbf{h}_{tm}$ is uncorrelated with the signal $\mathbf{y}_{tm}^{\mathtt{p}k}$ and the channel estimate $\hat{\mathbf{h}}_{tm}^k$. From (3.41), the LCMMSE channel estimate $\hat{\mathbf{h}}_{tm}^k$ can be calculated

$$\hat{\mathbf{h}}_{tm}^k = \mathbb{E}\left[\mathbf{h}_{tm} \mathbf{y}_{tm}^{\mathtt{p}kH}\right] \mathbb{E}[\mathbf{y}_{tm}^{\mathtt{p}k} \mathbf{y}_{tm}^{\mathtt{p}kH}]^{-1} \mathbf{y}_{tm}^{\mathtt{p}k} \tag{3.49}$$

$$= \frac{g_{tm}\beta_m \|\mathbf{p}_m\|^2 \sigma_{\mathtt{h}}^2}{N_0 \|\mathbf{p}_m\|^2 + \sum_{i \in \mathcal{S}_k} |\mathbf{p}_i^H \mathbf{p}_m|^2 g_{ti} \beta_i \sigma_{\mathtt{h}}^2} \mathbf{y}_{tm}^{\mathtt{p}k} \triangleq \eta_{tm}^k \mathbf{y}_{tm}^{\mathtt{p}k}. \tag{3.50}$$

3.8.1.3 MSBL

In each iteration of MSBL, two steps are performed. The first step, termed the E-step, updates the covariance $\mathbf{\Sigma}_{kt}^{j+1}$ and mean $\boldsymbol{\mu}_{ktn}^{j+1}$ of the posterior $p([\mathbf{Z}_t^k]_{:,n} | [\mathbf{Y}_t]_{:,n}, \boldsymbol{\gamma}_{kt}^j)$

$$\mathbf{\Sigma}_{kt}^{j+1} = \mathbf{\Gamma}_{kt}^j - \mathbf{\Gamma}_{kt}^j \mathbf{P}^{kH} (N_0 \mathbf{I}_\tau + \mathbf{P}^k \mathbf{\Gamma}_{kt}^j \mathbf{P}^{kH})^{-1} \mathbf{P}^k \mathbf{\Gamma}_{kt}^j, \tag{3.51}$$

$$\boldsymbol{\mu}_{ktn}^{j+1} = N_0^{-1}\boldsymbol{\Sigma}_{kt}^{j+1}\mathbf{P}^{kH}[\overline{\mathbf{Y}}_t^{\mathtt{p}k}]_{:,n},\ n \in [N]. \tag{3.52}$$

The second step, termed the M-step, updates the hyperparameter for the ith user in the tth RB as

$$[\boldsymbol{\gamma}_{kt}^{j+1}]_i = \frac{1}{N}\sum_{n=1}^{N}([\boldsymbol{\Sigma}_{kt}^{j+1}]_{i,i} + |[\boldsymbol{\mu}_{ktn}^{j+1}]_i|^2),\ i \in [M^k]. \tag{3.53}$$

This step estimates the variance of the channel of the ith user in the tth RB. Based on the estimate $\hat{g}_{ti}^k$ and the true g_{ti}, the set of users $[M^k]$ can be divided into four disjoint subsets

$$\mathcal{A}_t^k = \{i \in [M^k] \mid \hat{g}_{ti}^k g_{ti} = 1\}, \tag{3.54}$$

$$\mathcal{F}_t^k = \{i \in [M^k] \mid \hat{g}_{ti}^k (1 - g_{ti}) = 1\}, \tag{3.55}$$

$$\mathcal{M}_t^k = \{i \in [M^k] \mid (1 - \hat{g}_{ti}^k) g_{ti} = 1\}, \tag{3.56}$$

$$\mathcal{I}_t^k = \{i \in [M^k] \mid (1 - \hat{g}_{ti}^k)(1 - g_{ti}) = 1\}. \tag{3.57}$$

$\mathcal{A}_t^k$ is the set of true positive users, $\mathcal{F}_t^k$ is the set of false positive users, $\mathcal{M}_t^k$ is the set of false negative users, and $\mathcal{I}_t^k$ is the set of true negative users. False positive and false negative users form the errors in APM estimation. As the decoding iterations proceed, more users get decoded, and the errors in APM estimation decrease. The MSBL channel estimate $\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k}\mathbf{P}^k\hat{\boldsymbol{\Gamma}}_{kt}(\mathbf{P}^{kH}\mathbf{P}^k\hat{\boldsymbol{\Gamma}}_{kt} + N_0\mathbf{I}_{M^k})^{-1}$ is output in the E-step from Algorithm 1, where $\hat{\boldsymbol{\Gamma}}_{kt} = \mathrm{diag}(\boldsymbol{\gamma}_{kt}^{j_{\max}})$. The false negative users' channels do not get estimated even though they contribute towards $\mathbf{Y}_t^{\mathtt{p}k}$. The false positive users' channels get estimated even though they have not transmitted, and thus, an erroneous channel estimate is output for those users. Since $[\boldsymbol{\gamma}_{kt}]_i$ models the variance of the ith users signal in the tth RB, it models $g_{ti}\beta_i\sigma_{\mathtt{h}}^2$. Thus, the estimated hyperparameter $[\boldsymbol{\gamma}_{kt}^{j_{\max}}]_i$ would recover both $\hat{g}_{ti}^k$ and $\hat{\beta}_i^k$. Since the path loss is same across RBs, a higher quality estimate for the path loss can be estimated by averaging across RBs, and thus we obtain $\hat{\beta}_i^k = (\sum_{t=1}^{T}\hat{g}_{ti}^k[\boldsymbol{\gamma}_{kt}^{j_{\max}}]_i)/(\sigma_{\mathtt{h}}^2\sum_{t=1}^{T}\hat{g}_{ti}^k)$.

3.8.1.4 Error variances

The conditional covariance of $\mathbf{h}_{ti}$ is calculated conditioned on $\mathbf{z} = \hat{\mathbf{h}}_{ti}^k$. In MMSE, with $\mathbf{c}_{ti}^k = [\mathbf{C}_t^k]_{:,i}$ and $\mathbf{C}_t^k \triangleq \mathbf{P}_t^k \mathbf{B}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{B}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}$, we have

$$\mathbf{K}_{\mathbf{h}_{ti}\mathbf{h}_{ti}} = \mathbb{E}[\mathbf{h}_{ti}\mathbf{h}_{ti}^H] = \beta_i \sigma_{\mathtt{h}}^2 \mathbf{I}_N, \tag{3.58}$$

$$\mathbf{K}_{\mathbf{h}_{ti}\mathbf{z}} = \mathbb{E}[\mathbf{h}_{ti}\hat{\mathbf{h}}_{ti}^{kH}] = \mathbf{p}_i^H \mathbf{c}_{ti}^k g_{ti} \beta_i \sigma_{\mathtt{h}}^2 \mathbf{I}_N, \tag{3.59}$$

$$\mathbf{K}_{\mathbf{z}\mathbf{z}} = (N_0 \|\mathbf{c}_{ti}\|^2 + \textstyle\sum_{j \in \mathcal{S}_k} |\mathbf{p}_j^H \mathbf{c}_{ti}^k|^2 g_{tj} \beta_j \sigma_{\mathtt{h}}^2) \mathbf{I}_N. \tag{3.60}$$

Thus, the conditional covariance is

$$\begin{aligned} \mathbf{K}_{\mathbf{h}_{ti}\mathbf{h}_{ti}|\mathbf{z}} &= \mathbf{K}_{\mathbf{h}_{ti}\mathbf{h}_{ti}} - \mathbf{K}_{\mathbf{h}_{ti}\mathbf{z}} \mathbf{K}_{\mathbf{z}\mathbf{z}}^{-1} \mathbf{K}_{\mathbf{z}\mathbf{h}_{ti}} \\ &= \beta_i \sigma_{\mathtt{h}}^2 \left(\frac{N_0 \|\mathbf{c}_{ti}^k\|^2 + \sum_{j \in \mathcal{S}_k^i} |r_{jti}^k|^2 g_{tj} \beta_j \sigma_{\mathtt{h}}^2}{N_0 \|\mathbf{c}_{ti}^k\|^2 + \sum_{j \in \mathcal{S}_k} |r_{jti}^k|^2 g_{tj} \beta_j \sigma_{\mathtt{h}}^2} \right) \mathbf{I}_N \triangleq \delta_{ti}^k \mathbf{I}_N, \end{aligned} \tag{3.61}$$

where $r_{jti}^k \triangleq \mathbf{p}_j^H \mathbf{c}_{ti}^k$ and δ_{ti}^k accounts for pilot contamination. The conditional autocorrelation follows as

$$\mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tm}\mathbf{h}_{tm}^H] = \mathbf{K}_{\mathbf{h}_{tm}\mathbf{h}_{tm}|\mathbf{z}} + \mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tm}]\mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tm}]^H = \delta_{tm}^k \mathbf{I}_N + \hat{\mathbf{h}}_{tm}^k \hat{\mathbf{h}}_{tm}^{kH}. \tag{3.62}$$

The unconditional and conditional means of the estimation error are $\mathbb{E}[\tilde{\mathbf{h}}_{tm}^k] = \mathbb{E}[\hat{\mathbf{h}}_{tm}^k - \mathbf{h}_{tm}] = 0$ and $\mathbb{E}_{\mathbf{z}}[\tilde{\mathbf{h}}_{tm}^k] = \mathbb{E}_{\mathbf{z}}[\hat{\mathbf{h}}_{tm}^k - \mathbf{h}_{tm}] = \hat{\mathbf{h}}_{tm}^k - \hat{\mathbf{h}}_{tm}^k = 0$. The conditional autocovariance of the error therefore simplifies as

$$\mathbf{K}_{\tilde{\mathbf{h}}_{tm}^k \tilde{\mathbf{h}}_{tm}^k|\mathbf{z}} = \mathbb{E}_{\mathbf{z}}[\tilde{\mathbf{h}}_{tm}^k \tilde{\mathbf{h}}_{tm}^{kH}] = \mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tm}\mathbf{h}_{tm}^H] - \hat{\mathbf{h}}_{tm}^k \hat{\mathbf{h}}_{tm}^{kH} = \delta_{tm}^k \mathbf{I}_N, \tag{3.63}$$

and thus, δ_{tm}^k is also the variance of the estimation error. Substituting $\mathbf{C}_t^k = \mathbf{P}_t^k \operatorname{diag}(\eta_{ti_1}^k, \ldots, \eta_{ti_{M_t^k}}^k)$, we get the error variance for LCMMSE.

The MSBL estimate error is also uncorrelated with the estimate and the error variance can be derived similar to the MMSE scheme since the MSBL estimate is a "plug-in" MMSE estimate. Since only true positive users' channels are estimated, the error variance is

calculated only for the subset of true positive users (users with $\hat{g}_{ti}^k g_{ti} = 1$), and thus, each g_{ti} is accompanied by $\hat{g}_{ti}^k$ similar to [15] (See Chapter 4). Further, since the error variance models the true interference from other true positive users, the true path loss coefficient accompanies $\hat{g}_{ti}^k g_{ti}$. Hence we define $\mathbf{C}_t^k \triangleq \mathbf{P}^k \mathbf{D}_t^k (\mathbf{P}^{kH} \mathbf{P}^k \mathbf{D}_t^k + N_0 \mathbf{I}_{M^k})^{-1}$ and $\mathbf{D}_t^k \triangleq \text{diag}(d_{ti_1}^k, d_{ti_2}^k, \ldots, d_{ti_{M^k}}^k)$, with $d_{ti}^k = \hat{g}_{ti}^k g_{ti} \beta_i \sigma_{\mathtt{h}}^2$. Substituting for $\mathbf{C}_t^k$, we get the error variance for MSBL.

3.8.2 Proof of Theorem 3.2: SINR Evaluation

In order to evaluate the SINR, we first calculate the power of the received signal, which is calculated conditioned on the knowledge of the estimates $\mathbf{z} \triangleq \text{vec}(\hat{\mathbf{H}}_t^k)$ as $\mathbb{E}_{\mathbf{z}}[|\tilde{y}_{tm}^k|^2] = \mathbb{E}_{\mathbf{z}}[|\sum_{i=1}^4 T_i|^2]$. Since noise is uncorrelated with data, $\mathbb{E}_{\mathbf{z}}[T_1 T_4^H] = \mathbb{E}_{\mathbf{z}}[T_2 T_4^H] = \mathbb{E}_{\mathbf{z}}[T_3 T_4^H] = 0$. Since MMSE channel estimates are uncorrelated with their errors [49], $\mathbb{E}_{\mathbf{z}}[T_1 T_2^H] = 0$. Computing the remaining power components requires the evaluation of $\mathbb{E}_{\mathbf{z}}[x_i x_j]$ for $i \neq j$ which can be calculated as $\mathbb{E}_{\mathbf{z}}[x_i x_j] = \mathbb{E}_{\mathbf{z}}[x_i]\mathbb{E}_{\mathbf{z}}[x_j] = 0$. Thus, all the four terms are uncorrelated and the power in the received signal is just a sum of the powers of the individual components $\mathbb{E}_{\mathbf{z}}[|\tilde{y}_{tm}^k|^2] = \sum_{i=1}^4 \mathbb{E}_{\mathbf{z}}[|T_i|^2]$. We now compute the powers of each of the components. The useful signal power is

$$\mathbb{E}_{\mathbf{z}}[|T_1|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k g_{tm} x_m|^2] = P g_{tm}^2 |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k|^2. \tag{3.64}$$

The desired gain is written as

$$\mathtt{Gain}_{tm}^k \triangleq \frac{\mathbb{E}_{\mathbf{z}}[|T_1|^2]}{P\|\mathbf{a}_{tm}^k\|^2} = g_{tm} \frac{|\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k|^2}{\|\mathbf{a}_{tm}^k\|^2}. \tag{3.65}$$

The power of the estimation error is expressed as

$$\mathbb{E}_{\mathbf{z}}[|T_2|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}_{tm}^{kH} \tilde{\mathbf{h}}_{tm}^k g_{tm} x_m|^2] = P g_{tm}^2 \delta_{tm}^k \|\mathbf{a}_{tm}^k\|^2. \tag{3.66}$$

Next, the power of the inter-user interference term T_3 is

$$\begin{aligned}\mathbb{E}_{\mathbf{z}}[|T_3|^2] &= \mathbb{E}_{\mathbf{z}}\left[\left|\mathbf{a}_{tm}^{kH}\textstyle\sum_{i\in\mathcal{S}_k^m} g_{ti}\mathbf{h}_{ti}x_i\right|^2\right] \\ &= P\textstyle\sum_{i\in\mathcal{S}_k^m} g_{ti}^2\mathbf{a}_{tm}^{kH}\mathbb{E}_{\mathbf{z}}[\mathbf{h}_{ti}\mathbf{h}_{ti}^H]\mathbf{a}_{tm}^k \\ &= P\textstyle\sum_{i\in\mathcal{S}_k^m} g_{ti}^2\mathbf{a}_{tm}^{kH}(\delta_{ti}^k\mathbf{I}_N + \hat{\mathbf{h}}_{ti}^k\hat{\mathbf{h}}_{ti}^{kH})\mathbf{a}_{tm}^k \\ &= P\textstyle\sum_{i\in\mathcal{S}_k^m} g_{ti}^2(\|\mathbf{a}_{tm}^k\|^2\delta_{ti}^k + |\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^k|^2). \end{aligned} \tag{3.67}$$

Here, $\mathbb{E}_{\mathbf{z}}[|T_2|^2] + \mathbb{E}_{\mathbf{z}}[|T_3|^2]$ represents the contribution of estimation errors and multi-user interference components of the other users. Since g_{ti} is binary, its powers are dropped. We now split the normalized version of the above into the sum of the error component $\mathtt{Est}_{tm}^k$ and the multi-user interference $\mathtt{MUI}_{tm}^k$ as follows

$$\mathtt{Est}_{tm}^k \triangleq \sum\nolimits_{i\in\mathcal{S}_k} g_{ti}\delta_{ti}^k,\ \mathtt{MUI}_{tm}^k \triangleq \sum\nolimits_{i\in\mathcal{S}_k^m} g_{ti}\frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^k|^2}{\|\mathbf{a}_{tm}^k\|^2}. \tag{3.68}$$

The noise power is calculated as

$$\mathbb{E}_{\mathbf{z}}[|T_4|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}_{tm}^{kH}\mathbf{n}_t|^2] = N_0\|\mathbf{a}_{tm}^k\|^2. \tag{3.69}$$

A meaningful SINR expression can be written out by dividing the useful signal power from (3.65) by the sum of the interference and the noise powers (from (3.68), and (3.69)) [49]. Note that the interference component is comprised of the estimation error term and the signal powers of other users who have also transmitted in the same RB. For MMSE/LCMMSE, the corresponding SINR can be calculated by plugging in the channel estimates.

In MSBL, each of T_1, T_2, and T_3 is calculated among the subset of true positive users in the tth RB, i.e., users in $\mathcal{A}_t^k = \{i \in [M^k] | \hat{g}_{ti}^k g_{ti} = 1\}$. Hence, each of the powers previously derived for MMSE is accompanied by $\hat{g}_{ti}^k g_{ti}$. We need to account for false negative users, i.e., users in $\mathcal{M}_t^k = \{i \in [M^k] | (1 - \hat{g}_{ti}^k) g_{ti} = 1\}$. These users interfere with the decoding of other users and the SINR for such users is 0 since they will never get decoded. Such

users' signals are uncorrelated with the other terms, and thus, their power is

$$\begin{aligned}\mathbb{E}_{\mathbf{z}}[|T_5|^2] &= \mathbb{E}_{\mathbf{z}}[|\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}_t^k}\mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}g_{ti}x_i|^2] \\ &\overset{(b)}{=} P\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}_t^k} g_{ti}^2\mathbf{a}_{tm}^{kH}\mathbb{E}[\mathbf{h}_{ti}\mathbf{h}_{ti}^H]\mathbf{a}_{tm}^k \\ &= P\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}_t^k} g_{ti}^2\mathbf{a}_{tm}^{kH}(\beta_i\sigma_{\mathtt{h}}^2\mathbf{I}_N)\mathbf{a}_{tm}^k \\ &= P\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}_t^k} g_{ti}^2\beta_i\sigma_{\mathtt{h}}^2\|\mathbf{a}_{tm}^k\|^2, \end{aligned} \tag{3.70}$$

where the conditional expectation is dropped in (b) since the BS does not have the knowledge of the channel estimates of false negative users. The normalised power of the false positive users is $\mathtt{FNU}_{tm}^k \triangleq \sum_{i\in\mathcal{S}_k^m}(1-\hat{g}_{ti}^k)g_{ti}\beta_i\sigma_{\mathtt{h}}^2$.

3.8.3 Proof of Lemma 1: Deterministic Equivalent Analysis in the Context of Massive MIMO

It is known that, as the number of antennas gets large, both $\|\hat{\mathbf{h}}_{tm}^k\|^2$ and $|\hat{\mathbf{h}}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^k|^2$ converge almost surely (a.s.) to their deterministic equivalents [106]. Evaluating the deterministic equivalents as in [106] and plugging into the SINR expression instead of the original terms, we can find an approximation to the SINR in the high antenna regime. As N gets large, the SINR with MRC converges almost surely ($\rho_{tm}^k \xrightarrow{\text{a.s.}} \overline{\rho}_{tm}^k$) to

$$\overline{\rho}_{tm}^k = \frac{N\mathtt{Sig}_{tm}^k}{\epsilon_{tm}^k\left(N_0/P + \mathtt{IntNC}_{tm}^k\right) + \mathtt{IntC}_{tm}^k}, \tag{3.71}$$

where $\mathtt{Sig}_{tm}^k$ is the desired gain, $\mathtt{IntNC}_{tm}^k$ is the non-coherent interference, and $\mathtt{IntC}_{tm}^k$ is the coherent interference. For LCMMSE, $\mathtt{IntNC}_{tm}^k \triangleq g_{tm}\delta_{tm}^k + \sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i\sigma_{\mathtt{h}}^2$, $\mathtt{Sig}_{tm}^k \triangleq g_{tm}\beta_m^2\sigma_{\mathtt{h}}^4\|\mathbf{p}_m\|^4$, $\mathtt{IntC}_{tm}^k \triangleq N\sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i^2\sigma_{\mathtt{h}}^4|\mathbf{p}_m^H\mathbf{p}_i|^2$, and $\epsilon_{tm}^k \triangleq N_0\|\mathbf{p}_m\|^2 + \sum_{i\in\mathcal{S}_k} g_{ti}\beta_i\sigma_{\mathtt{h}}^2|\mathbf{p}_m^H\mathbf{p}_i|^2$. For MMSE, $\epsilon_{tm}^k \triangleq N_0\|\mathbf{c}_{tm}^k\|^2 + \sum_{i\in\mathcal{S}_k} g_{ti}\beta_i\sigma_{\mathtt{h}}^2|\mathbf{c}_{tm}^{kH}\mathbf{p}_i|^2$, $\mathtt{Sig}_{tm}^k \triangleq g_{tm}(\epsilon_{tm}^k)^2$, $\mathtt{IntC}_{tm}^k \triangleq N\sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i^2\sigma_{\mathtt{h}}^4|\mathbf{c}_{tm}^{kH}\mathbf{p}_i|^2$, $\mathtt{IntNC}_{tm}^k \triangleq g_{tm}\delta_{tm}^k + \sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i\sigma_{\mathtt{h}}^2$. For MSBL, $\epsilon_{tm}^k \triangleq N_0\|\mathbf{c}_{tm}^k\|^2 + \sum_{i\in\mathcal{S}_k} g_{ti}\beta_i\sigma_{\mathtt{h}}^2|\mathbf{c}_{tm}^{kH}\mathbf{p}_i|^2$, $\mathtt{IntNC}_{tm}^k \triangleq \hat{g}_{tm}^k g_{tm}\delta_{tm}^k + \sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i\sigma_{\mathtt{h}}^2$, $\mathtt{Sig}_{tm}^k \triangleq \hat{g}_{tm}^k g_{tm}(\epsilon_{tm}^k)^2$, and $\mathtt{IntC}_{tm}^k \triangleq N\sum_{i\in\mathcal{S}_k^m} g_{ti}\beta_i^2\sigma_{\mathtt{h}}^4|\mathbf{c}_{tm}^{kH}\mathbf{p}_i|^2$. Here, δ_{tm}^k and $\mathbf{c}_{tm}^k$ are obtained from the previous theorems. The above expressions are obtained by replacing each

of the terms involving $\hat{\mathbf{h}}_{tm}^{k}$ in the SINR with their respective deterministic equivalents.

3.8.4 Proof of Theorem 3.3: Success Probability

Let k denote the intra-RB decoding iteration. When perfect CSI is available at the BS and the users perform path loss inversion, the SINR of the mth user in an RB is computed as

$$\rho_m^k = \frac{P\|\mathbf{h}_m\|^4}{N_0\|\mathbf{h}_m\|^2 + P\sum_{i\in\mathcal{S}_k^m}|\mathbf{h}_m^H\mathbf{h}_i|^2}. \tag{3.72}$$

For $r = 1$, $\rho_1^1 = P\|\mathbf{h}_m\|^2/N_0$, and θ_1 reduces to

$$\theta_1 = \Pr(\rho_1^1 \geq \gamma_{\text{th}}) = \Gamma_{\text{inc}}(N, \rho_0^{-1}\gamma_{\text{th}})/\Gamma(N), \tag{3.73}$$

where $\rho_0 \triangleq P\sigma_{\mathtt{h}}^2/N_0$, $\Gamma_{\text{inc}}(s,x) = \int_x^\infty t^{s-1}\,e^{-t}\,\mathrm{d}t$ is the upper incomplete gamma function and $\Gamma(s)$ is the ordinary gamma function. The interference is written as $t_{mi} = |\mathbf{h}_m^H\mathbf{h}_i|^2/(\|\mathbf{h}_m\|^2\|\mathbf{h}_i\|^2)$, where $t_{mi} \sim \text{Beta}(\alpha = 1, \beta = N)$. We use $\xrightarrow{\text{a.s.}}$ to denote convergence in the almost surely sense. Since $\|\mathbf{h}_i\|^2/N \xrightarrow{\text{a.s.}} \sigma_{\mathtt{h}}^2$ and $\|\mathbf{h}_i\|^4/N^2 \xrightarrow{\text{a.s.}} \sigma_{\mathtt{h}}^4$ as $N \to \infty$ [106], we can approximate the SINR as

$$\rho_m^k \approx N(\rho_0^{-1} + N\textstyle\sum_{i\in\mathcal{S}_k^m} t_{mi})^{-1}. \tag{3.74}$$

Here, we have applied the theory of deterministic equivalents to only the channel norms and not to the interference. This is supported by the fact that the interference components converge to their deterministic equivalents slower than the norms converge to their deterministic equivalents [106].

For $r = 2$, since $t_{12} = t_{21}$, $\rho_1^1 = \rho_2^1 = N/(\rho_0^{-1} + Nt_{12})$. Thus, $\rho_{\max}^1 = N/(\rho_0^{-1} + Nt_{12})$ and $\rho_{\max}^2 = N\rho_0$ with $\rho_{\max}^1 \leq \rho_{\max}^2$. Thus, the success probability reduces to $\theta_r = \Pr(\rho_{\max}^1 \geq \gamma_{\text{th}})$. Let $t_0 \triangleq \gamma_{\text{th}}^{-1} - N^{-1}\rho_0^{-1}$. Hence, θ_2 is calculated as

$$\theta_2 \approx \Pr(\rho_{\max}^1 \geq \gamma_{\text{th}}) = \Pr(t_{12} \leq t_0)$$

$$= \mathbb{1}\{t_0 \geq 1\} + (1 - (1 - t_0)^N)\mathbb{1}\{0 \leq t_0 \leq 1\}. \tag{3.75}$$

For $r \geq 3$, ρ_m^k need not be a monotonically increasing function of k as seen in (3.74), and thus we cannot order the SINRs meaningfully to compute a closed form expression for θ_r. With $u_m = \sum_{i \in [r] \setminus m} t_{mi}$, the maximum SINR in the first intra-RB iteration is calculated as $\rho_{\max}^1 = \max_{m \in [r]} N(\rho_0^{-1} + N u_m)^{-1}$. Here, u_m is not independent across m and it is not clear which u_m is the minimum. Thus, we approximate $\rho_{\max}^1$ as ρ_1^1, and upon dropping the other SINR terms from (3.34), θ_r becomes

$$\theta_r \approx \Pr(\rho_1^1 \geq \gamma_{\text{th}}) = \Pr(u_1 \leq t_0). \tag{3.76}$$

We now discuss two approximations to u_m to evaluate θ_r, with the assumption that u_m is independent across m.

Since $\lim_{N \to \infty} \text{Beta}(\alpha = 1, \beta = N) = \exp(\lambda = N)$, we approximate $t_{mi} \sim \exp(N)$, which is a good approximation at high N [105]. Even with this approximation, u_m is identically Gamma distributed across users but not independent. Thus, with the independence assumption, u_m is i.i.d. Gamma distributed with shape parameter $r-1$ and rate parameter N, i.e., $u_m \overset{\text{i.i.d.}}{\sim} \text{Gamma}(r-1, N)$. Thus, we obtain the *Gamma* approximation:

$$\theta_r \approx 1 - \Gamma_{\text{inc}}(r-1, N t_0)/\Gamma(r-1). \tag{3.77}$$

Similarly, when we assume t_{mi} is Normal distributed, u_m is identically Normal distributed across users but not independent. Let $\mu_N = (N+1)^{-1}$ and $\sigma_N^2 = N(N+1)^{-2}(N+2)^{-1}$. If we approximate $t_{mi} \sim \mathcal{N}(\mu_N, \sigma_N^2)$ and u_m is independent across m, then $u_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}((r-1)\mu_N, (r-1)\sigma_N^2)$. Thus, we obtain the *Normal* approximation:

$$\theta_r \approx 1 - \mathcal{Q}\left(\frac{t_0 - (r-1)\mu_N}{\sqrt{r-1}\sigma_N}\right), \tag{3.78}$$

where $\mathcal{Q}(\cdot)$ is the standard Normal Q-function.

A simpler expression can be obtained for θ_r by applying the theory of deterministic equivalents to not just the channel norms but also to the interference. Thus, $|\mathbf{h}_i^H \mathbf{h}_m|^2/N \xrightarrow{\text{a.s.}} \sigma_{\mathtt{h}}^4$, as $N \to \infty$ [106]. Thus, the SINR becomes

$$\rho_m^k = N/(\rho_0^{-1} + r - k), \tag{3.79}$$

which is not random and is a deterministic function of N and ρ_0. This expression for SINR follows from Lemma 1. Thus, we obtain the *deterministic* approximation:

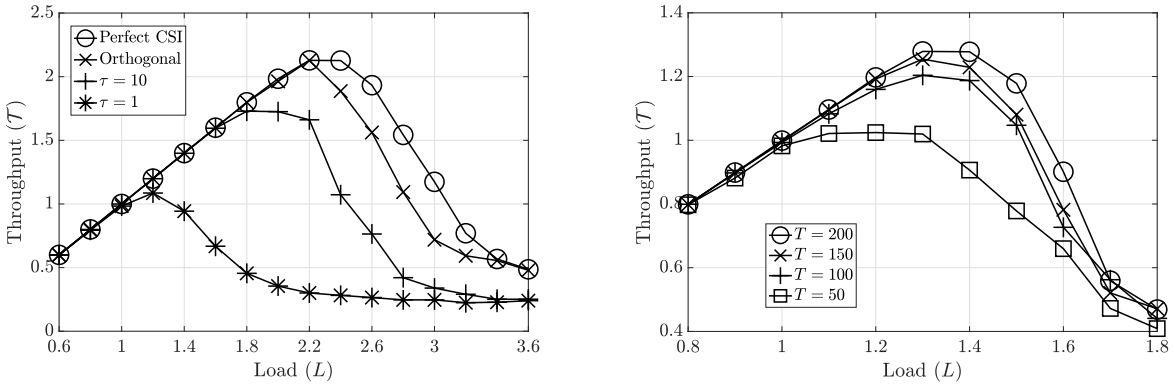
$$\theta_r = \Pr(\rho_1^1 \geq \gamma_{\text{th}}) = \mathbb{1}\{r \leq \lfloor N/\gamma_{\text{th}} - \rho_0^{-1} + 1 \rfloor\}. \tag{3.80}$$

3.9 Appendix

3.9.1 Preliminary Results

This section presents the preliminary results that were published in the conference version [12] of this chapter. The results in this section are presented for $T = 50$ RBs, $N = 4$ antennas, $N_s = 100$ simulations, capture threshold $\gamma_{\text{th}} = 4$, path loss exponent $\alpha = 3$, channel variance $\sigma_{\mathtt{h}}^2 = 1$, $r_{max} = 1000$ m, and $r_0 = 100$ m. The soliton distribution in [72] with parameters "N" $= k_s = 27$ and "a" $= a_s = 0.02$ is used to generate the repetition factor d_m for the mth user, whose pattern is formed by uniformly randomly choosing d_m RBs from T RBs. The signal power is taken to be $P = 1$, relative to the noise variance set as $N_0 = 10^{-4}$. Since the design principles and goals of other NOMA schemes such as PD-NOMA are completely different, comparison with other non RA based NOMA schemes has not been performed.

Fig. 3.14(a) shows the throughput, $\mathcal{T}$ (packets per RB), calculated under different cases. All the cases have a throughput that exceeds unity (which is the throughput of perfectly coordinated orthogonal access). As the load $L = M/T$ increases, the peak throughput of the system increases till it reaches a maximum. After that, it decreases since there are too many users sharing the same resources, thereby increasing the interference



(a) Effect of imperfect CSI on throughput.

(b) Throughput for different number of RBs.

Figure 3.14: Performance of IRSA.

power and hence decreasing SINR. It is seen that the peak occurs at $\mathcal{T} = 1.1$ with $L = 1.2$ for single-length pilots, $\mathcal{T} = 2.13$ with $L = 2.2$ for fully orthogonal pilots, and $\mathcal{T} = 2.13$ with $L = 2.2$ for perfect CSI. Thus, using single-length pilots results in a 48% drop in peak throughput. The significant drop in throughput under estimated channels underscores the importance of accounting for the effect of imperfect CSI in analyzing the performance.

In Fig. 3.14(b), the throughput of IRSA with single-length pilots is plotted against the load for different number of RBs. The peak throughput increases from $\mathcal{T} = 1.02$ at $L = 1.2$ for $T = 50$ to $\mathcal{T} = 1.28$ at $L = 1.3$ for $T = 200$. Since the truncated soliton size parameter k_s is retained at the same value for all T, each user has a larger number of RBs to choose from with the same k_s. Hence, the interference reduces, and the throughput increases, until it saturates. Note that, increasing T also increases the computational complexity of decoding and hence the desired number of RBs can be set according to the throughput requirements and the complexity constraints.

In the case of fully orthogonal pilots, MT symbols are required on average for channel estimation, whereas for single-length pilots, T symbols are required for channel estimation. Thus, for example, when $M = 60$ and $T = 50$, $(M-1)T = 2950$ extra training symbols required for orthogonal pilot-based channel estimation compared to single-length pilots based channel estimation.

3.9.2 Results with LCMMSE channel estimation

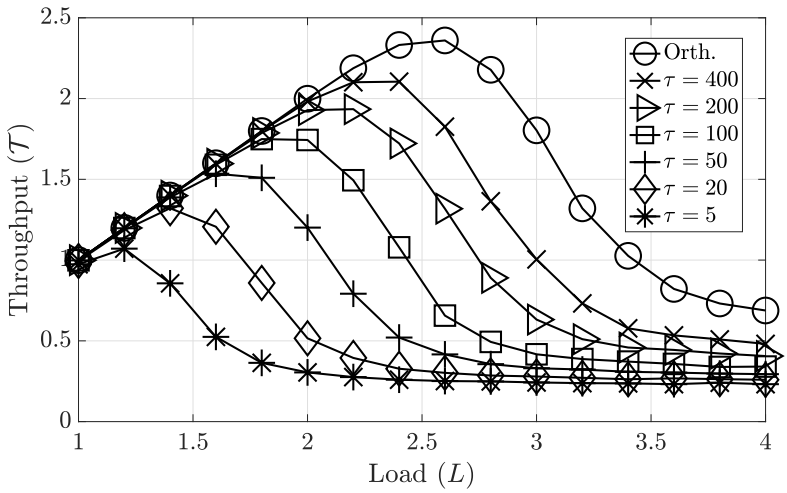


Figure 3.15: LCMMSE: Impact of L on the throughput.

This section presents the throughput results with the LCMMSE estimation scheme presented in Sec. 3.3. Fig. 3.15 shows the system throughput, $\mathcal{T}$ (packets per RE), evaluated for LCMMSE with different pilot lengths under estimated CSI, with $N = 16$, capture threshold $\gamma_{\text{th}} = 16$, and regularization parameter $\lambda = 1$. The throughputs exceed unity, which is the throughput of perfectly coordinated orthogonal access, for all $\tau \geq 5$. As the load L increases, the peak throughput of the system increases till it reaches a maximum. In the low load regime ($L < 2$), all the users' packets get successfully decoded through the SIC process, and thus the throughput increases linearly with the load. After reaching the maximum, it decreases since there are too many users sharing the same resources, thereby increasing the interference power and decreasing SINR. In the high load regime ($L > 3$), the system becomes interference-limited. As the pilot length τ increases, better quality channel estimates are obtained, and the corresponding SINR increases. The orthogonal pilots curve is obtained by allocating $\tau = M = \lfloor LT \rceil$ for each L. At $L = 1.5$, there are $M = 75$ total users that need to be served. For $\tau = 50$, the achievable throughput at $L = 1.5$ is nearly $\mathcal{T} = 1.5$. Thus, the throughput with orthogonal pilots (which requires $\tau = M$) can be achieved with fewer pilot symbols. Further, at low τ, there is a significant drop in the throughput with the non-orthogonal pilots compared to

orthogonal pilots (which achieves nearly the same performance as a genie-aided system where perfect CSI is available at the BS, not shown in the figure.) This significant drop in throughput under estimated channels underscores the importance of accounting for the effect of imperfect CSI in analyzing the performance of IRSA.

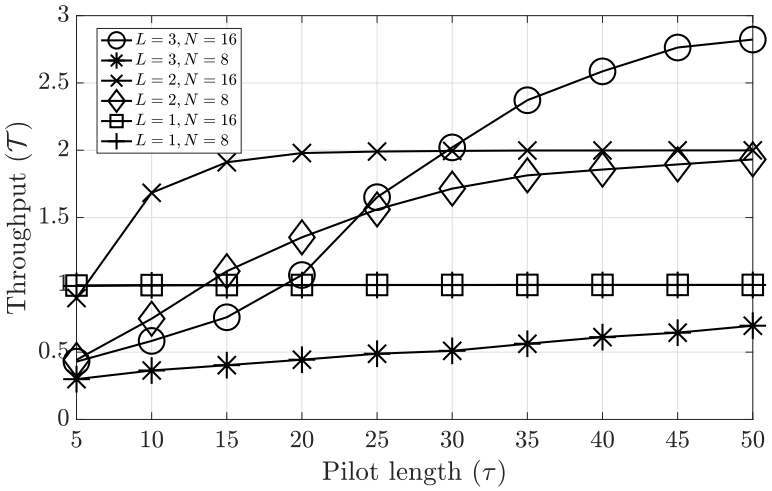


Figure 3.16: LCMMSE: Effect of τ on the throughput.

Fig. 3.16 shows the effect of pilot length on the system throughput at different loads and number of antennas. The capture threshold is set to $\gamma_{\text{th}} = 10$ and regularization parameter to $\lambda = 10^{-2}$ for the rest of this section. At all loads, the throughput saturates as the pilot length is increased, showing that beyond a certain pilot length, the throughput is limited by other system parameters such as the load, transmit power, or the number of antennas. For $N = 8$ and $L = 3$, the system throughput is low due to poor quality channel estimates. Increasing τ does not help since the system is already overloaded. That is, even though a larger number of measurements are available at high τ, $N = 8$ is insufficient to ensure successful delivery of all users' packets at high loads. When N is increased to 16, the system throughput dramatically improves with τ and approaches $\mathcal{T} = 2.8$ at $\tau = 50$. For $N = 16$, the optimal throughput of $\mathcal{T} = L$ is achieved with $\tau = 20$ (with either $N = 8$ or 16) for $L = 2$, which corresponds to $M = 100$ users, and with $\tau = 5$ for $L = 1$, which corresponds to $M = 50$ users.[13] For $L = 2$ and $N = 8$,

[13]Note that, the use of orthogonal pilots would require $\tau \geq 100$ and $\tau \geq 50$ for $M = 100$ and $M = 50$,

throughput greatly improves as τ increases and nearly optimal throughput is obtained at $\tau = 50$. To summarize, the pilot length has a significant impact on the performance of IRSA and yields near-optimal throughputs at significantly lower pilot lengths than that required for orthogonal pilot transmission.

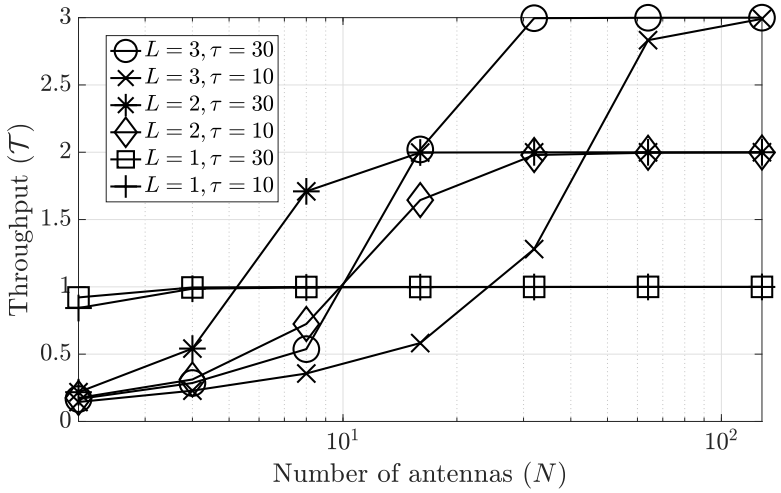


Figure 3.17: LCMMSE: Effect of N on the throughput.

In Fig. 3.17, the system throughput is plotted against the number of antennas at the BS for different system loads and pilot lengths. Under all configurations, it is observed that increasing N has a significant impact, and the peak throughput achieved reaches its maximum of $\mathcal{T} = L$. For $L = 1$, changing either τ or N does not have a significant impact and the throughput remains $\mathcal{T} = 1$ for $N \geq 4$. For $L = 2$, the throughput reaches the peak $\mathcal{T} = 2$ for $N \geq 32$. Similarly, for a high load of $L = 3$, the throughput reaches the peak, $\mathcal{T} = 3$, for $N \geq 128$. It is observed that at $L = 2,\ N = 16$ and $L = 3,\ N = 32$, improving τ greatly improves the throughput. Increasing the number of antennas increases the array gain and the decoding capability of the regularized zero forcing decoder at the BS, which in turn leads to more users getting decoded. In particular, the dramatic rise in the throughput from $N = 8$ to 32 for a high load of $L = 3$ shows the effectiveness of the number of antennas in improving the throughput.

Fig. 3.18 shows the impact of cell edge SNR on the throughput. In the noise-limited

respectively.

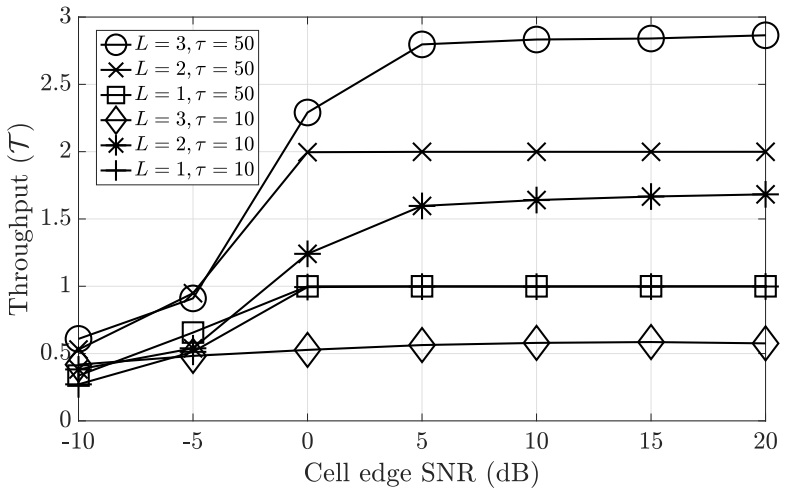


Figure 3.18: LCMMSE: Impact of SNR on the throughput.

regime (SNR $<$ 0 dB), an increase in SNR sharply increases the throughput. Beyond an SNR of 5 dB, increasing SNR only marginally increases the throughput for all L and τ, and the system becomes interference-limited for $\tau = 10$. This is because, at low τ, both signal and interference powers get scaled equally, and the SINR remains roughly constant. At $\tau = 50$, for $L = 1$ and 2, the optimal throughputs can be obtained at a cell edge SNR = 0 dB. However, the throughput for $L = 3$ saturates beyond an SNR of 5 dB and does not yield the optimal throughput of $\mathcal{T} = 3$ due to a high load and low N. In summary, the system throughput can be improved by increasing the pilot length, number of antennas, and cell edge SNR, but these need to be increased judiciously, keeping the other parameters in mind.

Fig. 3.19 shows the effect of regularization parameter λ on the throughput of the system when LCMMSE estimation is employed, with $L = 4$ and $N = 16$. As λ is varied from 10^{-6} to 10^{0}, the curves go from ZF on the left to RZF in the middle and finally to MRC on the right. ZF has poor performance at low τ due to low quality channel estimates. As τ is increased past 60, ZF starts performing better and achieves close to the optimal throughput of $\mathcal{T} = 4$. RZF shows a huge improvement over both ZF and MRC for $\lambda \in (10^{-4}, 10^{-2})$ and achieves near-optimal throughput for $\tau = 40$. The interference suppression capability of RZF helps it achieve a large improvement over both

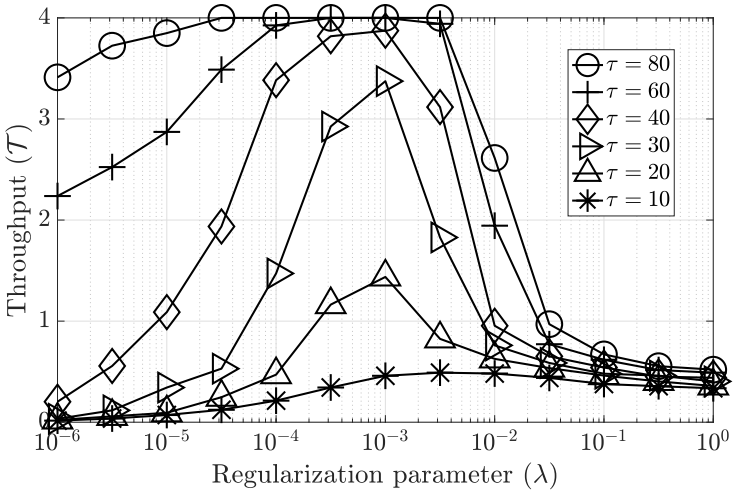


Figure 3.19: LCMMSE: Effect of λ and τ on the throughput.

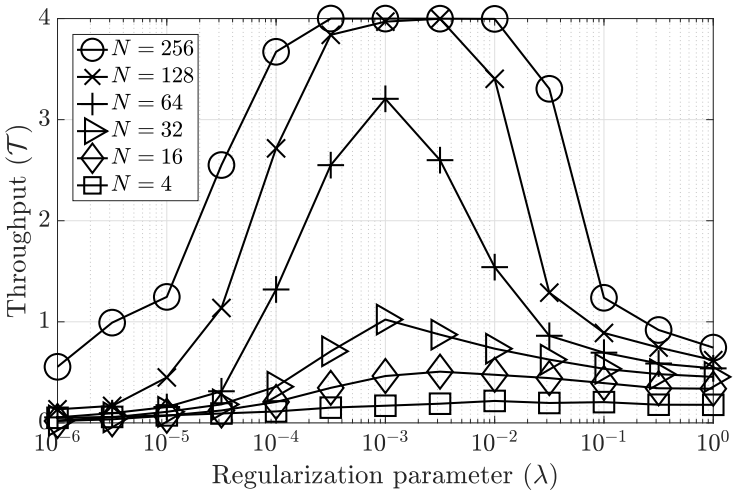


Figure 3.20: LCMMSE: Impact of λ and N on the throughput.

ZF and MRC, especially for $\tau = 40$. MRC does not have the interference suppression capability of RZF, and thus the performance saturates at a low value for all τ. Similar observations can be made on the joint effect of λ and N as seen in Fig. 3.20, where $L = 4$ and $\tau = 10$. Since $\tau = 10$ is fixed, no significant improvement in the throughput is seen up to $N = 64$. Increasing both τ and N together yields much better throughput. Further, since the performance of RZF remains constant over a wide range of λ, it is not important to choose its value very accurately, and any choice in this interval would result in near-optimal throughput.

3.9.3 Difficulty of DE

This section explains the difficulty of using the density evolution (DE) process to calculate closed-form theoretical expressions for the asymptotic performance of IRSA-type protocols. The DE algorithm is iterative in nature, and only the final failure probability p_∞ decides the effective throughput of the system. We can also employ the differential evolution algorithm [26] to find the optimal degree distribution. While this provides a numerical recipe to calculate the optimized degree distributions, it usually does not lead to further insights into the system performance, due to the numerical nature of the DE process.

The DE process does not yield a closed form expression for $\mathcal{T}$ or PLR not just for IRSA with our system model, but even for the most basic variants of IRSA (such as IRSA with a Gaussian multiple access channel (GMAC) or CRDSA).

1. For slotted ALOHA, which is the most basic variant of IRSA, users cannot be decoded if there are collisions. In this case, $\theta_r = \mathbb{1}\{r = 1\}$, and the failure probabilities become $q_i = 1, p_i = 1 - \exp(-L) = p_\infty$. Here $\mathbb{1}\{\cdot\}$ is the indicator function. Since each user transmits only once, the required degree distributions are $\phi(x) = x, \lambda(x) = 1, \psi(x) = \exp(-L(1-x))$. Thus, the packet loss rate is $\mathsf{PLR} = \phi(p_\infty) = 1 - \exp(-L)$, and the throughput can be calculated as $\mathcal{T} = Le^{(-L)}$. Hence, the peak throughput can be calculated by setting the derivative with respect to L to be equal to zero, which can be found as $\mathcal{T}^* = 1/e \simeq 0.37$ at inflection load

of $L^* = 1$.

2. For CRDSA, each user transmits only twice, and thus, the degree distributions are $\phi(x) = x^2, \lambda(x) = x$, and $\psi(x) = \exp(-2L(1-x))$. Similar to slotted ALOHA, users cannot be decoded if there are collisions, and thus $\theta_r = \mathbb{1}\{r = 1\}$. The failure probabilities can be found as $q_i = p_{i-1}, p_i = 1 - \exp(-2Lq_i)$, and thus, $q_{i+1} = 1 - \exp(-2Lq_i)$, and the throughput can be found as $\mathcal{T} = L\phi(p_\infty)$. In this case, we cannot find a closed form expression for the PLR or for the throughput, and we need to empirically find where the throughput peaks. Empirically, the inflection load can be found to be $L^* = 0.5$.

3. For IRSA with the GMAC, the collision model $\theta_r = \mathbb{1}\{r = 1\}$ is considered. Using a node-perspective user degree distribution $\phi(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$ [17] results in $\overline{d} = 3.6,\ \lambda(x) = 0.28x + 0.23x^2 + 0.49x^7$, and $\psi(x) = \exp(-3.6L(1-x))$. Thus, the failure probabilities can be found as $q_i = \lambda(p_{i-1}), p_i = 1 - \exp(-L\bar{d}q_i)$, and the throughput can be found as $\mathcal{T} = L\phi(p_\infty)$. Similar to the previous case, we cannot find a closed-form analytical expression for PLR or $\mathcal{T}$. Thus, we need to resort to empirical means to find a relationship between the throughput and the system parameters (such as L, $\phi(x)$, $\lambda(x)$, and $\bar{d}$).

4. For IRSA with the GMAC and a MUD [75], the K-collision model is considered, where $\theta_r = \mathbb{1}\{1 \leq r \leq K\}$. Here, the BS can decode K or fewer packets correctly. Thus, with any generic node-perspective user degree distribution, the failure probability p_i can be calculated as

$$p_i = 1 - e^{-\bar{c}q_i} \sum_{r=1}^{K} \frac{(\bar{c}q_i)^{r-1}}{(r-1)!} = \frac{\gamma_{\text{inc}}(K, L\bar{d}q_i)}{\Gamma(K)}. \tag{3.81}$$

Thus, $q_{i+1} = \lambda\left(\dfrac{\gamma_{\text{inc}}(K, L\bar{d}q_i)}{\Gamma(K)}\right)$ is the final recursive equation that determines the performance of the system. Once again, we cannot find a closed form expression for $\mathcal{T}$, and we need to empirically find a relationship between the throughput and the system parameters (such as K, L, $\phi(x)$, $\lambda(x)$, and $\bar{d}$).

Thus, slotted ALOHA is the only variant of IRSA for which we can obtain a closed form expression for the throughput. As we have demonstrated above, for all of CRDSA or IRSA with GMAC and a single-user decoder or IRSA with GMAC and a multi-user decoder, it is not possible to find closed form expressions for the throughput or the packet loss rate of the system. The system model we have considered in this chapter is far more complicated than any of these: we account for path loss, fading, MIMO channel (and thus, possibly MUD), pilot contamination, and CSI errors. Thus, since we do not have visibility into the DE process, it is hard to obtain a closed form expression for the throughput using DE for our system model also.

3.9.4 Pilot Power Boosting

In this section, we analyze the impact of *pilot power boosting*. So far, we had considered the pilot power P^{p} and data power P to be equal for simplicity. In pilot power boosting [107, 108], the pilot symbols are typically transmitted at a higher SNR compared to the data symbols. In Fig. 3.21 and Fig. 3.22, we plot the throughput of IRSA for $L = 2$ and $L = 3$, respectively. The setup for this is with MMSE channel estimation, $N = 8$, $\tau = 8$, $T = 50$, $\lambda = 10^{-2}$, and $\gamma_{\text{th}} = 10$. In both the figures, the x-axis is the cell edge SNR, which we consider to be equal to the data SNR, and the pilot SNR has been boosted appropriately above the data SNR according to the legends, e.g., $P^{\mathsf{p}}_{\text{dB}} = P_{\text{dB}} + 3$ dB for a 3 dB boost.

In Fig. 3.21, we observe that at very low cell edge SNR, the throughput is very low, and the optimal throughput of $\mathcal{T} = 2$ is achieved at a cell edge SNR of 5 dB. We observe that at very low cell edge SNR and at SNRs that are optimal for the decoding process, performing pilot power boosting does not improve the throughputs. This is because at low cell edge SNR, the system is noise-limited, and the SINR does not improve much by pilot power boosting. Pilot power boosting does not help much when the power of the channel estimation error is lower than noise variance. Similarly, at high cell edge SNRs, the optimal performance is already reached without any boosting. Thus, in the intermediary SNR region, the performance marginally improves by pilot power boosting. In this regime, increasing pilot power helps reduce the power of the channel estimation

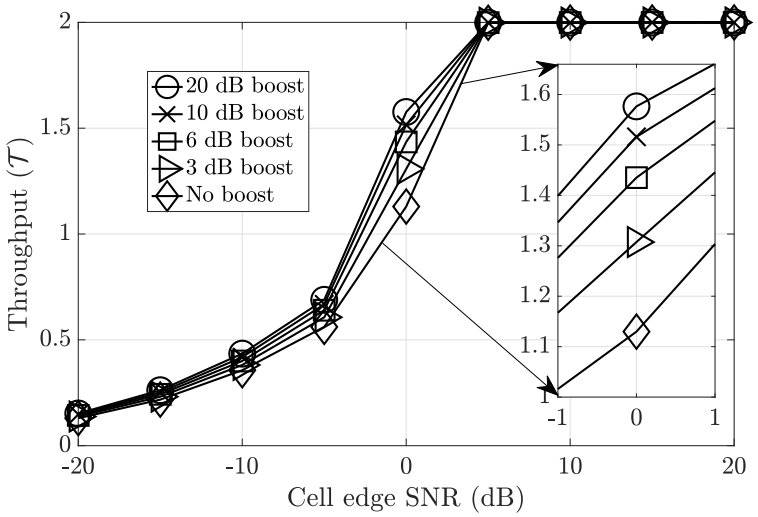


Figure 3.21: Pilot power boosting with $L = 2$.

errors and this in turn can help the packets achieve an SINR that exceeds the capture threshold. This is also the region in which the system is just around the inflection load (as revealed by the density evolution process), and thus, boosting pilot power helps increase the throughput.

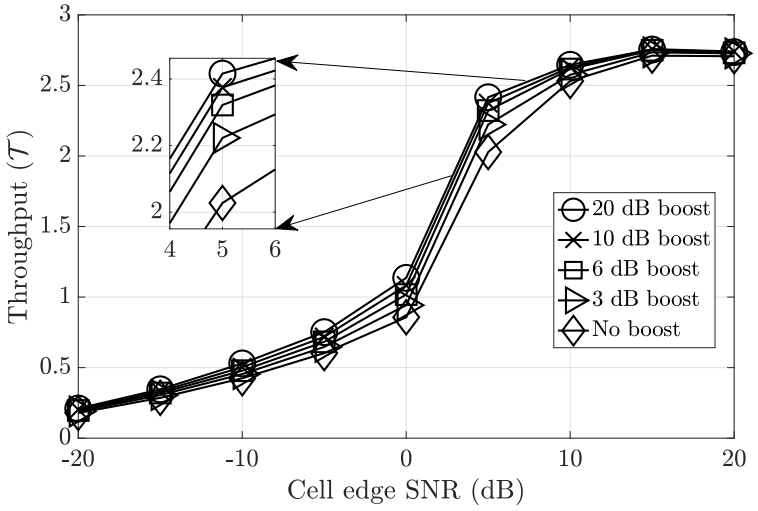


Figure 3.22: Pilot power boosting with $L = 3$.

In Fig. 3.22, we observe similar trends as seen previously, with one change. Here, unlike the previous figure, the optimal throughput of $\mathcal{T} = 3$ is never achieved. At a load

of $L = 3$ and high SNRs, the system is already too interference limited for the power boosting to affect it. However, pilot power boosting has the same impact at high SNRs as we observed in the previous plot. Thus, pilot power boosting does not have a significant impact on the decodability of the packet in most regimes. It helps only at intermediary SNRs, and hence we next focus on boosting only in this regime.

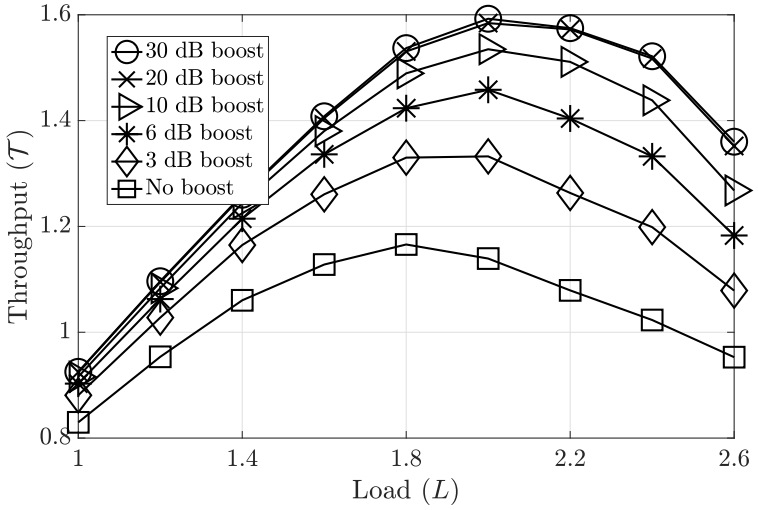


Figure 3.23: Pilot power boosting with cell edge SNR = 0 dB.

In Fig. 3.23, we plot the throughput of IRSA with pilot power boosting, under identical settings as the previous figures, and an intermediary cell edge SNR = 0dB. As seen in the previous figures, operating at this cell edge SNR yields a benefit when we perform boosting. Here, we observe that pilot power boosting from 3dB and up to 20dB drastically improves the performance. At $L = 1.8$, the throughput improves from $\mathcal{T} = 1.15$ with no boost to $\mathcal{T} = 1.55$ with 20 dB boost, which is a 34.78% improvement. At $L = 2$, the throughput improves from $\mathcal{T} = 1.13$ with no boost to $\mathcal{T} = 1.59$ with 20 dB boost, which is a 40.71% improvement. Thus, pilot power boosting can yield very good improvements, albeit only at certain intermediate SNRs.

3.9.5 How to generate the APM at the BS?

In this section, we describe how we can generate the APM at the BS, which justifies the assumption that the BS knows the APM. In conventional IRSA, each packet header has a pointer to the position of other replicas. Therefore, once a packet is successfully decoded, the other replicas can be located and the interference of the replicas can be removed from other REs.

In a practical system, the BS can generate the APM by using a common seed available at the users and the BS. The assumption that the APM is known at the BS is present in other existing works on IRSA as well. The first paper on IRSA [7] says "In practical implementations, the overhead due to the inclusion of pointers in the header of the burst may be reduced by adopting more efficient techniques. One may include in the header the repetition degree for the burst together with a random seed, out of which it is possible to reconstruct (by a pre-defined pseudo-random number generator) the positions of the burst replicas." The authors in [74] point out that "Each replica contains in the header information about the location of all other related replicas. In principle, this information can be agreed in advance through, e.g., a shared seed for a random number generator."; [109] mentions that "This can be achieved by some pseudo-random mechanism known by both the transmit and receive ends."; [110] says that "In practical implementations, the overhead due to the inclusion of pointers in the segment header may be reduced by adopting more efficient techniques. For fixed k, one may include in the segment header the code index h together with a random seed, out of which it is possible to reconstruct (by a pre-defined pseudo-random number generator) the positions of the n_h segments." All of these existing works describe the generation of the APM at the BS via a common pre-defined seed, either after decoding one packet or before decoding any packet. In the former case, one column of the APM is revealed either after one packet is decoded, while in the latter case, the APM is fully known at the BS.

We now describe how this can be implemented in practice. One of the simplest and

widely used pseudo-random number generator (PRNG) is the linear congruential generator [111]. This generator uses the recurrence relation

$$X_{n+1} = (aX_n + b) \bmod m, \tag{3.82}$$

where X_i is the ith pseudo-random number, X_0 is called the seed, a is called the multiplier, b is called the increment, and m is called the modulus [112]. Here, $X_0 \geq 0, m > 0, a > 0, \& \; b \geq 0$ completely specify the PRNG, with $X_0, a, b < m$. The detailed discussion of optimal multipliers [113] and fast parallel generation of numbers [114] is beyond the scope of this work. Simple PRNGs such as the above can be programmed into the hardware, and the generation of such a random number only takes about $1 - 3$ clock cycles. Further, we can compute the APM offline and store it beforehand at the BS.

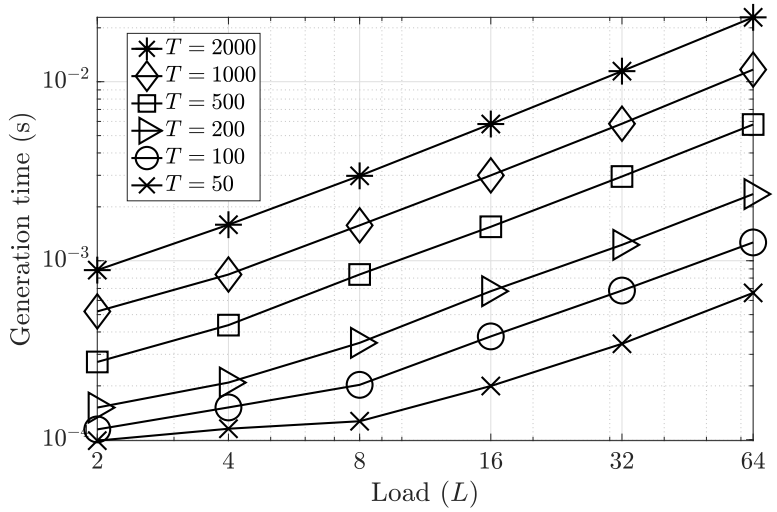


Figure 3.24: APM generation time vs. system load.

The seed to be used in the PRNG can be made of the frame index and the user ID. For example, the seed that is used to generate the APM coefficients for the ith user at the BS could be generated by appending the current frame index and the user ID, as Seed = [Current Frame Index; User ID]. For example, let there be a million users (6 decimal digits) and a thousand frames (3 decimal digits). This would require a total of 9 decimal digits to represent the seed. With a binary representation, a million users would need

around 20 bits to represent the user IDs, and a thousand frames would need around 10 bits to represent the frame index: it goes from 0 to 1023 and then cycles back to 0.

Using the above specified PRNG, we report the generation time of the random numbers required to compute the APM in Fig. 3.24. This is plotted in a log-log scale, and thus, the generation time increases linearly with the load. This setup is with typical PRNG settings such as $m = 2^{16} - 1$, $b = 11$, and $a = 47485$ [113]. Along the x-axis, we can see that the generation time increases linearly whenever the number of users is doubled. The number of users at any point can be calculated as $M = L \times T/p_a$, with $p_a = 0.1$ (this accounts for user activity). For $L = 64$, the points for $T = 2000$, 1000, 500, 200, 100, 50 correspond to $M = 1.28 \times 10^6$, 6.4×10^5, 3.2×10^5, 1.28×10^5, 6.4×10^4, and 3.2×10^4 users, respectively. Thus, the generation time for a million users is not too much and can be implemented with ultra-low complexity PRNGs such as the ones described above. There are also other RNGs that run faster, e.g., the Mersenne twister RNG.

If d_i is the repetition factor of the ith user, we need to generate $1+d_i$ random numbers for the ith user. That is, the first random number is needed to generate the repetition factor d_i itself from the degree distribution, and then, we need to generate d_i number of RE indices from the available T REs, without replacement. Thus, the BS would need to generate a total of $\sum_{i=1}^{M}(1 + d_i) \approx M(1 + \bar{d})$ random numbers at the BS on average in every frame. For $M = 10^6$ users, we need to generate $M(1 + \bar{d})$ random numbers. With $T = 1000$, we need 10 bits to store the above random numbers each. Thus, with $\bar{d} = 3/4/5$, we would need 40/50/60 Megabits to store the required RE indices in each frame. With 1024 frames, we would need 40/50/60 Gigabits, or equivalently 5/6.25/7.5 Gigabytes. Thus, it is true that the lookup table can be large for offline generation and storage. However, one could store partial random numbers, e.g., just the d_is and generate the remaining random number on the fly, as required, during the run-time of the algorithm.

Further, the BS is assumed to have enough computational power and storage for all the signal processing that needs to be performed in every frame; it is trying to decode a fraction of active users among millions of users and perform further processing. In such a

setting, if the BS is computationally powerful to do everything else in the entire receiver chain from UAD to channel estimation and data decoding, the generation and/or look-up of the APM is not a significant overhead for the BS. Also, the scaling of the UAD algorithm to a massive number of users is a high dimensional problem, which can be solved using several tools involving partial matrix processing at each stage. These aspects are beyond the scope of this thesis.

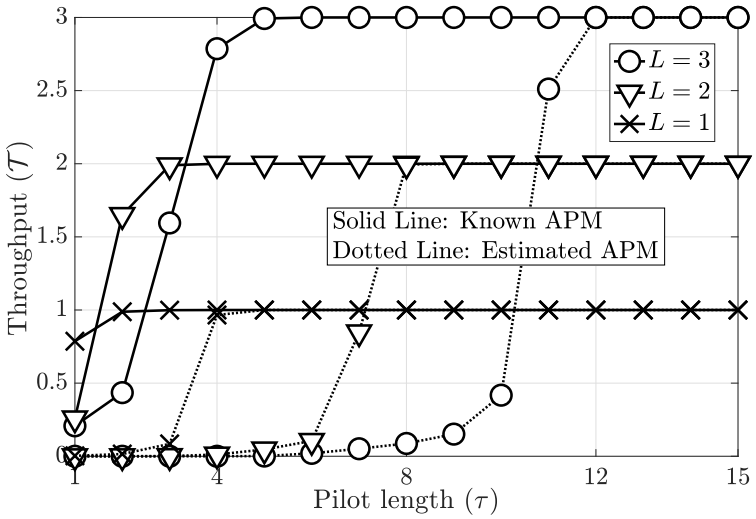


Figure 3.25: Comparison between known and estimated APM.

The MSBL scheme also estimates the APM from the received symbols as seen in Fig. 3.25. This estimation is done via the sparsity structure of IRSA and not by using pointers to packet replicas stored in the packet header. Thus, the performance with the estimated APM is a worst case scenario, and we can only perform better by using pointers to packet replicas or knowing the APM fully. The estimated APM curves are generated by applying MSBL on an RE-by-RE basis, and this does not need the knowledge of the APM or the path loss coefficients. We see that the estimated APM curves achieve the optimal throughputs (i.e., $\mathcal{T} = L$) for $L = 1, 2, 3$ at $\tau = 4, 8, 12$, up from $\tau = 2, 3, 6$ in the known-APM case, respectively, and beyond that, the performance is the same as that of known-APM. This shows that with a few additional pilot symbols, we can do away with the assumption of knowing the APM and path loss coefficients. Thus, the knowledge of the APM is not a strong assumption on the performance of IRSA, and it entails in

only a moderate increase in the pilot length required. Since the BS benefits due to the performance gain of the system (as a consequence of the assumption of known APM), it is advantageous to generate the APM at the BS and exploit this performance improvement, even at the cost of the computational overhead involved.

3.9.6 Choice of Pilot Sequences

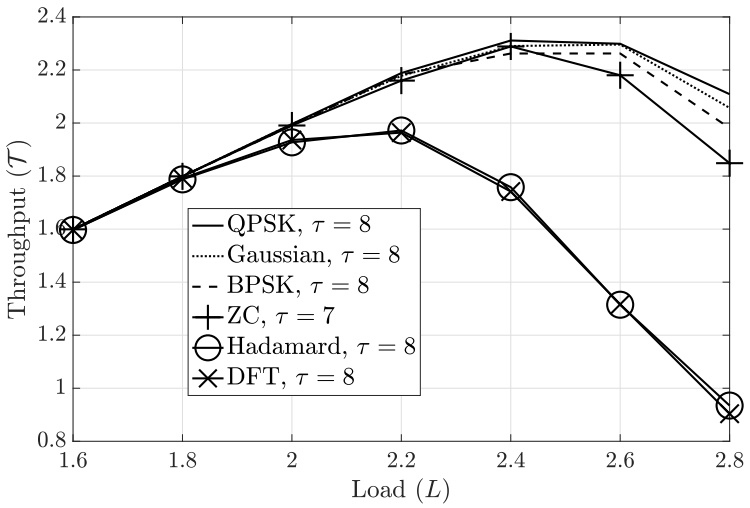


Figure 3.26: Performance comparison of different pilot sequences.

In this section, we explore the effect of different pilot sequences on the throughput of IRSA. In Fig. 3.26, we compare the throughput obtained when several orthogonal and non-orthogonal pilot sequence sets are used, with $T = 50$, $N = 16$, $\gamma_{\text{th}} = 6$, and $\lambda = 1$. The non-orthogonal pilots, termed as BPSK and QPSK, contain random pilot symbols belonging to the respective PSK constellations, and Zadoff-Chu (ZC) sequences are generated according to [115]. ZC sequences require prime pilot length τ [116, 117], we use $\tau = 7$. Mutually orthogonal pilot sequences, such as Hadamard and discrete Fourier transform (DFT), are limited by the length of the pilot sequence τ, i.e., τ mutually orthogonal pilot sequences of length τ can be generated. Thus, we perform orthogonal pilot reuse (OPR), where each user randomly selects a pilot sequence from the available set of τ pilots, similar to [63] and [118].

Random QPSK, Gaussian, and BPSK pilots have identical performance, and ZC sequences result in a marginally lower throughput at high loads. Similar observations are made in [53,119]. Any choice of the non-orthogonal pilots generally yield similar throughputs. Thus, the choice of pilots is not a major point of concern. In the $\mathcal{T} = L$ regime, the performance of non-orthogonal pilots is better than OPR. Too much pilot reuse, which is worse with OPR due to the smaller set of available pilots, deteriorates the performance. The use of non-orthogonal pilots provides better diversity, since there is a richer set of pilot sequences, leading to better performance [53].

Part 2

Chapter 4

User Activity Detection in IRSA

4.1 Introduction

Massive machine-type communications (mMTC) is expected to serve around a million devices per square kilometer [1]. Typical mMTC devices transmit short packets to a central base station (BS), and are sporadically active [2]. To facilitate efficient random access for such mMTC applications, distributed grant-free random access (GFRA) protocols need to be used, as they can serve a large number of users without incurring a large signaling overhead [5]. Since only a subset of users are active in any frame in mMTC [2], it is essential for the BS to detect the set of users that are active, before proceeding to perform channel estimation and data decoding. This process is termed *user activity detection (UAD).* Furthermore, without UAD, the BS would waste valuable resources attempting to decode a large number of users that have not transmitted any packets, i.e., users that are inactive. Knowing the subset of active users not only saves computational resources by helping the BS decide which users it needs to decode, it is also important for channel estimation, as will be seen in the sequel. Errors arising from the UAD process, namely, false positives and false negatives, deteriorate the channel estimates computed at the BS, which in turn affects the data decoding. Hence, it is crucial to account for these errors while analyzing the performance of GFRA protocols.

The contents of this chapter is published in a full length journal paper in the IEEE

Transactions on Signal Processing in 2023 [15].

4.1.1 Motivation

Irregular repetition slotted aloha (IRSA) is a popular GFRA protocol in which users transmit replicas of their packets in multiple resource blocks (RBs) in a frame [7]. Each RB can accommodate a whole data packet. In IRSA, each user samples their repetition factor d from a predefined distribution independently of other users and then transmits replicas of its packet in d RBs chosen uniformly at random from the set of all RBs in the frame [7]. The indices of the RBs in which the users transmit their packet replicas define the *access pattern matrix (APM),* which we assume is known at the BS. This assumption is explained in Sec. 4.2. Due to the structure of the APM, applying existing UAD algorithms to IRSA can lead to suboptimal performance. In particular, it is necessary to combine the information available in each RB in a principled manner, to accurately detect the active users.

Typically, UAD and channel estimation is performed by the BS using pilots transmitted by the users in their packet headers. If the users employ mutually orthogonal pilots, there is no pilot contamination, making UAD and channel estimation simple. However, the length of orthogonal pilots scales linearly with the total number of users, and hence the pilot overhead quickly overshadows the data payload size as the number of users gets large [6]. Thus, non-orthogonal pilots are used, and the resulting pilot contamination leads to both UAD errors and channel estimation errors. These effects must be accounted for while analyzing the performance of IRSA. The main goal of this chapter is to understand the effect of system parameters such as pilot length, SNR, and the number of antennas at the BS on the performance of IRSA, accounting for UAD errors, channel estimation errors, and pilot contamination.

4.1.2 Working of the IRSA protocol

Early works in IRSA considered the collision model in which only *singleton* RBs are decodable. Singleton RBs are RBs in which only a single user has transmitted, and since

there are no collisions in such RBs, users can be decoded with high probability. The decoding proceeds in iterations, and occurs via inter-RB successive interference cancellation (SIC), which refers to the removal of a packet replica from all other RBs where the same packet was transmitted. The decoding stops when there is no singleton RB available. The throughput of IRSA under the collision model is at most one packet per RB [7], which is achievable asymptotically with the number of RBs and users if the soliton distribution is used to generate the repetition factors [72].

In the case where the BS is equipped with multiple antennas, multiple users could potentially be decoded in an RB [49], and thus singleton RBs are not necessary for decoding. Any user with a sufficiently high instantaneous signal to interference plus noise ratio (SINR) can first be decoded, and the contribution of that user can be removed from the same RB. This process, termed intra-RB SIC, refers to the removal of interference of a packet replica from the same RB within which it was decoded. After the user with the highest SINR is decoded, other users could potentially be decoded as well. By performing both intra-RB and inter-RB SIC, the packet replicas of different users are removed from all RBs wherein the same user has transmitted a packet. This model, which we use in this chapter, is termed as the *SINR threshold* model, and it yields a higher throughput than the conventional singleton decoding model.

4.1.3 Existing Works in IRSA

IRSA has been studied with the SINR threshold model under scalar Rayleigh fading channels with perfect channel state information (CSI) [17] and pure path loss channels [8]. Coded slotted aloha (CSA), which is a variant of IRSA, was recently analyzed with imperfect SIC [88]. The authors in [63] studied CSA with an acknowledgement mechanism between frames. IRSA was analyzed with an SIC limit, i.e., a limit on the maximum number of packets that can be recovered in each RB, in [86]. The average age of information in IRSA in mMTC has also been studied [79]. We have previously [11, 12] analyzed the IRSA protocol accounting for channel estimation and pilot contamination, with perfect UAD (See Chapter 3). In contrast, this chapter focuses on UAD in IRSA, and analyzes

the impact of UAD errors on the throughput.

4.1.4 Existing Works for UAD

To the best of our knowledge, the problem of UAD in IRSA has not yet been considered in the literature. Further, none of the existing works study the performance of IRSA accounting for UAD errors, path loss, MIMO fading, pilot contamination, and channel estimation errors. An initial study into estimating the number of active users in IRSA was conducted in [93], which does not identify the subset of active users. UAD has been studied for massive random access outside the context of IRSA [13, 14]. The activity matrix to be estimated has jointly-sparse columns, i.e., columns that have the same sparse support [94]. Typical UAD solutions involve compressed sensing-based solutions [53] or a maximum aposteriori probability (MAP) detection [95]. The sparse Bayesian learning (SBL) framework has been employed to perform UAD in mMTC [96]. Faster SBL algorithms for UAD in mMTC have also been developed [97]. Other low complexity algorithms for UAD include approximate message passing [98] and orthogonal matching pursuit [99]. These approaches, however, cannot be used in IRSA due to the structure imposed by the APM. A naïve approach would be to perform UAD on an RB-by-RB basis and declare users inactive if they are found to be inactive in all the RBs. As we will show, this approach is inefficient and results in large error rates, especially when non-orthogonal pilots are used.

4.1.5 Contributions

This chapter proposes a novel UAD algorithm for IRSA, and analyzes the throughput of IRSA, accounting for UAD and channel estimation errors. Our main contributions are:

1. We develop a novel Bayesian algorithm to detect the set of active users in IRSA in Sec. 4.3. UAD in IRSA is a joint-sparse signal recovery problem with a measurement model with an important twist: different and unknown subsets of the row indices of the joint-sparse matrix participate in different measurements. Our algorithm is

an enhancement to the multiple sparse Bayesian learning (MSBL) algorithm [16] to cater to this scenario.[1]

2. We derive the channel estimates at the BS for users in all RBs in IRSA, acquired via non-orthogonal pilots, in Sec. 4.4. We also derive the Cramér-Rao bound (CRB) on the mean squared error (MSE) of the channels estimated by our proposed UAD algorithm. We show that a genie-aided minimum MSE (MMSE) estimator (that has knowledge of the second-order statistics and the user activities) achieves the CRB. Later, we also empirically show that the MSE of the channel estimates output by the proposed UAD algorithm meets the CRB.

3. Next, we analyze the SINR achieved by all the users in all RBs in Sec. 4.5, accounting for UAD errors, channel estimation errors, and pilot contamination. The SINR expression allows us to determine the throughput of IRSA, accounting for the effect of UAD errors.

Our numerical experiments in Sec. 4.6 show that there is at least a 4-fold reduction in the number of pilot symbols required to achieve a similar UAD performance as that of existing approaches. The loss in performance due to UAD errors can be recuperated by judiciously choosing the system parameters such as pilot length, number of antennas, and SNR. In essence, it is vital to account for both UAD and channel estimation when analyzing the throughput of IRSA.

Notation

The symbols a, $\mathbf{a}$, $\mathbf{A}$, $[\mathbf{A}]_{i,:}$, $[\mathbf{A}]_{:,j}$, $\mathbf{0}_N$, $\mathbf{1}_N$, and $\mathbf{I}_N$ denote a scalar, a vector, a matrix, the ith row of $\mathbf{A}$, the jth column of $\mathbf{A}$, all-zero vector of length N, all ones vector of length N, and an identity matrix of size $N \times N$, respectively. $[\mathbf{a}]_{\mathcal{S}}$ and $[\mathbf{A}]_{:,\mathcal{S}}$ denote the elements of $\mathbf{a}$ and the columns of $\mathbf{A}$ indexed by the set $\mathcal{S}$ respectively. $\text{diag}(\mathbf{a})$ is a diagonal matrix with diagonal entries given by $\mathbf{a}$, whereas $\text{blkdiag}(\mathbf{A}, \mathbf{B})$ is a block diagonal matrix with

[1]Our UAD algorithm can be applied to other variants of IRSA such as CSA since it entails only a minor change in the structure of the APM.

$\mathbf{A}$ and $\mathbf{B}$ as the diagonal blocks. $\mathbf{A} \otimes \mathbf{B}$ is the Kronecker product of $\mathbf{A}$ and $\mathbf{B}$, and $\mathbf{A} \succeq \mathbf{B}$ denotes that $\mathbf{A} - \mathbf{B}$ is positive semi-definite. $[N]$ denotes the set $\{1, 2, \ldots, N\}$. $|\cdot|$, $\|\cdot\|$, $\|\cdot\|_F$, $[\cdot]^T$, $[\cdot]^H$, $\mathbb{E}[\cdot]$, and $\mathbb{E}_{\mathbf{a}}[\cdot]$ denote the magnitude, ℓ_2 norm, frobenius norm, transpose, conjugate transpose, unconditional expectation, and the expectation conditioned on $\mathbf{a}$, respectively. The superscript $\mathtt{p}$ is used as a descriptive superscript in association with a symbol that is related to the *pilots*. All the other superscripts (or subscripts) that have not been defined as above are indices.

4.2 System Model

An IRSA system is considered with M single-antenna users communicating with a BS equipped with N antennas. The users are assumed to be spread randomly within a cell, with the BS located at the cell center. These users communicate with the BS over frames consisting of T RBs. The RBs can be slots, subcarriers or both. In every frame, a small subset of the M users, called *active* users, attempt to deliver a packet each to the BS. In a given frame, the *activity coefficient* of the mth user is denoted by a_m, where $a_m = 1$ if the mth user is active, and $a_m = 0$ otherwise. Note that a_m can change from one frame to the next, and the subset of active users (and hence a_m) is unknown at the BS. The users transmit replicas of their packet according to the random matrix $\mathbf{G} \in \{0, 1\}^{T \times M}$, which is called the *access pattern matrix (APM)*. Here, $g_{tm} = [\mathbf{G}]_{tm}$ is the access pattern coefficient of the mth user in the tth RB. If $g_{tm} = 1$, the mth user transmits its packet in the tth RB provided $a_m = 1$, and if $g_{tm} = 0$, the mth user does not transmit its packet in the tth RB even if $a_m = 1$. If $a_m = 0$, the mth user is *inactive* in the current frame, and does not transmit in any RB.

At the BS, the received signal in the tth RB is a superposition of the packets transmitted by the active users that have chosen to transmit in the tth RB. The packets of the users undergo both path loss and fading. We assume that the path loss component and the second-order statistics of the fading component are known at the BS, and that

the fading channel remains constant for the duration of an RB.[2] Each packet replica comprises a header containing pilot symbols and a payload which includes the coded data and cyclic redundancy check (CRC) symbols. In the pilot phase, if $a_m = 1$, the mth user transmits a $\tau-$length pilot sequence $\mathbf{p}_m \in \mathbb{C}^\tau$ in each packet replica (i.e., if $g_{tm} = 1$). Each pilot symbol has an average power $P^{\mathtt{p}}$, and the average power of the pilot sequence is $\mathbb{E}[\|\mathbf{p}_m\|^2] = \tau P^{\mathtt{p}}$. The received pilot signal $\mathbf{Y}_t^{\mathtt{p}} \in \mathbb{C}^{N\times\tau}$ at the BS across the N antennas in the tth RB is thus

$$\mathbf{Y}_t^{\mathtt{p}} = \sum\nolimits_{m=1}^{M} a_m g_{tm} \mathbf{h}_{tm} \mathbf{p}_m^H + \mathbf{N}_t^{\mathtt{p}}, \tag{4.1}$$

where $\mathbf{N}_t^{\mathtt{p}} \in \mathbb{C}^{N\times\tau}$ is the complex additive white Gaussian noise at the BS with $[\mathbf{N}_t^{\mathtt{p}}]_{:,j} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_N, N_0\mathbf{I}_N), \forall\, j \in [\tau]$ and $t \in [T]$, where N_0 is the noise variance. Here, $\mathbf{h}_{tm} = \sqrt{\beta_m}\mathbf{v}_{tm}$ is the uplink channel vector of the mth user in the tth RB, where β_m is the known path loss coefficient and $\mathbf{v}_{tm}$ is the unknown fading vector with $\mathbf{v}_{tm} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_N, \sigma_{\mathtt{h}}^2\mathbf{I}_N),\ \forall\ t \in [T]$ and $m \in [M]$, and channel variance $\sigma_{\mathtt{h}}^2$.

In the data phase, if $a_m = 1$, the mth user transmits a data symbol[3] x_m in each packet replica that it transmits. The data symbol satisfies $\mathbb{E}[x_m] = 0$ and $\mathbb{E}[|x_m|^2] = P$, where P denotes the data power. The received data signal $\mathbf{y}_t \in \mathbb{C}^N$, at the BS in the tth RB, is

$$\mathbf{y}_t = \sum\nolimits_{m=1}^{M} a_m g_{tm} \mathbf{h}_{tm} x_m + \mathbf{n}_t, \tag{4.2}$$

where $\mathbf{n}_t \in \mathbb{C}^N$ is the complex additive white Gaussian noise at the BS with $\mathbf{n}_t \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_N, N_0\mathbf{I}_N), \forall\ t \in [T]$.

In IRSA, if the mth user is active, it samples its repetition factor d_m from a predefined distribution, independently of other users. Then it chooses d_m RBs from a total of T RBs uniformly at random, and transmits replicas of its packet in these d_m RBs. The APM is formed as $[\mathbf{G}]_{tm} = g_{tm}, t \in [T], m \in [M]$, where $g_{tm} = 1$ if the mth user has chosen

[2]For simplicity of exposition, we consider i.i.d. Rayleigh fading between the users and the BS in each RB, although it is straightforward to extend the results to the correlated fading scenario.

[3]To derive SINR in any given RB, only one data symbol is written out from the multiple data symbols in each packet.

to transmit in the tth RB, and $g_{tm} = 0$ otherwise.[4] This generation of repetition factors is scalable to a large number of users and is completely distributed in nature, and is thus appropriate for mMTC. In practice, the random subset of RBs is generated using a pseudo-random number generator, whose seed completely determines the sequence [7].[5] This seed can be pre-programmed at each user, and made available to the BS. In this case, it is reasonable to assume that the BS has knowledge of $\mathbf{G}$. Also, the APM can be generated in an offline fashion and stored at the BS. However, it is important to note that although the BS knows the subset of RBs in which a user would transmit its packet replicas *if it were active,* the BS still does not have the knowledge of which users are active in a given frame.

4.2.1 SIC-based Decoding in IRSA

The decoding process in IRSA proceeds as follows. The BS first detects the set of active users (denoted by $\hat{a}_m$). Then, it estimates the channels for all the users detected to be active in the RBs for which $g_{tm} = 1$. It uses these channel estimates to combine the received data signal across the BS antennas, and attempts to decode the user's data packet, treating interference from other users as noise. If it successfully decodes any user, which can be verified via the CRC, it performs SIC in all RBs which that user has transmitted, with both inter-RB and intra-RB SIC. The channels are re-estimated for the remaining users, and this decoding process proceeds iteratively.

In this chapter, the decoding of any user's packet is abstracted into an SINR threshold model as in [8, 17]. That is, the packet can be decoded correctly if and only if the SINR of an active user's packet in an RB is greater than a threshold denoted by γ_{th}, the packet can be decoded correctly. The value of γ_{th} is usually chosen to be ≥ 1 for a narrowband system [17]; it is a parameter for the purposes of our analysis.

We now briefly describe how to evaluate the performance of IRSA under the abstract

[4]Note that users who are inactive in a given frame can also be virtually considered to have chosen the RBs in which they are scheduled to transmit, even though they do not transmit in any RB.

[5]For example, the seed could be a function of the current frame index and the user ID, such as, seed = [Current Frame Index; User ID]. Using simple pseudo-random number generators and with no computational speed-up, we can generate 10^6 random numbers within a few ms on a mid-range laptop.

SINR threshold-based decoding model. We first estimate the user activity coefficients for all users over the frame. For the users detected to be active, we compute channel estimates and SINR achieved in all RBs in which the users detected to be active have transmitted their packets. This SINR accounts for the CSI available at the BS and errors in the UAD process, as we will see in Sec. 4.5. If we find a user with SINR $\geq \gamma_{\text{th}}$ in some RB, we mark the data packet as having been decoded successfully, and remove that user from the set of users yet to be decoded. Also, the contribution of the user's packet is removed from all RBs that contain a replica of that packet. In the next iteration, the channels are re-estimated from the residual pilot symbols after SIC, the SINRs are recomputed in all RBs, and the decoding of users' packets continues. The iterations stop when no additional users are decoded in two successive iterations or if all users detected to be active have been successfully decoded. The system throughput $\mathcal{T}$ is calculated as the number of correctly decoded unique packets divided by the number of RBs. Note that the throughput accounts for packet losses that occur due to users that are incorrectly detected to be active, as well as due to failures in the SIC-based decoding process.

The rest of the chapter is organized as follows. Sec. 4.3 outlines the proposed UAD algorithm, and Sec. 4.4 describes the channel estimation process. The detailed derivation of the SINR accounting for both UAD errors and channel estimation errors is presented later in Sec. 4.5.

4.3 User Activity Detection

In this section, we describe our user activity detection (UAD) algorithm. For this purpose, we consider the conjugate transpose of the received pilot signal in the tth RB from (4.1) as $\overline{\mathbf{Y}}_t \triangleq \mathbf{Y}_t^{\mathrm{p}H}$, with $\overline{\mathbf{N}}_t \triangleq \mathbf{N}_t^{\mathrm{p}H}$. The signal $\overline{\mathbf{Y}}_t$ can be factorized into the product of the

product of two matrices as follows:

$$\underbrace{\overline{\mathbf{Y}}_t}_{\tau\times N} = \underbrace{[\mathbf{p}_1,\ldots,\mathbf{p}_M]}_{\mathbf{P}} \underbrace{\begin{bmatrix} a_1 g_{t1}\mathbf{h}_{t1}^H \\ \vdots \\ a_M g_{tM}\mathbf{h}_{tM}^H \end{bmatrix}}_{\mathbf{X}_t} + \underbrace{\overline{\mathbf{N}}_t}_{\tau\times N}. \tag{4.3}$$

Here, $\mathbf{P} \in \mathbb{C}^{\tau\times M}$ contains the known pilot sequences of the M users as its columns, and $\mathbf{X}_t \in \mathbb{C}^{M\times N}$ contains the tth row of the known APM $\mathbf{G}$, the unknown user activity coefficients, and the unknown channels. Note that the ith row of $\mathbf{X}_t$ is nonzero only if $a_i = 1$ and $g_{ti} = 1$, i.e., when the ith user is active and transmits in the tth RB.

Table 4.1: Hyperparameter notation in Algorithm 3.

Symbol	Quantity
$\boldsymbol{\gamma}$	Hyperparameter vector of all M users
$\boldsymbol{\Gamma}$	Diagonal matrix with $\boldsymbol{\gamma}$ as it's diagonal entries
$\boldsymbol{\gamma}_t$	Hyperparameter vector of the M_t users who would have transmitted in the tth RB had they been active
$\boldsymbol{\Gamma}_t$	Diagonal matrix with $\boldsymbol{\gamma}_t$ as it's diagonal entries
$\boldsymbol{\gamma}^j/\boldsymbol{\gamma}_t^j/\boldsymbol{\Gamma}_t^j$	Hyperparameters in the jth MSBL iteration
$\tilde{\boldsymbol{\gamma}}_t^j$	Auxiliary variable used to store $\boldsymbol{\gamma}^j$
γ_{pr}	Threshold used to declare support

Let $\mathcal{G}_t = \{i \in [M] \mid g_{ti} = 1\}$ be the set of users who would have transmitted in the tth RB, had they all been active in the current frame, and $M_t = |\mathcal{G}_t|$ be the number of such users. Since the BS has knowledge of $\mathcal{G}_t$, it can remove the contributions of the users who do not transmit in the tth RB. We thus obtain a column-reduced pilot matrix $\mathbf{P}_t \triangleq [\mathbf{P}]_{:,\mathcal{G}_t} \in \mathbb{C}^{\tau\times M_t}$ and a row-reduced channel matrix $\mathbf{Z}_t \triangleq [\mathbf{X}_t]_{\mathcal{G}_t,:} \in \mathbb{C}^{M_t\times N}$ in the tth

RB. Hence, (4.3) can be rewritten as

$$\underbrace{\overline{\mathbf{Y}}_t}_{\tau\times N} = \underbrace{\left[\mathbf{p}_{i_1},\ldots,\mathbf{p}_{i_{M_t}}\right]}_{\mathbf{P}_t} \underbrace{\begin{bmatrix} a_{i_1}\mathbf{h}_{ti_1}^H \\ \vdots \\ a_{i_{M_t}}\mathbf{h}_{ti_{M_t}}^H \end{bmatrix}}_{\mathbf{Z}_t} + \underbrace{\overline{\mathbf{N}}_t}_{\tau\times N}, \tag{4.4}$$

where $\mathcal{G}_t = \{i_1, i_2, \ldots, i_{M_t}\}$. The above results in an under-determined system of linear equations in the mMTC regime (since $\tau \ll M_t \ll M$), with rows of $\mathbf{Z}_t$ being either all zero or all nonzero. The columns of $\mathbf{Z}_t$ thus share a common support, i.e., they are joint-sparse. This structure is referred to as a multiple measurement vector (MMV) recovery problem in compressed sensing. Note that the above step reduces the dimension of the matrix to be estimated, but does not solve the UAD problem. The support recovery of $\mathbf{Z}_t$ from (4.4) can be performed with well known algorithms from compressed sensing literature to recover the activity coefficients in the each of the T RBs. By doing so, we would obtain an RB-specific activity estimate for each user. However, the activity coefficient for any user is the same across the T RBs, and thus we need a way to infer $\{a_i\}$ using information available in all T RBs. One naïve way to do this is to declare users to be active in the current frame if they are detected to be active in at least t' RBs, where t' is a parameter that can be optimized. As we will see, this leads to very poor performance compared to the algorithm presented in the sequel. In the following paragraph, we briefly discuss Multiple sparse Bayesian learning (MSBL) [16], which sets the stage for presenting our enhancement that combines the information gleaned from each RB in a principled manner. The notation we will now use is described in Table 4.1.

MSBL is an empirical Bayesian algorithm that recovers the joint-sparse columns of $\mathbf{Z}_t$ from linear underdetermined measurements $\overline{\mathbf{Y}}_t$. In MSBL, a hierarchical Gaussian prior is imposed on the columns of $\mathbf{Z}_t$ as

$$p(\mathbf{Z}_t;\boldsymbol{\gamma}_t) = \prod_{n=1}^{N} p([\mathbf{Z}_t]_{:,n};\boldsymbol{\gamma}_t) = \prod_{n=1}^{N} \mathcal{CN}(\mathbf{0}_{M_t},\boldsymbol{\Gamma}_t), \tag{4.5}$$

where the columns of $\mathbf{Z}_t$ are i.i.d. and $\mathbf{\Gamma}_t = \text{diag}(\boldsymbol{\gamma}_t)$. Here, the elements of $\boldsymbol{\gamma} \in \mathbb{R}_+^M$ are unknown hyperparameters and $\boldsymbol{\gamma}_t \triangleq [\boldsymbol{\gamma}]_{\mathcal{G}_t} \in \mathbb{R}_+^{M_t}$ picks the hyperparameters for the users who would have transmitted in the tth RB had they been active in the current frame. Such a hierarchical Bayesian model is known to result in sparse solutions for the maximum likelihood estimates of $\boldsymbol{\gamma}_t$ [16, 120]. Recovering the hyperparameters would yield the users' activities since $[\boldsymbol{\gamma}]_m$ models the variance of the mth user's channel. The hyperparameters in $\boldsymbol{\gamma}_t$ are estimated by maximizing the log-likelihood $\log p(\overline{\mathbf{Y}}_t; \boldsymbol{\gamma}_t)$, with $p(\overline{\mathbf{Y}}_t; \boldsymbol{\gamma}_t) = \prod_{n=1}^N p([\overline{\mathbf{Y}}_t]_{:,n}; \boldsymbol{\gamma}_t)$. Here, $p([\overline{\mathbf{Y}}_t]_{:,n}; \boldsymbol{\gamma}_t) = \mathcal{CN}(\mathbf{0}_\tau, \mathbf{\Sigma}_{\boldsymbol{\gamma}_t})$ because of the linear measurement model in (4.4), with $\mathbf{\Sigma}_{\boldsymbol{\gamma}_t} = N_0\mathbf{I}_\tau + \mathbf{P}_t\mathbf{\Gamma}_t\mathbf{P}_t^H$. Thus, the log-likelihood reads as

$$\log(p(\overline{\mathbf{Y}}_t; \boldsymbol{\gamma}_t)) \propto -N \log |\mathbf{\Sigma}_{\boldsymbol{\gamma}_t}| - \text{Tr}(\mathbf{\Sigma}_{\boldsymbol{\gamma}_t}^{-1} \overline{\mathbf{Y}}_t \overline{\mathbf{Y}}_t^H). \tag{4.6}$$

This is a non-convex function of $\boldsymbol{\gamma}_t$, and its maximizer cannot be found in closed form. In MSBL, expectation maximization (EM) is used to optimize the cost function iteratively [121].

Let j denote the iteration index in EM. In the jth MSBL iteration, the notations $\boldsymbol{\gamma}_t^j \triangleq [\boldsymbol{\gamma}^j]_{\mathcal{G}_t}$ and $[\boldsymbol{\gamma}_t^j]_i$ denote the hyperparameter vector of the users in the set denoted by $\mathcal{G}_t$ and the ith entry of $\boldsymbol{\gamma}_t^j$, respectively. The EM procedure consists of two steps in each iteration. The first step, termed the E-step, updates the covariance $\mathbf{\Sigma}_t^{j+1}$ and mean $\boldsymbol{\mu}_{tn}^{j+1}$ of the posterior $p([\mathbf{Z}_t]_{:,n} | [\mathbf{Y}_t]_{:,n}, \boldsymbol{\gamma}_t^j)$ as [120]

$$\mathbf{\Sigma}_t^{j+1} = \mathbf{\Gamma}_t^j - \mathbf{\Gamma}_t^j \mathbf{P}_t^H (N_0\mathbf{I}_\tau + \mathbf{P}_t\mathbf{\Gamma}_t^j\mathbf{P}_t^H)^{-1}\mathbf{P}_t\mathbf{\Gamma}_t^j, \tag{4.7}$$

$$\boldsymbol{\mu}_{tn}^{j+1} = N_0^{-1}\mathbf{\Sigma}_t^{j+1}\mathbf{P}_t^H[\overline{\mathbf{Y}}_t]_{:,n},\ n \in [N]. \tag{4.8}$$

The second step, known as the M-step, updates the hyperparameter for the ith user in the tth RB as

$$[\boldsymbol{\gamma}_t^{j+1}]_i = \frac{1}{N}\sum_{n=1}^N ([\mathbf{\Sigma}_t^{j+1}]_{i,i} + |[\boldsymbol{\mu}_{tn}^{j+1}]_i|^2),\ i \in [M_t]. \tag{4.9}$$

This M-step estimates the variance of the channel of the ith user in the tth RB, and this

hyperparameter update contains the information from the tth RB only. The above two steps are performed for all T RBs. Before the next E-step, the hyperparameter updates across the RBs must be combined.

4.3.1 The Proposed UAD Algorithm

The main novelty of our UAD algorithm lies in exploiting the access pattern coefficients across RBs to find a single hyperparameter update, which we term as the new M-step. For this purpose, let $\tilde{\boldsymbol{\gamma}}_t^{j+1} \in \mathbb{R}_+^M$ be an auxiliary variable for the tth RB that is updated as $[\tilde{\boldsymbol{\gamma}}_t^{j+1}]_{\mathcal{G}_t} = \boldsymbol{\gamma}_t^{j+1} \in \mathbb{R}_+^{M_t}$ and zero otherwise. The hyperparameter update for the mth user is obtained at the BS by combining the estimated hyperparameters for that user across all the RBs using the knowledge of g_{tm} as

$$[\boldsymbol{\gamma}^{j+1}]_m = \frac{1}{d_m} \sum_{t=1}^{T} g_{tm} [\tilde{\boldsymbol{\gamma}}_t^{j+1}]_m, \ m \in [M]. \tag{4.10}$$

Here $d_m = \sum_{t=1}^{T} g_{tm}$ is the repetition factor of the mth user. Note that, in conjunction with (4.9), this new M-step is equivalent to executing an M-step that maximizes the overall log-likelihood, $\sum_{t=1}^{T} \log p(\overline{\mathbf{Y}}_t; \boldsymbol{\gamma}_t)$, based on the knowledge of the APM at the BS. Effectively, since it estimates the variance of the channel of the mth user by averaging the estimated variances of the channels in each RB, it combines the information obtained from all RBs in computing the hyperparameter update. By iterating between the E- and M-steps, the EM algorithm converges to a saddle point or a local maximizer of the overall log-likelihood [121]. Further, the EM procedure has been empirically shown to correctly recover the support of $\mathbf{Z}_t$, provided τ and N are large enough [16]. In turn, this leads to significantly lower false positive and false negative rates in UAD, as we will empirically show later.

The overall UAD procedure is summarized in Algorithm 3. The algorithm is run for $j_{\max}$ iterations. As the iterations proceed, the hyperparameters corresponding to inactive users converge to zero, resulting in sparse estimates. At the end of the EM iterations, the estimated activity coefficient $\hat{a}_m$ for the mth user is obtained by thresholding $[\boldsymbol{\gamma}^{j_{\max}}]_m$ at

Algorithm 3: UAD in IRSA

Input: $\tau, N, T, M, N_0, \mathbf{G}, \mathbf{P}, \{\overline{\mathbf{Y}}_t\}_{t=1}^T, \{d_m\}_{m=1}^M, \gamma_{\text{pr}}, j_{\max}$

Initialize: $\boldsymbol{\gamma}^0 = \mathbf{1}_M$

Compute: $\mathcal{G}_t = \{m \in [M] \mid g_{tm} = 1\},\ M_t = |\mathcal{G}_t|, \mathbf{P}_t = [\mathbf{P}]_{:,\mathcal{G}_t}, t \in [T]$

for $j = 0, 1, 2, \ldots, j_{\max}$ **do**

  for $t = 1, 2, \ldots, T$ **do**

    Compute: $\boldsymbol{\Gamma}_t^j = \text{diag}([\boldsymbol{\gamma}^j]_{\mathcal{G}_t})$

$$\boldsymbol{\Sigma}_t^{j+1} = \boldsymbol{\Gamma}_t^j - \boldsymbol{\Gamma}_t^j \mathbf{P}_t^H (N_0 \mathbf{I}_\tau + \mathbf{P}_t \boldsymbol{\Gamma}_t^j \mathbf{P}_t^H)^{-1} \mathbf{P}_t \boldsymbol{\Gamma}_t^j$$

$$\boldsymbol{\mu}_{tn}^{j+1} = N_0^{-1} \boldsymbol{\Sigma}_t^{j+1} \mathbf{P}_t^H [\overline{\mathbf{Y}}_t]_{:,n},\ 1 \le n \le N$$

    $[\boldsymbol{\gamma}_t^{j+1}]_i = \frac{1}{N} \sum_{n=1}^N ([\boldsymbol{\Sigma}_t^{j+1}]_{i,i} + |[\boldsymbol{\mu}_{tn}^{j+1}]_i|^2), i \in [M_t]$

    $[\tilde{\boldsymbol{\gamma}}_t^{j+1}]_{\mathcal{G}_t} = \boldsymbol{\gamma}_t^{j+1},\ [\tilde{\boldsymbol{\gamma}}_t^{j+1}]_{[M]\setminus\mathcal{G}_t} = \mathbf{0}_{M-M_t}$

  end

  $[\boldsymbol{\gamma}^{j+1}]_m = \frac{\sum_{t=1}^T g_{tm} [\tilde{\boldsymbol{\gamma}}_t^{j+1}]_m}{\sum_{t=1}^T g_{tm}},\ 1 \le m \le M$

end

Output: $\hat{a}_m = \begin{cases} 1, & [\boldsymbol{\gamma}^{j_{\max}}]_m \ge \gamma_{\text{pr}} \\ 0, & [\boldsymbol{\gamma}^{j_{\max}}]_m < \gamma_{\text{pr}} \end{cases},\ 1 \le m \le M,$

$\hat{\mathbf{Z}}_t = [\boldsymbol{\mu}_{t1}^{j_{\max}} \boldsymbol{\mu}_{t2}^{j_{\max}} \ldots \boldsymbol{\mu}_{tN}^{j_{\max}}],\ 1 \le t \le T$

a value γ_{pr}. The algorithm also outputs the MAP estimates of the channels $\hat{\mathbf{X}}_t$ in each of the T RBs with $[\hat{\mathbf{X}}_t]_{\mathcal{G}_t,:} = \hat{\mathbf{Z}}_t$ and $[\hat{\mathbf{X}}_t]_{[M]\setminus\mathcal{G}_t,:} = \mathbf{0}_{(M-M_t)\times N}$, and the channel estimates of users across all RBs are stacked in $\hat{\mathbf{X}} = [\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_T]$.

We now discuss the complexity of our algorithm in terms of the number of floating point operations (flops). Each MSBL iteration has $\mathcal{O}(\tau^2 M)$ flops, if the pilot matrix is of size $\tau \times M$ [16]. In our algorithm, each iteration contains T RBs, where the size of the reduced pilot matrix is $\tau \times M_t$ in the tth RB. Also, the new M-step has lower complexity order than the E-step. Thus, the overall per-iteration complexity of Algorithm 3 is $\mathcal{O}(\tau^2 M_S)$, where $M_S = \sum_{t=1}^{T} M_t \approx \bar{d}M$, where $\bar{d}$ is the average repetition factor.

Based on the estimated activity $\hat{a}_i$ and the true activity a_i, the set of all users can be divided into four disjoint subsets

$$\mathcal{P} = \{i \in [M] \mid \hat{a}_i a_i = 1\}, \tag{4.11a}$$

$$\mathcal{F} = \{i \in [M] \mid \hat{a}_i(1 - a_i) = 1\}, \tag{4.11b}$$

$$\mathcal{M} = \{i \in [M] \mid (1 - \hat{a}_i)a_i = 1\}, \tag{4.11c}$$

$$\mathcal{I} = \{i \in [M] \mid (1 - \hat{a}_i)(1 - a_i) = 1\}. \tag{4.11d}$$

$\mathcal{P}$ is the true positive set of users, i.e., the users that are correctly detected to be active. $\mathcal{F}$ is the false positive set of users, i.e., the users that are detected to be active and are truly inactive. $\mathcal{M}$ is the false negative set of users, i.e., the users that are detected to be inactive, but are actually active. $\mathcal{I}$ is the true negative set of users, i.e., the users that are correctly detected to be inactive. False positive and false negative users together form the errors in the UAD process, and the error rates for such users are discussed in Sec. 4.6. After the active users are detected, the next task is to estimate the channels from the active users. However, before describing channel estimation, we take a small detour to explain why traditional compressed sensing approaches are not effective for frame-based UAD in IRSA-based multiple access.

4.3.2 Why One-Shot UAD Does Not Work

By stacking the received signal in (4.3) across all RBs, we can estimate the user activity coefficients in one-shot across all RBs. We now briefly explain why this performs poorly. The received pilot signals in all RBs can be stacked as

$$\overline{\mathbf{Y}} = [\overline{\mathbf{Y}}_1 \overline{\mathbf{Y}}_2 \dots \overline{\mathbf{Y}}_T] = \mathbf{P}\mathbf{X} + [\overline{\mathbf{N}}_1 \overline{\mathbf{N}}_2 \dots \overline{\mathbf{N}}_T], \tag{4.12}$$

$$\mathbf{X} = \begin{bmatrix} a_1 g_{11} \mathbf{h}_{11}^H & \dots & a_1 g_{T1} \mathbf{h}_{T1}^H \\ \vdots & \ddots & \vdots \\ a_M g_{1M} \mathbf{h}_{1M}^H & \dots & a_M g_{TM} \mathbf{h}_{TM}^H \end{bmatrix} \in \mathbb{C}^{M \times NT}. \tag{4.13}$$

The above structure is *not* an MMV recovery problem because the rows of $\mathbf{X}$ are not completely all zero or all nonzero. If the ith user is inactive, then the ith row of $\mathbf{X}$ is all zero. However if the ith user is active, then the ith row of $\mathbf{X}$ is not all nonzero. Only the blocks of the ith row corresponding to the RBs in which the ith user has transmitted in (i.e., where $g_{ti} = 1$) are all nonzero and the other blocks are all zero. Since IRSA results in the transmission of replicas in only a small subset of the T RBs, only a few blocks of the ith row are nonzero. Different blocks of each row of $\mathbf{X}$ corresponding to active users have different block-sparse supports. If an MMV recovery algorithm is applied across all RBs in one shot as in (4.13), a pilot length of $\tau = \Omega(M_a \log \frac{M}{M_a})$ can achieve a vanishing activity error rate as $N \to \infty$, where M_a is the average number of active users in each RB [122]. For example, with $M = 1500$ and $M_a = 150$, $\tau = \Omega(346)$ achieves vanishing error rates in a massive MIMO regime. These pilot lengths are infeasible in practice, and thus, in practical regimes of interest, one-shot UAD performs poorly.

4.4 Channel Estimation

In addition to performing UAD, Algorithm 3 also outputs an initial channel estimate for each user that is detected to be active, as a by-product. However, as the decoding iterations proceed, the interference cancellation can help improve the accuracy of the channel estimates, when the channels of the remaining users are re-estimated after each

SIC operation. We now derive MMSE channel estimates in each decoding iteration for all the users that have been detected to be active.[6] MMSE channel estimation is also required to compute meaningful expressions for the SINR [104].

Since MMSE estimates are recomputed in every iteration, the signals and channel estimates are indexed by the decoding iteration k. Let the set of users who have not yet been decoded in the first $k-1$ iterations be denoted by $\mathcal{S}_k$, with $\mathcal{S}_k^m \triangleq \mathcal{S}_k \setminus \{m\}$, and $\mathcal{S}_1 = [M]$. The received pilot signal at the BS, in the tth RB during the kth decoding iteration, is

$$\mathbf{Y}_t^{\mathrm{p}k} = \sum\nolimits_{i\in\mathcal{S}_k} a_i g_{ti} \mathbf{h}_{ti} \mathbf{p}_i^H + \mathbf{N}_t^{\mathrm{p}}. \tag{4.14}$$

In this section, we assume perfect SIC for simplicity of analysis; we study the performance variation under imperfect SIC in Sec. 4.6.5. This received signal is contributed from all users who are truly active in the current frame. The BS wishes to compute channel estimates for users who are detected to be active, i.e., for the users in $\hat{\mathcal{A}} = \{i \in [M] \mid \hat{a}_i = 1\}$, which is output by Algorithm 3. For this purpose, the received pilot signal is right combined with the pilot $\mathbf{p}_m$, $\forall m \in \hat{\mathcal{A}} \cap \mathcal{G}_t \cap \mathcal{S}_k$, to obtain the post-combining pilot signal as

$$\mathbf{y}_{tm}^{\mathrm{p}k} = \mathbf{Y}_t^{\mathrm{p}k} \mathbf{p}_m = \sum\nolimits_{i\in\mathcal{S}_k} a_i g_{ti} \mathbf{h}_{ti} (\mathbf{p}_i^H \mathbf{p}_m) + \mathbf{N}_t^{\mathrm{p}} \mathbf{p}_m, \tag{4.15}$$

which is further used for estimating the channel between the BS and the user in the tth RB [49]. The MMSE channel estimate is given by the following theorem.

Theorem 4.1 ▶ Channel Estimation in IRSA Accounting for UAD Errors.

The MMSE estimate $\hat{\mathbf{h}}_{tm}^k$ of the channel $\mathbf{h}_{tm}$ is calculated from the post-combining

[6] Specifically, this is the LCMMSE estimator similar to the previous chapter. The MMSE estimator is described in Sec. 4.9.1.

pilot signal as

$$\hat{\mathbf{h}}_{tm}^{k} = \eta_{tm}^{k} \mathbf{y}_{tm}^{\mathtt{p}k}, \quad \forall m \in \mathcal{S}_k, \tag{4.16}$$

where $\eta_{tm}^{k} \triangleq \dfrac{\hat{a}_m g_{tm} \beta_m \sigma_{\mathtt{h}}^2 \|\mathbf{p}_m\|^2}{N_0 \|\mathbf{p}_m\|^2 + \sum_{i \in \mathcal{S}_k} \hat{a}_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2 |\mathbf{p}_i^H \mathbf{p}_m|^2}$. Further, the estimation error $\tilde{\mathbf{h}}_{tm}^{k} \triangleq \hat{\mathbf{h}}_{tm}^{k} - \mathbf{h}_{tm}$ is uncorrelated with the channel $\mathbf{h}_{tm}$, and is distributed as $\mathcal{CN}(\mathbf{0}_N, \delta_{tm}^{k} \mathbf{I}_N)$. Here, δ_{tm}^{k} is the estimation error variance and is given by

$$\delta_{tm}^{k} = \beta_m \sigma_{\mathtt{h}}^2 \left(\frac{\sum_{i \in \mathcal{S}_k^m} |\mathbf{p}_i^H \mathbf{p}_m|^2 \hat{a}_i a_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2 + N_0 \|\mathbf{p}_m\|^2}{\sum_{i \in \mathcal{S}_k} |\mathbf{p}_i^H \mathbf{p}_m|^2 \hat{a}_i a_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2 + N_0 \|\mathbf{p}_m\|^2} \right).$$

Proof. See Sec. 4.8.1. □

Remark 1: The channel estimate is composed of a scaling coefficient η_{tm}^{k} and the post-combining pilot signal $\mathbf{y}_{tm}^{\mathtt{p}k}$. η_{tm}^{k} is computed at the BS and is a function of the estimated activity coefficients $\hat{a}_i$. Thus, false positive users feature in the denominator of η_{tm}^{k} and affect the channel estimates of other users. The BS also computes channel estimates for these false positive users.[7] Since false negative users are detected to be inactive, the BS does not account for the interference caused by them while computing η_{tm}^{k}. From (4.15), $\mathbf{y}_{tm}^{\mathtt{p}k}$ contains signals from other truly active users if pilots are not orthogonal, and is thus a function of the true activity coefficients a_i. Also, false negative users contribute to $\mathbf{y}_{tm}^{\mathtt{p}k}$, and thus both types of errors affect the estimates of other users.

Remark 2: In the above theorem, δ_{tm}^{k} accounts for the pilot contamination from other true positive users. False positive users are omitted from the expression for δ_{tm}^{k} because such users do not contaminate the pilots of other users. Only true positive users contribute to δ_{tm}^{k}. When orthogonal pilots are used, $\mathbf{p}_i^H \mathbf{p}_m = 0, \forall i \neq m$, there is no pilot contamination, and thus $\delta_{tm}^{k} = \beta_m \sigma_{\mathtt{h}}^2 N_0 / (\hat{a}_m a_m g_{tm} \beta_m \sigma_{\mathtt{h}}^2 \|\mathbf{p}_m\|^2 + N_0)$.

[7] Since false positive users will fail an error check, the BS can potentially try to identify such users as data decoding proceeds and compute better quality channel estimates. However, we make no such assumption, and thus, our channel estimation procedure models a worst-case scenario where false positive users contaminate the channel estimates of other true positive users.

4.4.1 Cramér-Rao Bound

In this subsection, we derive the Cramér-Rao bound (CRB) [123] on the mean squared error (MSE) of the channel estimated under the hierarchical Bayesian model given by (4.5). The signal $\overline{\mathbf{Y}}_t = \mathbf{P}_t\mathbf{Z}_t + \overline{\mathbf{N}}_t$ from (4.4) is first vectorized as

$$\underbrace{\overline{\mathbf{y}}_t}_{N\tau\times 1} \triangleq \mathrm{vec}(\overline{\mathbf{Y}}_t) = \underbrace{\mathbf{\Phi}_t}_{N\tau\times NM_t}\underbrace{\overline{\mathbf{z}}_t}_{NM_t\times 1} + \underbrace{\overline{\mathbf{n}}_t}_{N\tau\times 1}, \tag{4.17}$$

where $\mathbf{\Phi}_t \triangleq (\mathbf{I}_N \otimes \mathbf{P}_t)$, $\overline{\mathbf{z}}_t \triangleq \mathrm{vec}(\mathbf{Z}_t)$, and $\overline{\mathbf{n}}_t \triangleq \mathrm{vec}(\overline{\mathbf{N}}_t)$.

After stacking the received pilot signal across all RBs as $\overline{\mathbf{y}} = [\overline{\mathbf{y}}_1^T, \ldots, \overline{\mathbf{y}}_T^T]^T$, with $\overline{\mathbf{z}} = [\overline{\mathbf{z}}_1^T, \ldots, \overline{\mathbf{z}}_T^T]^T$, $\overline{\mathbf{n}} = [\overline{\mathbf{n}}_1^T, \ldots, \overline{\mathbf{n}}_T^T]^T$, and $\mathbf{\Phi} = \mathrm{blkdiag}\{\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_T\}$, we obtain

$$\overline{\mathbf{y}} = \mathbf{\Phi}\overline{\mathbf{z}} + \overline{\mathbf{n}}. \tag{4.18}$$

Here, we wish to estimate $\overline{\mathbf{z}} \in \mathbb{C}^{NM_S}$ from an observation $\overline{\mathbf{y}} \in \mathbb{C}^{NT\tau}$ via a measurement matrix $\mathbf{\Phi} \in \mathbb{C}^{NT\tau\times NM_S}$, with $M_S = \sum_{t=1}^{T} M_t$. Let $\mathbf{J}$ denote the $NM_S \times NM_S$ Fisher information matrix (FIM) associated with the vector $\overline{\mathbf{z}}$. It is easy to see that $\mathbf{J} = \mathrm{blkdiag}\{\mathbf{J}_1, \ldots, \mathbf{J}_T\}$, where $\mathbf{J}_t$ is the $NM_t \times NM_t$ sub-block of the FIM corresponding to the tth RB. Specifically, the CRB derived in this chapter is the hybrid Cramér-Rao bound [123], which is a bound analogous to the CRB for the estimation problem in MSBL. Due to the block diagonal structure of the FIM, the CRB for any estimate $\hat{\overline{\mathbf{z}}}_t$ of $\overline{\mathbf{z}}_t$ is given by

$$\mathbb{E}[(\hat{\overline{\mathbf{z}}}_t - \overline{\mathbf{z}}_t)(\hat{\overline{\mathbf{z}}}_t - \overline{\mathbf{z}}_t)^H] \succeq \mathbf{J}_t^{-1}. \tag{4.19}$$

Theorem 4.2 ▶ CRB for Channel Estimation in IRSA.

The sub-block of the FIM associated with the channel vector $\overline{\mathbf{z}}_t = \mathrm{vec}(\mathbf{Z}_t)$ in the tth RB is given by

$$\mathbf{J}_t = \mathbf{I}_N \otimes N_0^{-1}\left(\mathbf{P}_t^H\mathbf{P}_t + N_0\mathbf{\Gamma}_t^{-1}\right), \tag{4.20}$$

where $\mathbf{\Gamma}_t = \text{diag}([\boldsymbol{\gamma}]_{\mathcal{G}_t})$ picks the hyperparameters for the M_t users in the tth RB. Further, the CRB for any estimate $[\hat{\mathbf{Z}}_t]_{:,n}$ of $[\mathbf{Z}_t]_{:,n}$ in the tth RB across the nth antenna is given by

$$\mathbb{E}[([\hat{\mathbf{Z}}_t]_{:,n} - [\mathbf{Z}_t]_{:,n})([\hat{\mathbf{Z}}_t]_{:,n} - [\mathbf{Z}_t]_{:,n})^H] \succeq N_0 \left(\mathbf{P}_t^H \mathbf{P}_t + N_0 \mathbf{\Gamma}_t^{-1}\right)^{-1}, \quad 1 \leq n \leq N. \tag{4.21}$$

Proof. See Sec. 4.8.2. □

Remark: Note that the right hand side in (4.21) is independent of the antenna index. Also, from (4.19), the MSE of any estimate $\hat{\mathbf{Z}}_t$ of $\mathbf{Z}_t$ in the tth RB can be bounded below by $\text{Tr}(\mathbf{J}_t^{-1})$ as

$$\mathbb{E}[\|\hat{\mathbf{Z}}_t - \mathbf{Z}_t\|_F^2] \geq \text{Tr}\left(\mathbf{I}_N \otimes N_0 \left(\mathbf{P}_t^H \mathbf{P}_t + N_0 \mathbf{\Gamma}_t^{-1}\right)^{-1}\right) \tag{4.22}$$

$$= N\text{Tr}\left(\mathbf{\Gamma}_t - \mathbf{\Gamma}_t \mathbf{P}_t^H \left(N_0 \mathbf{I}_\tau + \mathbf{P}_t \mathbf{\Gamma}_t \mathbf{P}_t^H\right)^{-1} \mathbf{P}_t \mathbf{\Gamma}_t\right), \tag{4.23}$$

where the last step is obtained by using the Woodbury matrix identity and $\text{Tr}(\mathbf{I}_N \otimes \mathbf{A}) = N\text{Tr}(\mathbf{A})$. Considering the signals received across the entire frame, the effective MSE of the estimate $\hat{\mathbf{X}}$ of $\mathbf{X}$ can thus be bounded as

$$\text{MSE} = \mathbb{E}[\|\hat{\mathbf{X}} - \mathbf{X}\|_F^2] = \sum\nolimits_{t=1}^{T} \mathbb{E}[\|\hat{\mathbf{X}}_t - \mathbf{X}_t\|_F^2] \tag{4.24}$$

$$= \sum\nolimits_{t=1}^{T} \mathbb{E}[\|\hat{\mathbf{Z}}_t - \mathbf{Z}_t\|_F^2] \tag{4.25}$$

$$\geq NN_0 \sum\nolimits_{t=1}^{T} \text{Tr}\left(\mathbf{P}_t^H \mathbf{P}_t + N_0 \mathbf{\Gamma}_t^{-1}\right)^{-1}. \tag{4.26}$$

The channel variance can be calculated as

$$\mathbb{E}[\|\mathbf{X}\|_F^2] = \sum\nolimits_{t=1}^{T} \mathbb{E}[\|\mathbf{X}_t\|_F^2] = \sum\nolimits_{t=1}^{T} \text{Tr}(\mathbb{E}[\mathbf{Z}_t \mathbf{Z}_t^H]) \tag{4.27}$$

$$= \sum\nolimits_{t=1}^{T} \text{Tr}(\mathbf{I}_N \otimes \mathbf{\Gamma}_t) = N \sum\nolimits_{t=1}^{T} \text{Tr}(\mathbf{\Gamma}_t) \tag{4.28}$$

$$= N \sum\nolimits_{m=1}^{M} d_m [\boldsymbol{\gamma}]_m. \tag{4.29}$$

Hence, the normalized mean squared error (NMSE) of any channel estimate $\hat{\mathbf{X}}$ of $\mathbf{X}$ can be bounded as

$$\text{NMSE} \triangleq \frac{\mathbb{E}[\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2]}{\mathbb{E}[\|\mathbf{X}\|_F^2]} \tag{4.30}$$

$$\geq \frac{N_0}{\sum_{m=1}^{M} d_m[\boldsymbol{\gamma}]_m} \sum_{t=1}^{T} \text{Tr}\left(\mathbf{P}_t^H \mathbf{P}_t + N_0 \boldsymbol{\Gamma}_t^{-1}\right)^{-1}. \tag{4.31}$$

To better understand the above expressions, we consider the case of orthogonal pilots, i.e., $\mathbf{P}_t^H \mathbf{P}_t = \tau P^{\text{p}} \mathbf{I}_{M_t}$, applicable when $\tau \geq M_t$, $\forall t \in [T]$. In this case, the MSE is bounded as

$$\text{MSE} \geq N \sum_{t=1}^{T} \sum_{i=1}^{M_t} \left(\frac{\tau P^{\text{p}}}{N_0} + [\boldsymbol{\Gamma}_t^{-1}]_{i,i}\right)^{-1} \tag{4.32}$$

$$= N \sum_{m=1}^{M} d_m \left(\frac{\tau P^{\text{p}}}{N_0} + \frac{1}{[\boldsymbol{\gamma}]_m}\right)^{-1} = N \sum_{m=1}^{M} \left(\frac{d_m[\boldsymbol{\gamma}]_m}{1 + [\boldsymbol{\gamma}]_m \frac{\tau P^{\text{p}}}{N_0}}\right), \tag{4.33}$$

and the NMSE can be bounded as

$$\text{NMSE} \geq \frac{1}{\sum_{m=1}^{M} d_m[\boldsymbol{\gamma}]_m} \sum_{m=1}^{M} \left(\frac{d_m[\boldsymbol{\gamma}]_m}{1 + [\boldsymbol{\gamma}]_m \frac{\tau P^{\text{p}}}{N_0}}\right). \tag{4.34}$$

The above bound is for a given set of repetition factors $\{d_m\}$, hyperparameters $\boldsymbol{\gamma}$, the pilot SNR $\frac{\tau P^{\text{p}}}{N_0}$, and is independent of the number of antennas N. As $\tau \to \infty$, the MSE goes to zero.

We now describe an estimator that achieves the CRB.

Lemma 2. *Assuming the knowledge of the true hyperparameters, the CRB is achieved by the MMSE channel estimate:*

$$\hat{\mathbf{Z}}_t = \left(\mathbf{P}_t^H \mathbf{P}_t + N_0 \boldsymbol{\Gamma}_t^{-1}\right)^{-1} \mathbf{P}_t^H \overline{\mathbf{Y}}_t. \tag{4.35}$$

Proof. The MSBL algorithm iteratively calculates the MAP estimate. Since the posterior $p([\mathbf{Z}_t]_{:,n}|[\overline{\mathbf{Y}}_t]_{:,n};\boldsymbol{\gamma}_t)$ is Gaussian distributed, the MAP estimate is the same as the mean of the distribution, which coincides with the MMSE estimate in (4.35). Upon substituting the above estimate into the MSE expression in (4.23), it is easy to show that the CRB is achieved. □

Remark: The above estimator requires knowledge of $\boldsymbol{\Gamma}_t$, which in turn needs the user activity coefficients, and is thus a *genie-aided* estimator. In practice, one could use the hyperparameter estimates output by Algorithm 3 in place of $\boldsymbol{\Gamma}_t$ to obtain a "plug-in" MMSE estimator. However, such an estimator need not achieve the CRB. Nonetheless, as empirically shown in Sec. 4.6, the channel estimates obtained using (4.8) does achieve the CRB. (See Figs. 4.5 and 4.6.)

4.5 SINR Analysis

In this section, the SINR of each user in all the RBs where it has transmitted data is derived, accounting for pilot contamination, estimated user activities, and estimated channels. Let ρ_{tm}^k denote the SINR of the mth user in the tth RB in the kth decoding iteration. Similar to (4.2), the received data signal in the tth RB and kth iteration is

$$\mathbf{y}_t^k = \textstyle\sum_{i\in\mathcal{S}_k} a_i g_{ti} \mathbf{h}_{ti} x_i + \mathbf{n}_t. \tag{4.36}$$

Let $M_t^k = |\hat{\mathcal{A}} \cap \mathcal{G}_t \cap \mathcal{S}_k|$ be the number of users who are detected to be active and have transmitted in the tth RB, but have not been decoded in the first $k-1$ iterations. $\hat{\mathcal{A}}$ is obtained as an output of Algorithm 3. A combining matrix $\mathbf{A}_t^k \in \mathbb{C}^{N\times M_t^k}$ is used at the receiver in the tth RB and kth decoding iteration. For each $m \in [M_t^k]$, the vector $\mathbf{a}_{tm}^k = [\mathbf{A}_t^k]_{:,m}$ combines the received data signal as

$$\tilde{y}_{tm}^k = [\mathbf{A}_t^{kH}\mathbf{y}_t^k]_m = \mathbf{a}_{tm}^{kH}\mathbf{y}_t^k. \tag{4.37}$$

$$\tilde{y}_{tm}^{k} = \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}a_m g_{tm}x_m - \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^{k}a_m g_{tm}x_m + \sum\nolimits_{i\in\mathcal{S}_k^m\cap\mathcal{P}} \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}a_i g_{ti}x_i + \sum\nolimits_{i\in\mathcal{S}_k^m\cap\mathcal{M}} \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}a_i g_{ti}x_i + \mathbf{a}_{tm}^{kH}\mathbf{n}_t. \tag{4.38}$$

This post-combining data signal is used to decode the mth user and is composed of five terms as seen in (4.38). The term $T_1 \triangleq \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}a_m g_{tm}x_m$ is the desired signal of the mth user; the term $T_2 \triangleq \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^{k}a_m g_{tm}x_m$ is due to the estimation error $\tilde{\mathbf{h}}_{tm}^{k}$ of the mth user's channel; the term $T_3 \triangleq \sum_{i\in\mathcal{S}_k^m\cap\mathcal{P}} \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}a_i g_{ti}x_i$ models the inter-user interference from other true positive users (who have transmitted in the tth RB and have not yet been decoded); the term $T_4 \triangleq \sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}} \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}a_i g_{ti}x_i$ is the interference from false negative users (who have transmitted in the tth RB, but cannot be decoded since they are declared to be inactive); and $T_5 \triangleq \mathbf{a}_{tm}^{kH}\mathbf{n}_t$ is the additive noise.

To compute the SINR, the power of the post-combining data signal is calculated conditioned on the channel estimates [49]. This is equivalent to computing the power of the post-combining data signal conditioned on the post-combining pilot signal as $\mathbb{E}_{\mathbf{z}}[|\tilde{y}_{tm}^{k}|^2] = \mathbb{E}_{\mathbf{z}}[|T_1+T_2+T_3+T_4+T_5|^2]$. Here, $\mathbf{z}$ contains the post-combining pilot signals of all M_t^k users yet to be decoded. Since noise is uncorrelated with data, T_5 is uncorrelated with the other terms. As MMSE channel estimates are uncorrelated with their estimation errors [49], T_1 is uncorrelated with T_2. Since the data signals of different users are independent, T_3 and T_4 are independent of each other and the other terms as well. Thus, all the five terms are uncorrelated and the power in the received signal is simply the sum of the powers of the individual components

$$\mathbb{E}_{\mathbf{z}}[|\tilde{y}_{tm}^{k}|^2] = \sum\nolimits_{i=1}^{5}\mathbb{E}_{\mathbf{z}}[|T_i|^2]. \tag{4.39}$$

We now compute the SINR in the following theorem.

Theorem 4.3 ▶ SINR Accounting for Channel Estimation Errors and UAD Errors.

The signal to interference plus noise ratio (SINR) achieved by the mth user in the tth RB in the kth decoding iteration can be written as

$$\rho_{tm}^{k} = \frac{\texttt{Gain}_{tm}^{k}}{N_0 + \texttt{Est}_{tm}^{k} + \texttt{MUI}_{tm}^{k} + \texttt{FNU}_{tm}^{k}}, \ \forall m \in \mathcal{S}_k, \tag{4.40}$$

where $\texttt{Gain}_{tm}^{k}$ represents the useful signal power of the mth user, $\texttt{Est}_{tm}^{k}$ represents the interference power caused due to estimation errors of all true positive users, $\texttt{MUI}_{tm}^{k}$ represents the multi-user interference power of other true positive users, and $\texttt{FNU}_{tm}^{k}$ represents the interference power caused due to the false negative users. These can be expressed as

$$\texttt{Gain}_{tm}^{k} = P\hat{a}_m a_m g_{tm} \frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}|^2}{\|\mathbf{a}_{tm}^{k}\|^2}, \quad \texttt{MUI}_{tm}^{k} = P\textstyle\sum_{i\in\mathcal{S}_k^m} \hat{a}_i a_i g_{ti} \frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^{k}|^2}{\|\mathbf{a}_{tm}^{k}\|^2}, \tag{4.41a}$$

$$\texttt{Est}_{tm}^{k} = P\textstyle\sum_{i\in\mathcal{S}_k} \hat{a}_i a_i g_{ti} \delta_{ti}^{k}, \quad \texttt{FNU}_{tm}^{k} = P\textstyle\sum_{i\in\mathcal{S}_k^m} (1-\hat{a}_i) a_i g_{ti} \beta_i \sigma_{\texttt{h}}^2. \tag{4.41b}$$

Proof. See Sec. 4.8.3. □

Remark: The interference components in the SINR expression are contributed only by truly active users, i.e., the true positive and false negative users. False positive users do not contribute towards the received data signal. Even though they do not cause interference, false positive users still affect data decoding of other (true positive) users via their influence on the channel estimates, which also feature in the SINR expression. Further, the SINR for such false positive users is zero.[8] In contrast, false negative users contribute to the received pilot and data signals, affecting both the channel estimates and data decoding of true positive users. Since the BS does not detect or decode such users, their SINR is zero as well, and thus the system performance degrades due to such false negative users. True negative users do not contribute to the received pilot or data signal,

[8]The BS computes noise-based channel estimates for false positive users. Even if the SINR for such users happens to exceed γ_{th}, their packets will fail an error check, and thus their SINR can be set to zero.

and thus do not affect the decoding of other users. Thus, $\rho_{tm}^k = 0,\ \forall\ m\ \in \mathcal{F} \cup \mathcal{M} \cup \mathcal{I}$.

The SINR expression derived in Theorem 4.3 is applicable to any chosen combining scheme.[9] For example, with regularized zero forcing combining [49], $\mathbf{A}_t^k$ is computed as

$$\mathbf{A}_t^k = \hat{\mathbf{H}}_t^k (\hat{\mathbf{H}}_t^{kH} \hat{\mathbf{H}}_t^k + \lambda \mathbf{I}_{M_t^k})^{-1}, \tag{4.42}$$

where λ is the regularization parameter, and $\hat{\mathbf{H}}_t^k$ is an $N \times M_t^k$ matrix containing the channel estimates of the M_t^k users as its columns. The corresponding SINR is obtained by substituting the columns of the above combining matrix into (4.40). The system throughput can now be calculated from (4.40) via the decoding model described in Sec. 4.2.1. We note that, in practice, the BS does not compute the SINR; it simply tries to decode each user that is detected to be active, in the RBs it has chosen for transmission. However, the decoding succeeds only if the SINR exceeds the chosen threshold. Thus, we use the SINR threshold based abstraction to determine which packets are successfully decoded and hence the throughput.

4.6 Numerical Results

In this section, the UAD and channel estimation performance of Algorithm 3 and the impact of UAD errors on the throughput of IRSA are studied via Monte Carlo simulations. In each run, independent realizations of the user activities, user locations, the APM, and the fades experienced by the users are generated. The results in this section are for $T = 50$ RBs, $N_s = 10^3$ Monte Carlo runs, $j_{\max} = 100$ iterations, $\gamma_{\text{pr}} = 10^{-4}$, path loss exponent $\alpha = 3.76$, and channel variance $\sigma_{\mathtt{h}}^2 = 1$ [49]. The pilot sequences are generated as $\mathbf{p}_m \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_\tau, P^{\mathtt{p}} \mathbf{I}_\tau)$ as in [6]. The users are spread uniformly at random locations within a cell of radius $r_{\max} = 1000$ m, and the path loss is calculated as $\beta_m = (r_m / r_0)^{-\alpha}$, where r_m is the radial distance of the mth user from the BS and $r_0 = 100$ m is the reference distance. The soliton distribution [72] with $k_s = 27$ and $a_s = 0.02$ is used to

[9] The MMSE combiner is described in Sec. 4.9.1.

generate the repetition factors.[10]

The user activity coefficients are generated as $a_m \overset{\text{i.i.d.}}{\sim} \text{Ber}(p_a)$, where $p_a = 0.1$ is the per-user activity probability. The system load L is defined as the average number of active users per RB, $L = Mp_a/T$. The number of users contending for the T RBs is computed in each simulation based on the load L as $M = \lfloor LT/p_a \rceil$. The SNR for the mth user is calculated as $\text{SNR}_m = P\sigma_{\mathtt{h}}^2\beta_m/N_0$. The received SNR of a user at the edge of the cell at the BS is termed as the *cell edge SNR*. The power levels of all users is set to the same value, P, chosen such that the signal from a user at a distance $r_{\max}$ from the BS is received at the cell edge SNR. This ensures that all users' signals are received at an SNR greater than or equal to the cell edge SNR, in singleton RBs.[11] The power levels of users is set to $P = P^{\mathtt{p}} = 20$ dB [49] and N_0 is chosen such that the cell edge SNR is 10 dB, unless otherwise stated.[12]

4.6.1 Error Rates for UAD

In this subsection, the error rates for the recovery of user activity coefficients in IRSA is presented. The metrics used to characterize the UAD performance are false positive rate, $\text{FPR} \triangleq \frac{|\mathcal{F}|}{|\mathcal{F}|+|\mathcal{I}|}$, and false negative rate, $\text{FNR} \triangleq \frac{|\mathcal{M}|}{|\mathcal{M}|+|\mathcal{P}|}$. FPR is the fraction of inactive users declared to be active whereas FNR is the fraction of active users declared to be inactive.

Fig. 4.1 shows the receiver operating characteristic (ROC) plot, and compares the performance of the proposed algorithm with existing approaches such as the maximum likelihood (ML), non-negative least squares (NNLS), and MMV, proposed in [14]. Here, the threshold γ_{pr} is varied to generate the curves, and the FNR is plotted versus the FPR for $N = 4$ and $L = 3$, which corresponds to $M = 1500$ total users. The existing algorithms are applied to (4.4) to detect the ith user's activity $\hat{a}_i^t$ in the tth RB and the user is declared

[10]The soliton distribution achieves near optimal throughputs [8]. Here, we reuse the same distribution to generate d_m. For the optimal distributions, see Chapter 7.

[11]If the cell edge SNR is such that the cell edge user's packet is decodable, then all users' packets are decodable with high probability in singleton RBs.

[12]In cases where the cell edge SNR is varied, the noise variance N_0 is varied according to the required cell edge SNR.

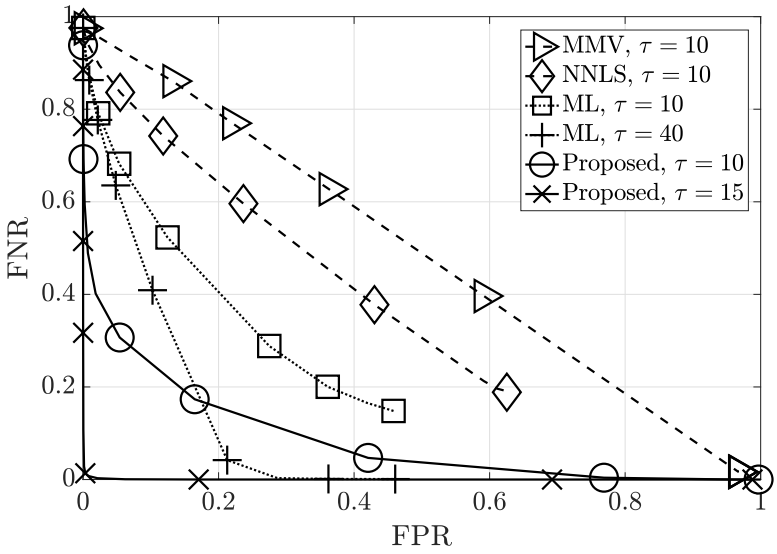


Figure 4.1: ROC of UAD: comparison with existing approaches.

active if it is detected to be active in at least t' RBs, i.e., $\hat{a}_i = \mathbb{1}\{\sum_{t=1}^{T} \hat{a}_i^t \geq t'\}$. We use $t' = 1$ since it yields the lowest FNR. Note that all of these algorithms estimate users' activities in each RB, whereas our algorithm combines the estimated hyperparameters in a principled manner as seen in (4.10), which is then used to infer the activities, and thus yields far fewer errors. The proposed algorithm outperforms all three approaches which have themselves shown an improvement over other compressed sensing based algorithms such as approximate message passing [14]. The ML approach with $\tau = 40$ intersects with the proposed algorithm with $\tau = 10$, and at the point of intersection, Algorithm 3 offers a 4-fold reduction in the pilot length compared to the ML approach while achieving the same UAD performance. Further, the proposed algorithm with $\tau = 15$ significantly outperforms all the other approaches, and achieves a near-ideal performance.

Next, in Fig. 4.2, we plot the error rates (i.e., the FNR and FPR) of Algorithm 3 versus the pilot length for varied L with $N = 16$. As the load is increased from $L = 1$ to $L = 2, 3$, the total number of users over the T RBs increases from $M = 500$ to $M = 1000, 1500$, and a longer pilot length is needed for accurate UAD. Thus, there is a significant improvement of the error rates with the pilot length τ. This is important, since short packets are used in mMTC, and using non-orthogonal pilots with as few as

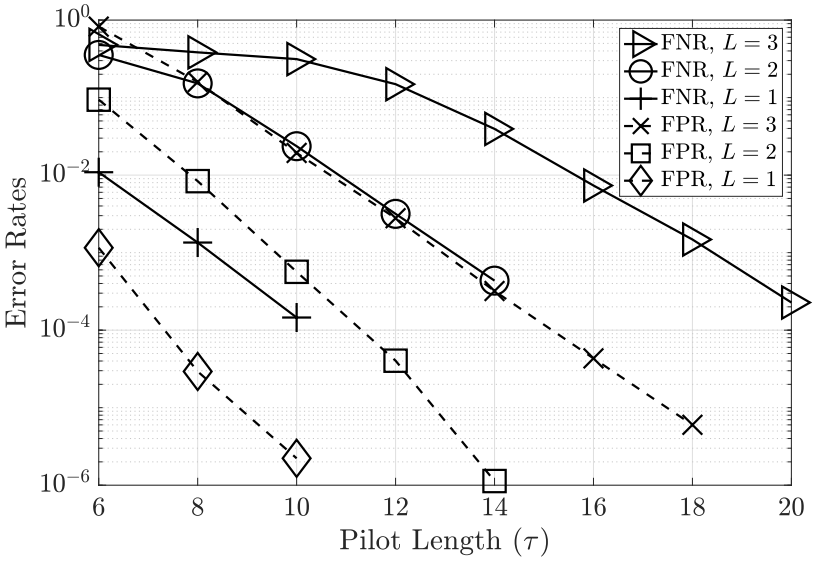


Figure 4.2: Impact of pilot length on error rates.

20 symbols yields very low error rates with as many as 1500 users. As noted earlier, with classical compressed sensing approaches for UAD, one would require $\Omega(346)$ pilot symbols for accurate UAD in the same settings.

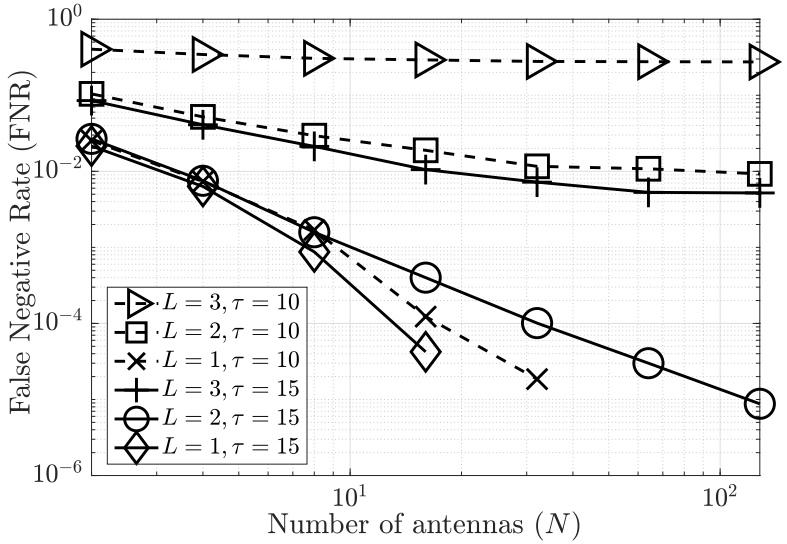


Figure 4.3: FNR for different pilot lengths and loads.

Next, we illustrate the variation of the FNR with the number of antennas for varied L and τ, in Fig. 4.3. The FNR is observed to increase with an increase in L. The FNR also

reduces with an increase in N or τ since the total number of measurements available in the received pilot signal increases, which improves the recovery of user activities in each RB. For $\tau = 10$, the FNR saturates with N for $L = 2/3$, whereas for $\tau = 15$, the FNR saturates at high L and reduces for low to medium L. This is because the performance of MSBL depends more critically on the number of rows τ in the received signal than the number of columns N [16]. Thus, at a given load, if τ is too low, the FNR improves only slowly with N, but if τ is large enough, the FNR improves dramatically with N. Hence, as the load increases, it is important to increase τ as well. In our approach, we solve a reduced problem in each RB as seen in (4.4), after accounting for the APM. Due to this, in the tth RB, $\tau = \Omega(M_t p_a \log \frac{M_t}{M_t p_a}) = \Omega(-M_t p_a \log p_a)$ would achieve a vanishing error rate. This guarantee is applicable when $\tau > M_t p_a$, which is the average number of non-zero entries to be recovered in each column of $\mathbf{Z}_t$ in (4.4). For $k_s = 27$, the average repetition factor is $\bar{d} = 4$, and thus on an average, $M_t = \frac{L\bar{d}}{p_a} = 120, 80, 40$ for $L = 3, 2, 1$, and of the order $\tau = 28, 19, 10$ pilot symbols are required, respectively, to achieve a low error rate for Algorithm 3 as N gets large.

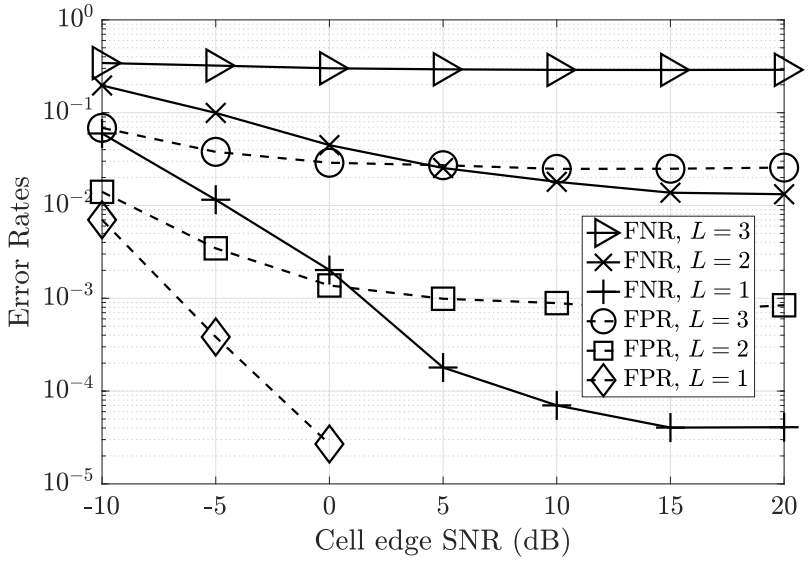


Figure 4.4: Effect of cell edge SNR on error rates.

In Fig. 4.4, the error rates are plotted against the cell edge SNR for varied L and $\tau = 10$. For low L, the error rates first linearly reduce with SNR and then saturate at

high SNR. The FPR for $L = 1$ requires longer simulations to capture the point where it saturates with SNR. For high L, both the error rates saturate very quickly with the SNR. As the load L is decreased, the error rates reduce since there are fewer users to be detected. As seen earlier, for a fixed L, increasing the pilot length can decrease the rates and the error rates reduce at the point of saturation. In the noise limited regime, i.e., SNR < -5 dB, the error rates are high since the Bayesian estimation process performs poorly at such low SNRs.

4.6.2 Normalized Mean Squared Error

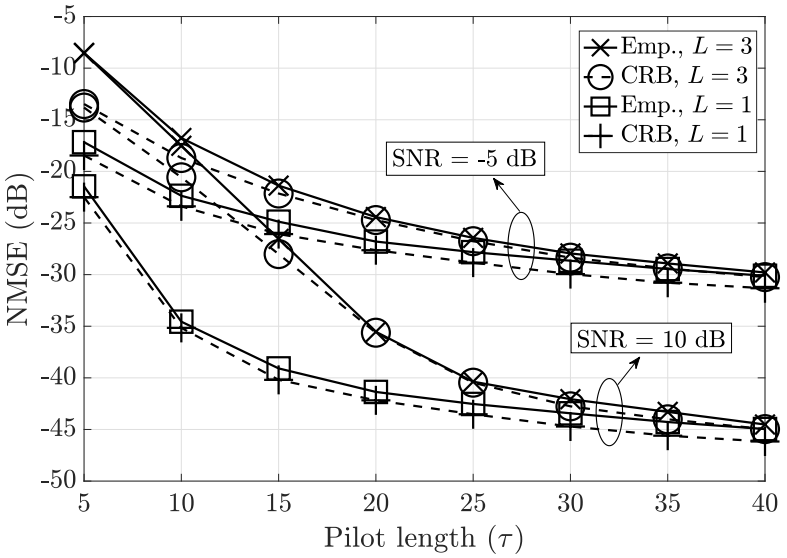


Figure 4.5: Impact of pilot length on NMSE.

Fig. 4.5 shows the impact of the pilot length τ on the normalized mean squared error (NMSE) of the channels estimated using Algorithm 3 (the curves labeled "Emp"). The NMSE is calculated as $\mathbb{E}[\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2]/\mathbb{E}[\|\mathbf{X}\|_F^2]$, where $\mathbf{X}$ is the channel matrix from (4.13) and $\hat{\mathbf{X}}$ is the corresponding matrix of channel estimates obtained from the UAD algorithm. It is observed that the NMSE converges to the same value at all L as τ is increased to 40, and the value the NMSE converges to decreases with SNR. As τ increases, UAD is perfect and the effect of pilot contamination is reduced, resulting in nearly the same NMSE at all loads. Also, at low τ, the NMSE is higher for $L = 3$ compared to

$L = 1$, since we have to estimate channels for a larger number of users – both UAD errors and pilot contamination contribute to a worsening of performance. The normalized CRB from (4.31) is also plotted for the system under all the configurations. It is seen that the gap between the true NMSE and the normalized CRB reduces as τ increases. The NMSE is insensitive to the value of N, as both the numerator and the denominator of the NMSE scale equally with N. Hence, we do not study the impact of N on the NMSE.

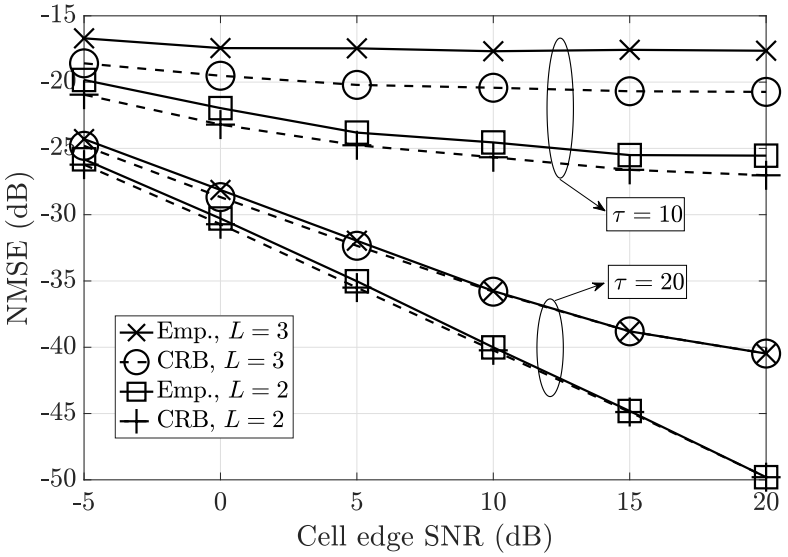


Figure 4.6: Effect of cell edge SNR on NMSE.

Fig. 4.6 shows the impact of SNR on NMSE. The NMSE saturates with an increase in SNR for both L at $\tau = 10$, since the UAD performance saturates and any increase in SNR does not improve the quality of the channel estimates. For $\tau = 20$, the NMSE linearly reduces with SNR up to 20 dB. At higher τ, the NMSE is lower since there are more measurements available in the received pilot signal to obtain both better UAD performance and high quality channel estimates. Further, the gap between the true NMSE and the normalized CRB reduces with an increase in SNR for $\tau = 20$. Thus, the CRB, which is achieved by the genie-aided estimator in (4.35), is also achieved by the estimates in Algorithm 3 as τ and SNR are increased.

4.6.3 Throughput Accounting for UAD and Channel Estimation

The performance of IRSA can be characterized by its throughput, which is defined as the number of packets that were successfully decoded at the BS as a fraction of the total number of RBs.[13] Note that, at a system load of L, the average throughput of the system is upper bounded by L packets per RB, since there are, on average, LT unique packets transmitted per frame of duration T RBs. In this subsection, the SINR analysis presented in Sec. 4.5 is used to evaluate the throughput of IRSA with UAD and estimated channels. The number of successfully decoded packets per RB for each simulation is calculated as described in Sec. 4.2.1, and the throughput of the system is found by averaging over the simulations.

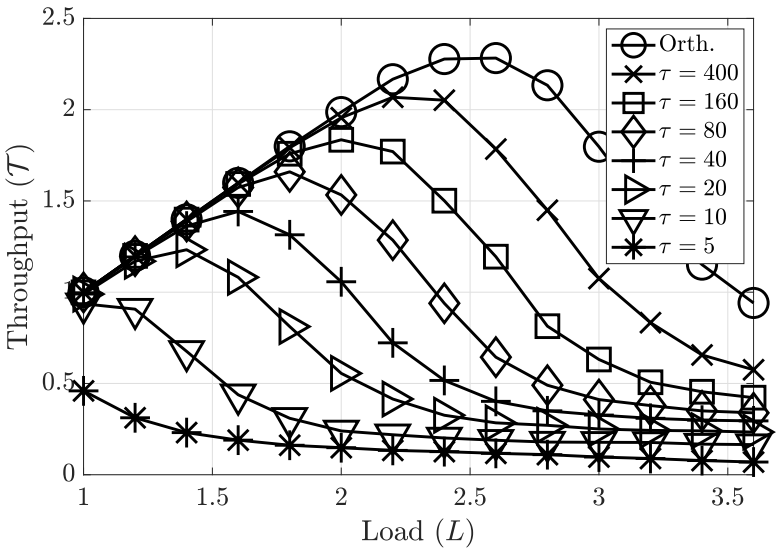


Figure 4.7: Effect of system load L.

Fig. 4.7 shows the system throughput, $\mathcal{T}$ (successfully decoded packets per RB), evaluated for different pilot lengths under UAD and estimated CSI, with threshold $\gamma_{\text{th}} = 16$ and regularization parameter $\lambda = 1$, as a function of the load L. For $\tau \geq 20$, the throughputs exceed unity, which is the throughput of perfectly coordinated orthogonal access. In the moderate load regime ($L < 2$), the system can serve more users, and thus the

[13]We note that the throughput $\mathcal{T}$ of IRSA is directly related to the packet loss rate PLR and the spectral efficiency $\mathcal{R}$ as $\mathcal{T} = L(1 - \mathsf{PLR})$ and $\mathcal{R} = \mathcal{T} \times \log_2(1 + \gamma_{\text{th}})$, respectively [7].

throughput increases linearly with load. As the load is increased further, the system becomes interference limited as there are too many users sharing the same resources, thereby decreasing the SINR and the throughput. Also, as the pilot length τ increases, UAD performance improves, better quality channel estimates are obtained, and the corresponding SINR increases. The orthogonal pilots curve is obtained by allocating $\tau = M = \lfloor LT/p_a \rceil$ for each L, and this achieves nearly the same performance as the case where perfect CSI is available at the BS. At $L = 2$, there are $M = 1000$ users that need to be served. For $\tau = 80$ and 400, the achievable throughputs are $\mathcal{T} = 1.5$ and 2, respectively. At a load of $L = 1.5$, the throughput obtained with $\tau = 80$ is identical to the one offered by the orthogonal pilots, which would need a pilot length of $\tau = M = 750$. This shows one can use significantly fewer number of pilot symbols and still achieve the same throughput as fully orthogonal pilots, at low to medium loads.

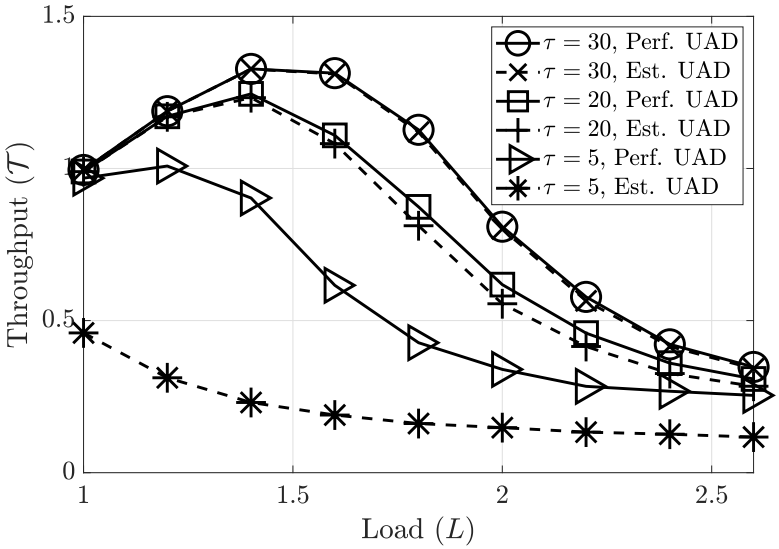


Figure 4.8: Impact of estimated UAD.

Fig. 4.8 quantifies the effect of UAD on the performance of IRSA, by plotting the throughput against the system load under *perfect* and *estimated* user activities. Here, $\gamma_{\text{th}} = 16$ and $\lambda = 1$ as in the previous figure. In both cases, the throughput increases linearly with L till it hits a maximum and then reduces. With a pilot length $\tau = 5$, the gap between estimated and perfect UAD is at its maximum of 0.7 (packets/RB) at $L = 1.2$.

As the pilot length is increased, the gap reduces to a maximum of 0.1 (packets/RB) at $L = 2$ for $\tau = 20$ and a negligibly small difference for $\tau = 30$. This shows that for lower pilot lengths, UAD performance has a significant effect on the throughput. For higher pilot lengths, the UAD is nearly perfect, and, in this regime, channel estimation and data decoding limits the performance. Thus, UAD is the easier problem in practical regimes of interest.

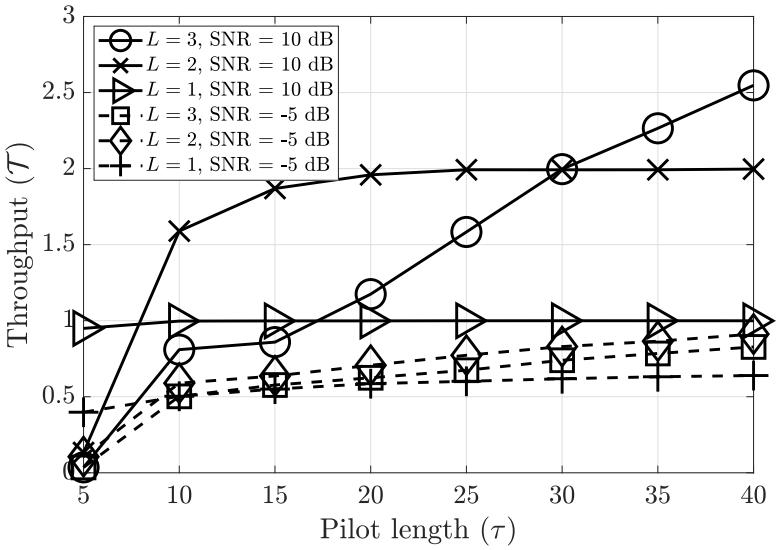


Figure 4.9: Impact of pilot length τ on throughput.

In Fig. 4.9, we investigate the effect of pilot length on the system throughput at different L and SNRs. The threshold is set to $\gamma_{\text{th}} = 10$ and regularization parameter to $\lambda = 10^{-2}$ for the rest of the results. At a cell edge SNR of -5 dB, the system throughput is very low due to poor UAD as well as poor quality channel estimates. The throughput saturates with an increase in τ for all loads. Even though more measurements are available at high τ, even if the UAD process is successful and the channel estimates are accurate, the low SNR results in data decoding failures, which limits the throughput. As the cell edge SNR is increased to 10 dB, the system performance dramatically improves. At this SNR, optimal throughputs of $\mathcal{T} = L$ is achieved with $\tau = 10/25$ for $L = 1/2$, respectively, which correspond to $M = 500/1000$ total users and on an average $Mp_a = 50/100$ active users, respectively. For $L = 3$, the optimal throughput is obtained at $\tau = 70$, which is not

depicted here. As seen previously, the UAD problem is dominant for very low τ for these loads, and for higher τ, channel estimation dominates the performance. To summarize, the pilot length has a significant impact on the performance of IRSA and is instrumental in yielding near-optimal throughputs.

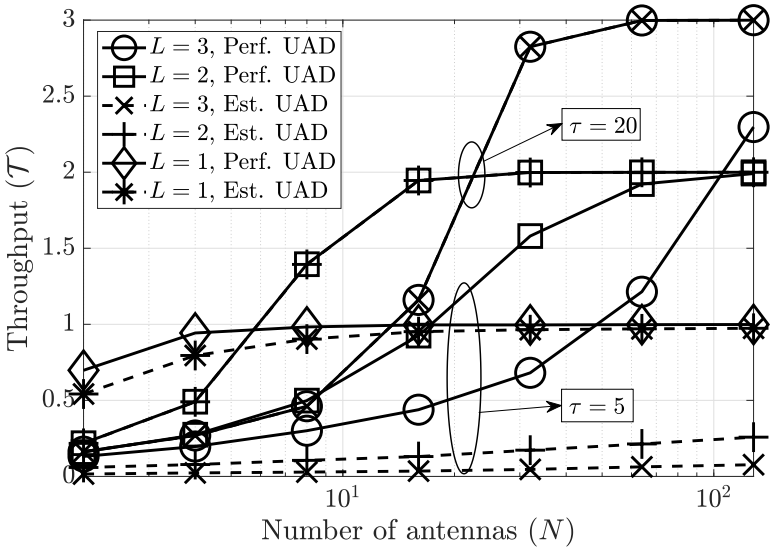


Figure 4.10: Effect of number of antennas N.

In Fig. 4.10, the system throughput is plotted against the number of antennas at the BS for different L and τ, under both perfect and estimated UAD. The gap between the perfect and estimated UAD throughputs for $L = 2, 3$ and $\tau = 5$ increases with N, and the gap is the highest at $N = 128$. This is because the UAD performance saturates with N for high L at low τ. Due to the combined effect of UAD errors, pilot contamination, and interference, low pilot lengths adversely impact both the UAD performance and system throughput. For $\tau = 20$, increasing N has a dramatic impact at high L, and the curves with perfect and estimated UAD overlap completely. Nearly optimal throughputs of $\mathcal{T} = L$ can be achieved with $N = 16, 32$ antennas for $L = 2, 3$. At $\tau \geq 20$, increasing the number of antennas improves UAD, and increases both the array gain and the decoding capability of the BS, leading to more users getting decoded with RZF. In particular, at $L = 3$, the rise in throughput as N is increased from 8 to 32 shows the impact of the number of antennas in improving the throughput.

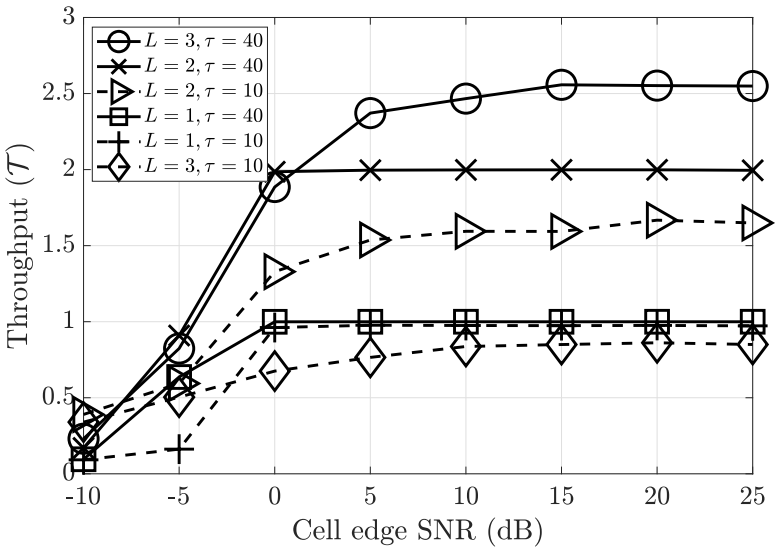


Figure 4.11: Impact of cell edge SNR.

In Fig. 4.11, we illustrate the impact of cell edge SNR on the throughput. In the noise-limited regime (SNR < 0 dB), an increase in SNR increases the throughput. Beyond an SNR of 0 dB, increasing SNR only marginally increases the throughput for all L and τ and the system becomes interference-limited for $\tau = 10$. This is because both signal and interference powers get scaled equally, and the SINR remains the same. At $\tau = 40$, for $L = 1$ and 2, optimal throughputs can be obtained at a cell edge SNR $= 0$ dB. However, the throughput for $L = 3$ saturates beyond 10 to 15 dB SNR and does not yield the optimal throughput of $\mathcal{T} = 3$ due to high L and low τ. In summary, the throughput can be improved by increasing the pilot length, number of antennas, and SNR judiciously: unilaterally increasing one of the three can lead to the throughput saturating at a value lower than $\mathcal{T} = L$.

4.6.4 Choice of Pilot Sequences for UAD

For a study on the uniqueness of pilot sequences in IRSA, see Sec. 5.9.9.

In Fig. 4.12, we plot the ROC curves for UAD in IRSA for different pilot sequences, with $N = 4$, $L = 3$, and $T = 50$. The non-orthogonal pilots, labeled as BPSK and QPSK,

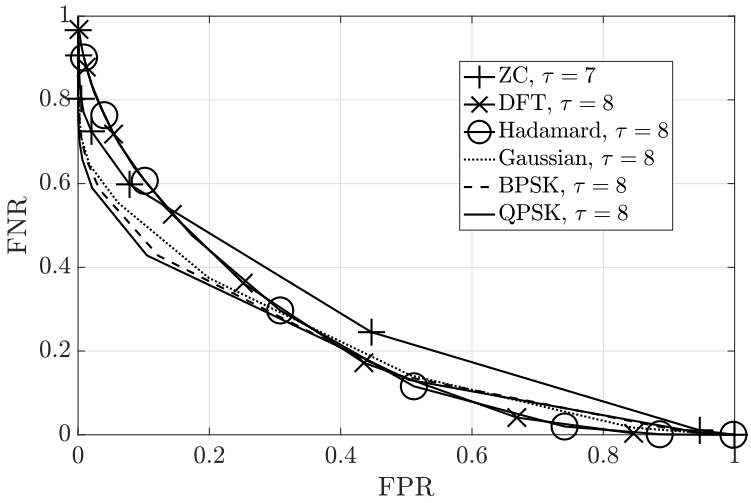


Figure 4.12: ROC comparison with different pilot sequences.

contain random pilot symbols belonging to the respective PSK constellations, and Zadoff-Chu (ZC) sequences are generated according to [115]. ZC sequences require prime τ; we use $\tau = 7$. With mutually orthogonal pilot sequences, such as Hadamard and discrete Fourier transform (DFT), τ sequences of length τ can be generated. Thus, we perform orthogonal pilot reuse (OPR), where each user randomly selects a pilot sequence from the available set of τ pilot sequences, similar to [63]. We see that all the pilot sequences have similar UAD performance. In particular, QPSK, BPSK, and Gaussian pilot sequences have nearly identical performance; DFT and Hadamard sequences have identical UAD performance. Thus, the choice of pilot sequences do not significantly affect the UAD performance. However, we observe that choosing non-orthogonal pilot sequences (with low correlation among distinct pilot sequences) results in marginally better throughput and nearly identical UAD performance compared to OPR.

Using OPR leads to identical pilots being chosen by many more users compared to non-orthogonal pilots (since the pool of pilot sequences with a given τ is much smaller with OPR). In IRSA, the collision probability, i.e., the probability that two colliding users have identical pilot sequences, is lower than in conventional grant-free random access since only a small subset of users transmit in any given RB. Similar to the calculation of the collision probability in grant-free random access [53], as an example, with $L = 5$, $\bar{d} = 4$,

$\tau = 20$, and $p_a = 0.1$, we have around $M_a = 200$ average number of active users in any RB, and the collision probability with random QPSK pilots is of the order of 10^{-8}. Thus, using non-orthogonal pilots has the advantage of lower collision probability and also better throughput.

4.6.5 Effect of Imperfect SIC

Performing SIC using the estimated channels and decoded data results in a residual signal depends on the channel estimation errors of users that have already been decoded. We elaborate on the analysis of pilot-aided and data-aided channel estimation in Sec. 4.9.2.

Under imperfect SIC, the post-combined data signal is given by

$$\tilde{y}_{tm}^{k} = \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}a_m g_{tm}x_m - \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^{k}a_m g_{tm}x_m + \sum\nolimits_{i\in\mathcal{S}_k^m\cap\mathcal{P}} \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}a_i g_{ti}x_i + \sum\nolimits_{i\in\mathcal{S}_1\setminus\mathcal{S}_k} \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{ti}^{k_i}a_i g_{ti}x_i + \sum\nolimits_{i\in\mathcal{S}_k^m\cap\mathcal{M}} \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}a_i g_{ti}x_i + \mathbf{a}_{tm}^{kH}\mathbf{n}_t. \quad (4.43)$$

Here, $\sum_{i\in\mathcal{S}_1\setminus\mathcal{S}_k} \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{ti}^{k_i}a_i g_{ti}x_i$ represents the residual interference due to channel estimation errors, and k_i denotes the iteration in which the ith user was decoded. The above equation is applicable under both LMMSE data aided channel estimation and MMSE pilot aided channel estimation. Thus, the SINR can be expressed as

$$\mathrm{SINR}_{tm}^{k} = \frac{\mathtt{Gain}_{tm}^{k}}{N_0 + \mathtt{Est}_{tm}^{k} + \mathtt{ImpSIC}_{tm}^{k} + \mathtt{MUI}_{tm}^{k} + \mathtt{FNU}_{tm}^{k}},\ \forall m \in \mathcal{S}_k. \quad (4.44)$$

Here, the term $\mathtt{ImpSIC}_{tm}^{k} = P\sum_{i\in\mathcal{S}_1\setminus\mathcal{S}_k}\hat{a}_i a_i g_{ti}\delta_{ti}^{k_i}$ is the extra term that arises due to imperfect SIC, where $\delta_{ti}^{k_i}$ is the power of the MMSE estimate error of the ith user in the tth RB who has been decoded in the k_ith decoding iteration.

Fig. 4.13 studies the effect of imperfect SIC on the performance of IRSA, with random BPSK pilots. For this, we use the SINR in (4.44), with MMSE channel estimation. We also assume perfect UAD here, since we wish to address the effect of imperfect SIC. The gap between the perfect SIC and imperfect SIC curves reduce as the pilot length is increased. The gap is negligible at $\tau = 20$, and is very high at $\tau = 5$. Thus, at higher

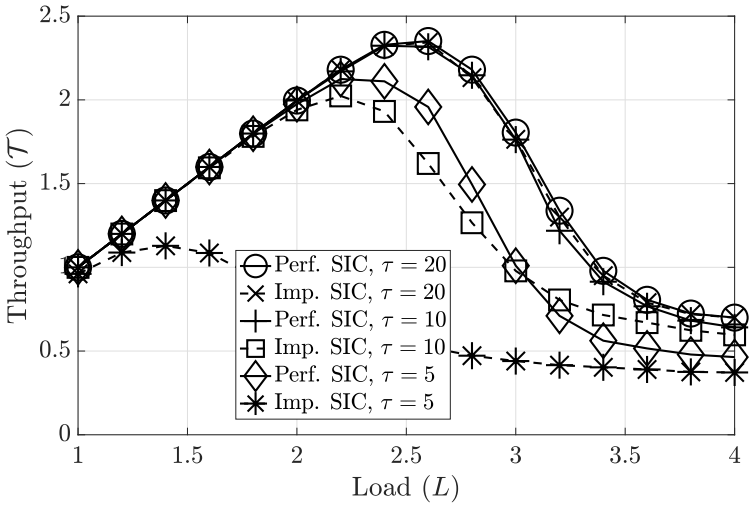


Figure 4.13: Effect of imperfect SIC.

pilot lengths, the effect of imperfect SIC due to channel estimation errors can be ignored.

4.6.6 Which UAD error is more harmful?

False positives and false negatives do not equally affect the performance of IRSA. We now analyze the impact of these UAD errors in each stage of the UAD algorithm.

1. The post-combined signals are

$$\mathbf{y}_{tm}^{\mathrm{p}k} = \mathbf{Y}_t^{\mathrm{p}k}\mathbf{p}_m = \textstyle\sum_{i\in\mathcal{S}_k} a_i g_{ti}\mathbf{h}_{ti}(\mathbf{p}_i^H\mathbf{p}_m) + \mathbf{N}_t^{\mathrm{p}}\mathbf{p}_m, \tag{4.45}$$

$$\mathbf{y}_t^k = \textstyle\sum_{i\in\mathcal{S}_k} a_i g_{ti}\mathbf{h}_{ti}x_i + \mathbf{n}_t. \tag{4.46}$$

   This contains terms from both true positive and false negative users, and thus, FNR affects the received pilot and data signals.

2. The LCMMSE channel estimate is

$$\hat{\mathbf{h}}_{tm}^k = \frac{\hat{a}_m g_{tm}\beta_m\sigma_{\mathrm{h}}^2\|\mathbf{p}_m\|^2}{N_0\|\mathbf{p}_m\|^2 + \sum_{i\in\mathcal{S}_k}\hat{a}_i g_{ti}\beta_i\sigma_{\mathrm{h}}^2|\mathbf{p}_i^H\mathbf{p}_m|^2}\mathbf{y}_{tm}^{\mathrm{p}k}, \quad \forall m \in \mathcal{S}_k. \tag{4.47}$$

   We can see that the estimated activity coefficients are present in both the numerator and the denominator of the scaling coefficient in the LCMMSE channel estimate.

This coefficient is contributed by both true positive and false positive users, and thus, FPR affects the channel estimates. For the MMSE estimate, a huge FPR can highly affect the covariance of the received signal and its inverse. Consequently, the channel estimates can be very poor and the performance can be very poor.

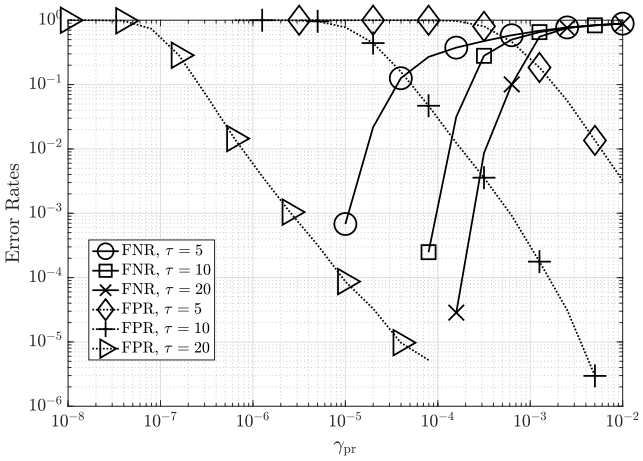
3. The SINR has an extra component in the denominator, $\texttt{FNU}_{tm}^{k}$, that models the contribution of false negative users: $\texttt{FNU}_{tm}^{k} = P\sum_{i\in\mathcal{S}_k^m}(1-\hat{a}_i)a_i g_{ti}\beta_i\sigma_{\texttt{h}}^2$. A high FNR can thus reduce the SINR, and consequently the throughput of IRSA. Poor channel estimates due to high FPR can also reduce the SINR.

It is not immediately clear which error affects the performance the most. Both the channel estimation and the data decoding processes are affected by both false positives and false negatives. We thus plot the errors as a function of γ_{pr} in Fig. 4.14(a), and the ensuing throughput in Fig. 4.14(b). The simulation settings for this are $T = 50$, $\bar{d} = 4$, $p_a = 0.1$, $L = 3$, and the other settings are chosen similar to the manuscript. As the pilot length τ increases, the error rates fall, the channel estimation quality and the SINR improves, and thus, the throughput improves to the optimal value. At varied γ_{pr}, different error rates have different effects on the throughput. For all the cases, we observe a huge drop in performance when the FPR>0.1, especially at very low γ_{pr}. This drop is not as pronounced when FNR>0.1 at very high γ_{pr}, and the FPR has a drastic effect on the throughput.

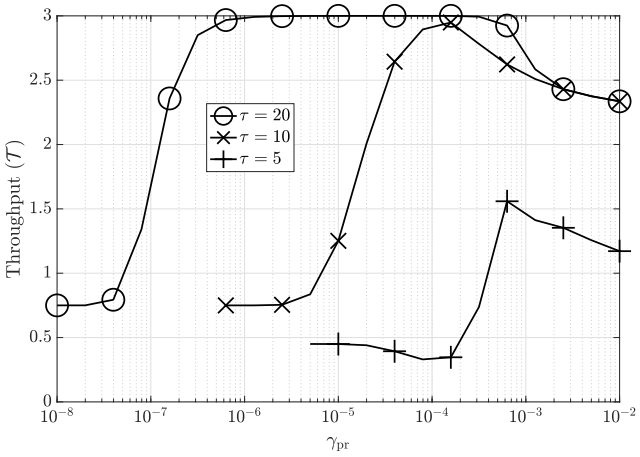
In summary, both the UAD errors have a huge effect on the performance of IRSA. In particular, FPR impacts channel quality and has a huge effect on the performance at very low γ_{pr}.

4.7 Summary

This chapter studied the impact of UAD on the throughput of IRSA, which is a GFRA protocol that involves repetition of packets across different RBs. A novel Bayesian algorithm was proposed to detect the set of active users in IRSA, which exploited the knowledge of the APM, and combined the hyperparameter updates across all RBs to



(a) UAD errors in IRSA



(b) Throughput of IRSA.

Figure 4.14: Impact of UAD errors on the performance of IRSA.

yield an improved UAD performance. Next, the channel estimates were derived accounting for UAD errors. A Cramér-Rao bound was then derived for the channels estimated under the hierarchical Bayesian model used to develop the proposed algorithm. Then, the SINR of all users was derived accounting for UAD, channel estimation errors, and pilot contamination. The effect of these errors on the throughput was studied via extensive simulations. Many new insights into the design of the IRSA protocol were discussed, namely, the complexity of UAD compared to channel estimation, and the improvement of both UAD and throughput with respect to τ, N, SNR, and L. The results underscored the importance of accounting for UAD errors and channel estimation, in studying the throughput offered by the IRSA protocol in mMTC. We assumed perfect RB- and frame-level synchronization across users and the BS; future work can consider relaxing this assumption. Exploiting the asynchronous nature of random access transmissions to detect active users and estimate their channels instead of orthogonal/non-orthogonal pilots is also an interesting direction for future work.

4.8 Proofs

4.8.1 Proof of Theorem 4.1: Channel Estimation

Since the channel coefficients are Gaussian distributed, the MMSE estimator[14] is $\hat{\mathbf{h}}_{tm}^{k} \triangleq \mathbb{E}_{\mathbf{z}}\left[\mathbf{h}_{tm}\right]$, where $\mathbf{z}$ contains the post-combining pilot signals for all users detected to be active. The channel estimation error $\tilde{\mathbf{h}}_{tm}^{k} \triangleq \hat{\mathbf{h}}_{tm}^{k} - \mathbf{h}_{tm}$ is uncorrelated with the received pilot signal and the estimate itself [49]. The conditional statistics of a Gaussian random vector $\mathbf{x}$ are

$$\mathbb{E}_{\mathbf{z}}\left[\mathbf{x}\right] = \mathbb{E}\left[\mathbf{x}\right] + \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}\left(\mathbf{z} - \mathbb{E}\left[\mathbf{z}\right]\right), \tag{4.48}$$

$$\mathbf{K}_{\mathbf{x}\mathbf{x}|\mathbf{z}} = \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}\mathbf{K}_{\mathbf{zx}}. \tag{4.49}$$

[14]Specifically, this is the LCMMSE estimator from the previous chapter. The MMSE estimator is described in Sec. 4.9.1.

Here, $\mathbf{K_{xx}}$, $\mathbf{K_{xx|z}}$, and $\mathbf{K_{xz}}$ are the unconditional covariance of $\mathbf{x}$, the conditional covariance of $\mathbf{x}$ conditioned on $\mathbf{z}$, and the cross-covariance of $\mathbf{x}$ and $\mathbf{z}$, respectively. From (4.48), the MMSE channel estimate $\hat{\mathbf{h}}_{tm}^k$ can be calculated as

$$\hat{\mathbf{h}}_{tm}^k = \mathbb{E}\left[\mathbf{h}_{tm}\mathbf{y}_{tm}^{\mathrm{p}kH}\right]\mathbb{E}[\mathbf{y}_{tm}^{\mathrm{p}k}\mathbf{y}_{tm}^{\mathrm{p}kH}]^{-1}\mathbf{y}_{tm}^{\mathrm{p}k} \tag{4.50}$$

$$= \frac{\hat{a}_m g_{tm}\beta_m\sigma_{\mathrm{h}}^2\|\mathbf{p}_m\|^2}{(N_0\|\mathbf{p}_m\|^2 + \sum_{i\in\mathcal{S}_k}\hat{a}_i g_{ti}\beta_i\sigma_{\mathrm{h}}^2|\mathbf{p}_i^H\mathbf{p}_m|^2)}\mathbf{y}_{tm}^{\mathrm{p}k} \triangleq \eta_{tm}^k\mathbf{y}_{tm}^{\mathrm{p}k}. \tag{4.51}$$

The above is computed based on the users detected to be active and is thus a function of estimated activity coefficients $\hat{a}_i$. From (4.49), the conditional covariance of the channel $\mathbf{h}_{tm}$ is calculated conditioned on $\mathbf{z}$, which contains the post-combining pilot signals for users detected to be active. Also, $\mathbf{K}_{\mathbf{h}_{tm}\mathbf{h}_{tm}} = \beta_m\sigma_{\mathrm{h}}^2\mathbf{I}_N$, $\mathbf{K}_{\mathbf{h}_{tm}\mathbf{z}} = \mathbb{E}[\mathbf{h}_{tm}\mathbf{y}_{tm}^{\mathrm{p}kH}]$ $= \|\mathbf{p}_m\|^2 a_m g_{tm}\beta_m\sigma_{\mathrm{h}}^2\mathbf{I}_N$. Thus, the conditional covariance is

$$\mathbf{K}_{\mathbf{h}_{tm}\mathbf{h}_{tm}|\mathbf{z}} = \mathbf{K}_{\mathbf{h}_{tm}\mathbf{h}_{tm}} - \mathbf{K}_{\mathbf{h}_{tm}\mathbf{z}}\mathbf{K}_{\mathbf{zz}}^{-1}\mathbf{K}_{\mathbf{z}\mathbf{h}_{tm}} \tag{4.52}$$

$$= (\beta_m\sigma_{\mathrm{h}}^2 - \eta_{tm}^k\|\mathbf{p}_m\|^2 a_m g_{tm}\beta_m\sigma_{\mathrm{h}}^2)\mathbf{I}_N \triangleq \delta_{tm}^k\mathbf{I}_N. \tag{4.53}$$

Here, $\delta_{tm}^k = \beta_m\sigma_{\mathrm{h}}^2\left(\frac{\sum_{i\in\mathcal{S}_k^m}|\mathbf{p}_i^H\mathbf{p}_m|^2\hat{a}_i a_i g_{ti}\beta_i\sigma_{\mathrm{h}}^2 + N_0\|\mathbf{p}_m\|^2}{\sum_{i\in\mathcal{S}_k}|\mathbf{p}_i^H\mathbf{p}_m|^2\hat{a}_i a_i g_{ti}\beta_i\sigma_{\mathrm{h}}^2 + N_0\|\mathbf{p}_m\|^2}\right)$ represents the interference caused due to estimation errors of other true positive users. It is a function of the pilots of the other true positive users only and not the pilots of false positive users. False positive users are omitted from the above because such users do not contaminate the pilots of other users. The conditional correlation follows from its definition as

$$\mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tm}\mathbf{h}_{tm}^H] = \mathbf{K}_{\mathbf{h}_{tm}\mathbf{h}_{tm}|\mathbf{z}} + \mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tm}]\mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tm}]^H = \delta_{tm}^k\mathbf{I}_N + \hat{\mathbf{h}}_{tm}^k\hat{\mathbf{h}}_{tm}^{kH}. \tag{4.54}$$

The unconditional and conditional means of the error are $\mathbb{E}[\tilde{\mathbf{h}}_{tm}^k] = \mathbb{E}[\hat{\mathbf{h}}_{tm}^k - \mathbf{h}_{tm}] = 0$ and $\mathbb{E}_{\mathbf{z}}[\tilde{\mathbf{h}}_{tm}^k] = \mathbb{E}_{\mathbf{z}}[\hat{\mathbf{h}}_{tm}^k - \mathbf{h}_{tm}] = \hat{\mathbf{h}}_{tm}^k - \hat{\mathbf{h}}_{tm}^k = 0$. The conditional covariance of the error is

$$\mathbf{K}_{\tilde{\mathbf{h}}_{tm}^k\tilde{\mathbf{h}}_{tm}^k|\mathbf{z}} = \mathbb{E}_{\mathbf{z}}[\tilde{\mathbf{h}}_{tm}^k\tilde{\mathbf{h}}_{tm}^{kH}] = \mathbb{E}_{\mathbf{z}}[(\hat{\mathbf{h}}_{tm}^k - \mathbf{h}_{tm})(\hat{\mathbf{h}}_{tm}^k - \mathbf{h}_{tm})^H] \tag{4.55}$$

$$= \mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tm}\mathbf{h}_{tm}^H] - \hat{\mathbf{h}}_{tm}^k\hat{\mathbf{h}}_{tm}^{kH} = \delta_{tm}^k\mathbf{I}_N. \tag{4.56}$$

Since $\mathbf{h}_{tm}^k \sim \mathcal{CN}(\mathbf{0}_N, \beta_m \sigma_{\mathtt{h}}^2 \mathbf{I}_N)$, the estimate $\hat{\mathbf{h}}_{tm}^k$ and the error $\tilde{\mathbf{h}}_{tm}^k$ are distributed as $\mathcal{CN}$ $(\mathbf{0}_N,\ \eta_{tm}^k \|\mathbf{p}_m\|^2 a_m g_{tm} \beta_m \sigma_{\mathtt{h}}^2 \mathbf{I}_N)$ and $\mathcal{CN}(\mathbf{0}_N, \delta_{tm}^k \mathbf{I}_N)$ respectively. Also, MMSE estimates are uncorrelated with their errors [49].

4.8.2 Proof of Theorem 4.2: Cramér-Rao Bound

The FIM sub-block associated with $\overline{\mathbf{z}}_t$ in the tth RB is defined as $\mathbf{J}_t = \mathbf{J}_{t1} + \mathbf{J}_{t2}$ [123], with

$$\mathbf{J}_{t1} = \mathbb{E}\left[\left(\frac{\partial \log p(\overline{\mathbf{z}}_t)}{\partial \overline{\mathbf{z}}_t^*}\right)\left(\frac{\partial \log p(\overline{\mathbf{z}}_t)}{\partial \overline{\mathbf{z}}_t^*}\right)^H\right], \tag{4.57}$$

$$\mathbf{J}_{t2} = \mathbb{E}\left[\mathbb{E}\left[\left(\frac{\partial \log p(\overline{\mathbf{y}}_t|\overline{\mathbf{z}}_t)}{\partial \overline{\mathbf{z}}_t^*}\right)\left(\frac{\partial \log p(\overline{\mathbf{y}}_t|\overline{\mathbf{z}}_t)}{\partial \overline{\mathbf{z}}_t^*}\right)^H \middle| \overline{\mathbf{z}}_t\right]\right]. \tag{4.58}$$

The conditional probability of $\overline{\mathbf{y}}_t$ given $\overline{\mathbf{z}}_t$ is $\mathcal{CN}(\mathbf{\Phi}_t \overline{\mathbf{z}}_t, N_0 \mathbf{I}_{\tau N})$, whereas the channel vector $\overline{\mathbf{z}}_t$ is distributed as $\mathcal{CN}(\mathbf{0}_{NM_t}, \mathbf{I}_N \otimes \mathbf{\Gamma}_t)$. Hence, the log of the conditional probabilities behave as

$$\log p(\overline{\mathbf{z}}_t) \propto -\overline{\mathbf{z}}_t^H (\mathbf{I}_N \otimes \mathbf{\Gamma}_t)^{-1} \overline{\mathbf{z}}_t, \tag{4.59}$$

$$\log p(\overline{\mathbf{y}}_t|\overline{\mathbf{z}}_t) \propto -\frac{\|\overline{\mathbf{y}}_t - \mathbf{\Phi}_t \overline{\mathbf{z}}_t\|_2^2}{N_0}. \tag{4.60}$$

Upon taking the derivative and then calculating the required expectations, it is straightforward to show that $\mathbf{J}_{t1} = \mathbf{I}_N \otimes \mathbf{\Gamma}_t^{-1}$ and $\mathbf{J}_{t2} = \mathbf{I}_N \otimes (\mathbf{P}_t^H \mathbf{P}_t / N_0)$. Further, the sub-blocks of $\mathbf{J}_t$ corresponding to different antennas are identical and equal to $\mathbf{P}_t^H \mathbf{P}_t / N_0 + \mathbf{\Gamma}_t^{-1}$. The result follows.

4.8.3 Proof of Theorem 4.3: SINR Computation

In order to compute the SINR, we first compute the power of the individual components. The desired signal power is

$$\mathbb{E}_{\mathbf{z}}[|T_1|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k a_m g_{tm} x_m|^2] = P a_m^2 g_{tm}^2 |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k|^2. \tag{4.61}$$

The powers of a_i and g_{ti} are dropped, since they are binary-valued. In order to account for zero data rates for false positive users, the desired signal power is non-zero only for true positive users and the desired gain is written as

$$\texttt{Gain}_{tm}^{k} \triangleq \frac{\mathbb{E}_{\mathbf{z}}[|T_1|^2]}{\|\mathbf{a}_{tm}^{k}\|^2} = P\hat{a}_m a_m g_{tm} \frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}|^2}{\|\mathbf{a}_{tm}^{k}\|^2}. \tag{4.62}$$

The power of the estimation error term is calculated as

$$\mathbb{E}_{\mathbf{z}}[|T_2|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^{k} a_m g_{tm} x_m|^2] = P a_m^2 g_{tm}^2 \delta_{tm}^{k} \|\mathbf{a}_{tm}^{k}\|^2. \tag{4.63}$$

Next, the power of the first inter-user interference term is

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}}[|T_3|^2] &= \mathbb{E}_{\mathbf{z}}\left[\left|\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{P}} \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti} a_i g_{ti} x_i\right|^2\right] \\
&= P\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{P}} a_i^2 g_{ti}^2 \mathbf{a}_{tm}^{kH}\mathbb{E}_{\mathbf{z}}[\mathbf{h}_{ti}\mathbf{h}_{ti}^{H}]\mathbf{a}_{tm}^{k} \\
&\overset{(a)}{=} P\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{P}} a_i^2 g_{ti}^2 \mathbf{a}_{tm}^{kH}(\delta_{ti}^{k}\mathbf{I}_N + \hat{\mathbf{h}}_{ti}^{k}\hat{\mathbf{h}}_{ti}^{kH})\mathbf{a}_{tm}^{k} \\
&= P\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{P}} a_i^2 g_{ti}^2 (\|\mathbf{a}_{tm}^{k}\|^2\delta_{ti}^{k} + |\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^{k}|^2),
\end{aligned} \tag{4.64}$$

where (a) follows from Theorem 4.1. Here, $\mathbb{E}_{\mathbf{z}}[|T_2|^2]+\mathbb{E}_{\mathbf{z}}[|T_3|^2]$ represents the contribution of estimation error components of all true positive users and multi-user interference components of other true positive users. We now split the normalized version of the above into the sum of the error component $\texttt{Est}_{tm}^{k}$ and the multi-user interference $\texttt{MUI}_{tm}^{k}$ as follows

$$\texttt{Est}_{tm}^{k} \triangleq P\textstyle\sum_{i\in\mathcal{S}_k} \hat{a}_i a_i g_{ti}\delta_{ti}^{k}, \tag{4.65}$$

$$\texttt{MUI}_{tm}^{k} \triangleq P\textstyle\sum_{i\in\mathcal{S}_k^m} \hat{a}_i a_i g_{ti} \frac{|\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^{k}|^2}{\|\mathbf{a}_{tm}^{k}\|^2}. \tag{4.66}$$

The power of the second inter-user interference term is

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}}[|T_4|^2] &= \mathbb{E}_{\mathbf{z}}\left[\left|\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}} \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti} a_i g_{ti} x_i\right|^2\right] \\
&\overset{(b)}{=} P\textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}} a_i^2 g_{ti}^2 \mathbf{a}_{tm}^{kH}\mathbb{E}[\mathbf{h}_{ti}\mathbf{h}_{ti}^{H}]\mathbf{a}_{tm}^{k}
\end{aligned}$$

$$= P\sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}} a_i^2 g_{ti}^2 \mathbf{a}_{tm}^{kH}(\beta_i\sigma_{\mathtt{h}}^2\mathbf{I}_N)\mathbf{a}_{tm}^k$$
$$= P\sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}} a_i^2 g_{ti}^2 \beta_i\sigma_{\mathtt{h}}^2\|\mathbf{a}_{tm}^k\|^2, \tag{4.67}$$

where the conditional expectation is dropped in (b) since the BS does not have the knowledge of the channel estimates of false negative users. The normalised power of the false negative users is calculated as

$$\mathtt{FNU}_{tm}^k \triangleq P\sum_{i\in\mathcal{S}_k^m}(1-\hat{a}_i)a_i g_{ti}\beta_i\sigma_{\mathtt{h}}^2. \tag{4.68}$$

Finally, the noise power is calculated as

$$\mathbb{E}_{\mathbf{z}}[|T_5|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}_{tm}^{kH}\mathbf{n}_t|^2] = N_0\|\mathbf{a}_{tm}^k\|^2. \tag{4.69}$$

Since the five terms in the received signal in (4.38) are mutually uncorrelated, a meaningful expression for the SINR can be obtained by dividing the useful signal power from (4.62) by the sum of the interference and the noise powers (which follow from (4.65), (4.66), (4.68), and (4.69)) [49, 104]. The SINR can thus be calculated as in (4.40) for all the users.

4.9 Appendix

4.9.1 MMSE Channel Estimation and MMSE Combining with UAD

In the channel estimation process in Sec. 4.4, we use the low complexity MMSE (LCMMSE) estimator. We now derive the MMSE estimator similar to the previous chapter, accounting for the UAD process. After the UAD phase, the BS performs channel estimation based on the received pilot signal. The received pilot and data signals are indexed by the decoding iteration since the signals are processed in iterations. For this purpose, we let k denote the current decoding iteration index, and $\mathcal{S}_k$ denote the number

of users not yet decoded up to the kth decoding iteration, with $\mathcal{S}_1 = [M]$ and $\mathcal{S}_k^i \triangleq \mathcal{S}_k \backslash \{i\}$. The received pilot signal at the BS in the tth RB in the kth decoding iteration is given by

$$\mathbf{Y}_t^{\mathtt{p}k} = \textstyle\sum_{i \in \mathcal{S}_k} a_i g_{ti} \mathbf{h}_{ti} \mathbf{p}_i^H + \mathbf{N}_t^{\mathtt{p}}, \tag{4.70}$$

where the first term contains signals from users who have transmitted in the current frame. Since the BS has detected only users in $\hat{\mathcal{A}} = \{i \in [M] | \hat{a}_i = 1\}$ to be active, it constructs channel estimates only for these users.

We now derive the MMSE channel estimates at the BS in each RB. Let $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$ be the set of users who would have transmitted in the tth RB had they been active. Let $\mathcal{M}_t^k = \mathcal{G}_t \cap \mathcal{S}_k \cap \hat{\mathcal{A}}$, where $M_t^k = |\mathcal{M}_t^k|$. This is the set of users who have been detected to be active in the tth RB but have not been decoded up to the kth iteration at the BS. Let us stack the channels of the M_t^k users as the columns of $\mathbf{H}_t^k \in \mathbb{C}^{N \times M_t^k}$, let $\mathbf{P}_t^k \in \mathbb{C}^{\tau \times M_t^k}$ denote a matrix that contains the pilot sequences of the M_t^k users as its columns, and let $\mathbf{B}_t^k \triangleq \sigma_{\mathtt{h}}^2 \mathrm{diag}(\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_{M_t^k}})$ be a diagonal matrix that contains the path loss coefficients of the M_t^k users, with $\mathcal{M}_t^k = \{i_1, i_2, \ldots, i_{M_t^k}\}$. Hence, the received pilot signal from (4.70) can be written as $\mathbf{Y}_t^{\mathtt{p}k} = \mathbf{H}_t^k \mathbf{P}_t^{kH} + \mathbf{N}_t^{\mathtt{p}}$. We now find the channel estimates using the signal $\mathbf{Y}_t^{\mathtt{p}k}$.

Theorem 4.4 ▶ MMSE Channel Estimation in IRSA Accounting for UAD Errors.

The minimum mean squared error (MMSE) channel estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ can be found as

$$\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k} (\mathbf{P}_t^k \mathbf{B}_t^k \mathbf{P}_t^{kH} + N_0 \mathbf{I}_\tau)^{-1} \mathbf{P}_t^k \mathbf{B}_t^k. \tag{4.71}$$

Specifically, the estimate of the channel $\mathbf{h}_{ti}$ of the ith user at the BS is calculated as $\hat{\mathbf{h}}_{ti}^k = [\hat{\mathbf{H}}_t^k]_{:,i}$. Further, the estimation error $\tilde{\mathbf{h}}_{ti}^k \triangleq \hat{\mathbf{h}}_{ti}^k - \mathbf{h}_{ti}$ is distributed as

$\tilde{\mathbf{h}}_{ti}^k \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^k \mathbf{I}_N)$, where δ_{ti}^k is calculated as

$$\delta_{ti}^k = \beta_i \sigma_{\mathtt{h}}^2 \left(\frac{N_0 \|\mathbf{c}_{ti}^k\|^2 + \sum_{j \in \mathcal{S}_k^i} |\mathbf{p}_j^H \mathbf{c}_{ti}^k|^2 \hat{a}_j a_j g_{tj} \beta_j \sigma_{\mathtt{h}}^2}{N_0 \|\mathbf{c}_{ti}^k\|^2 + \sum_{j \in \mathcal{S}_k} |\mathbf{p}_j^H \mathbf{c}_{ti}^k|^2 \hat{a}_j a_j g_{tj} \beta_j \sigma_{\mathtt{h}}^2} \right), \tag{4.72}$$

where $\mathbf{C}_t^k \triangleq \mathbf{P}_t^k \mathbf{D}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{D}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}$, $\mathbf{c}_{ti}^k \triangleq [\mathbf{C}_t^k]_{:,i}$, and $\mathbf{D}_t^k \triangleq \mathrm{diag}(d_{ti_1}, d_{ti_2}, \ldots, d_{ti_{M_t^k}})$, with $d_{ti} = \hat{a}_i a_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2$.

Proof. The proof is similar to the proof of Theorem 3.1 in Chapter 3. □

Remarks: The MMSE channel estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ can be written as seen in (4.71) as

$$\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k} (\mathbf{P}_t^k \mathbf{B}_t^k \mathbf{P}_t^{kH} + N_0 \mathbf{I}_\tau)^{-1} \mathbf{P}_t^k \mathbf{B}_t^k, \tag{4.73a}$$

$$\overset{(a)}{=} \mathbf{Y}_t^{\mathtt{p}k} \mathbf{P}_t^k \mathbf{B}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{B}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}, \tag{4.73b}$$

where (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$. Here, the estimate can be calculated via an inverse of either a $\tau \times \tau$ matrix or an $M_t^k \times M_t^k$ matrix as required.

4.9.1.1 SINR Calculation

Similar to (4.70), we can find the received data signal in the tth RB in the kth decoding iteration as

$$\mathbf{y}_t^k = \sum_{i \in \mathcal{S}_k} a_i g_{ti} \mathbf{h}_{ti} x_i + \mathbf{n}_t. \tag{4.74}$$

Here, we have that $\mathbb{E}[x_i] = 0$ and $\mathbb{E}[|x_i|^2] = p_i$.

$$\begin{aligned} \tilde{y}_{tm}^k &= a_m g_{tm} x_m \mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k - a_m g_{tm} x_m \mathbf{a}_{tm}^{kH} \tilde{\mathbf{h}}_{tm}^k \\ &\quad + \sum_{i \in \mathcal{S}_k^m \cap \mathcal{P}} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH} \mathbf{h}_{ti} + \sum_{i \in \mathcal{S}_k^m \cap \mathcal{M}} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH} \mathbf{h}_{ti} + \mathbf{a}_{tm}^{kH} \mathbf{n}_t. \end{aligned} \tag{4.75}$$

We use a generic combining vector $\mathbf{a}_{tm}^k$ to combine the received data signal across

antennas to obtain the post-combined data signal $\tilde{y}_{tm}^k \triangleq \mathbf{a}_{tm}^{kH}\mathbf{y}_t^k$ as seen in (4.75). The term $T_1 \triangleq a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k$ is the desired signal of the mth user; the term $T_2 \triangleq a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^k$ is due to the estimation error $\tilde{\mathbf{h}}_{tm}^k$ of the mth user's channel; the term $T_3 \triangleq \sum_{i\in\mathcal{S}_k^m\cap\mathcal{P}} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}$ models the inter-user interference from other true positive users (who have transmitted and have not yet been decoded); the term $T_4 \triangleq \sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH}\mathbf{h}_{ti}$ is the interference from false negative users (who have transmitted, but cannot be decoded since they are declared to be inactive); and $T_5 \triangleq \mathbf{a}_{tm}^{kH}\mathbf{n}_t$ is the additive noise. Since noise is uncorrelated with the other terms and the data streams of distinct users are uncorrelated, all the terms are uncorrelated with each other. The power in the received signal is a sum of the powers of the terms. Based on the post-combined data signal, we now compute the SINR.

Theorem 4.5 ▶ SINR in IRSA Accounting for UAD Errors.

The signal to interference plus noise ratio (SINR) achieved by the mth user at the CPU in the tth RB and the kth decoding iteration can be written as

$$\rho_{tm}^k = \frac{\mathtt{Gain}_{tm}^k}{N_0 + \mathtt{Est}_{tm}^k + \mathtt{MUI}_{tm}^k + \mathtt{FNU}_{tm}^k}, \forall m \in \mathcal{S}_k. \tag{4.76}$$

Here, $\mathtt{Gain}_{tm}^k$ is the desired signal power, $\mathtt{Est}_{tm}^k$ is the power of the channel estimation error, $\mathtt{MUI}_{tm}^k$ is the multi-user interference due to other true positive users, and $\mathtt{FNU}_{tm}^k$ is the interference due to false negative users. These can be calculated as

$$\mathtt{Gain}_{tm}^k \triangleq p_m \hat{a}_m a_m g_{tm} |\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k|^2 / \|\mathbf{a}_{tm}^k\|^2, \tag{4.77}$$

$$\mathtt{Est}_{tm}^k \triangleq \sum_{i\in\mathcal{S}_k} p_i \hat{a}_i a_i g_{ti} \delta_{ti}^k, \tag{4.78}$$

$$\mathtt{MUI}_{tm}^k \triangleq \sum_{i\in\mathcal{S}_k^m} p_i \hat{a}_i a_i g_{ti} |\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{ti}^k|^2 / \|\mathbf{a}_{tm}^k\|^2, \tag{4.79}$$

$$\mathtt{FNU}_{tm}^k \triangleq \sum_{i\in\mathcal{S}_k} p_i (1-\hat{a}_i) a_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2. \tag{4.80}$$

Proof. The proof is similar to the proof of Theorem 4.2 in Sec. 4.8.3. □

The channel estimates $\hat{\mathbf{h}}_{ti}^k$ and the error covariance δ_{ti}^k in the above expressions are

obtained from Thm. 4.4. The combining vector that maximizes the SINR in (4.76) is the MMSE combiner [49], which can be found as

$$\mathbf{A}_t^k = \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p}}^k (d_t^k \mathbf{I}_{M_t^k} + \hat{\mathbf{H}}_t^{kH} \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p}}^k)^{,-1} \tag{4.81}$$

$$= (d_t^k \mathbf{I}_N + \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p}}^k \hat{\mathbf{H}}_t^{kH})^{-1} \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p}}^k, \tag{4.82}$$

where $\mathbf{D}_{t,\mathtt{p}}^k \triangleq \mathrm{diag}(p_{i_1}, p_{i_2}, \ldots, p_{i_{M_t^k}})$ contains the power coefficients of the M_t^k users, and $d_t^k \triangleq N_0 + \sum_{i \in \mathcal{S}_k} p_i \hat{a}_i g_{ti} b_{ti}^k$, where

$$b_{ti}^k = \beta_i \sigma_{\mathtt{h}}^2 \left(\frac{N_0 \|\mathbf{f}_{ti}^k\|^2 + \sum_{j \in \mathcal{S}_k^i} |\mathbf{p}_j^H \mathbf{f}_{ti}^k|^2 \hat{a}_j g_{tj} \beta_j \sigma_{\mathtt{h}}^2}{N_0 \|\mathbf{f}_{ti}^k\|^2 + \sum_{j \in \mathcal{S}_k} |\mathbf{p}_j^H \mathbf{f}_{ti}^k|^2 \hat{a}_j g_{tj} \beta_j \sigma_{\mathtt{h}}^2} \right). \tag{4.83}$$

Here, $\mathbf{F}_t^k \triangleq \mathbf{P}_t^k \mathbf{E}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{E}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}$, with $\mathbf{f}_{ti}^k \triangleq [\mathbf{F}_t^k]_{:,i}$, and $\mathbf{E}_t^k \triangleq \mathrm{diag}(e_{ti_1}, e_{ti_2}, \ldots, e_{ti_{M_t^k}})$, where $e_{ti} \triangleq \hat{a}_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2$.

Thus, we have derived the MMSE channel estimator similar to the previous chapter, accounting for the UAD process.

4.9.2 Channel Estimation – Pilot Aided vs Data Aided

This section investigates the application of data aided channel estimation as opposed to pilot aided channel estimation. It also describes the effect of imperfect SIC due to channel estimation errors on the performance of IRSA.

First, we briefly discuss the handling of imperfect SIC in the literature. Coded slotted aloha (CSA), which is a variant of IRSA, has been recently analyzed with imperfect SIC [88]. Also, square-norm-based (SNB) interference subtraction and payload-aided-based (PAB) subtraction techniques have been used to analyze the effect of imperfect SIC in CSA [124]. The SNB scheme performs IC on the post-combined pilot signal $\mathbf{y}_{tm}^{\mathtt{p}k}$ using $\|\mathbf{h}_{tm}\|^2$, i.e., norm-squared of the channel. This assumes perfect CSI and neglects the cross products between channels of different users. The PAB scheme treats the decoded data symbols as virtual pilots and re-estimates the channels of the decoded user. This yields a significantly higher quality channel estimate compared to the one obtained via

only the pilots. The SBS SIC technique performs well in the massive MIMO setup, and it is outperformed by the PAB scheme, especially as the payload length increase. IRSA has also been analyzed with an SIC limit [86], i.e., a limit on maximum number of packets that can be recovered in each RB (which, however, is different from imperfect SIC).

We now describe a data-aided (DA) channel estimation process. Let $\mathbf{Y}_t^{\mathtt{d}} \in \mathbb{C}^{N \times \tau_{\mathtt{d}}}$ be the received data signal of users, where $\tau_{\mathtt{d}}$ is the number of data symbols in any packet. This can be written as

$$\mathbf{Y}_t^{\mathtt{d}} = \sum\nolimits_{i=1}^{M} a_i g_{ti} \mathbf{h}_{ti} \mathbf{x}_i^H + \mathbf{N}_t^{\mathtt{d}}, \tag{4.84}$$

where $\mathbf{x}_i \in \mathbb{C}^{\tau_{\mathtt{d}}}$ is the data vector of the ith user. After the mth user is decoded, $\mathbf{x}_m$ is known at the BS, and it can compute a least squares (LS) DA channel estimate similar to [124] as follows:

$$\hat{\mathbf{h}}_{tm}^{\mathrm{LS,DA}} = \mathbf{Y}_t^{\mathtt{d}} \mathbf{x}_m / \|\mathbf{x}_m\|^2. \tag{4.85}$$

However, the LS channel estimate is not uncorrelated with the estimation error [49], and thus, we cannot write out a meaningful SINR expression [104]. This is needed to quantify the SINR in order to use the SINR threshold model. Thus, we do not employ the LS channel estimate.

Since linear minimum mean squared error (LMMSE) estimates are uncorrelated with their estimation errors, we compute LMMSE estimates using decoded data symbols as a virtual pilot sequence. Firstly, the received data signal is right multiplied by the decoded data packet as

$$\mathbf{y}_{tm}^{\mathrm{DA}} = \mathbf{Y}_t^{\mathtt{d}} \mathbf{x}_m = a_m g_{tm} \mathbf{h}_{tm} \|\mathbf{x}_m\|^2 + \sum\nolimits_{i \neq m} a_i g_{ti} \mathbf{h}_{ti} \mathbf{x}_i^H \mathbf{x}_m + \mathbf{N}_t^{\mathtt{d}} \mathbf{x}_m, \tag{4.86}$$

where $\mathbf{y}_{tm}^{\mathrm{DA}}$ is the post-combined signal which is to be used for DA channel estimation. The LMMSE channel estimate can be obtained as

$$\hat{\mathbf{h}}_{tm}^{\mathrm{LMMSE,DA}} = \mathbb{E}\left[\mathbf{h}_{tm} \mathbf{y}_{tm}^{\mathrm{DA}H}\right] \left(\mathbb{E}\left[\mathbf{y}_{tm}^{\mathrm{DA}} \mathbf{y}_{tm}^{\mathrm{DA}H}\right]\right)^{-1} \mathbf{y}_{tm}^{\mathrm{DA}} \tag{4.87}$$

$$= \frac{\hat{a}_m g_{tm} \beta_m \sigma_{\mathtt{h}}^2}{N_0 + \hat{a}_m g_{tm} \beta_m \sigma_{\mathtt{h}}^2 \|\mathbf{x}_m\|^2 + P^{\mathtt{p}} \sum_{i \neq m} \hat{a}_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2} \, \mathbf{y}_{tm}^{\mathrm{DA}}. \tag{4.88}$$

For comparison, we now present the MMSE channel estimate that is obtained from the pilot symbols, i.e., the pilot-aided MMSE estimate:

$$\hat{\mathbf{h}}_{tm}^{\mathrm{MMSE,PA}} = \frac{\hat{a}_m g_{tm} \beta_m \sigma_{\mathtt{h}}^2}{N_0 + \hat{a}_m g_{tm} \beta_m \sigma_{\mathtt{h}}^2 \|\mathbf{p}_m\|^2 + \sum_{i \neq m} \hat{a}_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2 \frac{|\mathbf{p}_i^H \mathbf{p}_m|^2}{\|\mathbf{p}_m\|^2}} \, \mathbf{y}_{tm}^{\mathtt{p1}}. \tag{4.89}$$

We note a fundamental difference between the LMMSE DA and the MMSE PA estimates. In the former, the data sequences of other users are unknown while calculating the channel estimates, whereas in the latter, the pilot sequences of other users are known at the BS. This helps the MMSE PA estimation perform much better than LMMSE DA even when the number of data symbols is large.

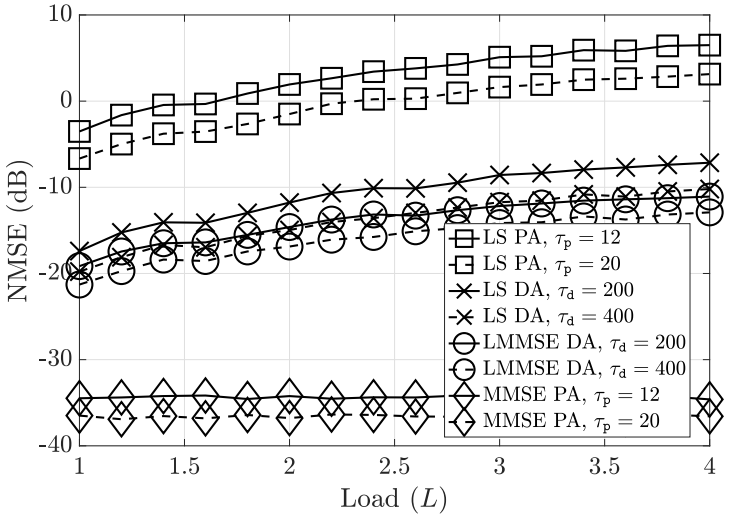


Figure 4.15: NMSE under different channel estimation schemes.

In Fig. 4.15, we plot the normalized mean squared error (NMSE) in channel reconstruction for four different channel estimation schemes. Here, $\tau_{\mathtt{d}}$ denotes the number of data symbols, and $\tau_{\mathtt{p}} = \tau$ denotes the number of pilot symbols. The DA estimates (labeled LS DA/LMMSE DA) are computed using purely the data symbols of the decoded users, and these estimates do not use the pilot symbols. In contrast, the pilot aided (PA) estimates (labeled LS/MMSE) are computed using purely the pilot symbols alone. The

data symbols are generated as random BPSK symbols for DA estimates, whereas for PA estimates, we use Hadamard pilots. We observe that the LS PA estimates perform poorly whereas the MMSE PA estimates perform the best, yielding up to -36 dB NMSE. The DA estimates remain below NMSE of -10 dB. As we increase either $\tau_{\mathtt{p}}$ or $\tau_{\mathtt{d}}$, the performance of all the estimates improve. Thus, we can use DA estimates similar to [124], but they yield a higher NMSE compared to MMSE estimates obtained using the pilot symbols only, which we have used here. This shows that MMSE estimation is quite powerful, and yields a channel estimation error that has very low variance. In turn, this results in lower residual noise after SIC. Thus, we just use the MMSE estimates from the pilot symbols and derive an expression for SINR, as detailed in 4.6.5.

4.9.3 Necessity of UAD in IRSA

This section details the necessity of performing UAD in IRSA. Firstly, we explain the need for a UAD algorithm. In traditional collision-based IRSA, the BS looks for an RB in which a single user has transmitted, i.e., an RB with no collisions [7]. That is, a user's packet can be decoded in an RB if and only if that user is the only one transmitting in that RB (i.e., a singleton slot). Since the user's packet is received without any collisions, the synchronization sequence transmitted by the users as part of the header of the packet itself can be used by the BS to perform accurate channel estimation and then subsequently decode the data packet. Thus, in a collision based model, UAD is not necessary, since decoding succeeds only in singleton slots.

In contrast, we use the SINR-threshold model for decoding in IRSA, with multiple antennas at the BS. Here, users can be decoded even when there are collisions, provided their SINR exceeds a threshold. This approach is more practical than the collision-based model [74], since multiple users could potentially be simultaneously decoded when the BS is equipped with multiple antennas. Now, even though multiple users can be decoded based on the SINR analysis, their pilot signals still collide, and thus, the synchronization signal (which can be viewed as a common pilot signal across the users) alone is not sufficient to perform either activity detection or channel estimation of multiple such users.

In this chapter, similar to [6, 53], the goal of using pilot sequences is to detect active users as well as estimate the channels of the users declared to be active. This approach has been pursued previously in the GFRA literature, in the case where multiple antennas are employed at the BS [6]. UAD is necessary when users' packets collide, and we wish to potentially recover all the packets using multiple antennas at the BS. We note that using non-orthogonal pilots is also enough for detecting the active users in each RB, estimating their channels, and performing decoding. We follow this viewpoint, and the protocol is still distributed in nature [53]. The BS tries to decode the users using the channel estimates that it computes in the pilot phase. As a consequence, without UAD, the BS would waste valuable resources (especially in the mMTC scenario) trying to decode users that have not even transmitted any packets, i.e., users that are inactive. Further, the quality of the channel estimates computed at the BS would be poor while trying to decode the users from collided packets. Knowing the subset of active users helps the BS decide which users it needs to decode, thereby saving computational resources and also aiding the channel estimation process. It also helps with conserving valuable resources at the BS, since it can focus on decoding the packets of users detected to be active.

Finally, we explain the relation of our analysis to unsourced random access (URA). In URA, introduced by Polyanskiy in [56], the base station (BS) only aims to decode the messages transmitted by the users, and the identity of the users is not important. Initial work on IRSA [7] and coded slotted aloha (CSA) [54] predate the idea of unsourced multiple access [56, 58, 59]. While it is true that IRSA can also be used as a URA protocol, it can also be used in massive random access without *requiring* the access to be "unsourced". In this chapter, we are not specifically looking at IRSA as a URA protocol. In particular, we do not insist that the BS be able to identify which user is transmitting, nor do we preclude it from being able to do so.

Part 3

Chapter 5

Censored-IRSA for Interference-Limited mMTC

5.1 Introduction

Massive machine-type communications (mMTC) is an evolving use-case in next generation wireless technologies that is expected to serve millions of devices per square km [125]. Typical mMTC devices sporadically transmit their data to a central base station (BS), and then sleep until the next time when they have data to transmit [3]. In order to serve mMTC applications efficiently, we need to use distributed massive random access protocols such as irregular repetition slotted aloha (IRSA) [7]. The performance of IRSA depends on the *load* of the system, which is the ratio of the number of users participating in a frame to the number of slots in the frame. Existing works in IRSA [7, 11] talk about an *inflection load*, which is the load beyond which the system becomes overloaded or interference-limited, resulting in a dramatic reduction of the throughput of IRSA. In this chapter, we address the issue of the poor throughput of IRSA in the overloaded regime by proposing a *distributed self-censoring* protocol which allows the system to maintain the throughput at the maximum possible value even as the load increases.

The contents of this chapter is published in part in a conference paper, for the SISO case, in IEEE ICC in 2023 [18], and a full-length journal paper, for the MIMO case, in

under preparation [19].

5.1.1 Interference Limitation in IRSA

In IRSA, each user samples a repetition factor from a predefined distribution, and then transmits those many replicas of their packets over multiple (randomly chosen) slots in a frame [82]. If the BS decodes a user in a slot, it uses the decoded data to perform successive interference cancellation (SIC) in all the other slots in which the user has transmitted a packet replica [79]. The decodability of a user in IRSA depends on its signal to interference plus noise ratio (SINR) [17]. The SINR of the user drops if the user has a poor channel state or there are too many collisions resulting in high multi-user interference (MUI). At low loads, the system is not MUI-limited, and the packet loss rate (PLR) is near-zero [86]. Beyond the *inflection load*, the system is overloaded, IRSA is MUI-limited, and the PLR rapidly goes to one [8]. In this case, allowing only users with good channel states to transmit increases their decodability by improving their SINRs via reducing MUI.

To tackle the MUI-limitation of IRSA in overloaded mMTC, we develop a novel *censored-IRSA (C-IRSA) protocol,* as follows. At the start of each frame, the BS transmits a pilot signal using which the users estimate their channel state information (CSI). Users with poor CSI *self-censor*, i.e., they refrain from transmitting, which reduces collisions and enables successful decoding of the users with good CSI. A user has good CSI if $\|\mathbf{v}\|^2 \geq \nu$,where $\mathbf{v}$ is the fading channel of the user and ν is a censor threshold that can be chosen at the BS based on the system load and periodically broadcast to the users. Note that this approach retains the fully distributed nature of IRSA. A high censor threshold can lead to too few users participating, resulting in low throughputs, whereas a low threshold can lead to too many collisions, again resulting in low throughput. Analyzing the trade-off between the censor threshold and the throughput is one of the focuses of this chapter.

5.1.2 Related Works

IRSA was initially proposed in [7] for the collision channel, wherein packets could only be decoded in singleton slots. Singleton slots refer to slots in which only a single packet has been received. The maximum throughput of IRSA, with a collision channel, was shown to be one, when the Soliton distribution is used to generate the repetition factors [72]. When the BS is equipped with multiple antennas, it can potentially decode multiple packets in a single slot, if the SINRs of the packets are sufficiently high, which can result in throughputs greater than 1. Thus, an *SINR threshold* model has been considered for IRSA, where users can be decoded if and only if their SINR exceeds a threshold [8, 74]. IRSA has been studied with several practical aspects such as the Rayleigh fading channel [17], with path loss [8], and with multiple antennas and *pilot contamination* [11]. In particular, in mMTC applications, since it is not possible to assign orthogonal pilots to all users, the resulting pilot contamination can significantly degrade the performance of IRSA [42].

The age-of-information metric has recently gained interest in IRSA [51, 79]. IRSA has been examined with energy harvesting [81], and analyzed in an information theoretic setting [82]. Several variants of aloha have been proposed like polar slotted aloha [83] and K-repetition [84, 85]. We have previously analyzed IRSA with channel estimation errors [11], pilot contamination [12], user activity detection [15], and multi-cell deployments [24]. Density evolution has been used to characterize the asymptotic throughput of IRSA [8, 11, 17]. Authors in [86] propose a feedback mechanism for IRSA based on which decoded users cease transmissions.

However, none of the above papers address the dramatic reduction in the throughput at high loads, which is the main focus in this chapter.

5.1.3 Contributions

The contributions of this chapter are as follows:

1. Firstly, in Sec. 5.2, we develop censored-IRSA (C-IRSA) to tackle the MUI-limitation of IRSA at high system loads. This involves self-censoring of users, wherein users

with poor CSI refrain from transmitting, which decreases the effective system load and ensures that the uncensored users are all successfully decoded.

2. Secondly, in Sec. 5.3, we empirically analyze the performance of C-IRSA accounting for path loss, channel estimation errors, MIMO fading, and pilot contamination.

3. Next, in Sec. 5.4, using density evolution (DE), we theoretically analyze the performance of C-IRSA when the users perform path loss inversion based power control.

4. In Sec. 5.5, we derive the optimal censor threshold accounting for path loss, MIMO, fading, channel estimation, and pilot contamination. We present two approaches: the first is a semi-analytic approach, whereas the second is an algorithmic approach that is PLR-optimal. Using these choices of the censor threshold, the PLR of uncensored users can be driven close to zero at all system loads, while maintaining the throughput of the system at its highest value.

Using extensive numerical simulations, we show that, C-IRSA operates at the full throughput at all loads, in contrast to vanilla IRSA which has near-zero throughput as the load is increased. In particular, at high loads, C-IRSA offers a $10\times$ throughput improvement over IRSA without user censoring.

Notation

The symbols a, $\mathbf{a}$, $\mathbf{A}$, $[\mathbf{A}]_{i,:}$, $[\mathbf{A}]_{:,j}$, $\mathbf{0}_N$, $\mathbf{1}_N$, and $\mathbf{I}_N$ denote a scalar, a vector, a matrix, the ith row of $\mathbf{A}$, the jth column of $\mathbf{A}$, all-zero vector of length N, all ones vector of length N, and an identity matrix of size $N \times N$, respectively. $[\mathbf{a}]_{\mathcal{S}}$ and $[\mathbf{A}]_{:,\mathcal{S}}$ denote the elements of $\mathbf{a}$ and the columns of $\mathbf{A}$ indexed by the set $\mathcal{S}$, respectively. $\mathrm{diag}(\mathbf{a})$ is a diagonal matrix with diagonal entries given by $\mathbf{a}$. The set of real and complex matrices of size $N \times M$ are denoted as $\mathbb{R}^{N\times M}$ and $\mathbb{C}^{N\times M}$. $\mathcal{N}(\mathbf{a}, \mathbf{A})$ and $\mathcal{CN}(\mathbf{a}, \mathbf{A})$ denote the real and complex Gaussian distribution, respectively, with mean $\mathbf{a}$ and covariance $\mathbf{A}$. $[N]$ denotes the set $\{1, 2, \ldots, N\}$. $|\cdot|$, $\|\cdot\|$, $[\cdot]^T$, $[\cdot]^*$, $[\cdot]^H$, $\mathbb{E}[\cdot]$, and $\mathbb{E}_{\mathbf{a}}[\cdot]$ denote the magnitude (or cardinality of a set), ℓ_2 norm, transpose, conjugate, hermitian, expectation, and the expectation

conditioned on $\mathbf{a}$, respectively. The superscript $\mathtt{p}$ is used as a descriptive superscript in association with a symbol that is related to the *pilots*. All the other superscripts (or subscripts) that have not been defined as above are indices.

5.2 System Model

We consider a typical IRSA setup where M single-antenna users want to communicate with a BS having N antennas, over frames consisting of T slots each. These M users are located arbitrarily within a cell centered at the BS location. mMTC applications use similar settings as narrowband internet of things, which uses a bandwidth of 180 kHz [3]. Over this band, the channel can be assumed to be flat and Rayleigh block fading. The BS allocates a pre-specified narrowband channel, and all the M users transmit their packets within this band.[1] The *system load,* L, is defined as the ratio of the number of users to the number of slots per frame, $L \triangleq M/T$. In any given frame, the users randomly select a subset of the T slots, and transmit replicas of their packets in those slots. In vanilla IRSA, when L is high, there are too many collisions in each slot, leading to a failure in the SIC decoding process and therefore poor performance.

In C-IRSA, users censor themselves from transmitting if they have a poor channel state. The BS computes the censor threshold, denoted by ν, based on the system load L.[2] The BS occasionally broadcasts the threshold ν to all the users. The BS also transmits a pilot signal at the start of each frame, using which the users estimate their channel state. The users participate in any IRSA frame if and only if the norm squared of its fading vector exceeds the censor threshold. We refer to the users who self-censor as *inactive* or *censored* users, and the other users as *active* or *uncensored* users. A censored user can sleep till the next time it has data to transmit, by when its channel state would change. In the overloaded regime, vanilla IRSA has near-zero throughput due to too many collisions. At such high L, performing censoring only helps improve the performance of

[1]The BS can distribute the load over several bands, but the problem formulation does not change as IRSA is used in each band.

[2]The BS can estimate the system load from the collisions in the previous IRSA frames. We discuss the optimal choice of ν in Sec. 5.5.

the system. Thus, we allow users who self-censor to drop their packets, and not perform any re-transmissions in subsequent frames.

In a given frame, the access of the T slots is represented as an *access pattern matrix* (APM) $\mathbf{G} \in \{0,1\}^{T\times M}$ [11]. If the mth user (if it were active) transmits its packet in the tth slot, then $g_{tm} \triangleq [\mathbf{G}]_{tm} = 1$; otherwise $g_{tm} = 0$. This user transmits a symbol x_m with $\mathbb{E}[x_m] = 0$ and $\mathbb{E}[|x_m|^2] = p_m$. Thus, the received *data* signal at the BS in the tth slot is

$$\mathbf{y}_t = \textstyle\sum_{m=1}^{M} a_m g_{tm} \mathbf{h}_m x_m + \mathbf{n}_t, \tag{5.1}$$

where $\mathbf{h}_m = \sqrt{\beta_m}\mathbf{v}_m$ is the uplink channel vector of the mth user, $\mathbf{v}_m \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_N, \sigma_{\mathtt{h}}^2\mathbf{I}_N)\ \forall\, m \in [M]$ is the uplink fading channel of the mth user (assumed independent across users and frames), β_m is the path loss coefficient of the mth user,[3] a_m is the activity coefficient of the mth user ($a_m = 1$ if the mth user is active and $a_m = 0$ otherwise), and $\mathbf{n}_t \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_N, N_0\mathbf{I}_N)$ is the complex AWGN at the BS, $\sigma_{\mathtt{h}}^2$ is the fading variance, and N_0 is the noise variance. We define the signal to noise ratio (SNR) as $\rho_0 \triangleq P\sigma_{\mathtt{h}}^2/N_0$, and the *cell edge SNR* as $\rho_{\mathrm{e}} \triangleq P\beta_{\text{edge}}\sigma_{\mathtt{h}}^2/N_0$, where P denotes the transmit power of a user at the cell edge. The cell edge SNR ρ_{e} refers to the received SNR of a user placed at the edge of the cell, if that were the only user transmitting, with β_{edge} denoting that user's path loss coefficient.

Similar to (5.1), we can write the received *pilot* signal at the BS in the tth slot as

$$\mathbf{Y}_t^{\mathtt{p}} = \textstyle\sum_{m=1}^{M} a_m g_{tm} \mathbf{h}_m \mathbf{p}_m^H + \mathbf{N}_t^{\mathtt{p}}, \tag{5.2}$$

where $\mathbf{N}_t^{\mathtt{p}}$ is the additive Gaussian noise, and $\mathbf{p}_m \in \mathbb{C}^\tau$ is the pilot sequence employed by the mth user. Here, τ is the length of the pilot sequence and $[\mathbf{N}_t^{\mathtt{p}}]_{nr} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0)$, $\forall\ n \in [N],\ r \in [\tau]$ and $t \in [T]$.

Remark 1: At the start of each frame, the BS transmits a pilot signal, using which the users estimate their channel state. The ith user participates in the frame if and only if the norm squared of its fading-only estimate $\hat{\mathbf{v}}_i$ exceeds the threshold ν. Thus,

[3]For a study of C-IRSA accounting for the effects of shadowing, see Sec. 5.9.8.

we have that $a_i = \mathbb{1}\{\|\hat{\mathbf{v}}_i\|^2 \geq \nu\}, \forall i \in [M]$. The set of active users is denoted by $\mathcal{A} \triangleq \{i \in [M] | \|\hat{\mathbf{v}}_i\|^2 \geq \nu\}$, and the number of active users is $M_a \triangleq |\mathcal{A}|$.[4] The *active load* L_a is defined as $L_a \triangleq M_a/T$.

Remark 2: We note that users self-censor based only on the fading vector $\mathbf{v}_i$, which does not include path loss. This makes the censoring process fair and not dependent on the user locations. Further, the users do not send any information to the BS, such as its estimated CSI. The BS still needs to detect the set of active users and then estimate their channels.

5.2.1 Decoding Process

The BS first carries out an activity detection phase based on which it knows the subset of users that have not self-censored. The BS then processes the received pilot and data signals iteratively. In every slot, the BS attempts to decode the users' packets. If a user is successfully decoded, which can be verified via a cyclic redundancy check, then the BS performs SIC in all slots in which that user has transmitted a packet [7]. This process repeats and the decoding proceeds in iterations until no new packets can be decoded.

In this work, the SINR threshold model: any packet is decoded correctly if and only if its SINR is above a threshold $\gamma_{\text{th}} \geq 1$ [17, 74]. With the SINR threshold model, the performance of C-IRSA can be computed as follows. First, the SINRs achieved by all users in all slots is computed. If there is a user with SINR $\geq \gamma_{\text{th}}$ in some slot, that packet is successfully decoded and its contribution is removed all other slots in which that user has transmitted a replica [8]. We then proceed to the next decoding iteration and recompute the SINRs for all users yet to be decoded. This process stops when no additional users are decoded in two successive iterations. The throughput $\mathcal{T}$ is calculated as the number of unique packets correctly decoded divided by the number of slots.

[4]Due to channel reciprocity, users can estimate their uplink channels using the downlink pilots. The CSI is used only for self-censoring, and is not sent to the BS. The details of this channel estimation process is presented in Section 5.9.3.

5.3 Performance Analysis of C-IRSA

At the BS, the users' packets are decoded using the SIC process as with vanilla IRSA. This entails a user activity detection (UAD) phase, followed by a channel estimation phase, and finally a data decoding phase. The BS first performs UAD to detect the subset of active users in each slot. This can be performed, for example, using the UAD algorithm proposed in [15].[5] For simplicity, we assume perfect UAD, i.e., the BS has perfect knowledge of $\mathcal{A}$. The results presented in this work can be easily extended to include UAD errors as seen in Chapter 4,[6] where it is shown that a short pilot sequence length used for channel estimation at the BS is also sufficient for accurate UAD.[7]

5.3.1 Channel Estimation

The BS first performs channel estimation based on the received pilot signal. The received pilot and data signals are indexed by the decoding iteration since the signals are processed in iterations. For this purpose, we let k denote the current decoding iteration index, and $\mathcal{S}_k$ denote the set of users not yet decoded up to the kth decoding iteration, with $\mathcal{S}_1 = [M]$ and $\mathcal{S}_k^i \triangleq \mathcal{S}_k \setminus \{i\}$. The received pilot signal at the BS in the tth slot in the kth decoding iteration is given by

$$\mathbf{Y}_t^{\mathrm{p}k} = \textstyle\sum_{i\in\mathcal{S}_k} a_i g_{ti} \mathbf{h}_i \mathbf{p}_i^H + \mathbf{N}_t^{\mathrm{p}}, \tag{5.3}$$

where the first term contains signals from uncensored users.

We now derive the channel estimates at the BS in each slot. Let $\mathcal{G}_t \triangleq \{i \in [M] | g_{ti} = 1\}$ be the set of users who would have transmitted in the tth slot had they been active. Let $\mathcal{M}_t^k \triangleq \mathcal{G}_t \cap \mathcal{S}_k \cap \mathcal{A}$ and $M_t^k \triangleq |\mathcal{M}_t^k|$. This is the set of M_t^k active users who have

[5] Details of the UAD process can be found in Chapter 4.

[6] Analysis including UAD errors is presented in Section 5.9.4.

[7] In practice, the UAD errors become negligibly small when the pilot length is chosen as $\tau = k_1 \log(L) + k_2$, where k_1 and k_2 are constants. This guarantee can be derived from [122], and it ensures that low pilot lengths are sufficient for extremely accurate UAD. We have explored the performance with tuned k_1 and k_2 for $\tau = k_1 \log(L) + k_2$, and have found that we achieve near ideal UAD. For more information, check out the results presented in Section 5.9.5.

transmitted in the tth slot but have not been decoded up to the kth iteration at the BS. We stack the channels of the M_t^k users as the columns of $\mathbf{H}_t^k \in \mathbb{C}^{N\times M_t^k}$, let $\mathbf{P}_t^k \in \mathbb{C}^{\tau\times M_t^k}$ denote a matrix that contains the pilot sequences of the M_t^k users as its columns, and let $\mathbf{B}_t^k \triangleq \sigma_{\mathtt{h}}^2 \mathrm{diag}(\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_{M_t^k}})$ be a diagonal matrix that contains the path loss coefficients of the M_t^k users, with $\mathcal{M}_t^k = \{i_1, i_2, \ldots, i_{M_t^k}\}$. Hence, the received pilot signal from (5.3) can be written as $\mathbf{Y}_t^{\mathtt{p}k} = \mathbf{H}_t^k \mathbf{P}_t^{kH} + \mathbf{N}_t^{\mathtt{p}}$. We now find the channel estimates using $\mathbf{Y}_t^{\mathtt{p}k}$.

Lemma 3. *The minimum mean squared error (MMSE) channel estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ is*

$$\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k}(\mathbf{P}_t^k \mathbf{B}_t^k \mathbf{P}_t^{kH} + N_0 \mathbf{I}_\tau)^{-1} \mathbf{P}_t^k \mathbf{B}_t^k. \tag{5.4}$$

Specifically, the MMSE estimate of the channel $\mathbf{h}_i$ of the ith user at the BS is calculated as $\hat{\mathbf{h}}_{ti}^k \triangleq [\hat{\mathbf{H}}_t^k]_{:,i}$. Further, the estimation error $\tilde{\mathbf{h}}_{ti}^k \triangleq \hat{\mathbf{h}}_{ti}^k - \mathbf{h}_i$ is distributed as $\tilde{\mathbf{h}}_{ti}^k \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^k \mathbf{I}_N)$, where δ_{ti}^k is calculated as

$$\delta_{ti}^k = \beta_i \sigma_{\mathtt{h}}^2 \left(\frac{N_0 \|\mathbf{c}_{ti}^k\|^2 + \sum_{j\in\mathcal{S}_k^i} |\mathbf{p}_j^H \mathbf{c}_{ti}^k|^2 a_j g_{tj} \beta_j \sigma_{\mathtt{h}}^2}{N_0 \|\mathbf{c}_{ti}^k\|^2 + \sum_{j\in\mathcal{S}_k} |\mathbf{p}_j^H \mathbf{c}_{ti}^k|^2 a_j g_{tj} \beta_j \sigma_{\mathtt{h}}^2} \right), \tag{5.5}$$

with $\mathbf{C}_t^k \triangleq \mathbf{P}_t^k \mathbf{D}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{D}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}$, $\mathbf{c}_{ti}^k \triangleq [\mathbf{C}_t^k]_{:,i}$, and $\mathbf{D}_t^k \triangleq diag(d_{ti_1}, d_{ti_2}, \ldots, d_{ti_{M_t^k}})$, with $d_{ti} \triangleq a_i g_{ti} \beta_i \sigma_{\mathtt{h}}^2$.

Proof. This can be derived in a similar fashion as the proof of Theorem 3.1 in Chapter 3. □

Remark 3: We note that the channel estimate is indexed by the slot index, since we obtain different channel estimates in different slots. The MMSE channel estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ can be written as seen in (5.4) as

$$\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k}(\mathbf{P}_t^k \mathbf{B}_t^k \mathbf{P}_t^{kH} + N_0 \mathbf{I}_\tau)^{-1} \mathbf{P}_t^k \mathbf{B}_t^k, \tag{5.6a}$$

$$\overset{(a)}{=} \mathbf{Y}_t^{\mathtt{p}k} \mathbf{P}_t^k \mathbf{B}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{B}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}, \tag{5.6b}$$

where (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$. Here, the estimate can be calculated as an inverse of either a $\tau \times \tau$ matrix or an $M_t^k \times M_t^k$ matrix. Also, higher the value of δ_{ti}^k, higher is the estimation error as well as pilot contamination.

5.3.2 SINR Calculation

Similar to (5.3), we can write the received data signal in the tth slot in the kth decoding iteration as

$$\mathbf{y}_t^k = \textstyle\sum_{i\in\mathcal{S}_k} a_i g_{ti} \mathbf{h}_i x_i + \mathbf{n}_t. \tag{5.7}$$

We use a generic combining vector $\mathbf{a}_{tm}^k$ to combine the received data signal across antennas to obtain the post-combined data signal $\tilde{y}_{tm}^k \triangleq \mathbf{a}_{tm}^{kH}\mathbf{y}_t^k$ as

$$\begin{aligned}\tilde{y}_{tm}^k = {} & a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k - a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^k \\ & + \textstyle\sum_{i\in\mathcal{S}_k^m} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH}\mathbf{h}_i + \mathbf{a}_{tm}^{kH}\mathbf{n}_t.\end{aligned} \tag{5.8}$$

The term $T_1 \triangleq a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k$ is the desired signal of the mth user; the term $T_2 \triangleq a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^k$ is due to the estimation error $\tilde{\mathbf{h}}_{tm}^k$ of the mth user's channel; the term $T_3 \triangleq \sum_{i\in\mathcal{S}_k^m} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH}\mathbf{h}_i$ models the inter-user interference from other active users; and $T_4 \triangleq \mathbf{a}_{tm}^{kH}\mathbf{n}_t$ is the additive noise. Since noise is uncorrelated with the other terms and the data streams of distinct users are uncorrelated, all the terms are uncorrelated with each other. The power in the received signal is the sum of the powers of the terms. Based on the post-combined data signal, we now compute the SINR.

Lemma 4. *The signal to interference plus noise ratio (SINR) achieved by the mth user at the BS in the tth slot and the kth decoding iteration in C-IRSA can be written as*

$$\rho_{tm}^k = \frac{\mathtt{Gain}_{tm}^k}{N_0 + \mathtt{Est}_{tm}^k + \mathtt{MUI}_{tm}^k}, \forall m \in \mathcal{S}_k. \tag{5.9}$$

Here $\texttt{Gain}_{tm}^k$ *is the desired signal power,* $\texttt{Est}_{tm}^k$ *is the power of the channel estimation error,* $\texttt{MUI}_{tm}^k$ *is the multi-user interference. These can be calculated as*

$$\texttt{Gain}_{tm}^k \triangleq p_m a_m g_{tm} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k|^2 / \|\mathbf{a}_{tm}^k\|^2, \tag{5.10a}$$

$$\texttt{Est}_{tm}^k \triangleq \textstyle\sum_{i \in \mathcal{S}_k} p_i a_i g_{ti} \delta_{ti}^k, \tag{5.10b}$$

$$\texttt{MUI}_{tm}^k \triangleq \textstyle\sum_{i \in \mathcal{S}_k^m} p_i a_i g_{ti} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{ti}^k|^2 / \|\mathbf{a}_{tm}^k\|^2. \tag{5.10c}$$

Proof. This can be derived in a similar fashion as the proof of Theorem 4.3 in Chapter 4. □

Remark 4: The channel estimates $\hat{\mathbf{h}}_{ti}^k$ and the error covariance δ_{ti}^k in the above expressions are obtained from Lemma 3. The combiner that maximizes the SINR in (5.9) is the MMSE combiner [49], which can be found as

$$\mathbf{A}_t^k = \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathrm{p}}^k (d_t^k \mathbf{I}_{M_t^k} + \hat{\mathbf{H}}_t^{kH} \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathrm{p}}^k)^{-1} \tag{5.11}$$

$$= (d_t^k \mathbf{I}_N + \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathrm{p}}^k \hat{\mathbf{H}}_t^{kH})^{-1} \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathrm{p}}^k, \tag{5.12}$$

where $\mathbf{D}_{t,\mathrm{p}}^k \triangleq \mathrm{diag}(p_{i_1}, p_{i_2}, \ldots, p_{i_{M_t^k}})$ contains the power coefficients of the M_t^k users, and $d_t^k \triangleq N_0 + \sum_{i \in \mathcal{S}_k} p_i a_i g_{ti} \delta_{ti}^k$.

The empirical performance of C-IRSA can be found as detailed in Alg. 4. The decoding is run for $k_{\max}$ iterations, and the output is the system throughput, $\mathcal{T}$, the PLR of the active users, PLR_a, and the system PLR, PLR. We now characterize the theoretical performance of C-IRSA and then discuss the optimal choice of the censor threshold.

5.4 Theoretical Analysis of C-IRSA

In the previous section, we described the empirical evaluation of the performance of C-IRSA, as detailed in Alg. 4. In this section, we describe the process of density evolution (DE) [7, 11], which is used to characterize the theoretical performance of C-IRSA. DE is applicable as M_a and $T \to \infty$ with a fixed $L_a = M_a/T$ [17], i.e., it yields the asymptotic

Algorithm 4: Performance Evaluation of C-IRSA

Input: $\tau, N, T, M, \rho_0, N_0, k_{\max}, \mathbf{G}, \mathbf{P}, \{\mathbf{Y}_t^{\mathtt{p}}\}_{t=1}^{T}, \mathcal{A}$

Initialize: $\mathcal{S}_1 = [M]$, $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$

for $k = 1, 2, \ldots, k_{\max}$ **do**

  for $t = 1, 2, \ldots, T$ **do**

    Find $\mathcal{M}_t^k = \mathcal{G}_t \cap \mathcal{S}_k \cap \mathcal{A}$, $\mathbf{P}_t^k = [\mathbf{P}]_{:,\mathcal{M}_t^k}$, $\mathbf{Y}_t^{\mathtt{p}k}$

    Obtain channel estimates $\hat{\mathbf{H}}_t^k$ from (5.4)

    Evaluate the SINR $\rho_{ti}^k,\ \forall i \in \mathcal{S}_k$ from (5.9)

    If $\rho_{ti}^k \geq \gamma_{\text{th}}$, remove user i from $\mathcal{S}_k$ and perform SIC in all slots where $g_{ti} = 1$

  end

end

Output: $\mathsf{PLR} = |\mathcal{S}_{k_{\max}}|/M,\ \mathcal{T} = M(1 - \mathsf{PLR})/T$, $\mathsf{PLR}_a = |\mathcal{A} \cap \mathcal{S}_{k_{\max}}|/|\mathcal{A}|$.

throughput. Hence, we describe the DE process in terms of the *active load,* L_a. We only outline the high-level steps in the analysis here; detailed discussion of the DE process can be found in several references [7,8,11,17]. For the DE analysis in this section, for analytical tractability, we assume that users perform path loss inversion (PLI) based power control, and that the BS has perfect knowledge of the CSI.

SIC decoding has been viewed as message passing on a bipartite graph [7,8], and thus, C-IRSA can be decoded on graphs as well. The underlying bipartite graph contains the user nodes on one side, the slot nodes on the other, and the edges between them. There exists an edge between a user node and a slot node if and only if that user has transmitted a packet in the slot. During decoding, an edge is removed if the user connected to it has been decoded in any of the slots. The process of an edge removal corresponds to an SIC operation. Decoding is successful, if all edges in the graph are removed by the end of the decoding process. A failure is declared if not all edges are removed and no new edge is removed from the graph in two consecutive iterations.

With every user, there is an associated *repetition factor*, which is the number of replicas that user has transmitted in a given frame. With every slot, there is an associated

collision factor, which is the number of packets that have collided in that slot. The *node-perspective user degree distribution* is defined as the set of probabilities $\{\phi_d\}_{d=2}^{d_{\max}}$, where ϕ_d is the probability that a user has a repetition factor d; with minimum and maximum repetition factors of 2 and $d_{\max}$, respectively. The *edge-perspective user degree distribution* is defined as the set of probabilities $\{\lambda_d\}_{d=2}^{d_{\max}}$, where $\lambda_d = d\phi_d/\phi'(1)$ is the probability that an edge is connected to a user with repetition factor d. The corresponding polynomial representations of the node- and edge- perspective user degree distributions are

$$\phi(x) = \sum_{d=2}^{d_{\max}} \phi_d x^d, \quad \lambda(x) = \sum_{d=2}^{d_{\max}} \lambda_d x^{d-1}, \tag{5.13}$$

respectively. The average repetition is $\bar{d} \triangleq \sum_d d\phi_d$.

We now use the above degree distributions to find a pair of interdependent *failure probabilities* denoted by "p_i" and "q_i" in the ith decoding iteration. When a decoding failure happens, an edge passes a failure message between the user and the slot nodes. This happens when the user connected to an edge is not decoded in the slot that is connected to the same edge, in the current decoding iteration. We denote the probability that an edge carries a failure message from a slot node to a user node by p_i, and the probability that an edge carries a failure message from a user node to a slot node by q_i. The failure probability q_i is calculated using the edge-perspective user degree distribution as

$$q_i = \sum_{d=2}^{d_{\max}} \lambda_d p_{i-1}^{d-1} = \lambda(p_{i-1}). \tag{5.14}$$

Here, p_{i-1}^{d-1} is the probability that an edge carries a failure message in the ith iteration given that it is connected to a user node with repetition factor d. If all the other $d-1$ incoming edges to that user node carry failure messages in the previous iteration, then the edge will carry a failure message from that user node in the ith iteration. The failure probability p_i is calculated as in [11, 17] as

$$p_i = 1 - e^{-L_a \bar{d} q_i} \sum_{r=1}^{\infty} \theta_r \frac{(L_a \bar{d} q_i)^{r-1}}{(r-1)!} \triangleq f(q_i). \tag{5.15}$$

Here, θ_r is the probability that a reference packet gets decoded in any iteration in a slot of degree r using only *intra-slot SIC* [17]. Intra-slot SIC refers to interference cancellation within the same slot a user is decoded in, whereas inter-slot SIC refers to interference cancellation in a different slot in which a user is decoded. The evaluation of θ_r, which is the crucial step in the DE process, is described in Theorem 5.1.

In DE, $q_i = \lambda(p_{i-1})$ and $p_i = f(q_i)$ are calculated recursively as functions of each other using (5.14) and (5.15). The iterations are initialized with either $q_0 = 1$ or $p_0 = f(1)$. At the end of the iterations, the failure probability is obtained as $p_\infty = \lim_{i\to\infty} p_i$. The probability that a packet from a user with repetition factor d does not get decoded at all is $(p_\infty)^d$. Thus, the asymptotic PLR of the active users (PLR_a), which is the fraction of packets of active users that are not decoded at the BS, can be found as

$$\mathsf{PLR}_a = \phi(p_\infty) = \textstyle\sum_{d=2}^{d_{\max}} \phi_d (p_\infty)^d. \tag{5.16}$$

Now, for *CSI based censoring* with a threshold ν, the fraction of active users is denoted by $\bar{\mathrm{F}}(\nu)$. Since the channel states of users are i.i.d., we can calculate $\bar{\mathrm{F}}(\nu)$ as $\bar{\mathrm{F}}(\nu) = \Pr(\|\mathbf{v}\|^2 \geq \nu)$. The active load L_a of the system is then $L_a = L\bar{\mathrm{F}}(\nu)$.[8] Since the fraction of censored users in the system is $\mathrm{F}(\nu) = 1 - \bar{\mathrm{F}}(\nu)$, the effective PLR of the system (including censored users) can be calculated as

$$\mathsf{PLR} = \mathrm{F}(\nu) + \bar{\mathrm{F}}(\nu)\mathsf{PLR}_a. \tag{5.17}$$

Using the asymptotic PLR, we can now obtain the throughput $\mathcal{T}$ of the users in the system as

$$\mathcal{T} = L(1 - \mathsf{PLR}) = L_a(1 - \mathsf{PLR}_a). \tag{5.18}$$

The iterations $p_i = f(\lambda(p_{i-1}))$ converge to $p_\infty = 0$ if the active load $L_a < L_a^*$, asymptotically [7,11]. Here, L_a^* is called the *active inflection load* of the system, and it corresponds

[8] Similar to [17], L_a is calculated using the fraction of the average number of active users as opposed to the instantaneous number of active users.

to a *system inflection load* of $L^* = L_a^*/\bar{\mathrm{F}}(\nu)$, with a threshold ν. For $L_a < L_a^*$, since $p_\infty = 0$, we have $\mathsf{PLR}_a = 0$, $\mathsf{PLR} = \mathrm{F}(\nu)$, and $\mathcal{T} = L\bar{\mathrm{F}}(\nu) = L_a$. For any $L_a \geq L_a^*$, PLR_a does not converge to 0, and $\mathcal{T}$ decreases monotonically with L_a. Also, from (5.17), we see that $\mathsf{PLR} \geq \mathrm{F}(\nu)$, and thus, $\mathcal{T} \leq L\bar{\mathrm{F}}(\nu)$.

Note that in (5.15), p_i depends on the active load L_a, which itself is a function of the threshold ν as $L_a = L\bar{\mathrm{F}}(\nu)$. Consequently, the active inflection load L_a^*, the system inflection load L^*, the active PLR, the system PLR, and the throughput $\mathcal{T}$ are all dependent on ν.

We now describe the evaluation of the success probability θ_r.

5.4.0.1 Single Input Single Output (SISO)

The result below is applicable for C-IRSA when the BS and UEs are equipped with single antennas.

Theorem 5.1 ▶ Success Probability in Censored-IRSA with SISO.

For the Rayleigh block-fading channel with an SNR ρ_0, a censor threshold ν, and an SINR threshold γ_{th}, the probability that a reference packet gets decoded in a slot of degree r using only intra-slot SIC, can be obtained as

$$\theta_r = \sum_{k=1}^{r} \frac{\exp(r\nu - (r-k)\nu\bar{\gamma}_{\mathrm{th},k} - \rho_0^{-1}(\bar{\gamma}_{\mathrm{th},k} - 1))}{r\ \bar{\gamma}_{\mathrm{th},k}^{r-(k+1)/2}}, \tag{5.19}$$

where $\bar{\gamma}_{\mathrm{th},k} = (1+\gamma_{\mathrm{th}})^k$, and $\nu \leq \rho_0^{-1}\gamma_{\mathrm{th}}$.

Proof. See Section 5.8.1. □

Remark 5: When $\nu = 0$, i.e., there is no censoring, the expression for θ_r matches with the results by Clazzer et. al. [17].

5.4.0.2 Multiple Input Multiple Output (MIMO)

Closed form expressions for the success probability θ_r are difficult to compute [8, 11] due to several reasons. With MIMO, there is no clear ordering of the peak SINRs across intra-slot SIC iterations. Further, the channel estimates of the users are correlated across the intra-slot SIC iterations as well as across the user indices, which makes order statistics not directly applicable. Finally, θ_r is dependent on a large number of random channel vectors, the order statistics of the peak SINRs, and the pilot sequences of the users. Thus, we now present two approximations to θ_r, which are valid when perfect CSI is available at the BS and the BS uses maximal ratio combining (MRC) to decode the users. These lead to interpretable expressions for the SINR and θ_r, and provide approximations to the throughput with estimated CSI.

Theorem 5.2 ▶ Success Probability in Censored-IRSA with MIMO.

When perfect CSI is available at the BS, and MRC is used for decoding users in C-IRSA, θ_r can be calculated as follows. Firstly, θ_1 can be exactly found as

$$\theta_1 = \begin{cases} \Gamma_{\text{inc}}(N, \rho_0^{-1}\gamma_{\text{th}})/\Gamma_{\text{inc}}(N,\nu), & \nu \leq \rho_0^{-1}\gamma_{\text{th}}, \\ 1, & \nu > \rho_0^{-1}\gamma_{\text{th}}. \end{cases} \tag{5.20}$$

where $\Gamma_{\text{inc}}(s,x) = \int_x^\infty t^{s-1}\, e^{-t}\, \mathrm{d}t$ is the upper incomplete gamma function. Next, θ_2 can be calculated as

$$\theta_2 = \mathbb{1}\{t_0(\nu) > 1\} + (1-(1-t_0(\nu))^N)\mathbb{1}\{0 \leq t_0(\nu) \leq 1\}. \tag{5.21}$$

Here, $t_0(\nu) \triangleq \gamma_{\text{th}}^{-1} - (\bar{H}(\nu)\rho_0)^{-1}$, $\bar{H}(\nu) \triangleq N + \nu^N/S_N(\nu)$, and $S_N(\nu) \triangleq (N-1)!\sum_{k=0}^{N-1}(\nu^k/k!)$, with $S_N(0) \triangleq 1$.

For $r \geq 3$, two approximations to θ_r with large N are described below. The *Gamma* approximation is

$$\textit{Gamma:} \quad \theta_r = 1 - \Gamma_{\text{inc}}(r-1, Nt_0(\nu))/\Gamma(r-1). \tag{5.22}$$

With $\mu_N \triangleq (N+1)^{-1}$, and $\sigma_N^2 \triangleq N(N+1)^{-2}(N+2)^{-1}$, we obtain the *Normal* approximation

$$\textit{Normal:} \quad \theta_r = 1 - \mathcal{Q}\left(\frac{t_0(\nu) - (r-1)\mu_N}{\sqrt{r-1}\sigma_N}\right), \tag{5.23}$$

where $\mathcal{Q}(\cdot)$ is the standard Normal Q-function.

Proof. See Section 5.8.2. □

Remark 6: The above approximations are derived using the theory of deterministic equivalents, and they provide closed form expressions for θ_r, and are valid when N is large [106].

5.5 Choice of Censor Threshold

In this section, we discuss the optimal choice of the censor threshold. Firstly, we develop a semi-analytic approach that fixes a target load L_{tgt} based on a target threshold ν_{tgt}. Next, we develop an optimization based approach to calculate the censor threshold in Alg. 5, which computes two censor thresholds ν_1 and ν_2 in the linear throughput and MUI-limited regions of IRSA, respectively. Under both the approaches, the active PLR of C-IRSA can be made to approach zero while maintaining the throughput of the system at its highest. For notational convenience, this section has been written for perfect CSI, but the results are also applicable for estimated CSI.

5.5.1 Semi-Analytic Approach to Find the Censor Threshold

In this subsection, we describe a semi-analytic approach to find the optimal censor threshold for C-IRSA, under both SISO and MIMO cases. The semi-analytic approach is applicable with path loss, channel estimation errors, MMSE combining, pilot contamination, etc.

In order to find the censor threshold, we first choose a target PLR for the active users, $\mathsf{PLR}_{a,\text{tgt}}$, which is a maximum permissible PLR among the active users. Let L_{tgt} be

the *target load*, which is the minimum L at which the system achieves an active PLR of $\mathsf{PLR}_{a,\mathrm{tgt}}$, with $\nu = \nu_{\mathrm{tgt}}$. At L_{tgt}, the active load is $L_a = L_{\mathrm{tgt}}\bar{\mathrm{F}}(\nu_{\mathrm{tgt}})$, with a corresponding throughput $\mathcal{T}_{\mathrm{tgt}}$. For a load $L \geq L_{\mathrm{tgt}}$, we wish to continue to operate at the same PLR of $\mathsf{PLR}_{a,\mathrm{tgt}}$, and to keep the throughput fixed at $\mathcal{T}_{\mathrm{tgt}}$. This can be done by maintaining the same active load L_a at L and L_{tgt}. Thus, we need to choose ν such that

$$L_a = L\bar{\mathrm{F}}(\nu) = L_{\mathrm{tgt}}\bar{\mathrm{F}}(\nu_{\mathrm{tgt}}). \tag{5.24}$$

5.5.1.1 SISO

For the SISO case, we have that $\bar{\mathrm{F}}(\nu) = \Pr(|v|^2 \geq \nu)$. Thus, $\mathrm{F}(x)$ and $\bar{\mathrm{F}}(x) \triangleq 1 - \mathrm{F}(x)$ are the CDF and the CCDF, respectively, of the exponential distribution (of $|v|^2 \sim \exp(1)$) evaluated at x. Since $\bar{\mathrm{F}}(x) = \exp(-x)$, we obtain

$$\nu = \log(L/L_a) = \log(L/L_{\mathrm{tgt}}) + \nu_{\mathrm{tgt}}. \tag{5.25}$$

The above is valid when $L \geq L_{\mathrm{tgt}}$. When $L < L_{\mathrm{tgt}}$, the threshold that maximizes the active inflection load L_a^* is $\nu_{\mathrm{tgt}} = \rho_0^{-1}\gamma_{\mathrm{th}}$. An intuitive reason for this is that the probability of decoding a user, if that user was the only one transmitting in a slot, is $\theta_1 = \Pr(|h_1|^2 \geq \rho_0^{-1}\gamma_{\mathrm{th}} \mid |h_1|^2 \geq \nu) = \exp(\nu - \rho_0^{-1}\gamma_{\mathrm{th}}) \cdot \mathbb{1}\{\nu \leq \rho_0^{-1}\gamma_{\mathrm{th}}\} + \mathbb{1}\{\nu > \rho_0^{-1}\gamma_{\mathrm{th}}\}$, when the threshold is ν. So if we set $\nu > \rho_0^{-1}\gamma_{\mathrm{th}}$ or $\nu < \rho_0^{-1}\gamma_{\mathrm{th}}$, we are censoring more or fewer users than required, respectively. Thus, the optimal choice of the censor threshold is given by the function $g(\cdot,\cdot)$ defined as

$$\nu = g(L, L_{\mathrm{tgt}}) \triangleq \begin{cases} \rho_0^{-1}\gamma_{\mathrm{th}}, & L < L_{\mathrm{tgt}}, \\ \log(L/L_{\mathrm{tgt}}) + \rho_0^{-1}\gamma_{\mathrm{th}}, & L \geq L_{\mathrm{tgt}}. \end{cases} \tag{5.26}$$

For $\nu_{\mathrm{tgt}} = \rho_0^{-1}\gamma_{\mathrm{th}}$, the system inflection load is $L^* = L_a^*/\bar{\mathrm{F}}(\rho_0^{-1}\gamma_{\mathrm{th}})$. For $L_{\mathrm{tgt}} < L^*$, the set of functions $\{g(\cdot,\cdot)\}$ achieve $\mathsf{PLR}_a \leq \mathsf{PLR}_{a,\mathrm{tgt}}$ among the set of active users. In practice, we set a low target PLR of $\mathsf{PLR}_{a,\mathrm{tgt}} \approx 10^{-3}$ or 10^{-4} to ensure near-zero PLR_a. We summarize the result in the theorem below.

Theorem 5.3 ▶ Optimal Censor Function in Censored-IRSA with SISO.

Let $\nu_{\text{tgt}} = \rho_0^{-1}\gamma_{\text{th}}$ and L_{tgt} be the minimum load at which the system achieves a PLR of $\mathsf{PLR}_{a,\text{tgt}}$ using Alg. 4. Then, for any given L, the optimal choice of the censor threshold is given by $\nu = g(L, L_{\text{tgt}})$, where

$$\nu = g(L, L_{\text{tgt}}) \triangleq \begin{cases} \rho_0^{-1}\gamma_{\text{th}}, & L < L_{\text{tgt}}, \\ \log(L/L_{\text{tgt}}) + \rho_0^{-1}\gamma_{\text{th}}, & L \geq L_{\text{tgt}}. \end{cases} \tag{5.27}$$

The above choice of the censor threshold maximizes that throughput while maintaining a maximum PLR of $\mathsf{PLR}_{a,\text{tgt}}$ among the uncensored users.

5.5.1.2 MIMO

For the MIMO case, we have that $\bar{\mathrm{F}}(\nu) = \Pr(\|\mathbf{v}\|^2 \geq \nu)$. Thus, $\mathrm{F}(x)$ and $\bar{\mathrm{F}}(x) \triangleq 1 - \mathrm{F}(x)$ are the CDF and the CCDF, respectively, of the Gamma distribution (of $\|\mathbf{v}\|^2 \sim$ Gamma (rate = 1, shape = N)) evaluated at x. Thus, we have $\bar{\mathrm{F}}(x) \triangleq \exp(-x)\sum_{k=0}^{N-1}(x^k/k!)$.

When the BS is equipped with multiple antennas, it can potentially decode multiple packets in a slot, if the SINRs of the packets are sufficiently high. This is referred to as the *capture effect,* and is the main difference between the SISO and MIMO cases. Packet capture can also occur due to receive power disparity between users arising from the differences in the distances of the users to the BS. Thus, we need to analyze the optimal censor threshold both when we have multiple antennas at the BS as well as when users do not perform PLI. Recall that censoring is done using only the fades, and not including the path loss, to maintain fairness across users.

The target threshold ν_{tgt} maximizes the active inflection load L_a^*. For the SISO case, $\nu_{\text{tgt}} = \rho_0^{-1}\gamma_{\text{th}}$ is the least ν that maximizes L_a^*. In the general case, we choose $\nu_{\text{tgt}} = \rho_{\text{e}}^{-1}\gamma_{\text{th}}$, where ρ_{e} is the cell edge SNR. Here, we choose the cell edge SNR since it ensures that a user at the cell edge gets decoded if that were the only user transmitting. An intuitive reason for choosing $\nu_{\text{tgt}} = \rho_{\text{e}}^{-1}\gamma_{\text{th}}$ is that, as before, the probability of decoding a user, if that user were the only one transmitting in a slot, is $\theta_1 = \Pr(\|\mathbf{v}_1\|^2 \geq \rho_{\text{e}}^{-1}\gamma_{\text{th}} \mid \|\mathbf{v}_1\|^2 \geq \nu)$,

when the threshold is ν. So if we set $\nu > \rho_{\mathrm{e}}^{-1}\gamma_{\mathrm{th}}$ or $\nu < \rho_{\mathrm{e}}^{-1}\gamma_{\mathrm{th}}$, we are censoring more or fewer users than required, respectively.

When $L \geq L_{\mathrm{tgt}}$, the system contains too many users, and is MUI-limited. In this regime, censoring enhances the throughput via a combination of improved channel states among the active users and lower MUI. If we choose the censor threshold similar to SISO, the choice of the threshold with MIMO is given by the function $h(\cdot,\cdot)$ defined as

$$h(L, L_{\mathrm{tgt}}) \triangleq \begin{cases} \nu_{\mathrm{tgt}}, & L < L_{\mathrm{tgt}}, \\ \mathrm{H}(L, L_{\mathrm{tgt}}, \nu_{\mathrm{tgt}}), & L \geq L_{\mathrm{tgt}}. \end{cases} \tag{5.28}$$

Here, the function $\mathrm{H}(\cdot,\cdot,\cdot)$ is defined as

$$\mathrm{H}(L, L_{\mathrm{tgt}}, \nu_{\mathrm{tgt}}) \triangleq \bar{\mathrm{G}}(L_a/L) = \bar{\mathrm{G}}(\bar{\mathrm{F}}(\nu_{\mathrm{tgt}})L_{\mathrm{tgt}}/L). \tag{5.29}$$

Here $x = \bar{\mathrm{G}}(y)$ is the inverse CCDF of the Gamma distribution (with rate 1 and shape N), which is the x such that $y = \bar{\mathrm{F}}(x)$. For $N = 1$, $L \geq L_{\mathrm{tgt}}$, and $\nu_{\mathrm{tgt}} = \rho_{\mathrm{e}}^{-1}\gamma_{\mathrm{th}}$, we have $\nu = \mathrm{H}(L, L_{\mathrm{tgt}}, \nu_{\mathrm{tgt}}) = \log(L/L_{\mathrm{tgt}}) + \rho_{\mathrm{e}}^{-1}\gamma_{\mathrm{th}}$, which coincides with the expression in (5.27).

In (5.28), we censor with a threshold $\nu = \nu_{\mathrm{tgt}}$ for $L < L_{\mathrm{tgt}}$. Censoring need not always help reduce PLR for $L < L_{\mathrm{tgt}}$. This can happen in two cases. In the first case, if the system has zero PLR at a low L, then no censoring is needed since the system is already operating optimally.[9] This is possible, for example, when there are multiple antennas at the BS, due to capture effect. In the second case, the system can be power-limited when the transmit powers of users are not high enough, and thus, the MUI component is low to begin with at low L. Censoring does not reduce MUI significantly, and ensures that there are fewer active users upon censoring, and thus, the throughput reduces. Hence, we do not need to censor users in this case. Thus, we see that the censor function from (5.28) need not always be the optimal function, and we need to account for cases where we do not need to perform any censoring for $L < L_{\mathrm{tgt}}$.

[9] In practice, $\mathsf{PLR} = 0$ is not feasible, but rather only a sufficiently low PLR (for example, $< 10^{-3}$) is achievable. For convenience, in this chapter, we use $\mathsf{PLR} = 0$ to mean a sufficiently low PLR.

We now derive a generalized censor function that maximizes the throughput of C-IRSA at all loads. For this, we define the parameter L_0, which denotes the load up to which users do not self-censor, with $L_0 < L_{\text{tgt}}$. L_0 can be computed as the system inflection load L^* corresponding to the threshold $\nu = 0$.[10] We summarize our choice of the threshold in the theorem below.

Theorem 5.4 ▶ Optimal Censor Function in Censored-IRSA with MIMO.

Let L_0 be the system inflection load L^* corresponding to the threshold $\nu = 0$, $\nu_{\text{tgt}} = \rho_{\text{e}}^{-1}\gamma_{\text{th}}$ be the target threshold, and L_{tgt} be the minimum load at which the system achieves a PLR of $\mathsf{PLR}_{a,\text{tgt}}$ with $\nu = \nu_{\text{tgt}}$, obtained using Alg. 4. The generalized censor function in IRSA is given by $\nu = i(L, L_{\text{tgt}}, L_0)$, where

$$i(L, L_{\text{tgt}}, L_0) \triangleq \begin{cases} 0, & L < L_0, \\ \dfrac{\log(L/L_0)}{\log(L_{\text{tgt}}/L_0)}\nu_{\text{tgt}}, & L_0 \le L < L_{\text{tgt}}, \\ \mathrm{H}(L, L_{\text{tgt}}, \nu_{\text{tgt}}), & L \ge L_{\text{tgt}}, \end{cases} \tag{5.30}$$

Proof. See Section 5.8.3. □

Remark 7: The proposed functions are identical for $L \ge L_{\text{tgt}}$, and different for $L < L_{\text{tgt}}$. The $h(\cdot,\cdot)$ function is an extension of $g(\cdot,\cdot)$, and uses a fixed threshold for $L < L_{\text{tgt}}$. However, $i(\cdot,\cdot,\cdot)$, which has $\nu = 0$ for $L < L_0$, does not use a fixed threshold for $L_0 \le L < L_{\text{tgt}}$. Choosing $L_0 = 0$ does not result in identical $h(\cdot,\cdot)$ and $i(\cdot,\cdot,\cdot)$. In fact, the logarithmic interpolation in $i(\cdot,\cdot,\cdot)$ is not defined for $L_0 = 0$. Further, as we will see in Sec. 5.6, $i(\cdot,\cdot,\cdot)$ maximizes the throughput, whereas $h(\cdot,\cdot)$ minimizes PLR_a.

Remark 8: Recall that we set ν_{tgt} as $\rho_0^{-1}\gamma_{\text{th}}$ with PLI, as opposed to $\rho_{\text{e}}^{-1}\gamma_{\text{th}}$ without PLI. The latter ensures that a user at the cell edge is successfully decoded if they were the only user transmitting in a slot. With PLI, the received power of all users are the same up to

[10] In practice, L_0 can be modelled as a parameter which can be fine-tuned. This choice of L_0 is addressed in Fig. 5.3. Also, we have investigated the performance of C-IRSA with various other censor functions in Sec. 5.9.6.

the variations due to fading. PLI results in higher power consumption per transmission by cell edge users compared to without PLI. Another benefit of not inverting the path loss is the additional packet captures due to the disparity in the received power levels across users.

Via numerical experiments, we have found that the results in Theorem 5.4 hold true when the system model accounts for pilot contamination, MMSE combining, channel estimation errors, and UAD errors. Thus, the semi-analytic approach described here is applicable under any of these practical non-idealities.

5.5.2 Algorithmic Approach to Find the Censor Threshold

In the previous subsection, we presented a semi-analytic approach to find the censor threshold, which uses Alg. 4. In this subsection, we develop an active PLR-optimal algorithm for the same. This algorithmic approach does not require choosing either L_0 or L_{tgt}, and it optimizes the censor threshold by minimizing the active PLR at every L. This algorithmic approach is based on the DE process. Let $\texttt{PLR}_a(\nu)$ be the function (see (5.16)) that outputs the active PLR for the C-IRSA protocol with a censor threshold ν. Since censoring only helps reduce the active PLR, the function $\texttt{PLR}_a(\nu)$ is a non-increasing function of ν. Let $L^*(\nu)$ denote the system inflection load with a censor threshold ν. Thus, for all $L \leq L^*(\nu)$, we have $\texttt{PLR}_a(\nu) \approx 0$, and $\forall L > L^*(\nu)$, we have $\texttt{PLR}_a(\nu) > 0$. As we vary $\nu \in \mathbb{R}^+$, the corresponding $L^*(\nu)$ is also a non-decreasing function on $\mathbb{R}^+$. Since we can operate with any $\nu \in \mathbb{R}^+$, every L can be viewed as a system inflection load for some threshold ν_0, i.e., $L = L^*(\nu_0)$ for some ν_0. Our goal thus reduces to finding ν_0, since the throughput equals L till $L = L^*(\nu_0)$ and reduces thereafter. We now develop an optimization algorithm in order to find this threshold ν_0.

We term the region $L \leq L^*(\nu)$ as the linear throughput region and $L > L^*(\nu)$ as the MUI-limited region of C-IRSA. In the linear throughput region, the system has either optimal censoring or excessive self-censoring since $\texttt{PLR}_a(\nu) = 0$ in this region. Thus, we need to identify a threshold $\nu_1 \geq \nu_0$ such that $\texttt{PLR}_a(\nu_1) = 0$. In the MUI-limited region, enough users have not self-censored. Thus, we need to identify a threshold $\nu_2 < \nu_0$ such

that $\texttt{PLR}_a(\nu_2) > 0$. The optimal value of ν_0 can be identified by performing a line search between ν_1 and ν_2.

Algorithm 5: Algorithm to compute the PLR-optimal censor threshold in C-IRSA

Input: $\nu_1, \nu_2, \nu_\varepsilon, \nu_\delta, p_\varepsilon, \texttt{PLR}_a(\cdot)$

Output: Optimal threshold ν_{algo}

1 **if** $\texttt{PLR}_a(0) \leq p_\varepsilon$ **then**
2 | **Output:** $\nu_{\text{algo}} = 0$ // Output $\nu = 0$ if $\mathsf{PLR}_a = 0$
3 **end**
4 **while** $\texttt{PLR}_a(\nu_1) > p_\varepsilon$ **do**
5 | $\nu_1 = \nu_1 + \nu_\delta$ // Increase ν_1 if $\mathsf{PLR}_a > 0$
6 **end**
7 **while** $\texttt{PLR}_a(\nu_2) \leq p_\varepsilon$ **do**
8 | $\nu_2 = \max\{\nu_2 - \nu_\delta, 0\}$ // Decrease ν_2 if $\mathsf{PLR}_a = 0$
9 | **break** if $\nu_2 = 0$
10 **end**
11 **Initialize:** $\nu = \nu_2$
12 **if** $\texttt{PLR}_a(\nu) > p_\varepsilon$ and $|\nu_1 - \nu_2| > \nu_\varepsilon$ **then**
13 | $\nu_2 = \nu, \nu = (\nu_1 + \nu_2)/2$ // Update ν_2 and ν
14 | **go to** Step 9
15 **else**
16 | **if** $\texttt{PLR}_a(\nu) \leq p_\varepsilon$ and $|\nu_1 - \nu_2| > \nu_\varepsilon$ **then**
17 | | $\nu_1 = \nu, \nu = (\nu_1 + \nu_2)/2$ // Update ν_1 and ν
18 | | **go to** Step 9
19 | **else**
20 | | **Output:** $\nu_{\text{algo}} = \nu_1$ // Optimal solution reached
21 | **end**
22 **end**

We present the PLR-optimal approach in Alg. 5. Here, the active PLR function $\texttt{PLR}_a(\cdot)$ can be found using the DE process as detailed in Sec. 5.4 (specifically using (5.14), (5.15), and (5.16)). We also input tolerance parameters p_ε and ν_ε for the PLR function and the censor threshold respectively, to account for finite precision. If $\nu = 0$ is the optimal threshold, then it is output immediately. This happens for example when $L \leq L^*(\nu = 0)$. The rest of the algorithm proceeds when $L > L^*(\nu = 0)$. The thresholds ν_1 and ν_2 can be initialized with a high $\nu_1 \geq \nu_2$ and either a low ν_2 or $\nu_2 = 0$. If the threshold ν_1 is too low or if $\texttt{PLR}_a(\nu_1) > 0$, then ν_1 is incremented in steps of ν_δ ($\nu_1 = \nu_1 + \nu_\delta$) to ensure $\texttt{PLR}_a(\nu_1) = 0$. If the threshold ν_2 is too high or if $\texttt{PLR}_a(\nu_2) = 0$, then ν_2 is decremented in steps of ν_δ ($\nu_2 = \max\{\nu_2 - \nu_\delta, 0\}$) to ensure $\texttt{PLR}_a(\nu_2) > 0$.

Once we have identified two thresholds ν_1 and ν_2 such that $\texttt{PLR}_a(\nu_1) = 0$, $\texttt{PLR}_a(\nu_2) > 0$, and $\nu_1 \geq \nu_2$, we perform a bisection search between ν_1 and ν_2. We ensure updates to ν_1 and ν_2 are always maintained such that $\texttt{PLR}_a(\nu_1) = 0$, $\texttt{PLR}_a(\nu_2) > 0$, and $\nu_1 \geq \nu_2$, i.e., we ensure that ν_1 is such that the system is in the linear throughput region and ν_2 is such that the system is MUI-limited. Since ν_1 and ν_2 are updated in every iteration to take a step toward each other, the algorithm is guaranteed to converge to the optimal threshold ν_{algo}.

5.5.2.1 Computational Complexity

The computational complexity of the algorithm depends only on the computation of the active PLR function $\texttt{PLR}_a(\cdot)$, since the other operations in the algorithm are only simple addition operations. We recall that $d_{\max}$ is the maximum repetition factor, and $i_{\max}$ is the maximum number of iterations within the DE process. Let $r_{\max}$ be the index up to which θ_r is numerically significant, i.e., $\theta_r > 10^{-3}$. Each iteration in the DE process, namely (5.14) and (5.15), require $d_{\max}(d_{\max}+1)/2 - 1$ and $r_{\max}(r_{\max}+1)/2 + 1$ multiplication operations, respectively. This amounts to a total of $N_{\texttt{PLR}_a}$ multiplication operations per a single computation of $\texttt{PLR}_a(\cdot)$, which can be found as

$$N_{\texttt{PLR}_a} = 0.5(d_{\max}(d_{\max}+1)(i_{\max}+1) + r_{\max}(r_{\max}+1)i_{\max} + 2(d_{\max}-2)). \tag{5.31}$$

The above linearly depends on the number of iterations $i_{\max}$, the square of the maximum repetition factor $d_{\max}$, and the square of the maximum number of collisions $r_{\max}$ for the success probability θ_r. For typical system parameters with $d_{\max} = 4$ and $r_{\max} = 10$, this amounts to a maximum of $N_{\texttt{PLR}_a} = (141 i_{\max} + 24)/2$ multiplication operations in each computation of the active PLR function $\texttt{PLR}_a(\cdot)$. DE usually converges in $i_{\max} = 10$ iterations, which yields a complexity of 717 multiplication operations per computation of $\texttt{PLR}_a(\cdot)$.

5.5.2.2 Choice of Algorithm Parameters

The threshold ν_1 is initialized as a high value sufficient enough to ensure that the system is interference-free amongst the subset of active users, whereas ν_2 is initialized as a low value sufficient enough to ensure that the system is MUI-limited among the active users. For example, $\nu_1 = \rho_0^{-1}\gamma_{\text{th}}$ and $\nu_2 = 0.1\rho_0^{-1}\gamma_{\text{th}}$. The algorithm tunes these initializations and verifies if they are indeed valid. If invalid, the thresholds are varied in steps of $\nu_\delta = \rho_0^{-1}\gamma_{\text{th}}$, till they are validated. The tolerance parameters p_ε and ν_ε can be chosen to be small values such as 10^{-3} or 10^{-4}.

Remark 9: Since the optimal thresholds discussed so far are continuous functions, any change to the system load will change the optimal threshold only by a small quantity. That is, if the algorithm outputs a threshold ν_{algo}, when the load marginally changes, the existing threshold ν_{algo} can be used to next initialize the threshold ν_1. This ensures a faster convergence. Further, the algorithm needs to be run only once at the BS for every L, and not in every frame.

Remark 10: The DE analysis presented in this work contains closed form expressions for θ_r and thus, is an iterative recipe to compute the theoretical active PLR when perfect CSI is available at the BS and the BS uses MRC to combine signals of users across antennas. However, the developed algorithm can be used under any system model, as long as we can evaluate the active PLR function $\texttt{PLR}_a(\cdot)$. Thus, the algorithm itself can be used with MMSE combining, channel estimation errors, UAD errors, pilot contamination, etc., when the PLR function $\texttt{PLR}_a(\cdot)$ can be computed, possibly empirically. Further, the DE

process which yields the function $\mathtt{PLR}_a(\cdot)$ is asymptotic in nature. Thus, for finite frame lengths, we can operate with 5% higher ν_{algo} which operates at a slightly lower $L^*(\nu_{\text{algo}})$.

5.6 Numerical Results

In this section, we evaluate the empirical and theoretical performance of C-IRSA via Monte Carlo runs, and provide insights into the dependence of the system performance on the various parameters. In each run, we generate independent realizations of the user locations, the APM, and the fades of the users. For the empirical results, the number of users is computed as $M = \lfloor LT \rceil$; whereas the theoretical results are dependent only on L, as described in Sec. 5.4. The throughput for each run is calculated using Alg. 4 as described in Sec. 5.3, and the effective throughput is averaged over the runs.

The results in this section are for 10^4 Monte Carlo runs, $T = 50$ slots, $\sigma_{\mathtt{h}}^2 = 1$, $\tau = 10$, $N = 4$, MMSE combining, and SINR threshold $\gamma_{\text{th}} = 30$ [11]. The path loss is calculated as β_i (dB) $= -37.6 \log(r_i/10)$, where r_i is the distance of the ith user from the BS in meters. The location of each user is uniformly sampled from within a cell of radius $r_{\max} = 1$ km centered at the BS. The Soliton distribution [72] $\phi(x) = 0.625x^2 + 0.25x^3 + 0.125x^4$ is used to generate the repetition factors of the users [11].[11] The repetition factor d_i is used to form the access vector for the ith user, by uniformly randomly choosing d_i slots from T slots without replacement [7]. The packet replicas are transmitted in these d_i slots. The power levels of all the users is set to $P = 0$ dBm, $P^{\mathtt{p}} = 3$ dBm, and N_0 is chosen such that $\rho_{\text{e}} = 10$ dB, unless otherwise stated. This ensures that all users' signals are received at an SINR that is at least ρ_{e} on average, in singleton slots. The pilots are chosen as τ-length vectors with random symbols belonging to the QPSK constellation, normalized to have power $\tau P^{\mathtt{p}}$.[12]

We first elucidate the choice of the target load L_{tgt} using PLR_a in Fig. 5.1. The PLR_a of the system increases with L for $\nu = 0, 3$, and 4, and becomes 1 at high loads, since

[11] The repetition distribution is optimized in Chapter 7.

[12] For the performance of C-IRSA with other pilot sequences, see Sec. 5.9.10. We have also studied the uniqueness of pilot sequences in IRSA and C-IRSA in Sec. 5.9.9.

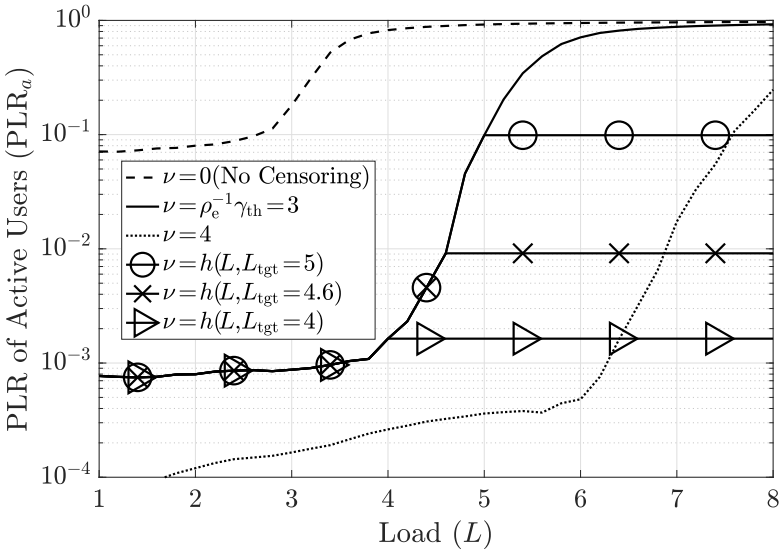


Figure 5.1: Choice of target load L_{tgt}.

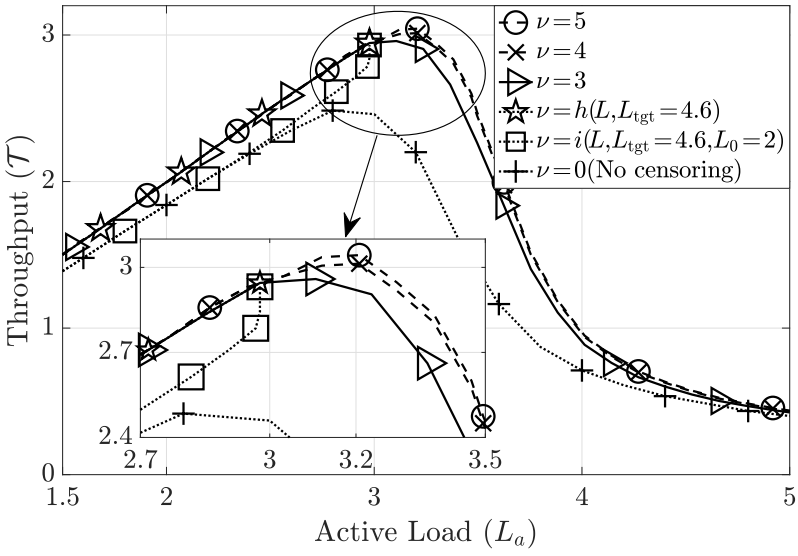


Figure 5.2: Impact of active load L_a on $\mathcal{T}$.

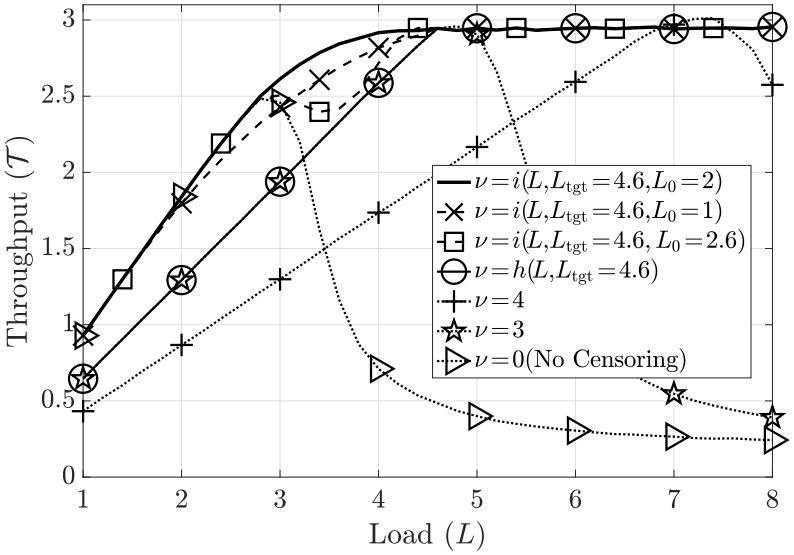


Figure 5.3: Throughput of C-IRSA.

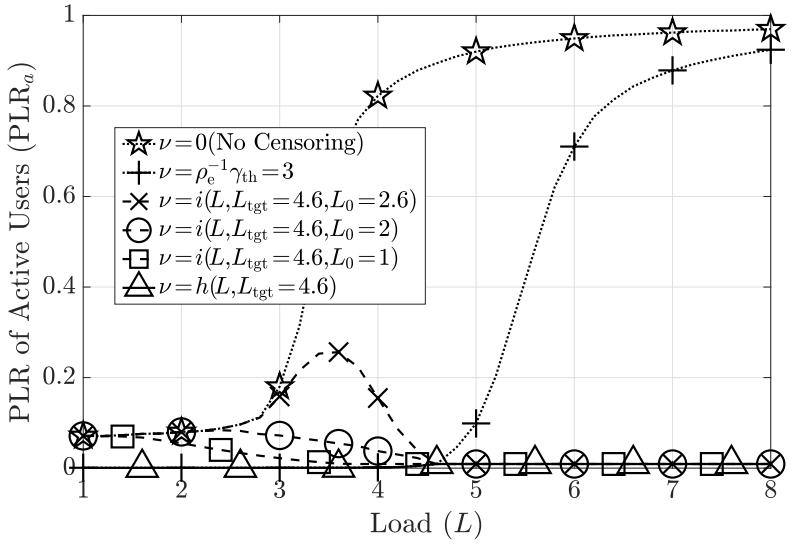


Figure 5.4: Effect of threshold ν on PLR_a.

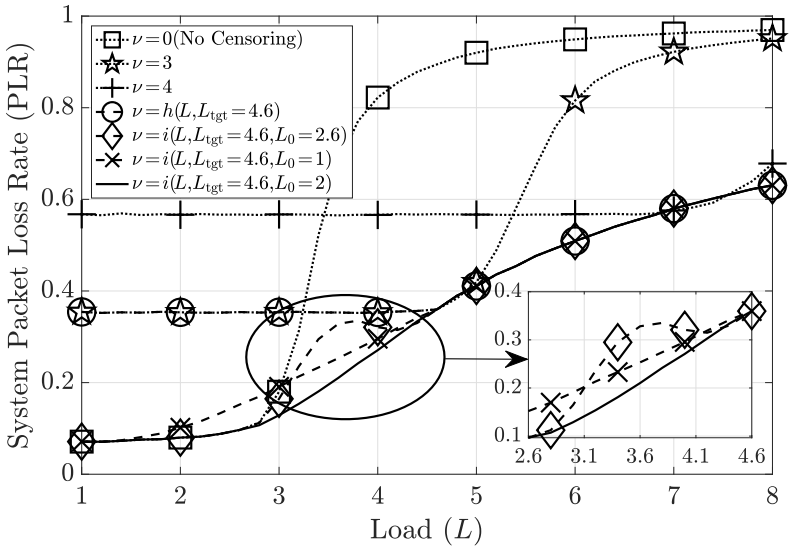


Figure 5.5: Effect of threshold ν on PLR.

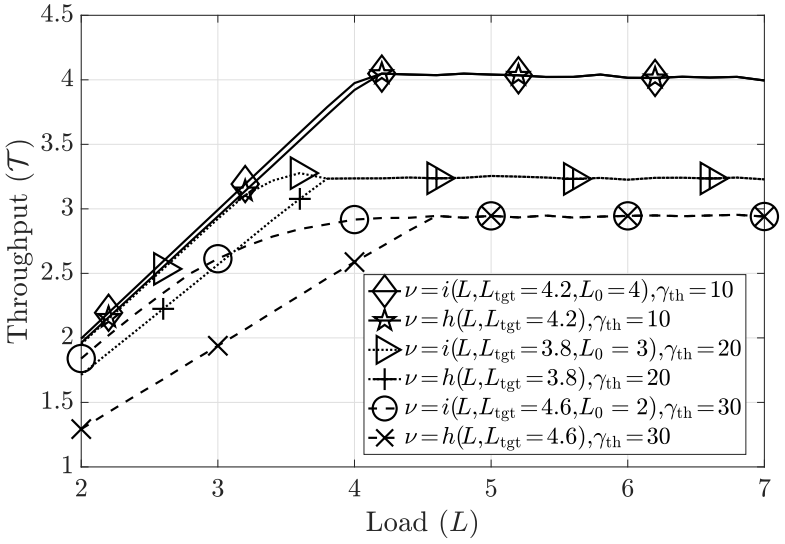


Figure 5.6: Impact of SINR threshold γ_{th} on $\mathcal{T}$.

the system is MUI-limited at higher L. Vanilla IRSA is equivalent to C-IRSA with no censoring, i.e., with $\nu = 0$. It is seen that vanilla IRSA has a high PLR_a at all loads, and, with censoring, the PLR_a reduces. The curves with $\nu = h(L, L_{\text{tgt}})$ follow the performance of $\nu = \rho_{\text{e}}^{-1}\gamma_{\text{th}} = 3$ up to a load of $L = L_{\text{tgt}}$, and beyond that PLR_a stays constant at every load. We first fix a $\mathsf{PLR}_{a,\text{tgt}}$ and the minimum L with which the system achieves $\mathsf{PLR}_a = \mathsf{PLR}_{a,\text{tgt}}$, with $\nu = \rho_{\text{e}}^{-1}\gamma_{\text{th}}$, forms the target load. For example, $\mathsf{PLR}_{a,\text{tgt}} = 10^{-1}$ or 10^{-2} yields $L_{\text{tgt}} = 5$ or 4.6.

In Fig. 5.2, we show the impact of the active load L_a on $\mathcal{T}$. All the curves linearly increase till their respective inflection points, after which they drop down since the system in MUI-limited. The linear increase in $\mathcal{T}$ is the region in which $\mathsf{PLR}_a = 0$ [7], i.e., when $L_a \leq 3$, all active users are decoded. With $N = 4$, the BS can potentially decode four users per slot. However, due to the SINR threshold model, small pilot lengths, pilot contamination, estimated CSI, etc., the number of users that can be decoded per slot is less than 4. On the other hand, the disparity in the received powers across users can occasionally allow the BS to decode even more than 4 users per slot. The net effect is that up to 3 active users can be decoded in practice. At $L_a = 2.7$, $\nu = 0$ achieves $\mathcal{T} = 2.4$, whereas $\nu = 2.7$ achieves full throughput of $\mathcal{T} = L_a = 2.7$. The throughput improves as ν is increased since users with poor channel states are self-censored. Even with a little amount of censoring, C-IRSA performs better than IRSA. Thus, C-IRSA helps overcome packet losses due to both poor CSI and MUI. For $\nu \geq \gamma_{\text{th}}/\rho_{\text{e}}$, $\mathcal{T}$ very marginally increases and the system achieves nearly the same $\mathcal{T}$ for $\nu = 5$ and 4 as for $\nu = 3$. This increase is due to capture effect from both path loss and MIMO fading. Further, the threshold ν must be such that the system is always operated at $L_a \leq L_a^* = 3$. Also, by optimally choosing the threshold using $\nu = h(L, L_{\text{tgt}})$ as described in Sec. 5.5, we can obtain the same $\mathcal{T}$ as that obtained with $\nu = 3$ at higher loads also. Note that in the MUI-limited regime, the PLR of IRSA is non-zero, and both users with poor channel states and packets that experience collisions cannot be decoded correctly. Censoring improves $\mathcal{T}$ on both counts by choosing users whose packets are more likely to be decoded correctly as well as reducing the number of collisions.

In Fig. 5.3, we depict the impact of the developed censor functions on the throughput of C-IRSA. Vanilla IRSA, C-IRSA with $\nu = 3$, and C-IRSA with $\nu = 4$ achieve a peak performance of $\mathcal{T} = 2.5$ at $L = 2.8$, $\mathcal{T} = 2.95$ at $L = 4.6$, and $\mathcal{T} = 3$ at $L = 7.2$, respectively. For all three curves, $\mathcal{T}$ increases linearly with the load up to the peak, and beyond that, the throughput drops to zero. C-IRSA with $\nu = 4$ has too many users which are self-censoring, and we could potentially reduce ν. Thus, we could choose ν for every L such that we obtain an envelope of all curves for $\nu \geq 3$, which yields the same performance as $\nu = i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$. The throughput of C-IRSA with $\nu = i(L, L_{\text{tgt}}, L_0)$ outperforms $\nu = h(L, L_{\text{tgt}})$ for all $L < L_{\text{tgt}} = 4.6$, and then remains constant for $L \geq L_{\text{tgt}} = 4.6$. Up to $L = 2.8$, no censoring is needed, i.e., $\nu = 0$ is sufficient. For $2.8 < L < 4.6$, $\mathcal{T}$ is maximized with $\nu = i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$. For all $L \geq 4.6$, $\mathcal{T}$ remains constant at $\mathcal{T} = 2.95$, with both $\nu = i(L, L_{\text{tgt}} = 4.6, L_0)$ (with any choice of L_0) and $\nu = h(L, L_{\text{tgt}} = 4.6)$. Choosing a high $L_0 = 2.6$ yields a censor function with a low slope (as seen in (5.30)) which censors too few users, whereas choosing a low $L_0 = 1$ yields a censor function with a high slope which censors too many users. L_0 can be fine tuned to yield the peak performance, e.g., with $L_0 = 2$. At $L = 8$, the throughput of vanilla IRSA is $\mathcal{T} = 0.25$, whereas C-IRSA achieves $\mathcal{T} = 2.95$, which is over a $10\times$ improvement. Thus, at high L, the system can be operated at its maximum potential in C-IRSA, compared to vanilla IRSA which has near-zero throughput.

We illustrate the effect of the threshold ν on PLR_a in Fig. 5.4. For $\nu = 3$, we have $\mathsf{PLR}_a = 0$ and thus, $\mathsf{PLR} = \bar{\mathrm{F}}(3) = 0.65$ up to $L = 4.6$. Choosing any $L_{\text{tgt}} < 4.6$ always yield $\mathsf{PLR}_a = 0$ at all L since the active load will then always be maintained at $L_a \leq 2.95$. We thus choose $L_{\text{tgt}} = 4.6$ to maximize $\mathcal{T}$, which can be obtained from our analysis as $L_{\text{tgt}} = L_a^*/\bar{\mathrm{F}}(\rho_{\text{e}}^{-1}\gamma_{\text{th}}) = 2.95/\bar{\mathrm{F}}(3) = 4.6$. Further, when $L \leq 4.6$, we see that $\nu = i(L, L_{\text{tgt}} = 4.6, L_0)$ (with any choice of L_0) has a non-zero PLR_a, whereas $\nu = h(L, L_{\text{tgt}} = 4.6)$ achieves $\mathsf{PLR}_a = 0$. Both the functions have $\mathsf{PLR}_a = 0$ for $L > 4.6$. This is the exact opposite trend on the throughput as seen in Fig. 5.3. This shows that in order to maximize the throughput of the system, with a small hit on the PLR_a, we choose the censor threshold as $\nu = i(L, L_{\text{tgt}}, L_0)$. In order to ensure zero PLR_a with a small hit on $\mathcal{T}$, we choose the

censor threshold as $\nu = h(L, L_{\text{tgt}})$.

In Fig. 5.5, we present the system PLR corresponding to the throughput curves in Fig. 5.3. It is observed that the lowest PLR is obtained with the $\nu = i(\cdot,\cdot,\cdot)$ function with parameters $L_{\text{tgt}} = 4.6$ and $L_0 = 2$. Tuning the value of L_0 helps us achieve the lowest system PLR. These trends are similar to the ones observed in Fig. 5.3 since the throughput and the system PLR have a one-to-one relation (see (5.18)).

In Fig. 5.6, we vary the threshold γ_{th}, with a fixed $\rho_{\text{e}} = 10$. We observe that the peak throughputs of the system increase as we decrease γ_{th}. This is because more users have SINRs that cross γ_{th}. Further, the gap between the functions $i(\cdot,\cdot,\cdot)$ and $h(\cdot,\cdot)$ reduces as we reduce γ_{th}. In general, the less MUI-limited the system becomes at lower L, the closer L_0 and L_{tgt} become, which implies the functions $i(\cdot,\cdot,\cdot)$ and $h(\cdot,\cdot)$ both yield nearly the same performance. This can happen, e.g., with high N or high ρ_{e}. In these cases, we need not censor at low L and censor at high L using the function in (5.29).

The impact of various system parameters on the generalized censor function is presented in Sec. 5.9.7.

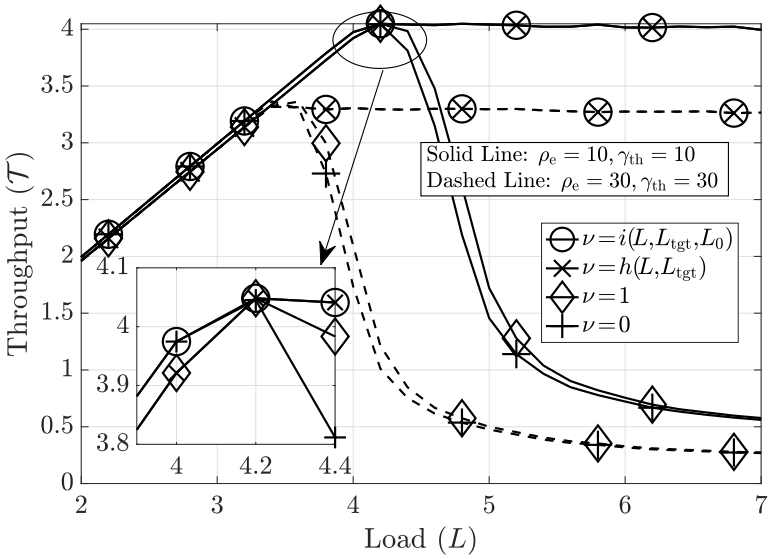


Figure 5.7: Impact of threshold ν on $\mathcal{T}$ with $\rho_{\text{e}}^{-1}\gamma_{\text{th}} = 1$.

In Fig. 5.7, we present the throughput curves for varied censor thresholds with a fixed value of $\rho_{\text{e}}^{-1}\gamma_{\text{th}} = 1$. The curves shift downwards when the parameters are changed from

$\rho_{\mathrm{e}} = \gamma_{\mathrm{th}} = 10$ to $\rho_{\mathrm{e}} = \gamma_{\mathrm{th}} = 30$. This is because fewer users are decoded when the threshold is increased. Further, the gap between $\nu = h(L, L_{\mathrm{tgt}})$ and $\nu = i(L, L_{\mathrm{tgt}}, L_0)$ remains the same for $L < L_{\mathrm{tgt}}$ and beyond L_{tgt}, they stay the same. The performance of $\nu = i(L, L_{\mathrm{tgt}}, L_0)$ is, however, slightly better than $\nu = h(L, L_{\mathrm{tgt}})$ for $L < L_{\mathrm{tgt}}$. This is because, in this regime, not censoring at low loads is good for the system. Note that in this plot, we choose L_0 to be very close to L_{tgt} since the gap is very small.

5.6.1 Asymptotic Results with DE

In this subsection, we evaluate the theoretical throughput of C-IRSA as discussed in Sec. 5.4 and provide insights into the impact of various system parameters on the performance. For this subsection, we assume that perfect CSI is available at the BS, the users perform PLI, and the BS uses MRC for decoding users. The results are presented for $N = 8$ antennas, $\rho_0 = 10$ dB, SINR threshold $\gamma_{\mathrm{th}} = 10$ [11], and 50 maximum DE iterations. For Alg. 5, we use the tolerance parameters $p_\varepsilon = \nu_\varepsilon = 10^{-3}$, step size $\nu_\delta = \rho_0^{-1}\gamma_{\mathrm{th}}$, with initializations $\nu_1 = 2\rho_0^{-1}\gamma_{\mathrm{th}}$ and $\nu_2 = 0.1\rho_0^{-1}\gamma_{\mathrm{th}}$.

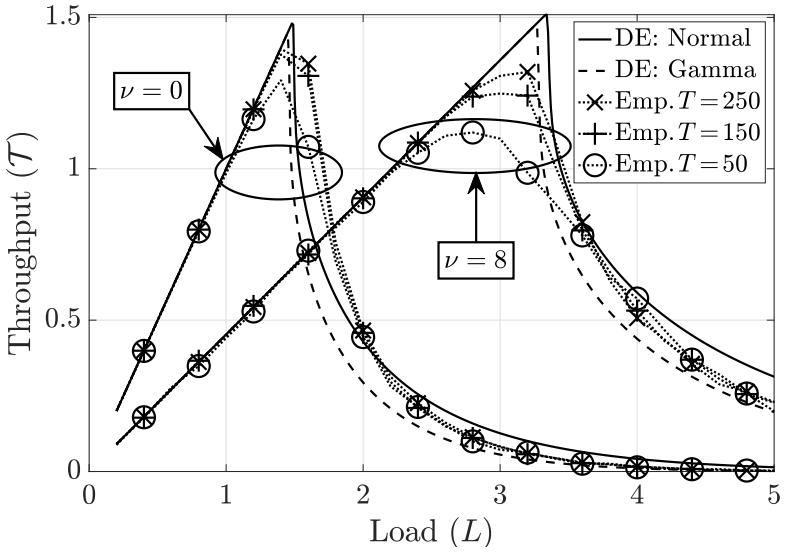


Figure 5.8: Verification of the DE approximations.

In Fig. 5.8, we verify the asymptotic DE $\mathcal{T}$ obtained using the Normal and the Gamma approximations with the empirical $\mathcal{T}$. For both $\nu = 0$ and 8, the Normal approximation

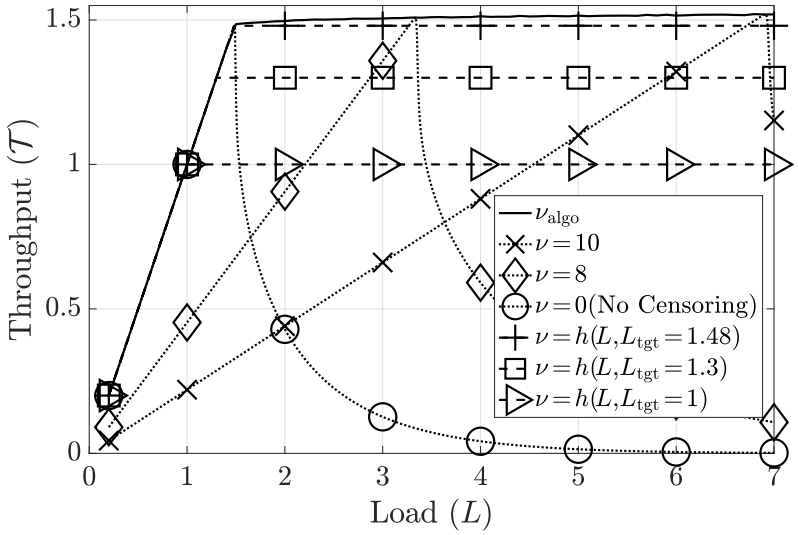


Figure 5.9: Theoretical performance of C-IRSA with $N = 8$.

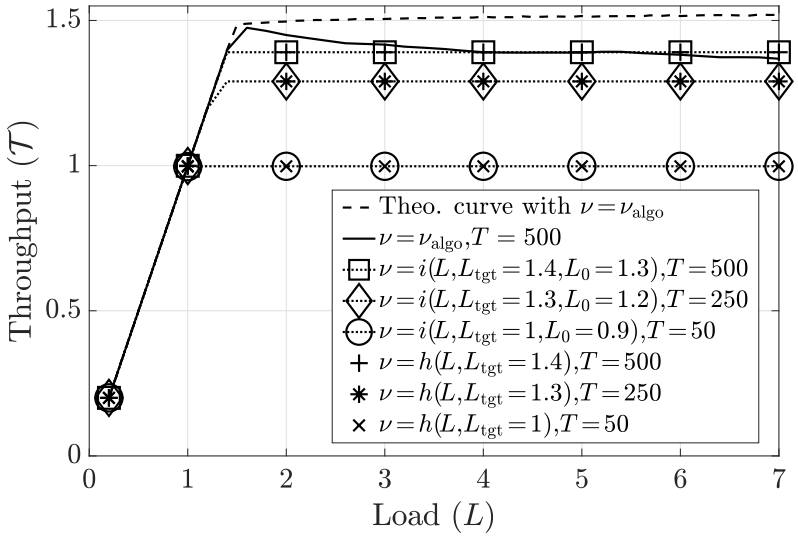


Figure 5.10: Empirical performance of C-IRSA with $N = 8$.

results in $\mathcal{T}$ that is slightly in excess of the empirical $\mathcal{T}$, albeit the inflection loads are approximated well. The Gamma approximation slightly underperforms compared to the Normal approximation but it also achieves the same performance as the empirical throughputs at high L. For both $\nu=0$ and 8, the peak empirical $\mathcal{T}$ increases with an increase in T, and they get closer towards the asymptotic $\mathcal{T}$. The asymptotic $\mathcal{T}$ drops very quickly after the inflection load L^*. The Normal approximation yields an inflection load of $L^*=1.48$ for $\nu=0$, and $L^*=3.33$ for $\nu=8$. At these inflection loads, the peak asymptotic throughputs are $\mathcal{T}=1.48$ and $\mathcal{T}=1.5$, respectively.

Fig. 5.9 shows the theoretical performance of C-IRSA with the Normal approximation for θ_r. As ν is increased from 0 to 8 to 10, there is a marginal increase in the peak throughput. We have skipped the curve for $\nu = 1$ since it overlaps with $\nu = 0$. The function $h(L, L_{\text{tgt}})$ follow the $\nu=0$ curve up to $L=L_{\text{tgt}}$ and remain constant for $L \geq L_{\text{tgt}}$. Specifically, $L_{\text{tgt}} = 1, 1.3$, and 1.48 achieve a peak performance of $\mathcal{T} = 1, 1.3$, and 1.48 at all $L \geq L_{\text{tgt}}$. Thus, there is a 100% increase in the throughput compared to vanilla IRSA. The optimized threshold obtained using Alg. 5 is labelled as ν_{algo} and it achieves the envelope of all the curves with a fixed ν. There is a very small gap between the peak performance with $L_{\text{tgt}}=1.48$ and the throughput with the optimized threshold. Fig. 5.10 shows the equivalent empirical performance of C-IRSA with the same parameters as in Fig. 5.9. Here, choosing $\nu=h(\cdot,\cdot)$ achieves the same performance as $\nu=i(\cdot,\cdot,\cdot)$ with L_0 very close to L_{tgt}. The theoretical $\mathcal{T}$ from the previous plot coincides with the empirical $\mathcal{T}$ for $L_{\text{tgt}}=1$ and 1.3. The censor function for the theoretical curve can be obtained with $L_{\text{tgt}}=1.48$ and $\mathcal{T}=1.48$, whereas for the empirical curve we have $L_{\text{tgt}}=1.4$ and a slightly lower peak throughput of $\mathcal{T}=1.4$. This is because DE curves are achieved asymptotically. Thus, in practice, we back off from 1.48 to $L_{\text{tgt}}=1.4$ or 1.3. The solid curve with legend $\nu=\nu_{\text{algo}}$ uses the threshold that is optimized asymptotically, but is applied with $T=500$, and thus, it drops a little as L increases. Both $h(\cdot,\cdot)$ or $i(\cdot,\cdot,\cdot)$ achieve nearly the same performance as the optimized threshold and can be used in practice.

In Fig. 5.11, we show the theoretical performance with $N = 2$. $\nu = 0$ achieves the lowest peak $\mathcal{T}$, and both $\nu = \nu_{\text{algo}}$ and $i(L, L_{\text{tgt}} = 1.2, L_0 = 0.2)$ achieve nearly similar

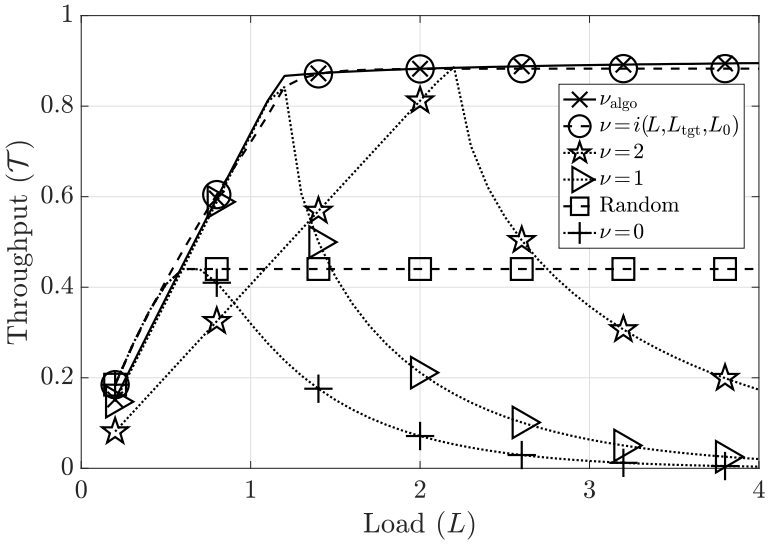


Figure 5.11: Theoretical throughput with $N = 2$.

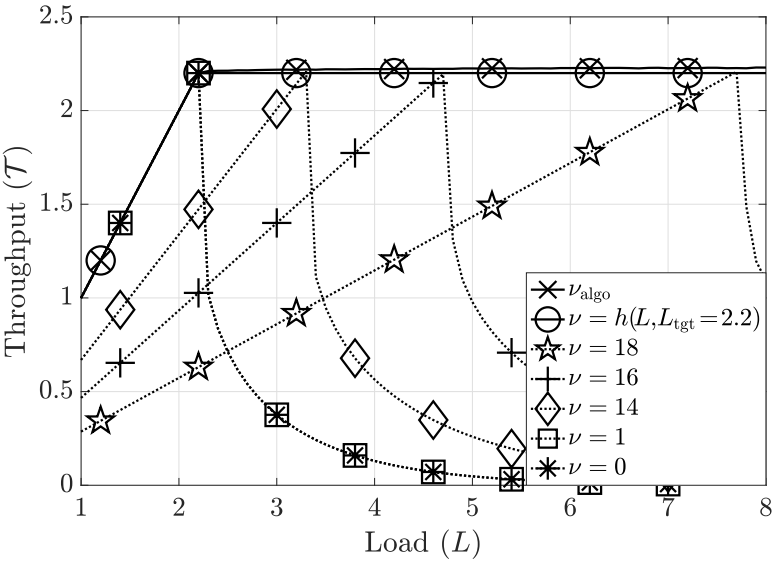


Figure 5.12: Theoretical throughput with $N = 16$.

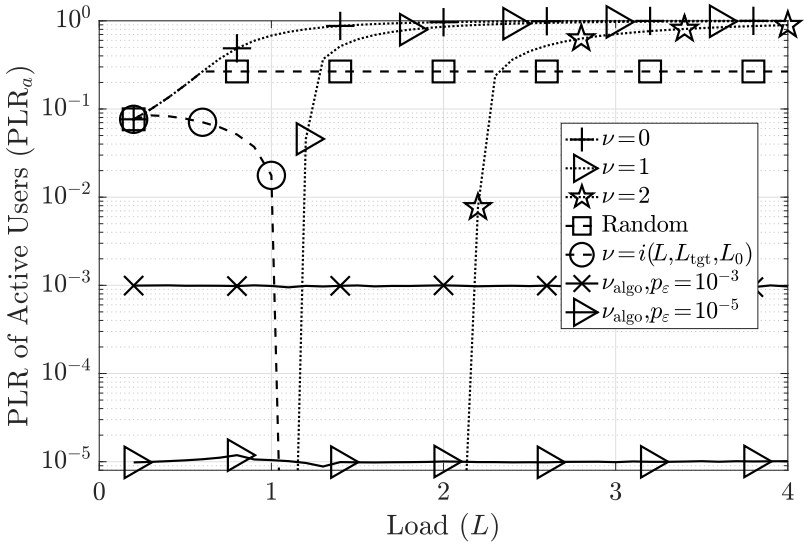


Figure 5.13: Theoretical active PLR with $N = 2$.

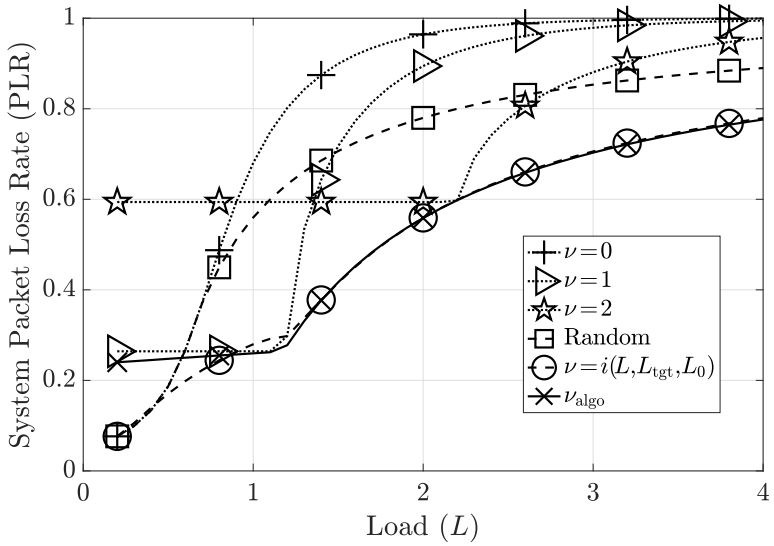


Figure 5.14: Theoretical PLR with $N = 2$.

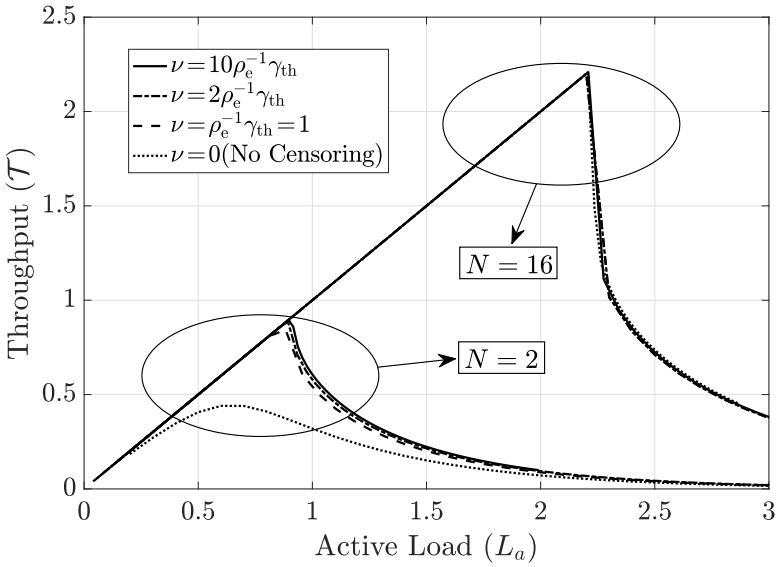


Figure 5.15: Effect of active load L_a on asymptotic $\mathcal{T}$.

performance. It is seen that a small change in ν from 1 to 2 results in a huge shift of the curves towards the right. Previously, in Fig. 5.9, with $N = 8$, increasing the threshold by a large amount from $\nu=0$ to $\nu=8$ shifts $\mathcal{T}$ to the right, after which increasing by a small amount to $\nu=10$ shifts $\mathcal{T}$ by a large amount. This is an effect of channel hardening [106], which is the phenomenon that as N increases the norm squared of the fade converges to a constant, i.e.,$\|\mathbf{v}_i\|^2 \to N\sigma_{\mathtt{h}}^2$. Since we perform fade-based censoring as $\|\mathbf{v}_i\|^2 \geq \nu$, the threshold has to be made close to $N\sigma_{\mathtt{h}}^2$ ($\pm$ a standard deviation) to ensure a significant impact on $\mathcal{T}$. Similar trends are observed for $N = 16$ as seen in Fig. 5.12.

Fig. 5.13 shows the theoretical PLR_a with $N = 2$, whereas the PLR curves are seen in Fig. 5.14. $\nu = 0$ achieves the highest PLR_a and $\nu = i(L, L_{\text{tgt}} = 1.2, L_0 = 0.2)$ achieves $\mathsf{PLR}_a = 0$ beyond $L = 1.2$. Alg. 5 yields $\mathsf{PLR}_a = p_\varepsilon$ at all L, and thus the tolerance parameter p_ε can be tuned by choosing a target active PLR. In practice, choosing $L_{\text{tgt}} = 0.9L^*$ or $0.8L^*$ achieves near-zero PLR_a at all L.

Impact of Random Censoring:

The censoring of users can be done in a random fashion as opposed to CSI-based censoring. That is, users independently participate in each frame with a probability p_a, and self-censor with probability $1 - p_a$. This yields an active load of $L_a = Lp_a$. For a load $L \geq L^*$, the optimal random censoring can be done by choosing $p_a = L^*/L$, which ensures $L_a = L^*$ for all L. Specifically, with optimal random censoring, we need to choose $p_a = \min\{1, 0.6/L\}$, which operates the system at $L_a^* = 0.6$ and achieves $\mathcal{T} = 0.45$ at all L. For random censoring, θ_r is obtained by substituting for $\nu = 0$ in (5.19) and in Theorem 5.2. Analysis for optimal random censoring is presented in Sec. 5.9.2.

The curve marked "Random" in Figs. 5.11 and 5.13 uses the above optimal random censoring and achieves the same throughput as peak vanilla IRSA for every $p_a \in (0, 1]$. For the same active load L_a, the channel states of the uncensored users with CSI-based censoring are better than the channel states of the active users with random censoring. The PLR_a of random censoring is very high whereas CSI-based censoring yields arbitrarily low PLR_a.

In Fig. 5.15, we study the effect of active load L_a on asymptotic $\mathcal{T}$. For $N = 2$, increasing ν from 0 to $\rho_{\mathrm{e}}^{-1}\gamma_{\mathrm{th}} = 1$ shows a drastic improvement and after that it shows a negligible improvement. For $N = 16$, however, increasing the threshold has a very negligible effect since the system is already well performing without censoring in the linear throughput regime. For such a system, performing random censoring as opposed to CSI based censoring could be beneficial.

5.7 Summary

In this work, we developed Censored-IRSA (C-IRSA), which overcame the interference limitation of IRSA at high loads. In C-IRSA, users self-censor based on their CSI and an adaptive threshold that is periodically broadcast by the BS. The protocol retains the fully distributed, random access nature of IRSA. First, the MMSE channel estimates and the SINR were derived in C-IRSA accounting for MIMO, fading, path loss, pilot

contamination, and estimation errors. The empirical performance of C-IRSA was then analyzed using the SINR threshold model. We analyzed the theoretical performance of C-IRSA using DE, and derived closed-form expressions for the success probability θ_r under the Normal and Gamma approximations. We developed two semi-analytic functions $i(L, L_{\text{tgt}}, L_0)$ and $h(L, L_{\text{tgt}})$, with which the system throughput was maximized and the active PLR was minimized, respectively, when $L < L_0$. With both these functions, the throughput was maximized when $L \geq L_{\text{tgt}}$. Next, we developed an algorithm to find the active PLR-optimal censor threshold. With this, the active PLR can be driven close to zero, whilst achieving the highest throughput. At high loads, C-IRSA operates at the full capacity compared to IRSA which has near-zero throughput. Finally, we discussed several insights into the design of C-IRSA: the choice of the target load L_{tgt}, the load L_0, and the optimal censor function.

An interesting extension we can explore for C-IRSA is based not just on fading, but also based on the time since a user attempted a transmission. This would involve censoring a user with good channel state to ensure that a user with bad channel state, who has also not transmitted for a long duration of time, can transmit its packet. This can be achieved, for example, using proportional fairness. With this, the censoring is performed not just based on the channel state of the user in the current frame, but also based on the average throughput a specific user has seen in the previous C-IRSA frames.

5.8 Proofs

5.8.1 Proof of Theorem 5.1: Success Probability for SISO

We now characterize θ_r, which is the probability of decoding a reference packet in a *single slot* where r users have transmitted their packets. Since there is only one slot under consideration, users are decoded via intra-slot SIC. These r packets are from r active users who are not yet decoded. The reference packet is one of the r packets, and it gets decoded only if the packets having a higher SINR get successfully decoded first. Hence, they must also satisfy the SINR $\geq \gamma_{\text{th}}$ constraint. Thus, θ_r is the probability

that the reference packet and the packets with higher SINRs all get decoded. Since θ_r is evaluated based on the SINR of multiple active users in a slot, we consider a slot wherein r active users have transmitted their packets, and use the active load L_a (and not L) to characterize the performance.

We denote the set of active users who have not yet been decoded in the first $k-1$ intra-slot decoding iterations by $\mathcal{S}_k$, and $\mathcal{S}_k^m \triangleq \mathcal{S}_k \backslash \{m\}$, with $\mathcal{S}_1 = [r]$. The SINR of the mth user in the kth intra-slot decoding iteration, ρ_m^k, is calculated as $\rho_m^k = |h_m|^2/(\rho_0^{-1} + \sum_{i \in \mathcal{S}_k^m} |h_i|^2)$. Let $\rho_{\max}^k$ denote the SINR of the user with the highest SINR in the kth intra-slot decoding iteration, calculated as $\rho_{\max}^k = \max_{m \in \mathcal{S}_k} \ \rho_m^k$. Let s be the index of the intra-slot decoding iteration in which the reference packet is decoded, with $1 \leq s \leq r$. Thus, θ_r is calculated as $\theta_r = \Pr(\rho_{\max}^1 \geq \gamma_{\text{th}}, \rho_{\max}^2 \geq \gamma_{\text{th}}, \ldots, \rho_{\max}^s \geq \gamma_{\text{th}})$. Since the reference packet is tagged uniformly at random from the users, the reference packet is equally likely to get decoded in any decoding iteration. We denote the probability that the k packets with the highest SINRs across decoding iterations all exceed the threshold γ_{th} by $\theta_{rk} \triangleq \Pr(\rho_{\max}^1 \geq \gamma_{\text{th}}, \rho_{\max}^2 \geq \gamma_{\text{th}}, \ldots, \rho_{\max}^k \geq \gamma_{\text{th}})$. We can calculate θ_r using θ_{rk} as $\theta_r = (\sum_{k=1}^r \theta_{rk})/r$. Without loss of generality, let the channels of the users be ordered as $|h_1|^2 \geq |h_2|^2 \geq \ldots \geq |h_r|^2$. Now,

$$\theta_{rk} = \Pr\Bigg(\frac{|h_1|^2}{\rho_0^{-1} + \sum_{i=2}^r |h_i|^2} \geq \gamma_{\text{th}}, \frac{|h_2|^2}{\rho_0^{-1} + \sum_{i=3}^r |h_i|^2} \geq \gamma_{\text{th}}, \\ \ldots, \frac{|h_k|^2}{\rho_0^{-1} + \sum_{i=k+1}^r |h_i|^2} \geq \gamma_{\text{th}} \Bigg| |h_j|^2 \geq \nu, \forall j \in [r] \Bigg). \tag{5.32}$$

The above is a conditional probability, conditioned on $|h_j|^2 \geq \nu$, since we are considering only uncensored users. Thus, θ_{rk} from (5.32) can be calculated equivalently as

$$\theta_{rk} = \Pr(t_1 \geq \gamma_{\text{th}}(\rho_0^{-1} + \textstyle\sum_{i=2}^r t_i), t_2 \geq \gamma_{\text{th}}(\rho_0^{-1} + \sum_{i=3}^r t_i), \ldots, t_k \geq \gamma_{\text{th}}(\rho_0^{-1} + \sum_{i=k+1}^r t_i)).$$

Here, t_i is a random variable follows a *truncated exponential* distribution with the density function $\mathrm{f}(t) = \exp(\nu - t) \cdot \mathbb{1}\{\nu \leq t < \infty\}$. Assuming $\nu \leq \rho_0^{-1}\gamma_{\text{th}}$, with $\bar{\gamma}_{\text{th},i} = (1 + \gamma_{\text{th}})^i$,

θ_{rk} can be calculated as

$$\begin{aligned}\theta_{rk} &= e^{r\nu}\int_{\nu}^{\infty} e^{-t_r}\mathrm{d}t_r \int_{\nu}^{\infty} e^{-t_{r-1}}\mathrm{d}t_{r-1}\cdots\int_{\nu}^{\infty} e^{-t_{k+1}}\mathrm{d}t_{k+1}\\ &\quad\times\int_{\gamma_{\mathrm{th}}(\rho_0^{-1}+\sum_{i=k+1}^{r}t_i)}^{\infty} e^{-t_k}\mathrm{d}t_k\cdots\int_{\gamma_{\mathrm{th}}(\rho_0^{-1}+\sum_{i=2}^{r}t_i)}^{\infty} e^{-t_1}\mathrm{d}t_1\\ &= \frac{\exp(r\nu-(r-k)\nu\bar{\gamma}_{\mathrm{th},k}-\rho_0^{-1}\gamma_{\mathrm{th}}(\sum_{i=1}^{k}\bar{\gamma}_{\mathrm{th},i-1}))}{\bar{\gamma}_{\mathrm{th},k}^{r-(k+1)/2}}.\end{aligned} \tag{5.33}$$

Thus, we get

$$\theta_r = \sum_{k=1}^{r}\frac{\exp(r\nu-(r-k)\nu\bar{\gamma}_{\mathrm{th},k}-\rho_0^{-1}(\bar{\gamma}_{\mathrm{th},k}-1))}{r\ \bar{\gamma}_{\mathrm{th},k}^{r-(k+1)/2}}. \tag{5.34}$$

5.8.2 Proof of Theorem 5.2: Success Probability for MIMO

We now characterize θ_r, which is the probability of decoding a reference packet in a *single slot* where r users have transmitted their packets. Since there is only one slot under consideration, users are decoded via intra-slot SIC. These r packets are from r active users who are not yet decoded. The reference packet is one of the r packets, and it gets decoded only if the packets having a higher SINR get successfully decoded first. Hence, they must also satisfy the SINR $\geq \gamma_{\mathrm{th}}$ constraint. Thus, θ_r is the probability that the reference packet and the packets with higher SINRs all get decoded. Since θ_r is evaluated based on the SINR of multiple active users in a slot, we consider a slot wherein r active users have transmitted their packets, and use the active load L_a (and not L) to characterize the performance.

We now setup some notation for precisely characterizing θ_r. Let k denote the intra-slot decoding iteration. We denote the set of active users who have not yet been decoded in the first $k-1$ intra-slot decoding iterations by $\mathcal{S}_k$, and $\mathcal{S}_k^m \triangleq \mathcal{S}_k \setminus \{m\}$, with $\mathcal{S}_1 = [r]$. With PLI, the channel of the user is the same as the fading state, i.e., $\mathbf{h}_i = \mathbf{v}_i$, and hence we use both equivalently. When perfect CSI is available at the BS and the users perform PLI, the SINR of the mth user in the kth intra-slot decoding iteration, ρ_m^k, is calculated as

$$\rho_m^k = \frac{P\|\mathbf{h}_m\|^4}{N_0\|\mathbf{h}_m\|^2 + \sum_{i\in\mathcal{S}_k^m} P|\mathbf{h}_m^H\mathbf{h}_i|^2}. \tag{5.35}$$

Let $\rho^k_{\max}$ denote the SINR of the user with the highest SINR in the kth intra-slot decoding iteration, calculated as $\rho^k_{\max} = \max_{m \in \mathcal{S}_k} \rho^k_m$. Let s be the index of the intra-slot decoding iteration in which the reference packet is decoded, with $1 \leq s \leq r$. Thus, with a threshold ν, θ_r is calculated as

$$\theta_r = \Pr(\rho^1_{\max} \geq \gamma_{\text{th}}, \rho^2_{\max} \geq \gamma_{\text{th}}, \ldots, \rho^s_{\max} \geq \gamma_{\text{th}} \mid \|\mathbf{h}_j\|^2 \geq \nu, \forall j \in [r]). \tag{5.36}$$

Since the reference packet is tagged uniformly at random from the users, the reference packet is equally likely to get decoded in any decoding iteration. The above is a conditional probability, conditioned on $\|\mathbf{h}_j\|^2 \geq \nu$, since we are considering only users who have not censored.

For $r = 1$, $\rho^1_1 = P\|\mathbf{h}_1\|^2/N_0$, and θ_1 reduces to $\Pr(\rho^1_1 \geq \gamma_{\text{th}} \mid \|\mathbf{h}_1\|^2 \geq \nu)$. This can be computed in closed form as

$$\theta_1 = \begin{cases} \Gamma_{\text{inc}}(N, \rho_0^{-1}\gamma_{\text{th}})/\Gamma_{\text{inc}}(N, \nu), & \nu \leq \rho_0^{-1}\gamma_{\text{th}}, \\ 1, & \nu > \rho_0^{-1}\gamma_{\text{th}}. \end{cases} \tag{5.37}$$

where $\rho_0 \triangleq P\sigma^2_{\mathtt{h}}/N_0$, and $\Gamma_{\text{inc}}(s, x) = \int_x^\infty t^{s-1}\, e^{-t}\, \mathrm{d}t$ is the upper incomplete gamma function.

For $r \geq 2$, we resort to computing the deterministic equivalents [106] of the norms of the channels as well as the interference components to compute approximations to the SINR. We first find the deterministic equivalents of the norms. Let $\|\mathbf{h}_i\|^2|_{\|\mathbf{h}_i\|^2 \geq \nu}$ denote the random variable $\|\mathbf{h}_i\|^2$ conditioned on the event $\{\|\mathbf{h}_i\|^2 \geq \nu\}$. We use $\xrightarrow{\text{a.s.}}$ to denote convergence in the almost sure sense. Since $\|\mathbf{h}_i\|^2 \xrightarrow{\text{a.s.}} N\sigma^2_{\mathtt{h}}$ [106], we can find that $\|\mathbf{h}_i\|^2|_{\|\mathbf{h}_i\|^2 \geq \nu} \xrightarrow{\text{a.s.}} \bar{H}(\nu)\sigma^2_{\mathtt{h}}$, where

$$\bar{H}(\nu) = \Gamma_{\text{inc}}(N+1, \nu)/\Gamma_{\text{inc}}(N, \nu) = N + \nu^N/S_N(\nu), \tag{5.38}$$

with $S_N(\nu) \triangleq (N-1)!\sum_{k=0}^{N-1}(\nu^k/k!)$ and $S_N(0) \triangleq 1$.

We now find the deterministic equivalents of the interference components. Let us

write the channel of the ith user $\mathbf{h}_i$ as the product of the norm $n_i = \|\mathbf{h}_i\|$ and the direction $\mathbf{d}_i = \mathbf{h}_i/\|\mathbf{h}_i\|$, i.e., $\mathbf{h}_i = n_i\mathbf{d}_i$. Since the norm and the direction of a circularly symmetric complex Normal random vector are independent [105], n_i and $\mathbf{d}_i$ are independent. Thus, when we employ a censor threshold ν, only the distribution of the norms of the channels changes but the distribution of the directions of the channels remains the same. That is, $\|\mathbf{h}_i\|^2$ is Gamma distributed whereas $\|\mathbf{h}_i\|^2|_{\|\mathbf{h}_i\|^2\geq\nu}$ is *truncated Gamma* distributed. Further, the distribution of $\mathbf{d}_i$ and $\mathbf{d}_i|_{\|\mathbf{h}_i\|^2\geq\nu}$ remain the same. Thus, the interference $t_{mi} \triangleq |\mathbf{h}_m^H\mathbf{h}_i|^2/(\|\mathbf{h}_m\|^2\|\mathbf{h}_i\|^2) = |\mathbf{d}_m^H\mathbf{d}_i|^2$ is Beta distributed as $t_{mi} \sim \text{Beta}(\alpha = 1, \beta = N)$, both with and without censoring.

For $r \geq 2$, the SINR with MRC and large N, can be approximated as

$$\rho_m^k \approx ((\bar{H}(\nu)\rho_0)^{-1} + \textstyle\sum_{i\in\mathcal{S}_k^m} t_{mi})^{-1}. \tag{5.39}$$

Here, we have applied the theory of deterministic equivalents to only the channel norms and not to the interference, and we have an SINR that is affected only by the randomness in the multi-user interference components. This is supported by the fact that the interference components converge to their deterministic equivalents slower than the norms converge to their deterministic equivalents [106].

For $r = 2$, since $t_{12} = t_{21}$, $\rho_1^1 = \rho_2^1 = ((\bar{H}(\nu)\rho_0)^{-1} + t_{12})^{-1}$. Thus, $\rho_{\max}^1 = ((\bar{H}(\nu)\rho_0)^{-1} + t_{12})^{-1}$ and $\rho_{\max}^2 = \bar{H}(\nu)\rho_0$, with $\rho_{\max}^1 \leq \rho_{\max}^2$. Thus, the success probability reduces to $\theta_r = \Pr(\rho_{\max}^1 \geq \gamma_{\text{th}})$. Let $t_0(\nu) \triangleq \gamma_{\text{th}}^{-1} - (\bar{H}(\nu)\rho_0)^{-1}$. Hence, θ_2 is calculated as

$$\begin{aligned}\theta_2 &\approx \Pr(\rho_{\max}^1 \geq \gamma_{\text{th}}) = \Pr(t_{12} \leq t_0(\nu)) \\ &= \mathbb{1}\{t_0(\nu) > 1\} + (1 - (1 - t_0(\nu))^N)\mathbb{1}\{0 \leq t_0(\nu) \leq 1\}.\end{aligned} \tag{5.40}$$

For $r \geq 3$, ρ_m^k need not be a monotonically increasing function of k as seen in (5.39), and thus we cannot order the SINRs meaningfully to compute θ_r in closed-form. With $u_m = \sum_{i\in[r]\setminus m} t_{mi}$, the maximum SINR in the first intra-slot iteration is $\rho_{\max}^1 = \max_{m\in[r]}((\bar{H}(\nu)\rho_0)^{-1} + u_m)^{-1}$. Here, u_m is not independent across m and it is not clear which u_m is the minimum. Thus, we approximate $\rho_{\max}^1$ as ρ_1^1, and upon dropping the other SINR terms from (5.36),

θ_r becomes

$$\theta_r \approx \Pr(\rho_1^1 \geq \gamma_{\text{th}}) = \Pr(u_1 \leq t_0(\nu)). \tag{5.41}$$

We now discuss two approximations to u_m to evaluate θ_r, with the assumption that u_m is independent across m.

Since $\lim_{N\to\infty} \text{Beta}(\alpha = 1, \beta = N) = \exp(\lambda = N)$, we approximate $t_{mi} \sim \exp(N)$, which is a good approximation at high N [105]. Even with this approximation, u_m is identically Gamma distributed across users but not independent. With the independence assumption, u_m is i.i.d. Gamma distributed with shape $r-1$ and rate N, i.e., $u_m \overset{\text{i.i.d.}}{\sim} \text{Gamma}(r-1, N)$. Thus, we obtain the *Gamma* approximation:

$$\theta_r \approx 1 - \Gamma_{\text{inc}}(r-1, Nt_0(\nu))/\Gamma(r-1). \tag{5.42}$$

Similarly, when we assume t_{mi} is Normal distributed, u_m is identically Normal distributed across users but not independent. Let $\mu_N = (N+1)^{-1}$ and $\sigma_N^2 = N(N+1)^{-2}(N+2)^{-1}$. If we approximate $t_{mi} \sim \mathcal{N}(\mu_N, \sigma_N^2)$ and u_m is independent across m, then $u_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}((r-1)\mu_N, (r-1)\sigma_N^2)$. Thus, we obtain the *Normal* approximation:

$$\theta_r \approx 1 - \mathcal{Q}\left(\frac{t_0(\nu) - (r-1)\mu_N}{\sqrt{r-1}\sigma_N}\right), \tag{5.43}$$

where $\mathcal{Q}(\cdot)$ is the standard Normal Q-function.

5.8.3 Proof of Theorem 5.4: Generalized Censor Function for MIMO

In order to derive a generalized censor function, we analyze the system behaviour below and above the target load L_{tgt}.

5.8.3.1 $L \geq L_{\text{tgt}}$

The system is MUI-limited when $L \geq L_{\text{tgt}}$, and censoring improves the performance. This enhancement happens via a combination of improved channel states among the active users and lesser MUI. In this regime, similar to the SISO threshold, we can compute ν by inverting the CCDF in (5.24). For this, we define the function $\mathrm{H}(\cdot,\cdot,\cdot)$ as

$$\mathrm{H}(L, L_{\text{tgt}}, \nu_{\text{tgt}}) \triangleq \bar{\mathrm{G}}(L_a/L) = \bar{\mathrm{G}}(\bar{\mathrm{F}}(\nu_{\text{tgt}})L_{\text{tgt}}/L). \tag{5.44}$$

Here $x = \bar{\mathrm{G}}(y)$ is the inverse CCDF of the Gamma distribution (with rate 1 and shape N), which is the x such that $y = \bar{\mathrm{F}}(x)$.

5.8.3.2 $L < L_{\text{tgt}}$

Censoring need not always help improve the performance when $L < L_{\text{tgt}}$. We handle this by introducing a load parameter L_0, which denotes the load up to which users do not self-censor, with $L_0 < L_{\text{tgt}}$. There are two cases when censoring is not needed. The first case happens when $\mathsf{PLR} = 0$ at low L, for which no censoring is needed. This is because the throughput is at the optimal value of $\mathcal{T} = L$ already. One instance when this can happen is when there are multiple antennas at the BS and the BS can decode all the users via a combination of SIC and interference-suppression due to multiple antennas. When $\mathsf{PLR} = 0$ at low loads, L_0 can be found as the system inflection load L^* with $\nu = 0$.

In the second case, the system can be power-limited when the transmit powers of users are not high enough, and thus, the MUI component is low to begin with at low L. While there are fewer active users upon censoring, the MUI does not reduce. In this case, $\mathsf{PLR} \neq 0$ at low L, censoring does not increase $\mathcal{T}$, and thus, we do not need to censor users at low L. When $\mathsf{PLR} > 0$ at low loads, we can still choose L_0 as the inflection load corresponding to $\nu = 0$, but this can be fine tuned if needed. In both the cases mentioned above, users need not self-censor for $L < L_0$.

If we choose the censor threshold similar to SISO, the choice of the threshold with

MIMO is given by the function $h(\cdot,\cdot)$ defined as

$$h(L, L_{\text{tgt}}) \triangleq \begin{cases} \nu_{\text{tgt}}, & L < L_{\text{tgt}}, \\ \mathrm{H}(L, L_{\text{tgt}}, \nu_{\text{tgt}}), & L \geq L_{\text{tgt}}. \end{cases} \tag{5.45}$$

Thus, we need to modify (5.45) so that $\nu = 0$ for $L < L_0$ in these two cases.

So far, we have addressed how to compute ν for $L \geq L_{\text{tgt}}$ and $L < L_0$. We now discuss how to compute ν for $L_0 \leq L < L_{\text{tgt}}$. In the SISO case, L_a^* is maximized at $\nu_{\text{tgt}} = \rho_0^{-1}\gamma_{\text{th}}$. However, for MIMO, as the censor threshold is increased, the subset of active users experience a higher degree of capture effect since there are fewer users. This behaviour is prominent especially at loads very close to the inflection loads L^* and L_a^*, and this variation in the censor threshold occurs only in a small region between L_0 and L_{tgt}.

Choosing a fixed $\nu = \nu_{\text{tgt}}$ for $L_0 \leq L < L_{\text{tgt}}$ yields a censor threshold that is a discontinuous function of L at $L = L_0$ and $L = L_{\text{tgt}}$. This is not ideal, since in practice, the throughput/PLR linearly change with L and they do not exhibit such discontinuous behaviour. So, to make it a continuous function, in this small region, we choose ν based on a logarithmic interpolation function as

$$(\log(L/L_0)/\log(L_{\text{tgt}}/L_0))\nu_{\text{tgt}}. \tag{5.46}$$

We have investigated many functions in this regime (See Sec. 5.9.6), and observed that the log-interpolation yields the best throughput. The log-interpolation with a properly tuned L_0 outperforms linear-interpolation as well as other functions.

5.9 Appendix

5.9.1 Preliminary Results

This section presents the preliminary results that were published in the conference version [18] of this chapter, specifically for C-IRSA with SISO and perfect CSI.The results are

presented for 10^4 Monte Carlo runs, SNR $\rho_0 = 10$ dB, SINR threshold $\gamma_{\text{th}} = 10$ [11]. We use the truncated Soliton distribution [72] $\phi(x) = 0.625x^2 + 0.25x^3 + 0.125x^4$ to generate the repetition factors of the users [11]. The repetition factor d_i is used to form the access vector for the ith user, by uniformly randomly choosing d_i slots from T slots without replacement [7]. The packet replicas are transmitted in these d_i slots.

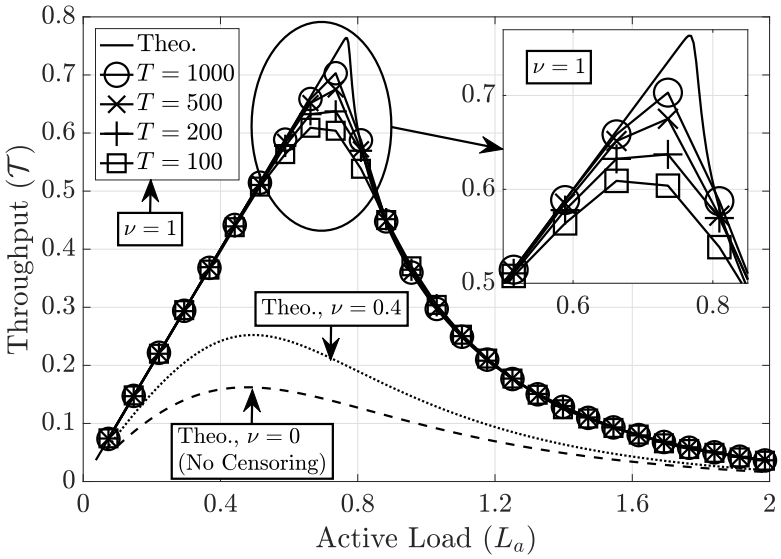


Figure 5.16: Impact of T on the throughput.

Fig. 5.16 shows the impact of T on the empirical throughput with $\nu = \rho_0^{-1}\gamma_{\text{th}} = 1$. The theoretical asymptotic throughput curves for $\nu = 0, 0.4$, and 1, obtained via DE, are also shown. The curves linearly increase till a peak, after which they drop quickly to zero as the system becomes MUI-limited. The asymptotic $\mathcal{T}$ is maximized at $L_a^* = \mathcal{T} = 0.76$, for $\nu = 1$. The linear increase in $\mathcal{T}$ marks the region in which $\mathsf{PLR}_a = 0$ [7]: when $L_a \leq 0.76$, all active users are decoded. Conventional IRSA corresponds to no censoring ($\nu = 0$). At $L_a = 0.4$, $\nu = 0$ achieves $\mathcal{T} = 0.15$, whereas $\nu = 1$ achieves full throughput of $\mathcal{T} = L_a = 0.4$. The asymptotic throughput dramatically improves as ν is increased from 0 to 1, because users with poor channel states are self-censored. Even with a little amount of censoring, C-IRSA performs better than IRSA. Thus, C-IRSA helps overcome packet losses due to both poor CSI and MUI.

We have seen that the choice of the threshold ν must be such that $L_a \leq L_a^*$. In

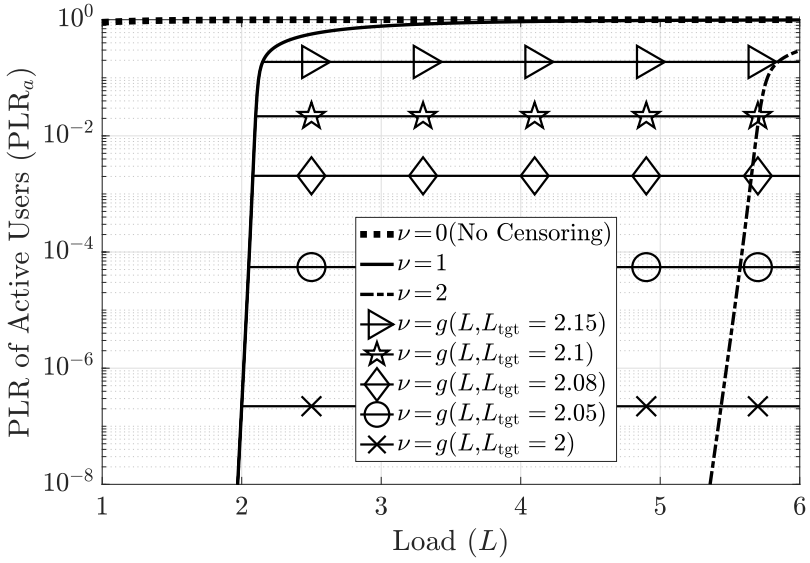


Figure 5.17: Choice of target load L_{tgt} using theoretical PLR_a.

Fig. 5.17, we depict the influence of the choice of the target load, L_{tgt}, using the asymptotic active PLR, PLR_a. The PLR is close to 1 with no censoring. The PLR_a of the system increases with L for $\nu = 0, 1$, and 2, and becomes 1 at high loads. The curves with $\nu = g(L, L_{\text{tgt}})$ follow the performance of $\nu = \rho_0^{-1}\gamma_{\text{th}} = 1$ up to a load of $L = L_{\text{tgt}}$, and beyond that PLR_a stays constant at every load. Fixing a $\mathsf{PLR}_{a,\text{tgt}}$ yields the choice of L_{tgt} and the corresponding threshold $\nu = g(L, L_{\text{tgt}})$. The asymptotic PLR increases very quickly around the inflection load L^*. In practice, however, choosing $L_{\text{tgt}} = 0.9L^*$ or $0.8L^*$ works well.

In Fig. 5.18, we show the effect of the active load L_a on the empirical throughput $\mathcal{T}$, with $T = 250$. Conventional IRSA (no censoring, i.e., $\nu = 0$) achieves very low throughputs since the system is highly interference limited. Similar to the previous plot, where the theoretical throughput increased with increase in ν, the empirical throughput also increases with an increase in from $\nu = 0$ to $\nu = \gamma_{\text{th}}/\rho_0 = 1$. For $\nu \geq \gamma_{\text{th}}/\rho_0$, the throughput of the system stays constant with respect to the active load and the system achieves the same throughput for $\nu = 2$ as for $\nu = 1$. From the plot, we also see that we should choose a threshold ν such that we always operate the system at active load of $L_a \leq L_a^* = 0.65$. Also, by optimally choosing the threshold using $\nu = g(L, L_{\text{tgt}})$ as described in Sec. 5.5, we can

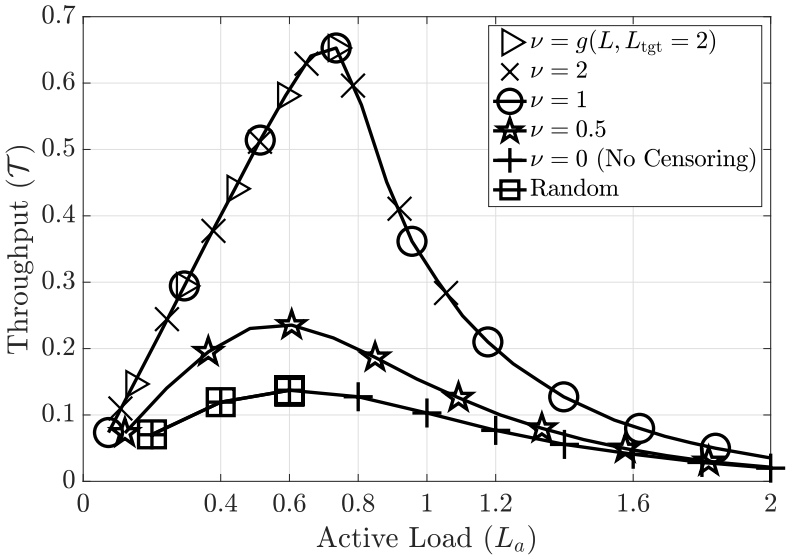


Figure 5.18: Effect of active load L_a on $\mathcal{T}$.

obtain the same throughput as that obtained with $\nu = 1$. Note that in the MUI-limited regime, the PLR of IRSA is nonzero, and both users with poor channel states as well as users who collide with many users cannot be decoded correctly. Censoring improves the performance of the system on both counts by choosing users whose packets are more likely to be decoded correctly as well as reducing the number of collisions.

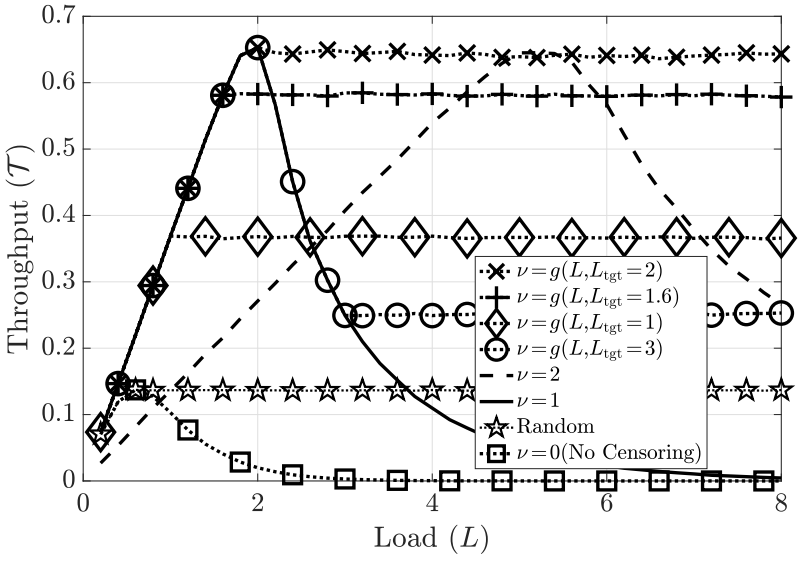


Figure 5.19: Impact of threshold ν on $\mathcal{T}$.

So far, we have observed that both the theoretical and empirical throughputs are maximized at $\nu = \rho_0^{-1}\gamma_{\text{th}}$ for every L_a. We now study the effect of censoring and the system load L on the empirical throughput in Fig. 5.19, with $T = 250$. With $\nu = 0$, i.e., no censoring, the throughput of IRSA becomes zero at $L = 3$. With $\nu = \rho_0^{-1}\gamma_{\text{th}} = 1$, the throughput of the system increases linearly with load up to $\mathcal{T} = 0.65$ at $L = 2$, and beyond that, the throughput drops to zero. This is also observed with $\nu = 2$, which achieves a peak throughput of $\mathcal{T} = 0.65$ at $L = 5$. The linearity of the curve up to $L = 5$ indicates that too many users are self-censoring, and we could reduce ν.

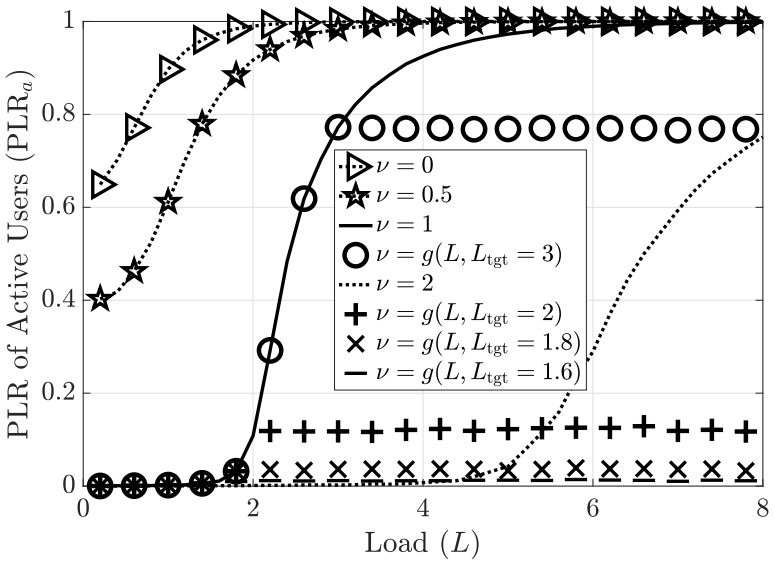


Figure 5.20: Impact of threshold ν on PLR_a.

We present the PLR curves corresponding to the above throughput curves in Figs. 5.20 and 5.21. For $\nu = 1$, we have $\mathsf{PLR}_a = 0$ and $\mathsf{PLR} = \bar{\mathrm{F}}(1)$ up to $L = 2$; for $\nu = 2$, we have $\mathsf{PLR}_a = 0$ and $\mathsf{PLR} = \bar{\mathrm{F}}(2)$ up to $L = 5$. Thus, we could choose ν for every L such that we obtain an envelope of all curves for $\nu \geq 1$, which yields the same performance as that of the curve marked $\nu = g(L, L_{\text{tgt}} = 2)$.[13] All the curves marked $\nu = g(L, L_{\text{tgt}})$ follow the performance of $\nu = 1$ up to L_{tgt}, beyond which $\mathcal{T}$ stays constant for every L. Since

[13]The theoretical throughputs for Figs. 5.18 and 5.19 match the above observations. Also, the results are presented for $\rho_0^{-1}\gamma_{\text{th}} = 1$. The trends are similar for any other $\rho_0^{-1}\gamma_{\text{th}}$, and $\mathcal{T}$ is maximized at $\nu = \rho_0^{-1}\gamma_{\text{th}}$ for every L_a.

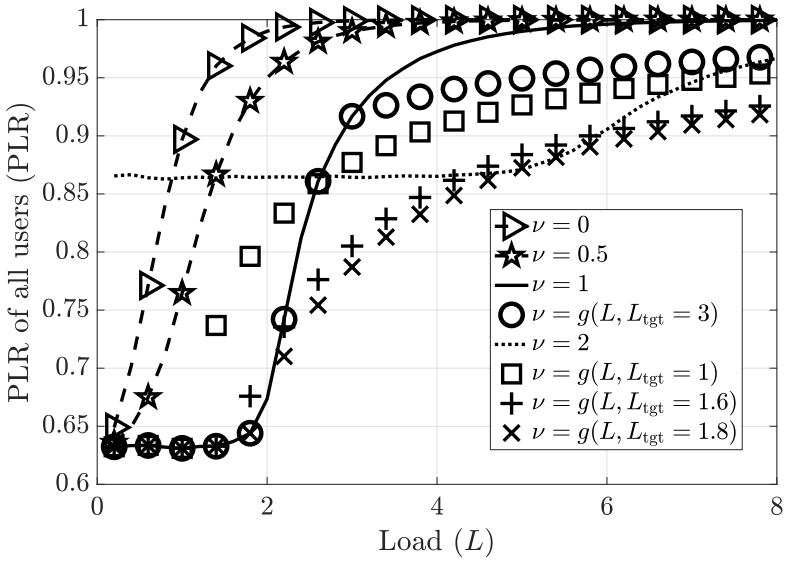


Figure 5.21: Effect of threshold ν on PLR.

$L^* = 2$, choosing $L_{\text{tgt}} = 3$ is not preferred since the system is operating at a high PLR. Choosing $L_{\text{tgt}} = 1, 1.6$, and 2 all yield $\mathsf{PLR}_a = 0$ at all L since the active load $L_a \leq 0.65$. We thus choose $L_{\text{tgt}} = 2$ to maximize $\mathcal{T}$, which can be obtained from our analysis as $L_{\text{tgt}} = L_a^*/\bar{\text{F}}(\rho_0^{-1}\gamma_{\text{th}}) = 0.65/\bar{\text{F}}(1) = 2$. Since the DE curves are achieved asymptotically, in practice, we back off from L_{tgt} by 10% to 20% to $L_{\text{tgt}} = 1.8$ or 1.6, to achieve zero PLR_a at all L. At high L, we see that C-IRSA with $L_{\text{tgt}} \leq L^*$ operates with $\mathcal{T} = 0.65$, whereas conventional IRSA has $\mathcal{T} = 0$. Thus, the system can be operated at its maximum potential in C-IRSA compared to vanilla IRSA which has zero throughput.

5.9.2 Impact of Random Censoring

The censoring of users can be done in a random fashion as opposed to CSI-based censoring. That is, users independently participate in each frame with a probability p_a, and self-censor with probability $1 - p_a$. This yields an active load of $L_a = Lp_a$. For a load $L \geq L^*$, the optimal random censoring can be done by choosing $p_a = L^*/L$. This ensures that the active load is $L_a = Lp_a = L^*$. Thus, the active load stays constant at L^* for all L. This choice of the activity probability is applicable to any underlying PHY layer system model (e.g., SISO with PLI, SISO without PLI, MIMO with PLI, MIMO without PLI and pilot

contamination, etc.,) since the optimal choice depends only on the inflection load. The inflection load itself varies based on the underlying system model parameters, but the choice of the optimal activity probability is dependent only on the inflection load L^*.

We now describe the computation of the success probability θ_r under random censoring for both SISO and MIMO.

Lemma 5. *For the SISO Rayleigh block-fading channel with an SNR of ρ_0, and an SINR threshold γ_{th}, the probability that a reference packet gets decoded in a slot of degree r using only intra-slot SIC, with random censoring, can be obtained as*

$$\theta_r = \sum_{k=1}^{r} \frac{\exp(-\rho_0^{-1}((1+\gamma_{\text{th}})^k - 1))}{r\ (1+\gamma_{\text{th}})^{rk-k(k+1)/2}}. \tag{5.47}$$

Proof. In random censoring, there is no CSI threshold. Substituting for $\nu = 0$ in (5.19), we obtain the above. □

Lemma 6. *When perfect CSI is available at the BS, and MRC is used for decoding, with random censoring, θ_r can be calculated as follows. Firstly, θ_1 can be exactly found as*

$$\theta_1 = \Gamma_{\text{inc}}(N, \rho_0^{-1}\gamma_{\text{th}})/\Gamma(N), \tag{5.48}$$

where $\rho_0 \triangleq P\sigma_{\mathtt{h}}^2/N_0$, $\Gamma_{\text{inc}}(s,x) = \int_x^\infty t^{s-1}\, e^{-t}\, \mathrm{d}t$ is the upper incomplete gamma function, and $\Gamma(s)$ is the ordinary gamma function. For $r \geq 2$, the SINR with MRC and large N can be computed as $\rho_m^k = N(\rho_0^{-1} + N\sum_{i\in\mathcal{S}_k^m} t_{mi})^{-1}$, where $t_{mi} \triangleq |\mathbf{h}_m^H \mathbf{h}_i|^2/(\|\mathbf{h}_m\|^2\|\mathbf{h}_i\|^2)$. With $t_0 \triangleq \gamma_{\text{th}}^{-1} - N^{-1}\rho_0^{-1}$, θ_2 can be calculated as

$$\theta_2 = \mathbb{1}\{t_0 \geq 1\} + (1-(1-t_0)^N)\mathbb{1}\{0 \leq t_0 \leq 1\}. \tag{5.49}$$

Three approximations to θ_r for $r \geq 3$ and large N are described below. Approximating $\rho_{\max}^1$ as ρ_1^1, and assuming u_m as i.i.d. Gamma distributed with shape $r-1$ and

rate N, we obtain the Gamma *approximation:*

$$\text{Gamma:} \quad \theta_r = 1 - \Gamma_{\text{inc}}(r-1, Nt_0)/\Gamma(r-1). \tag{5.50}$$

Approximating $\rho^1_{\max} = \rho^1_1$ and $u_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}((r-1)\mu_N, (r-1)\sigma^2_N)$, where $\mu_N \triangleq (N+1)^{-1}$, and $\sigma^2_N \triangleq N(N+1)^{-2}(N+2)^{-1}$, we obtain the Normal *approximation:*

$$\text{Normal:} \quad \theta_r = 1 - \mathcal{Q}\left(\frac{t_0 - (r-1)\mu_N}{\sqrt{r-1}\sigma_N}\right), \tag{5.51}$$

where $\mathcal{Q}(\cdot)$ is the standard Normal Q-function.

Proof. In random censoring, there is no CSI threshold. Substituting for $\nu = 0$ in Theorem 5.2, we obtain the above. □

Let us use an arbitrary censoring scheme in which p_a fraction of the users are active. The active load L_a of the system is then $L_a = Lp_a$. The effective PLR of the system (including censored users) can be calculated as

$$\mathsf{PLR} = (1 - p_a) + p_a \mathsf{PLR}_a. \tag{5.52}$$

The throughput $\mathcal{T}$ of the users in the system can now be obtained from the asymptotic PLR as

$$\mathcal{T} = L(1 - \mathsf{PLR}) = L_a(1 - \mathsf{PLR}_a). \tag{5.53}$$

The iterations $p_i = f(\lambda(p_{i-1}))$ converge to $p_\infty = 0$ if the active load $L_a < L^*_a$, asymptotically [7,11]. Here, L^*_a is called the *active inflection load* of the system, and it corresponds to a *system inflection load* of $L^* = L^*_a/p_a$. For $L_a < L^*_a$, since $p_\infty = 0$, we have $\mathsf{PLR}_a = 0$, $\mathsf{PLR} = (1 - p_a)$, and $\mathcal{T} = Lp_a = L_a$. For any $L_a \geq L^*_a$, PLR_a does not converge to 0, and $\mathcal{T}$ decreases monotonically with L_a. Also, from (5.52), we see that $\mathsf{PLR} \geq (1 - p_a)$, and thus, $\mathcal{T} \leq Lp_a$.

The curve marked "Random" in Fig. 5.18 uses random censoring and achieves the same throughput as vanilla IRSA for every $p_a \in (0, 1]$. For the same active load L_a, the channel states of the uncensored users with CSI-based censoring are better than the channel states of the active users with random censoring. With optimal random censoring, in order to operate the system at $\mathcal{T}_{\text{tgt}} = 0.15$ at $L_a^* = 0.6$, we need to choose $p_a = \min\{1, 0.6/L\}$. With this choice of p_a, we obtain the curve marked "Random" in Fig. 5.19, which achieves $\mathcal{T} = 0.15$ at all L. Thus, the optimal CSI-based censoring in C-IRSA achieves a peak throughput of $\mathcal{T} = 0.65$ whereas optimal random censoring in IRSA has a peak throughput of $\mathcal{T} = 0.15$, which is a $4\times$ improvement.

Similar trends are observed for the MIMO case, as seen with the curves marked "Random" in Figs. 5.11 and 5.13.

5.9.3 Channel Estimation at the Users

We formalize the process of channel estimation at the users and the consequent self-censoring. Prior to the start of each frame, let the BS transmit a set of N orthogonal pilots $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N\}$ as N consecutive symbols across the N antennas. With $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N]$, the received downlink signal $\mathbf{y}_i^{\mathtt{D}} \in \mathbb{C}^{1\times N}$ at the ith user across the N symbols is

$$\mathbf{y}_i^{\mathtt{D}} = \mathbf{h}_i^H \boldsymbol{\Phi} + \mathbf{n}_i^{\mathtt{D}}, \tag{5.54}$$

where $\mathbf{n}_i^{\mathtt{D}} \in \mathbb{C}^{1\times N}$ is the AWGN at the ith user. The ith user can obtain the minimum mean squared error (MMSE) estimate of its fading vector $\mathbf{v}_i \in \mathbb{C}^N$ as

$$\hat{\mathbf{v}}_i = \sqrt{\beta_i}\sigma_{\mathtt{h}}^2(\beta_i\sigma_{\mathtt{h}}^2\boldsymbol{\Phi}^H\boldsymbol{\Phi} + N_0\mathbf{I}_N)^{-1}\boldsymbol{\Phi}^H\mathbf{y}_i^{\mathtt{D}H}. \tag{5.55}$$

The ith user participates in the frame if and only if the norm squared of its fading estimate $\hat{\mathbf{v}}_i$ exceeds the threshold ν. Thus, we have that $a_i = \mathbb{1}\{\|\hat{\mathbf{v}}_i\|^2 \geq \nu\}, \forall i \in [M]$.

5.9.4 Analysis of C-IRSA Accounting for UAD Errors

In mMTC systems, users transmit data as and when they have data to transmit, and they largely remain inactive. Due to this sporadic activity of the users, the BS needs to figure out which users are active at any instant of time. In typical IRSA systems, the users' activities are contributed only from the sporadic activity property in mMTC. In C-IRSA, the users are active only when two conditions are satisfied: users have chosen to transmit data in the current frame and they have good channel states in the current frame. The former arises due to the sporadic activity of the users in mMTC, whereas in the latter, the "goodness" of the channel states is measured by the property that they have a fading channel $\mathbf{v}$ such that $\|\mathbf{v}\|^2 \geq \nu$.

In the C-IRSA protocol, the users' packets are decoded at the BS using the SIC process as with the conventional IRSA protocol. This entails a UAD phase, followed by a channel estimation phase, and finally a data decoding phase. The BS first carries out an activity detection phase based on which it knows the subset of users that have not self-censored. The BS then processes the received pilot and data signals iteratively. In every slot, the BS attempts to decode the users' packets. If a user is successfully decoded, which can be verified via a cyclic redundancy check, then the BS performs SIC in all slots in which that user has transmitted a packet [7]. This process repeats and the decoding proceeds in iterations until no new packets can be decoded.

5.9.4.1 UAD Phase

In the UAD phase, the BS first performs UAD (possibly, using the UAD algorithm proposed in Chapter 4) to detect the subset of active users in each slot. We now setup some notation for analyzing the performance of C-IRSA with UAD. Let the total number of users in the system be M_{tot} and the corresponding load be the *total load* $L_{\text{tot}} \triangleq M_{\text{tot}}/T$. Let the number of users in the system who wish to transmit in the current frame in C-IRSA be M and the corresponding load be the *system load* $L \triangleq M/T$. M denotes the number of users who wish to transmit in the current frame, regardless of whether they have a good or a bad channel state. M out of the M_{tot} users wish to transmit in

the current frame due to the sporadic activity property in mMTC. Let the number of users who have not censored themselves in the current frame in C-IRSA be M_a and the corresponding load be the *active load* $L_a \triangleq M_a/T$. M_a denotes the number of users who wish to transmit in the current frame, who also have a good channel state. The censoring is performed based on L and L_a as described before.

Let the true activity coefficient of the ith user be denoted by a_i, which is 1 if the ith user is truly active in the frame and has not censored itself, and 0 otherwise. Let the set of truly active users (users who are active in the frame and have not censored themselves) be denoted by $\mathcal{A} \triangleq \{i \in [M_{\text{tot}}] | a_i = 1\}$. Let the estimated activity coefficient of the ith user be denoted by $\hat{a}_i$, which is 1 if the ith user is truly active in the frame and has not censored itself, and 0 otherwise. Let the set of users detected to be active by the UAD algorithm be denoted by $\hat{\mathcal{A}} \triangleq \{i \in [M_{\text{tot}}] | \hat{a}_i = 1\}$. Let b_i denote the sporadic activity coefficient of the ith user. That is, if the ith user has data to transmit in the current frame, irrespective of whether the user is censored or not, $b_i = 1$. If the ith user does not have data to transmit in the current frame, then $b_i = 0$. We see that $a_i = 1$ if and only if $b_i = 1$ and $\|\hat{\mathbf{v}}_i\|^2 \geq \nu$, i.e., $a_i = b_i \cdot \mathbb{1}\{\|\hat{\mathbf{v}}_i\|^2 \geq \nu\}$.

The errors in the UAD process are the false positives, which refer to inactive users who have been falsely declared to be active, and the false negatives, which refer to active users who have been missed to be declared active. We denote $\mathcal{F}$ as the set of false positives and $\mathcal{M}$ as the set of false negatives. These can be written as

$$\mathcal{F} = \{i \in [M_{\text{tot}}] | \hat{a}_i(1 - a_i) = 1\}, \tag{5.56}$$

$$\mathcal{M} = \{i \in [M_{\text{tot}}] | (1 - \hat{a}_i)a_i = 1\}. \tag{5.57}$$

In contrast with the above, true positives refer to active users who have been truly declared to be declared active, and true negatives refer to inactive users who have been truly declared to be inactive. We denote $\mathcal{P}$ as the set of true positives and $\mathcal{I}$ as the set of true negatives. These are defined as

$$\mathcal{P} = \{i \in [M_{\text{tot}}] | \hat{a}_i a_i = 1\}, \tag{5.58}$$

$$\mathcal{I} = \{i \in [M_{\text{tot}}] | (1 - \hat{a}_i)(1 - a_i) = 1\}. \tag{5.59}$$

Chapter 4 presents an in-depth discussion about the algorithm and the effect of the UAD errors (namely the false positives and the false negatives) on the performance of IRSA.

5.9.4.2 Channel Estimation Phase

After the UAD phase, the BS performs channel estimation based on the received pilot signal. The received pilot and data signals are indexed by the decoding iteration, since the signals are processed in iterations. For this purpose, we let k denote the current decoding iteration index, and $\mathcal{S}_k$ denote the number of users not yet decoded up to the kth decoding iteration, with $\mathcal{S}_1 = [M_{\text{tot}}]$ and $\mathcal{S}_k^i \triangleq \mathcal{S}_k \setminus \{i\}$. The received pilot signal at the BS in the tth slot in the kth decoding iteration is given by

$$\mathbf{Y}_t^{\mathtt{p}k} = \textstyle\sum_{i \in \mathcal{S}_k} a_i g_{ti} \mathbf{h}_i \mathbf{p}_i^H + \mathbf{N}_t^{\mathtt{p}}, \tag{5.60}$$

where the first term contains signals from users who have transmitted in the current frame. Since the BS has detected only users in $\hat{\mathcal{A}} = \{i \in [M_{\text{tot}}] | \hat{a}_i = 1\}$ to be active, it constructs channel estimates only for the users in $\hat{\mathcal{A}}$. Thus, the BS could possibly construct channel estimates for the false positive users as well, in addition to true positive users.

We now derive the MMSE channel estimates at the BS in each slot. Let $\mathcal{G}_t = \{i \in [M_{\text{tot}}] | g_{ti} = 1\}$ be the set of users who would have transmitted in the tth slot had they been active. Let $\mathcal{M}_t^k = \mathcal{G}_t \cap \mathcal{S}_k \cap \hat{\mathcal{A}}$ and $M_t^k \triangleq |\mathcal{M}_t^k|$. $\mathcal{M}_t^k$ is the set of users who have been detected to be active in the tth slot but have not been decoded up to the kth iteration at the BS (includes true positive and false positive users). Let us stack the channels of the M_t^k users as the columns of $\mathbf{H}_t^k \in \mathbb{C}^{N \times M_t^k}$, let $\mathbf{P}_t^k \in \mathbb{C}^{\tau \times M_t^k}$ denote a matrix that contains the pilot sequences of the M_t^k users as its columns, and let $\mathbf{B}_t^k \triangleq \sigma_{\mathtt{h}}^2 \text{diag}(\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_{M_t^k}})$ be a diagonal matrix that contains the path loss coefficients of the M_t^k users, with $\mathcal{M}_t^k = \{i_1, i_2, \ldots, i_{M_t^k}\}$. Hence, the received pilot signal from (5.60) can be written as $\mathbf{Y}_t^{\mathtt{p}k} = \mathbf{H}_t^k \mathbf{P}_t^{kH} + \mathbf{N}_t^{\mathtt{p}}$.

We now find the channel estimates using the signal $\mathbf{Y}_t^{\mathtt{p}k}$.

Theorem 5.5 ▶ Channel Estimation in Censored-IRSA with Activity Errors.

The minimum mean squared error (MMSE) channel estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ can be found as

$$\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k}(\mathbf{P}_t^k\mathbf{B}_t^k\mathbf{P}_t^{kH} + N_0\mathbf{I}_\tau)^{-1}\mathbf{P}_t^k\mathbf{B}_t^k. \tag{5.61}$$

Specifically, the estimate of the channel $\mathbf{h}_i$ of the ith user at the BS is calculated as $\hat{\mathbf{h}}_{ti}^k = [\hat{\mathbf{H}}_t^k]_{:,i}$. Further, the estimation error $\tilde{\mathbf{h}}_{ti}^k \triangleq \hat{\mathbf{h}}_{ti}^k - \mathbf{h}_i$ is distributed as $\tilde{\mathbf{h}}_{ti}^k \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^k\mathbf{I}_N)$, where δ_{ti}^k is calculated as

$$\delta_{ti}^k = \beta_i\sigma_{\mathtt{h}}^2\left(\frac{N_0\|\mathbf{c}_{ti}^k\|^2+\sum_{j\in\mathcal{S}_k^i}|\mathbf{p}_j^H\mathbf{c}_{ti}^k|^2\hat{a}_j a_j g_{tj}\beta_j\sigma_{\mathtt{h}}^2}{N_0\|\mathbf{c}_{ti}^k\|^2+\sum_{j\in\mathcal{S}_k}|\mathbf{p}_j^H\mathbf{c}_{ti}^k|^2\hat{a}_j a_j g_{tj}\beta_j\sigma_{\mathtt{h}}^2}\right), \tag{5.62}$$

where $\mathbf{C}_t^k \triangleq \mathbf{P}_t^k\mathbf{D}_t^k(\mathbf{P}_t^{kH}\mathbf{P}_t^k\mathbf{D}_t^k + N_0\mathbf{I}_{M_t^k})^{-1}$, $\mathbf{c}_{ti}^k \triangleq [\mathbf{C}_t^k]_{:,i}$, and $\mathbf{D}_t^k \triangleq \mathrm{diag}(d_{ti_1}, d_{ti_2}, \ldots, d_{ti_{M_t^k}})$, with $d_{ti} = \hat{a}_i a_i g_{ti}\beta_i\sigma_{\mathtt{h}}^2$.

Proof. The proof is similar to the proof of Theorem 4.1 in Chapter 4. □

Remarks: The MMSE channel estimate $\hat{\mathbf{H}}_t^k$ of $\mathbf{H}_t^k$ can be written as seen in (5.61) as

$$\hat{\mathbf{H}}_t^k = \mathbf{Y}_t^{\mathtt{p}k}(\mathbf{P}_t^k\mathbf{B}_t^k\mathbf{P}_t^{kH} + N_0\mathbf{I}_\tau)^{-1}\mathbf{P}_t^k\mathbf{B}_t^k, \tag{5.63a}$$

$$\overset{(a)}{=} \mathbf{Y}_t^{\mathtt{p}k}\mathbf{P}_t^k\mathbf{B}_t^k(\mathbf{P}_t^{kH}\mathbf{P}_t^k\mathbf{B}_t^k + N_0\mathbf{I}_{M_t^k})^{-1}, \tag{5.63b}$$

where (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$. Here, the estimate can be calculated via an inverse of either a $\tau \times \tau$ matrix or an $M_t^k \times M_t^k$ matrix as required.

5.9.4.3 Data Decoding Phase

Similar to (5.60), we can find the received data signal in the tth slot in the kth decoding iteration as

$$\mathbf{y}_t^k = \textstyle\sum_{i\in\mathcal{S}_k} a_i g_{ti} \mathbf{h}_i x_i + \mathbf{n}_t. \tag{5.64}$$

We use a generic combining vector $\mathbf{a}_{tm}^k$ to combine the received data signal across antennas to obtain the post-combined data signal $\tilde{y}_{tm}^k \triangleq \mathbf{a}_{tm}^{kH}\mathbf{y}_t^k$ as seen in (5.65).

$$\begin{aligned}\tilde{y}_{tm}^k &= a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k - a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^k \\ &\quad + \textstyle\sum_{i\in\mathcal{S}_k^m\cap\mathcal{P}} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH}\mathbf{h}_i + \sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH}\mathbf{h}_i + \mathbf{a}_{tm}^{kH}\mathbf{n}_t.\end{aligned} \tag{5.65}$$

Here, the term $T_1 \triangleq a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k$ is the desired signal of the mth user; the term $T_2 \triangleq a_m g_{tm} x_m \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^k$ is due to the estimation error $\tilde{\mathbf{h}}_{tm}^k$ of the mth user's channel; the term $T_3 \triangleq \sum_{i\in\mathcal{S}_k^m\cap\mathcal{P}} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH}\mathbf{h}_i$ models the inter-user interference from other true positive users (who have not censored and have not yet been decoded); the term $T_4 \triangleq \sum_{i\in\mathcal{S}_k^m\cap\mathcal{M}} a_i g_{ti} x_i \mathbf{a}_{tm}^{kH}\mathbf{h}_i$ is the interference from false negative users (who have not censored, but cannot be decoded since they are declared to be inactive); and $T_5 \triangleq \mathbf{a}_{tm}^{kH}\mathbf{n}_t$ is the additive noise. Since noise is uncorrelated with the other terms and the data streams of distinct users are uncorrelated, all the terms are uncorrelated with each other. Thus, the power in the received signal is a sum of the powers of the terms [49]. Based on the post-combined data signal, we now compute the signal to interference plus noise ratio (SINR).

Theorem 5.6 ▶ SINR Calculation in Censored-IRSA with Activity Errors.

The signal to interference plus noise ratio (SINR) achieved by the mth user at the BS in the tth slot and the kth decoding iteration in C-IRSA can be written as

$$\rho_{tm}^k = \frac{\mathtt{Gain}_{tm}^k}{N_0 + \mathtt{Est}_{tm}^k + \mathtt{MUI}_{tm}^k + \mathtt{FNU}_{tm}^k}, \forall m \in \mathcal{S}_k. \tag{5.66}$$

Here, $\texttt{Gain}_{tm}^k$ is the desired signal power, $\texttt{Est}_{tm}^k$ is the power of the channel estimation error, $\texttt{MUI}_{tm}^k$ is the multi-user interference due to other true positive users, and $\texttt{FNU}_{tm}^k$ is the interference due to false negative users. These can be calculated as

$$\begin{aligned}
\texttt{Gain}_{tm}^k &\triangleq p_m \hat{a}_m a_m g_{tm} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k|^2 / \|\mathbf{a}_{tm}^k\|^2, \\
\texttt{Est}_{tm}^k &\triangleq \textstyle\sum_{i \in \mathcal{S}_k} p_i \hat{a}_i a_i g_{ti} \delta_{ti}^k, \\
\texttt{MUI}_{tm}^k &\triangleq \textstyle\sum_{i \in \mathcal{S}_k^m} p_i \hat{a}_i a_i g_{ti} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{ti}^k|^2 / \|\mathbf{a}_{tm}^k\|^2, \\
\texttt{FNU}_{tm}^k &\triangleq \textstyle\sum_{i \in \mathcal{S}_k} p_i (1 - \hat{a}_i) a_i g_{ti} \beta_i \sigma_{\texttt{h}}^2.
\end{aligned}$$

Proof. The proof is similar to the proof of Theorem 4.3 in Chapter 4. □

The channel estimates $\hat{\mathbf{h}}_{ti}^k$ and the error covariance δ_{ti}^k in the above expressions are obtained from Theorem 5.5. The combining vector that maximizes the SINR in (5.66) is the MMSE combiner [49], which can be found as

$$\begin{aligned}
\mathbf{A}_t^k &= \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\texttt{p}}^k (d_t^k \mathbf{I}_{M_t^k} + \hat{\mathbf{H}}_t^{kH} \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\texttt{p}}^k)^{,-1} \\
&= (d_t^k \mathbf{I}_N + \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\texttt{p}}^k \hat{\mathbf{H}}_t^{kH})^{-1} \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\texttt{p}}^k,
\end{aligned}$$

where $\mathbf{D}_{t,\texttt{p}}^k \triangleq \text{diag}(p_{i_1}, p_{i_2}, \ldots, p_{i_{M_t^k}})$ contains the power coefficients of the M_t^k users, and $d_t^k \triangleq N_0 + \sum_{i \in \mathcal{S}_k} p_i \hat{a}_i g_{ti} b_{ti}^k$, where

$$b_{ti}^k = \beta_i \sigma_{\texttt{h}}^2 \left(\frac{N_0 \|\mathbf{f}_{ti}^k\|^2 + \sum_{j \in \mathcal{S}_k^i} |\mathbf{p}_j^H \mathbf{f}_{ti}^k|^2 \hat{a}_j g_{tj} \beta_j \sigma_{\texttt{h}}^2}{N_0 \|\mathbf{f}_{ti}^k\|^2 + \sum_{j \in \mathcal{S}_k} |\mathbf{p}_j^H \mathbf{f}_{ti}^k|^2 \hat{a}_j g_{tj} \beta_j \sigma_{\texttt{h}}^2} \right).$$

Here, $\mathbf{F}_t^k \triangleq \mathbf{P}_t^k \mathbf{E}_t^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{E}_t^k + N_0 \mathbf{I}_{M_t^k})^{-1}$, with $\mathbf{f}_{ti}^k \triangleq [\mathbf{F}_t^k]_{:,i}$, and $\mathbf{E}_t^k \triangleq \text{diag}(e_{ti_1}, e_{ti_2}, \ldots, e_{ti_{M_t^k}})$, where $e_{ti} \triangleq \hat{a}_i g_{ti} \beta_i \sigma_{\texttt{h}}^2$.

We use the SINR threshold model to abstract the decoding of a user's packet: any packet is decoded correctly if and only if its SINR is above a threshold $\gamma_{\text{th}} \geq 1$ [17, 74]. With the SINR threshold model, the performance of C-IRSA can be computed as follows. First, the SINRs achieved by all users in all slots is computed. If there is a user with SINR

$\geq \gamma_{\text{th}}$ in some slot, that packet is successfully decoded and its contribution is removed all other slots in which that user has transmitted a replica [8]. We then proceed to the next decoding iteration and recompute the SINRs for all users yet to be decoded. This process stops when no additional users are decoded in two successive iterations. The throughput $\mathcal{T}$ is calculated as the number of unique packets correctly decoded divided by the number of slots.

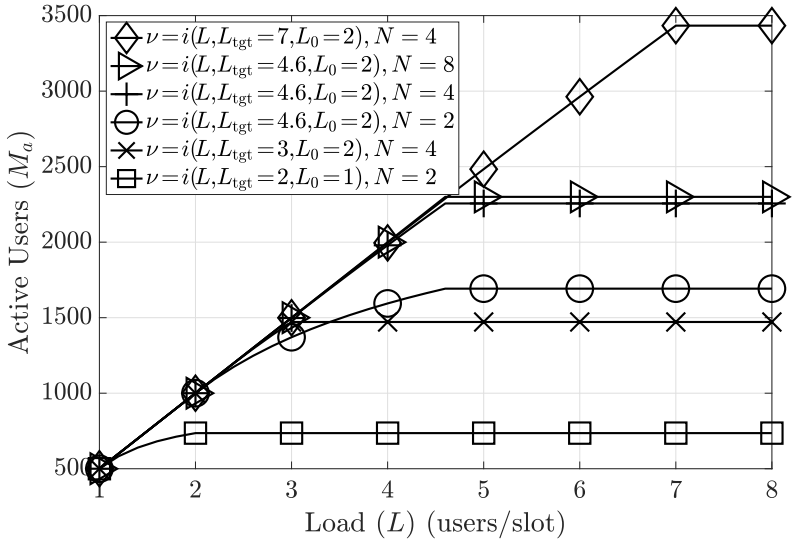
We have thus obtained channel estimates and the SINR in C-IRSA accounting for UAD errors.

5.9.5 Numerical Results in C-IRSA Accounting for UAD Errors

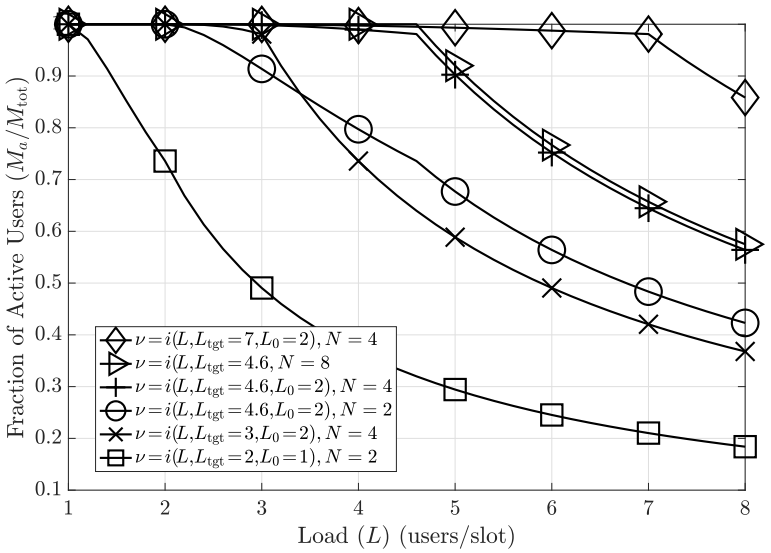
In this subsection, we study the performance of our UAD algorithm from Chapter 4 applied to C-IRSA. We also analyze the impact of UAD errors on the throughput of C-IRSA via Monte Carlo simulations. The metrics used to characterize the UAD performance are false positive rate, FPR $\triangleq \frac{|\mathcal{F}|}{|\mathcal{F}|+|\mathcal{I}|}$, and false negative rate, FNR $\triangleq \frac{|\mathcal{M}|}{|\mathcal{M}|+|\mathcal{P}|}$. FPR is the fraction of inactive users declared to be active whereas FNR is the fraction of active users declared to be inactive. The UAD algorithm is run for $j_{\max} = 100$ maximum iterations. The UAD algorithm has an activity threshold parameter $\gamma_{\text{pr}} = 10^{-8}$, which is used to declare the estimated activities of the users.[14]

The results in this section are for $T = 50$ slots and $N_s = 10^4$ Monte Carlo runs [49]. The sporadic activities of the users are modelled as being Bernoulli distributed, i.e., $b_i \overset{\text{i.i.d.}}{\sim} \text{Ber}(p_a)$, where $p_a = 0.1$ is the per-user sporadic activity probability [53]. Since we are evaluating the performance *averaged* over Monte Carlo simulations, we calculate the system load L in each frame as the *average* number of users expected to be active in each frame per slot, i.e., $L = M_{\text{tot}}p_a/T = L_{\text{tot}}p_a$. The number of users contending for the T slots is computed in each simulation based on the load L as $M_{\text{tot}} = \lfloor LT/p_a \rceil$. The other system parameters are chosen similar to the settings in Section 5.6.

[14]We can obtain different FNRs and FPRs by varying the activity threshold γ_{pr}, similar to Chapter 4. This threshold can be used to trade-off the false negatives for the false positives and vice versa as required, similar to Chapter 4. If needed, γ_{pr} can be tuned at each load to achieve the lowest balanced error rate, which is the average of the FNR and the FPR.

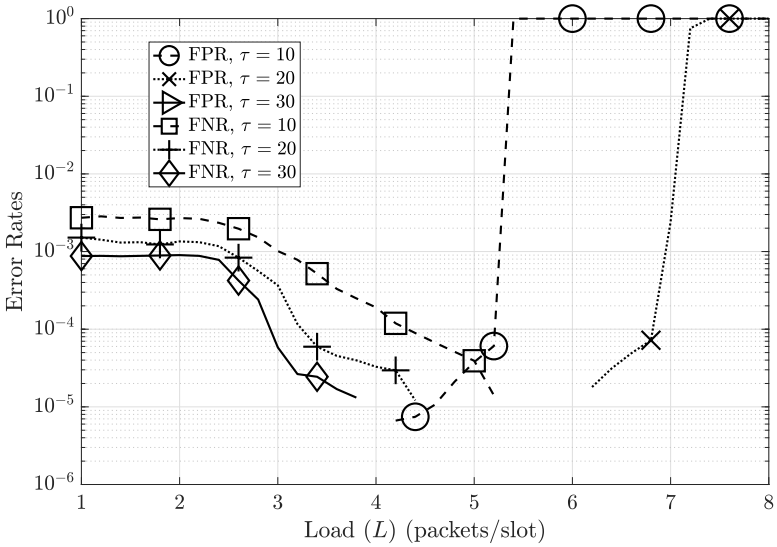


(a) Effect of $\nu = i(L, L_{\text{tgt}}, L_0)$ on M_a.

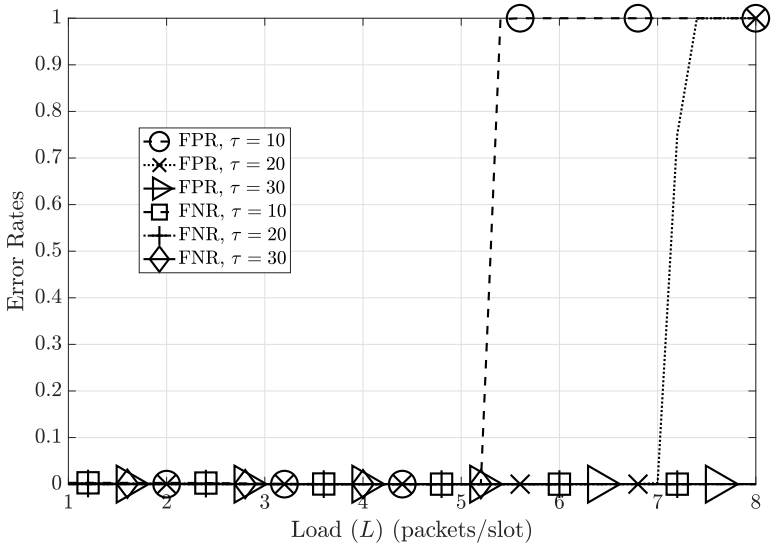


(b) Effect of $\nu = i(L, L_{\text{tgt}}, L_0)$ on M_a/M_{tot}.

Figure 5.22: Effect of $\nu = i(L, L_{\text{tgt}}, L_0)$ on the number of active users in C-IRSA.

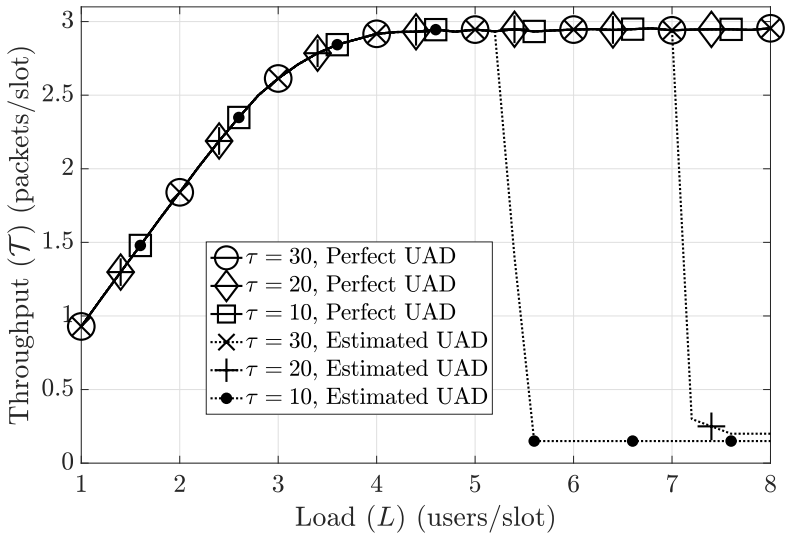


(a) UAD Performance in C-IRSA.

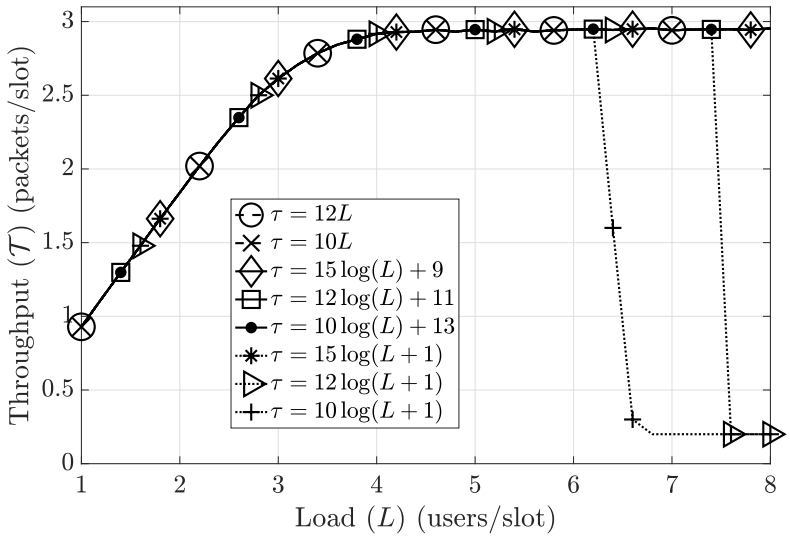


(b) Performance depicted with a linear y-axis.

Figure 5.23: UAD in C-IRSA.



(a) Perfect UAD vs Estimated UAD in C-IRSA.



(b) Impact of pilot length choices on C-IRSA with estimated UAD.

Figure 5.24: Impact of UAD errors on C-IRSA.

Before we study the UAD performance in C-IRSA, we study the effect of the censor function $\nu = i(L, L_{\text{tgt}}, L_0)$ on the active users M_a and the fraction of active users to the total number of users M_a/M_{tot}. As the load is increased from $L = 1$ to $2, 3, \ldots, 8$, the total number of users over the T slots increases from $M_{\text{tot}} = 500$ to $1000, 1500, \ldots, 4000$. In Fig. 5.22(a), we plot the active users as a function of the load L. For each of the curves, when $L < L_0$, all the users are active, i.e., $M_a = M = LT/p_a$. This is because we do not censor when $L < L_0$. For each of the curves, when $L_0 \leq L < L_{\text{tgt}}$, we see that M_a starts increasing and it hits a peak value at $L = L_{\text{tgt}}$. This is because we start censoring when $L_0 \leq L < L_{\text{tgt}}$, and thus, $M_a < M = LT/p_a$. Beyond $L > L_{\text{tgt}}$, the main idea of the censor function $\nu = i(L, L_{\text{tgt}}, L_0)$ is to maintain a fixed number of active users. Thus, we see a constant M_a for every $L > L_{\text{tgt}}$. In Fig. 5.22(b), we plot the ratio of active users to the total number of users as a function of the load L. In the previous figure, we saw the trends of the number of active users M_a as the load L is varied. Here, we observe that all the curves trend downwards as the load increases. Thus, as L is increased beyond L_0, using the censor function $\nu = i(L, L_{\text{tgt}}, L_0)$ monotonically reduces M_a/M_{tot} and keeps M_a fixed.

The performance of compressed sensing algorithms such as orthogonal matching pursuit [99], approximate message passing [98], sparse Bayesian learning (SBL) [68, 69, 120], multiple sparse Bayesian learning (MSBL) [16], and our algorithm as well depends not just on the number of sparse entries but also on the fraction of the number of non-zero entries to the total number of entries [122]. Specifically, the result is as follows: If a multiple measurement vector recovery algorithm (such as ours) is applied, a pilot length of $\tau = \Omega(M_a \log(M_{\text{tot}}/M_a))$ can achieve a vanishing activity error rate as $N \to \infty$ [122].[15] Thus, the error rates that we expect to see for C-IRSA must adhere to the above result.

In Fig. 5.23(a), we plot the UAD error rates in C-IRSA versus the load L under the generalized censor function $\nu = i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$. With a pilot length of $\tau = 10$, the FNR drops below an error rate of 10^{-4} at $L = 4.2$. In contrast with this, the FNR

[15]The $\Omega(\cdot)$ notation provides a best case guarantee as opposed to the $\mathcal{O}(\cdot)$ notation which provides a worst case guarantee. The formal definition of $\Omega(\cdot)$ is $\Omega(g(n)) = \{f(n) | \exists c, n_0 > 0$ such that $0 \leq cg(n) \leq f(n) \forall n \geq n_0\}$.

drops below 10^{-4} for $\tau = 20$ at $L = 3.2$ and for $\tau = 10$ at $L = 2.9$. This demonstrates the efficacy of increasing the pilot length in improving the UAD performance. The FPR for $\tau = 10$ crosses 10^{-4} at $L = 5.2$, beyond which the FPR becomes unity. Similarly, the FPR for $\tau = 20$ crosses 10^{-4} at $L = 6.8$, beyond which the FPR becomes unity. The FPR for $\tau = 30$ is zero for the depicted range of L, and does not show up in the plot since the y-axis is on a logarithmic scale. However, we can see that FPR is zero as plotted in Fig. 5.23(b), where the y-axis is on a linear scale. The FPR for $\tau = 30$ will cross 10^{-4} at some high load beyond $L = 8$. Thus, we observe that increasing the pilot length decreases both the FPRs and the FNRs, and hence the pilot length highly affects the UAD performance of C-IRSA.

We see that the FPR is zero at low to moderate loads and becomes unity at high loads. This is because a pilot length of $\tau = 10$ is insufficient at such high loads, and the UAD performance suffers as a consequence. At this high load, the result we previously saw dictates that $\tau = \Omega(M_a \log(M_{\text{tot}}/M_a))$ can achieve a vanishing activity error rate as $N \to \infty$ [122]. Since we use the censor function $\nu = i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$, we start censoring users at $L_0 = 2$ and we wish to continue the same number of active users beyond $L_{\text{tgt}} = 4.6$. Since M_a is fixed for all $L > L_{\text{tgt}}$, and M_{tot} is increasing, a pilot length of $\tau = 10$ becomes insufficient at $L = 5.2$, and a pilot length of $\tau = 20$ becomes insufficient at $L = 6.8$, whereas a pilot length of $\tau = 30$ is sufficient for $0 < L \leq 8$. FPR becomes non-zero since M_{tot} is increasing, which requires higher τ to achieve near-zero error rates. Further, the FNR is fairly constant upto $L_0 = 2$ (since we do not censor), and then drops beyond $L_0 = 2$. This is because we are in the regime where the pilot length is sufficient and the percentage of active users starts dropping. Also, the activity threshold is high enough to ensure that no user is missed. This comes at the cost of high FPR at higher loads.

We compare the performance of estimated UAD and perfect UAD in C-IRSA in Fig. 5.24(a). For $\tau = 10$, we see that with estimated UAD, the throughput drops to near-zero beyond $L = 5.2$. For $\tau = 20$, we see that with estimated UAD, the throughput drops to near-zero beyond $L = 7.2$. These are the same loads at which we observed the

FPR becomes unity in Fig. 5.23(a). For $\tau = 30$, we see that the performance of estimated UAD is the same when we assume perfect UAD. The low FNRs observed in Fig. 5.23(a) do not have a significant impact on the throughput. In fact, we have observed that to have a significant impact on the throughput, the error rates need to be higher than 0.1. But we have observed in Fig. 5.23(b), that only the FPRs for $\tau = 10$ and 20 are higher than 0.1 and consequently, the corresponding throughputs are near-zero. Otherwise, all the error rates are indeed very low and thus, the throughput results presented before are indeed valid given the fact that we choose an appropriate pilot length.

When we keep the ratio M_a/M_{tot} fixed, $\tau = \Omega(M_a)$ pilot symbols are sufficient. But in IRSA and C-IRSA, the number of collisions in each slot is far fewer than M_a, and as a result very low pilot lengths are sufficient for accurate UAD. In fact, we had observed earlier that a pilot length of $\tau = 10$ was indeed sufficient for peak channel estimation performance, and increasing τ beyond 10 did not have a significant impact on the throughput (See Chapter 4). The UAD problem is easier to solve when compared to channel estimation (See Chapter 4), and thus, any pilot length that is sufficient for channel estimation is indeed sufficient for accurate UAD. In practice, the optimal pilot length must be calculated for every L, and this optimal pilot length not only provides good UAD performance but also good channel estimation.

A simple solution to achieve very accurate UAD is to perhaps consider a fixed pilot length of $\tau = 30$ or even higher. While this keeps the throughput at the peak as seen in Fig. 5.24(a), it also leads to low spectral efficiencies (SEs) at low loads. At low loads, such a high pilot length is excessive and fewer symbols are used for data, and thus, lower SEs are achieved. Thus, we must consider variable pilot lengths as opposed to fixed pilot lengths as above. In Fig. 5.24(b), we plot the performance of C-IRSA with estimated UAD under different variable choices for the pilot length. In all of the cases plotted in the figure, we round up the choice of the pilot lengths to the nearest integer with the ceil function to ensure integer valued pilot lengths. That is, the legend of $\tau = 10L$, $\tau = 10\log(L+1)$, and $\tau = 10\log(L) + 13$ are actually implemented as $\tau = \lceil 10L \rceil$, $\tau = \lceil 10\log(L+1) \rceil$, and $\tau = \lceil 10\log(L) + 13 \rceil$. We first analyze the case when the pilot length varies linearly with

the load, i.e., when $\tau = k_1 L$, where k_1 is a constant. If we choose $\tau = 10L$ or $\tau = 12L$, we see that the throughput is maximized and it achieves the best performance of the system. But this leads to excessive pilot lengths of $\tau = 40/80$ or $\tau = 48/96$ at $L = 4/8$, respectively, for $\tau = 10L$ or $\tau = 12L$. This is again excessive for both UAD and channel estimation, and leads to low SEs.

The recovery guarantee of $\tau = \Omega(M_a \log(M_{\text{tot}}/M_a))$ can help us calculate the pilot lengths that are required for accurate UAD. The guarantee $\tau = \Omega(M_a \log(M_{\text{tot}}/M_a))$ can be rewritten in terms of the load as $\tau = \Omega(M_a \log(L_{\text{tot}}T/M_a)) = \Omega(M_a \log(LT/(p_a M_a)))$.[16] When M_a is fixed, this guarantee becomes $\tau = \Omega(\log(Lk_2))$, where k_2 is a constant. Thus, we explore the case when the pilot length is varied as a logarithmic function of the load L as $\tau = k_3 \log(L)$, where k_3 is a constant. To ensure that the pilot length does not become negative for $L < 1$, we can choose the pilot length alternatively as $\tau = k_4 \log(L + 1)$, where k_4 is a constant. In Fig. 5.24(b), we plot the performance with $\tau = 10 \log(L + 1)$, $\tau = 12 \log(L + 1)$, and $\tau = 15 \log(L + 1)$. While $\tau = 15 \log(L + 1)$ maintains the performance at the peak, $\tau = 10 \log(L + 1)$ and $\tau = 12 \log(L + 1)$ do not maintain the peak performance beyond $L = 6.2$ and $L = 7.4$, respectively. This is because these curves have FPR$= 1$ beyond $L = 6.2$ and $L = 7.4$, and thus, their throughput becomes near-zero.

When M_a is fixed, since the recovery guarantee was $\tau = \Omega(\log(Lk_2))$, where k_2 is a constant, we can alternately choose the pilot length as $\tau = k_5 \log(L) + k_6$, where k_5 and k_6 are constants. In accordance with this, in Fig. 5.24(b), we plot the performance with $\tau = 10 \log(L) + 13$, $\tau = 12 \log(L) + 11$, and $\tau = 15 \log(L) + 9$. The throughputs corresponding to all three of these choices coincide with each other. All the choices of k_5 and k_6 in these functions are fine tuned to ensure that the UAD error rates are near-zero.

In fact, since the recovery guarantee is asymptotic, there exists a choice of pilot length $\tau = k_5 \log(L) + k_6$ with minimal k_5 and k_6 such that UAD is extremely accurate at all

[16] In order to vary the total number of users $M_{\text{tot}} = LT/p_a$, we can alternately vary T or the activity probability p_a, instead of varying L. In this case, the recovery guarantees become $\tau = \Omega(\log(T))$ or $\tau = \Omega(-\log(p_a))$, respectively. In either case, the load L is fixed, and the complexity of the compressed sensing recovery problem is fixed as M_{tot} increases. We have considered the more difficult problem which deals with increasing L, which involves a higher complexity of the compressed sensing recovery problem [67].

L [122]. This choice needs to be found in practice, and can be fine tuned as we observed above. For upto loads of $L = 8$, $\tau = 30$ or $\tau = 10\log(L) + 13$ is sufficient. This implies that we can indeed assume perfect UAD upto such pilot lengths. Further, we do not need very large pilot lengths for accurate UAD. The generalized censor function also does not depend on the pilot length.

5.9.6 Other Censor Functions

We now investigate the performance with various other censor functions defined as

$$h_1(L, L_{\text{tgt}}) \triangleq \begin{cases} 4, & L < L_{\text{tgt}}, \\ \mathrm{H}(L, L_{\text{tgt}}, \nu = 4), & L \geq L_{\text{tgt}}, \end{cases} \tag{5.67}$$

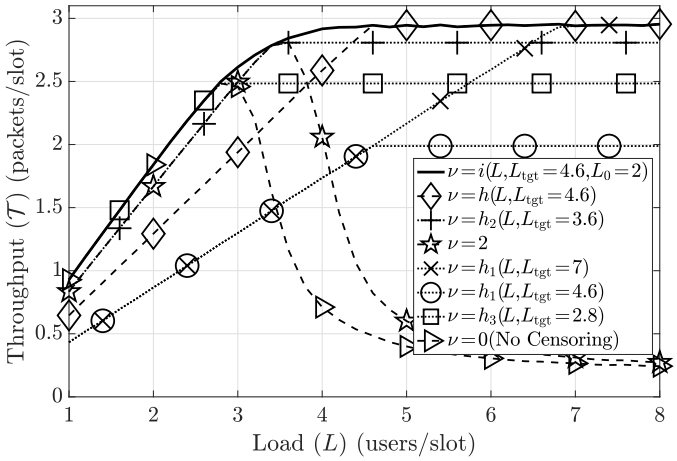
$$h_2(L, L_{\text{tgt}}) \triangleq \begin{cases} 2, & L < L_{\text{tgt}}, \\ \mathrm{H}(L, L_{\text{tgt}}, \nu = 2), & L \geq L_{\text{tgt}}, \end{cases} \tag{5.68}$$

$$h_3(L, L_{\text{tgt}}) \triangleq \begin{cases} 0, & L < L_{\text{tgt}}, \\ \mathrm{H}(L, L_{\text{tgt}}, \nu = 0), & L \geq L_{\text{tgt}}. \end{cases} \tag{5.69}$$

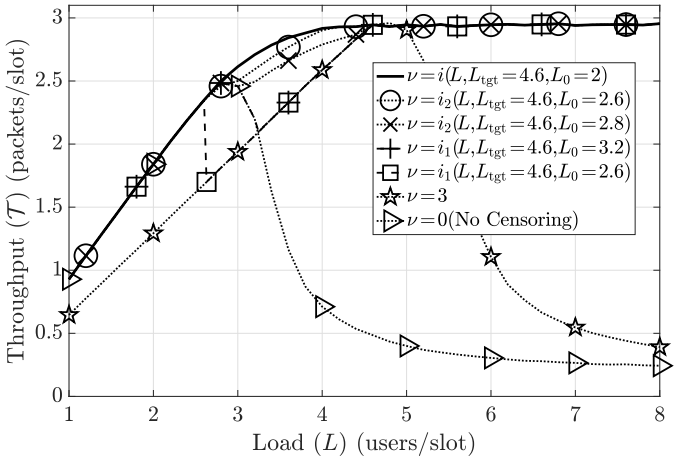
The functions $h_1(\cdot,\cdot)$, $h_2(\cdot,\cdot)$, and $h_3(\cdot,\cdot)$ are obtained from the $h(\cdot,\cdot)$ function, when the target threshold is chosen as a fixed value as $\nu = 4$, 2, and 0, respectively. We have also explored other variants of the generalized censor function defined as

$$i_1(L, L_{\text{tgt}}, L_0) \triangleq \begin{cases} 0, & L < L_0, \\ \nu_{\text{tgt}}, & L_0 \leq L < L_{\text{tgt}}, \\ \mathrm{H}(L, L_{\text{tgt}}, \nu_{\text{tgt}}), & L \geq L_{\text{tgt}}, \end{cases} \tag{5.70}$$

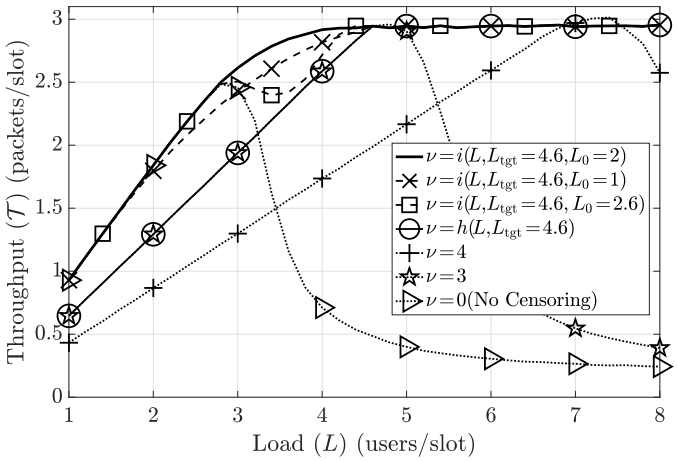
$$i_2(L, L_{\text{tgt}}, L_0) \triangleq \begin{cases} 0, & L < L_0, \\ \left(\dfrac{L - L_0}{L_{\text{tgt}} - L_0}\right) \nu_{\text{tgt}}, & L_0 \leq L < L_{\text{tgt}}, \\ \mathrm{H}(L, L_{\text{tgt}}, \nu_{\text{tgt}}), & L \geq L_{\text{tgt}}. \end{cases} \tag{5.71}$$



(a) Comparison between $i(\cdot,\cdot,\cdot)$, $h(\cdot,\cdot)$, $h_1(\cdot,\cdot)$, $h_2(\cdot,\cdot)$, and $h_3(\cdot,\cdot)$.



(b) Comparison between $i(\cdot,\cdot,\cdot)$, $i_1(\cdot,\cdot,\cdot)$, and $i_2(\cdot,\cdot,\cdot)$.



(c) Comparison between different L_0 with $i(\cdot,\cdot,\cdot)$.

Figure 5.25: Performance comparison of varied censor functions.

Here, $i_1(\cdot,\cdot,\cdot)$ is obtained from $h(\cdot,\cdot)$ if we do not censor below L_0 and $i_2(\cdot,\cdot,\cdot)$ is obtained from $i(\cdot,\cdot,\cdot)$ if we perform linear interpolation instead of logarithmic interpolation.

In Fig. 5.25, we evaluate the performance of C-IRSA with varied censor functions defined as above. Firstly, in Fig. 5.25(a), we compare the performance between $i(\cdot,\cdot,\cdot)$, $h(\cdot,\cdot)$, $h_1(\cdot,\cdot)$, $h_2(\cdot,\cdot)$, and $h_3(\cdot,\cdot)$. $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$ performs the best and achieves the highest throughput at every L. When $L < L_0$, censoring is not always needed. If we censor with $\nu = \nu_{\text{tgt}}$, we achieve the performance with $h(L, L_{\text{tgt}} = 4.6)$. The performance of $h(L, L_{\text{tgt}} = 4.6)$ coincides with the performance of $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$ when $L > L_{\text{tgt}}$. $h_1(\cdot,\cdot)$, $h_2(\cdot,\cdot)$, and $h_3(\cdot,\cdot)$ operate with different ν_{tgt} compared to $h(\cdot,\cdot)$. Their performances coincide with their respective ν_{tgt} of 4, 2, and 0 upto the respective L_{tgt}, and beyond L_{tgt} their performance remains fixed at the same throughput achieved at $L = L_{\text{tgt}}$. $h_1(\cdot,\cdot)$ performs poorly when $L < L_{\text{tgt}}$ since it operates with a very high ν_{tgt}. The performance of $h_2(L, L_{\text{tgt}} = 3.6)$ is very close to the performance of $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$, both above and below $L = 3.6$. $h_3(L, L_{\text{tgt}} = 2.8)$ performs good upto $L < L_{\text{tgt}}$, and beyond that it performs poorly w.r.t. $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$.

In summary, this figure validates the reason why we should not censor at low loads. At low loads, $\nu = 0$ performs the best and hence, there is no requirement of censoring users with a high ν_{tgt}. At high loads, a high ν_{tgt} can help. We can extract more juice out of the system by crossing the gap between $h_3(L, L_{\text{tgt}} = 2.8)$ and $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$. This is why we need to resort to censoring between $L_0 \leq L < L_{\text{tgt}}$. The generalized censor function takes care of all of this within one continuous function.

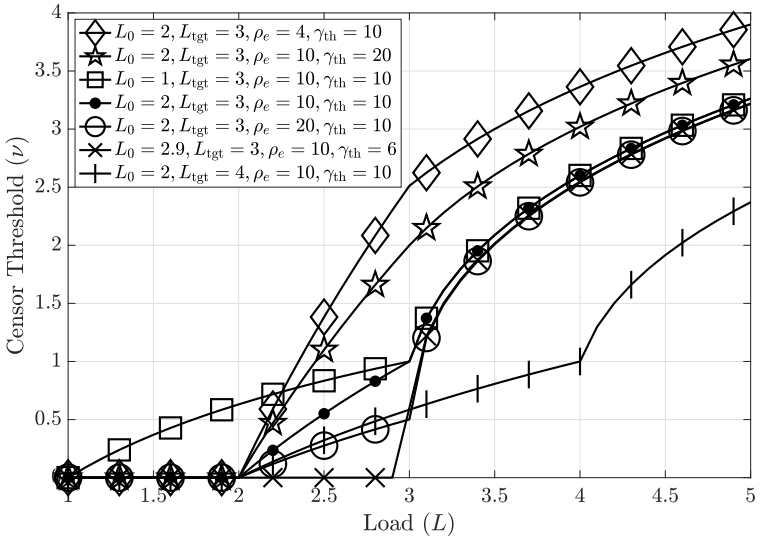
We now study various functions and see which performs the best when $L_0 \leq L < L_{\text{tgt}}$ in Fig. 5.25(b). The first variant of $h(\cdot,\cdot)$ is the $i_1(\cdot,\cdot,\cdot)$ function, which is a discontinuous function which blindly sets $\nu = 0$ below a chosen L_0 and maintains $\nu = \nu_{\text{tgt}}$ between $L_0 \leq L < L_{\text{tgt}}$. $i_1(L, L_{\text{tgt}} = 4.6, L_0 = 2.6)$ sees an abrupt drop in the throughput at $L = 2.6$ because of this discontinuity. $i_1(L, L_{\text{tgt}} = 4.6, L_0 = 3.2)$ follows the performance of $\nu = 0$ up to $L = 3.2$ and then follows the performance of $h(\cdot,\cdot)$. Both perform poorly when $L_0 \leq L < L_{\text{tgt}}$ compared to $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$. We need to obtain better censor thresholds in this region.

In order to make the throughput slowly continuously vary with the load, we resort to interpolation. We see that linear interpolation performs very well, and nearly achieves the best performance. Specifically, $i_2(L, L_{\text{tgt}} = 4.6, L_0 = 2.6)$ and $i_2(L, L_{\text{tgt}} = 4.6, L_0 = 2.8)$ nearly achieve the performance of $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$. However, we could never reach the peak performance of $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$ no matter how much L_0 was fine tuned. When $L < L_{\text{tgt}}$ is close to $L_{\text{tgt}} = 4.6$, we see that $i_2(L, L_{\text{tgt}} = 4.6, L_0 = 2.6)$ just starts achieving the best performance of $i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$. This suggests that we need to choose a higher censoring threshold when L just exceeds L_0. This can be achieved, for example, if we choose a logarithmic interpolation function as seen in the SISO case. This is precisely the reason why we choose the logarithmic interpolation, and as seen in the figure, it achieves the peak performance. The logarithmic interpolation approach closely approximates the algorithmic approach presented in the paper, which is PLR_a-optimal. This validates our choice of the logarithmic interpolation.

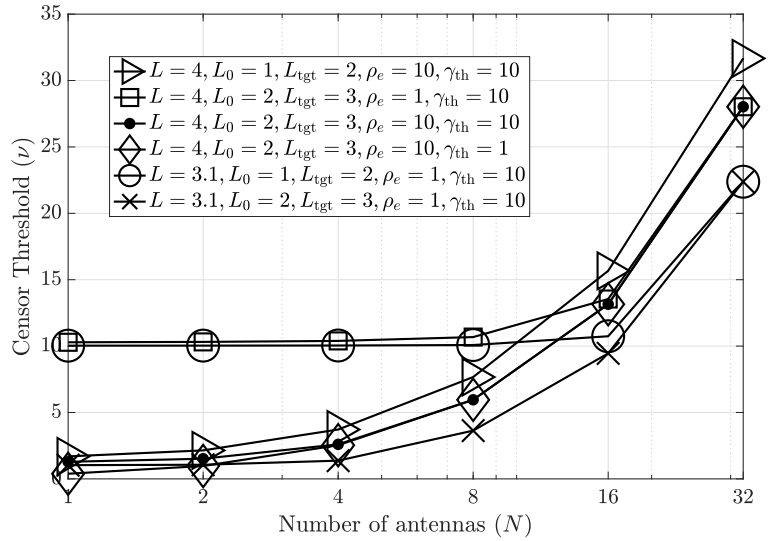
Finally, in Fig. 5.25(c), we fine-tune L_0 with the logarithmic interpolation. With $L_0 = 1$ or 2.6, the performance is not at the peak. The system is censoring too much with $L_0 = 2.6$ and too less with $L_0 = 1$. We can fine-tune L_0, and see with which value of L_0, the system performs the best. In this case, $L_0 = 2$ performs the best.

5.9.7 Impact of System Parameters on the Generalized Censor Function

We now plot the generalized censor function as a function of the various system parameters. In Fig. 5.26(a), we plot the generalized censor function as a function of the system load L. Each of the curves have a logarithmic interpolation for $L_0 \leq L < L_{\text{tgt}}$ as described in the previous comment. The censor function is a monotonically non-decreasing function of L. As we increase the system load, as expected, the censor function increases to censor more users. The curves shift downwards as $\rho_e^{-1}\gamma_{\text{th}}$ decreases from 2.5 to 2 to 1 to 0.5. This is because a decrease in target threshold monotonically decreases the optimal censor function via the $\mathrm{H}(\cdot,\cdot,\cdot)$ function. In the $L_0 \leq L < L_{\text{tgt}}$ region, increasing L_0 decreases the slope of the censoring function and hence reduces the value of the generalized censor

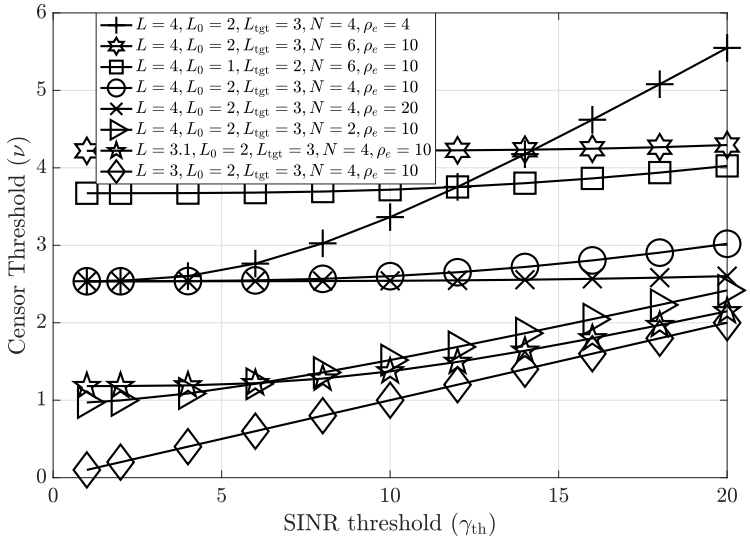


(a) Impact of load.

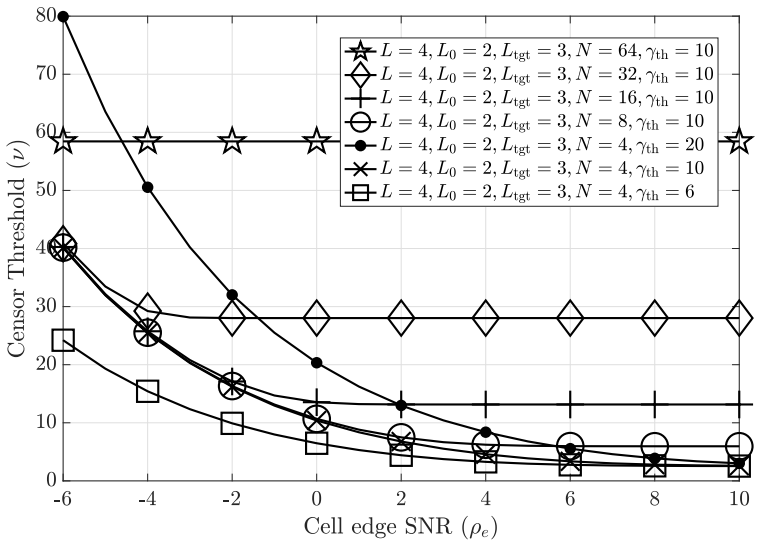


(b) Impact of number of antennas.

Figure 5.26: Impact of system parameters on the generalized censor function $i(L, L_{\text{tgt}}, L_0)$.



(a) Impact of SINR threshold.



(b) Impact of cell edge SNR.

Figure 5.27: Impact of system parameters on the generalized censor function $i(L, L_{\text{tgt}}, L_0)$.

function.

Fig. 5.26(b) studies the impact of the number of antennas N on the generalized censor function. The censor function is a monotonically non-decreasing function of γ_{th} for $L \geq L_{\text{tgt}}$. This is because an increase in N monotonically decreases (non-increases) the optimal censor function via the $\mathrm{H}(\cdot,\cdot,\cdot)$ function. We also observe that at high N, the optimal censoring function tends to concentrate near N. Channel hardening [106] is the phenomenon that as N increases the norm squared of the fade converges to a constant, i.e.,$\|\mathbf{v}_i\|^2 \to N\sigma_{\mathtt{h}}^2$. Since we perform fade-based censoring as $\|\mathbf{v}_i\|^2 \geq \nu$, the threshold has to be made close to $N\sigma_{\mathtt{h}}^2$ ($\pm$ a standard deviation) to ensure a significant impact on $\mathcal{T}$. The effect of channel hardening gets more pronounced at higher N as seen in Fig. 5.26(b).

In Fig. 5.27(a), we study the impact of the SINR threshold γ_{th} on the generalized censor function. The censor function is a monotonically non-decreasing function of γ_{th}. This is because an increase in the SINR threshold γ_{th} monotonically increases (non-decreases) the optimal censor function via the $\mathrm{H}(\cdot,\cdot,\cdot)$ function. As we increase the SINR threshold γ_{th}, fewer users cross the SINR threshold, and as a consequence more users are censored to ensure the remaining users who have good channel states cross the SINR decoding threshold. Fig. 5.27(b) studies the impact of the cell edge SNR ρ_e on the generalized censor function. The censor function is a monotonically non-increasing function of ρ_e. This is because an increase in the SNR ρ_e monotonically decreases (non-increases) the optimal censor function via the $\mathrm{H}(\cdot,\cdot,\cdot)$ function. In both of these plots, changing N changes the slope of the optimal censoring function. At low SNRs, we need to censor users with a very high threshold to ensure only users with better channel states have good decodability and hence get decoded.

5.9.8 C-IRSA accounting for Shadowing

So far, we have accounted for only the path loss coefficient in the system model and not the shadowing component, since we model the large scale fading coefficient (LSFC) as β_m (dB) $= -37.6\log_{10}(r_i/r_0)$, where r_0 represents the reference distance. The parameter β_m, which represents the LSFC, can model the contribution of both path loss as well as shadowing.

It is quite prevalent to model the LSFC, in typical communication systems, as β_m (dB) $= -37.6\log_{10}(r_i/r_0) + \texttt{Shad.}$, where `Shad.` represents the shadowing component [21]. In C-IRSA, users estimate their small scale fading coefficient (SSFC) and perform censoring based on the estimated fading coefficient. Similar to estimating their SSFCs, the users can estimate their LSFCs every once in a while. Since LSFCs change very slowly with time [126], they need not be estimated every frame like the SSFCs, but can be estimated irregularly, say every 50 or every 100 frames.

Multiple techniques have been used in existing works to estimate the shadow powers. A Kalman-filter-based shadow power estimation and prediction algorithm is developed in [127] for the Rayleigh-lognormal scenario. A sequential Bayesian method has been proposed for dynamic estimation and prediction of shadow powers from instantaneous signal powers in composite fading-shadowing wireless communication channels [128]. Further, [128] also reviews 10 other papers which perform very accurate shadow power estimation. Existing works show that it can be estimated over time by averaging over the beacon signals at the start of multiple frames. A few works do this via moving window based averaging, and it captures the effect of both shadowing and path loss [129]. Path loss estimation based on distances is terrain and frequency dependent [130, 131]. LSFC estimation can be incorporated into the protocol as well.

We now present the performance of C-IRSA accounting for shadowing as well as path loss in Fig. 5.28. Here, we model the LSFC as $\beta_i\,(\text{dB}) = -37.6\log_{10}(r_i/r_0) + \texttt{Shad}$, where $\texttt{Shad} \sim \mathcal{N}(0, \sigma^2_{\text{Shad}})$ is a standard log-normal shadowing component, with shadowing variance σ^2_{Shad}. The censoring is still performed based only on the SSFC and not the LSFC, as earlier. The users can estimate their LSFCs under any of the previous mentioned techniques. We choose the censor thresholds as $\nu = i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$. In Fig. 5.28, we compare the performance of C-IRSA without accounting for shadowing (with $\sigma^2_{\text{Shad}} = 0$) with the performance accounting for shadowing (with non-zero σ^2_{Shad}). We observe that $\sigma^2_{\text{Shad}} = 9$ is nearly coincidental with $\sigma^2_{\text{Shad}} = 0$, and both achieve a peak throughput of $\mathcal{T} = 2.93$.[17] With a realistic value of $\sigma^2_{\text{Shad}} = 16$ [49], we observe that the peak throughput

[17]We have excluded the curves with $\sigma^2_{\text{Shad}} = 1$, $\sigma^2_{\text{Shad}} = 2$, and $\sigma^2_{\text{Shad}} = 4$, since they are nearly

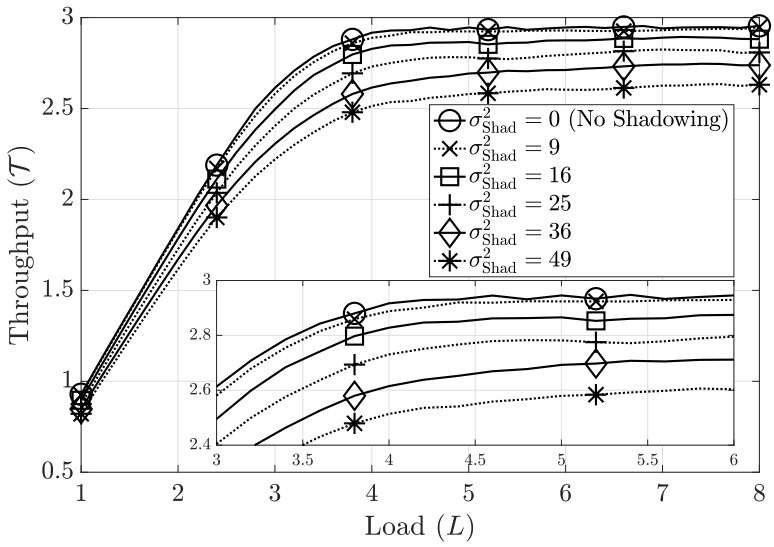


Figure 5.28: Performance of C-IRSA with shadowing.

only slightly drops to $\mathcal{T} = 2.87$, which is a very small performance reduction when we account for shadowing. When σ^2_{Shad} is higher, few users experience deep fades, and the performance drops as expected. Similar to conventional communication systems, if users experience deep fades, then the users can change their frequency of operation and transmit in a different frequency band.

5.9.9 Uniqueness of Pilot Sequences

In IRSA, in each slot, there are $L\bar{d}$ packet collisions on an average in every frame [7]. Even if we were to do UAD on a slot by slot basis, we would need only $L\bar{d}$ unique pilot sequences in each slot on an average. For example, if we use an average repetition factor of $\bar{d} = 3$, then in each slot, there would be $3L$ packet collisions on an average; with $L = 2, 4, 8$, this would be 6, 12, and 24 packet collisions in each slot. If we employ QPSK pilots, with a pilot sequence length of $\lceil \log_4(L\bar{d}) \rceil$, we could potentially "pre-assign" unique sequences.

Let the pilot sequence used by the mth user be $\mathbf{p}_m \in \mathbb{C}^\tau$. If we use a QPSK constellation to generate the pilot sequences, then the entries of $\mathbf{p}_m$ lie in $\{(\pm 1 \pm j)/\sqrt{2}\}$. Let

coincidental with $\sigma^2_{\text{Shad}} = 0$.

$\Pr_{2,\tau}$ denote the probability that any 2 colliding users in a slot have picked an identical pilot sequence of length τ. We study this probability under different scenarios.

1. For a conventional mMTC system, M users transmit concurrently in the same time slot, and the BS needs to detect which subset of the $[M]$ users are active. In this case, the probability that any 2 users out of the M users pick identical pilot sequences of length τ is calculated as [53]

$$\Pr_{2,\tau} = \begin{cases} 1 - \prod_{k=1}^{M-1} \left(1 - \frac{k}{4^\tau}\right), & M \leq 4^\tau, \\ 1, & M > 4^\tau. \end{cases} \tag{5.72}$$

   The above implies that if the number of users M is greater than 4^τ, then it is not possible to assign unique pilot sequences to all the M users, and there are at least 2 users who have identical pilot sequences. For example, with $M = 1000$ users, we need a low pilot length of $\tau = \lceil \log_4(M) \rceil = 5$ symbols to ensure that unique pilot sequences are picked by the users, in the conventional mMTC setup.

2. In IRSA, the number of users transmitting in a specific slot is much lower than the total number of users in a conventional mMTC setup. If the length of the frame is T slots, the number of users is M, with the load being $L \triangleq M/T$, and the average repetition factor of the users is $\bar{d}$, then the number of users transmitting in any slot is $L\bar{d}$ on an average [7,8]. For an arbitrarily large M, the number of packet collisions in any slot is instantaneously close to $L\bar{d}$ relative to M. For ease of calculation, let us consider that $L\bar{d}$, or more precisely $\lceil L\bar{d} \rceil$, users are indeed colliding in any slot. In this case, in IRSA, the probability that any 2 users out of the $\lceil L\bar{d} \rceil$ users pick identical pilot sequences of length τ is calculated as

$$\Pr_{2,\tau} = \begin{cases} 1 - \prod_{k=1}^{\lceil L\bar{d} \rceil - 1} \left(1 - \frac{k}{4^\tau}\right), & \lceil L\bar{d} \rceil \leq 4^\tau, \\ 1, & \lceil L\bar{d} \rceil > 4^\tau. \end{cases} \tag{5.73}$$

For a frame with load $L = 3$ and average repetition factor $\bar{d} = 2$, 3, and 4, the number of users colliding in each slot are $L\bar{d} = 6$, 9, and 12, on an average, respectively. Hence a pilot length of $\tau = \lceil \log_4(L\bar{d}) \rceil = 2$, 2, and 2, respectively, is theoretically sufficient for the pilot sequences of two colliding users to not be identical in any slot. For an average repetition factor $\bar{d} = 4$, with load $L = 3$, 10, 30, 100, and 1000, the number of users colliding in each slot are $L\bar{d} = 12$, 40, 120, 400, and 4000, on an average, respectively. For this, we only need pilot sequences of length $\tau = \lceil \log_4(L\bar{d}) \rceil = 2$, 3, 4, 5, and 6, respectively, to ensure any two users do not pick the same pilot sequence in any time slot. This is applicable for any M such that $L = M/T$ is fixed to the above values. This implies that we can choose an arbitrarily high M (and correspondingly a high $T = LM$) to ensure no two users pick the same pilot sequences in IRSA with as low a pilot length as less than 10 symbols.

3. In C-IRSA, with a censor threshold ν, the effective active load of the system in any frame is $L_a = L\bar{\mathrm{F}}(\nu)$. The repetition factor generation is same in both IRSA and C-IRSA. This implies that $L\bar{\mathrm{F}}(\nu)\bar{d}$ users collide in each slot on an average in C-IRSA, compared to $L\bar{d}$ users on an average in IRSA. In this case, the probability that any 2 users out of the $\lceil L\bar{\mathrm{F}}(\nu)\bar{d} \rceil$ users pick identical pilot sequences of length τ is calculated as

$$\mathrm{Pr}_{2,\tau} = \begin{cases} 1 - \prod_{k=1}^{\lceil L\bar{\mathrm{F}}(\nu)\bar{d} \rceil - 1} \left(1 - \frac{k}{4^\tau}\right), & \lceil L\bar{\mathrm{F}}(\nu)\bar{d} \rceil \leq 4^\tau, \\ 1, & \lceil L\bar{\mathrm{F}}(\nu)\bar{d} \rceil > 4^\tau. \end{cases} \tag{5.74}$$

Since $\bar{\mathrm{F}}(\nu) \leq 1$, the number of users colliding in each slot in C-IRSA is always lesser than in IRSA, i.e., $L\bar{\mathrm{F}}(\nu)\bar{d} \leq L\bar{d}$. For a frame with load $L = 3$, $N = 4$ antennas, average repetition factor $\bar{d} = 2$, and a censor threshold of $\nu = 0$, 1, 2, 3, and 4, the number of users colliding in each slot are $\lceil L\bar{\mathrm{F}}(\nu)\bar{d} \rceil = 6$, 6, 6, 4, and 3, on an average, respectively. Hence, a pilot length of $\tau = \lceil \log_4(L\bar{\mathrm{F}}(\nu)\bar{d}) \rceil = 2$, 2, 2, 1 and 1, respectively, is *theoretically* sufficient for the pilot sequences of two colliding users to

not be identical in any slot. For an average repetition factor $\bar{d} = 4$, $N = 4$ antennas, and a censor threshold of $\nu = 3$, with load $L = 3$, 10, 30, 100, and 1000, the number of users colliding in each slot are $\lceil L\bar{\mathrm{F}}(\nu)\bar{d} \rceil = 8$, 26, 78, 259, and 2589, on an average, respectively. Here we only need pilot sequences of length $\tau = \lceil \log_4(L\bar{d}) \rceil = 2$, 3, 4, 5, and 6, respectively, to ensure any two users do not pick the same pilot sequence in any time slot. This is applicable for any M such that $L = M/T$ is fixed to the above values. This implies that we can choose an arbitrarily high M (and correspondingly a high $T = LM$) to ensure no two users pick the same pilot sequences in C-IRSA with as low a pilot length as less than 10 symbols.

4. When users use random Gaussian pilots, the pilot sequences are almost surely unique. This is true for any mMTC system, IRSA system or C-IRSA system. Since the pilot symbols in a Gaussian pilot sequence have an infinite support, the probability that two users pick identical pilot sequences is almost surely zero, i.e.,

$$\mathrm{Pr}_{2,\tau} = \text{Probability that any 2 users pick the same sequence} = 0. \tag{5.75}$$

5.9.10 Impact of Different Pilot Sequences

So far, we have only talked about the uniqueness of the pilot sequences. In practice, the UAD performance depends not just on the uniqueness of the pilot sequences, but also depends on other properties of the pilots. We have demonstrated in Chapter 4, that the length of the pilot sequence is the most critical factor in deciding the UAD performance of the protocol. The number of antennas is the next critical factor. There are other properties such as low cross correlation [117] or low mutual coherence [116] among the pilots, that can be used to design sequences that are as "orthogonal" as possible [119], and thereby achieve the best possible performance. Examples of sequences with low mutual coherence are Zadoff-Chu sequences [117] or orthogonal pilot reuse [63, 118], where users reuse the pilots amongst a pool of orthogonal sequences. We now study the performance of IRSA and C-IRSA with these pilot sequences.

In 5.29, we plot the throughput of C-IRSA for different choices of the pilot sequences.

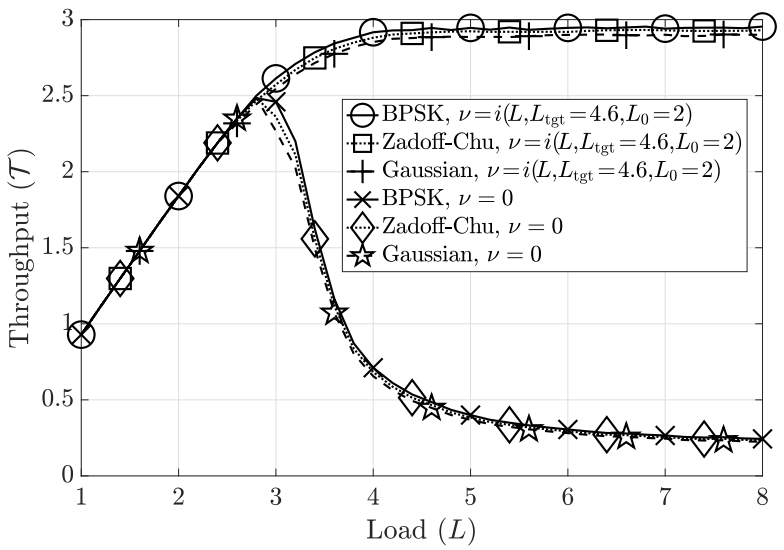


Figure 5.29: Throughput of C-IRSA with different pilot sequences.

The non-orthogonal pilots, labeled as BPSK and QPSK, contain random pilot symbols belonging to the respective PSK constellations, and Zadoff-Chu (ZC) sequences are generated according to [115]. ZC sequences require prime τ; we use $\tau = 7$. We plot the performance using random BPSK pilots of length $\tau = 10$, ZC pilots of length $\tau = 11$, and Gaussian pilots of length $\tau = 10$ as described earlier. The curves for $\nu = 0$ are nearly identical across the varied choice of the pilots; the curves for $\nu = i(L, L_{\text{tgt}} = 4.6, L_0 = 2)$ are nearly identical across the varied choice of the pilots. The throughput of C-IRSA remains at its highest beyond the inflection load for each of the curves. These figures indicate that as long as we choose such low coherence pilots, the specific choice of the pilots is not that integral to the performance of the UAD algorithm or to the performance of the protocol itself.

Part 4

Chapter 6

Analysis of IRSA in Multi-Cell and Cell-Free Systems

6.1 Introduction

Massive machine-type communications (mMTC) require random access protocols that serve a massive numbers of users [3, 4]. One such protocol is irregular repetition slotted aloha (IRSA), a successive interference cancellation (SIC) aided protocol, in which users transmit multiple packet replicas in randomly chosen resource blocks (RBs) [7]. Channel estimation in IRSA is accomplished using training or pilot sequences transmitted by the users at the start of their packets. Assigning mutually orthogonal pilots to users avoids pilot contamination, but is prohibitive in mMTC, since the pilot overhead would be proportional to the total number of users [9]. Thus, *pilot contamination (PC)*, which reduces the accuracy of channel estimation and makes the estimates correlated [10], is unavoidable in mMTC, and significantly degrades the throughput of IRSA. PC is caused by both within-cell and out-of-cell users, termed intra-cell PC and inter-cell PC, respectively. The goal of this chapter is to analyze the performance of IRSA, accounting for both intra-cell PC and inter-cell PC.

Cell-free (CF) architectures have been proposed for expanding the coverage of communication systems [20]. In a typical CF system, instead of conventional BSs deployed at

the centers of cells and serving only the users within the cell, several access points (APs) are used to jointly and cooperatively serve the users [21]. These APs are spread across the entire region of interest where users have to be served [132]. The APs are connected to a central processing unit (CPU) which is responsible for data aggregation and network coordination [22]. mMTC has the goal of increased connectivity and packet success rates. This is especially challenging to achieve when there are several cell-edge users who may not be decoded in mMTC due to high path losses. Further, these devices are expected to consume low power and have long battery lives, because of which they cannot transmit at high powers to compensate for the path loss. CF architectures naturally overcome this issue by exploiting the *macro-diversity gain* (MDG), which helps decode these edge users [23]. Thus, mMTC scenarios are a natural application for using CF architectures, and studying IRSA for mMTC in a CF setup is another key goal of this chapter.

Since the received signal power at the BS decays rapidly with the propagation distance, the users that are close to an AP (i.e., in the cell center) will experience a higher signal-to-noise ratio (SNR) than those that are close to the cell-edge. Further, the users at the cell edge are also affected by interference from neighbouring BSs, thus, the signal-to-interference-plus-noise ratio (SINR) can be substantially lower than the SNR at these locations. Thus, there is a huge disparity in the throughputs achieved by the cell edge users compared to the users located closer towards the cell center. This issue of path loss is still present in a massive MIMO (mMIMO) system, wherein the BSs are equipped with a massive number of antennas that aid interference suppression. Cell-edge users still face the same issue of high path loss, and especially in mMTC, they cannot transmit at higher powers to compensate for the path loss in order to increase the battery life. This is another reason why we study CF systems for mMTC, which help in decoding these users who would otherwise not be decoded in conventional small cell systems.

The contents of this chapter are published in part for the multi-cell setup in a conference paper in IEEE SPAWC in 2022 [24], and a journal paper for the cell-free setup is under preparation [25].

6.1.1 Existing Works in IRSA

Initial studies on IRSA focused on MAC [7] and path loss channels [8]. IRSA has been analyzed in a single-cell (SC) setup, accounting for intra-cell PC, estimation errors, path loss, and MIMO fading [11, 12]. Multi-user interference from users within the same cell is termed intra-cell interference and from users across cells is termed inter-cell interference. In the SC setup, only intra-cell interference affects the decoding of users since users do not face inter-cell interference. In practice, multiple BSs are deployed to cover a large region, and thus inter-cell interference is inevitable [100]. Furthermore, MC processing (e.g., MC MMSE combining of signals) schemes can achieve better performance compared to SC processing, since it accounts for inter-cell interference [49].

6.1.2 Existing Works in CF mMIMO

Analysis of CF mMIMO systems is a topic that has recently received intense research attention. We review only the existing works that are relevant to this chapter.

In CF systems, the APs are connected to the CPU via a fronthaul link [20]. As opposed to a conventional massive MIMO system, where a single BS has a large number of antennas to serve users, a CF system has multiple APs that are geographically spread with fewer antennas per AP [116]. Collocated mMIMO architectures, where all service antennas are located in a compact area, have the advantage of low backhaul requirements. In contrast, in distributed mMIMO systems, the antennas are spread out over a large area. The CPU is connected to the core network via backhaul links, whereas the fronthaul links between the APs and the CPUs is used for sharing received signals, CSI, and phase reference signals [132]. Further, CF systems have smaller variations in the SNR compared to SC massive MIMO systems, and have an advantage of joint processing at the CPU [21].

Scalability is an important aspect of CF mMIMO systems [133]. Scalability here is defined as the ability to add more APs and users to the system without having to increase the capabilities of existing APs [134]. Cellular systems achieve scalability via a small cell approach. Different levels of receiver cooperation levels have been studied in CF systems [133].

Densification is another topic of interest in small cell networks [135]. Interference-limitation of communication systems due to path loss is typically handled by using the divide-and-conquer approach, where the network is split up into small cells [22]. This reduces the effect of path loss and improves the performance. Existing works have considered varied schemes in CF systems and compared it with small cell networks [22]. In conventional cellular architectures, for a given spatial antenna density, a single cell massive MIMO system usually performs poorly compared with the small cell setup due to a high degree of path loss. For the same spacial antenna density, CF architectures achieve a huge gain in performance as described previously due to cooperative processing. In general, densifying always helps improve the performance of the system, but it is not clear if this relation still holds for IRSA in MC or CF mMTC systems.

6.1.3 Existing Works in GFRA for CF Systems

Very few works have analyzed GFRA protocols for the CF setup. Most of the existing works are for UAD in CF setups. Authors in [136] have proposed UAD algorithms for GFRA in a CF mMIMO setup. Distributed algorithms for UAD in CF mMIMO have been recently proposed [137]. CF systems have shown to have a better UAD performance compared to collocated MIMO [138]. Authors in [139] have explored cooperative and non-cooperative ML and MAP detection of users. Approximate message passing has been used to jointly detect the users and estimate their channels in a CF IoT setup [140]. A strongest-user collision resolution protocol has been proposed for CF IoT [141].

None of the above works analyze IRSA in the MC or CF setup, which forms the main novelty of our work.

6.1.4 Contributions

The main contributions in this chapter are as follows:

1. We first analyse IRSA in the MC setup. We derive the channel estimates and the SINR in MC IRSA accounting for path loss, MIMO fading, intra-cell pilot contamination (PC), and inter-cell PC.

2. We next analyze IRSA in the CF setup, with all of the above non-idealities. Specifically, we study three CF schemes for IRSA: one with local processing at each AP, termed local-cell-free (LCF); next with fully centralized processing at the CPU, termed centralized-cell-free (CCF); and finally, with hybrid processing at both the APs and the CPU, termed hybrid-cell-free (HCF). These schemes have different levels of partial signal processing at the APs and the CPU.

3. We provide insights into the effect of system parameters such as number of antennas, number of APs (or BSs), pilot length, and SNR on the throughput of MC IRSA and CF IRSA. We study the effect of BS and AP densification in MC and LCF IRSA, respectively, where we observe an inverse behaviour in the throughput compared to CCF and HCF IRSA. Specifically, densification improves the performance of CCF and HCF IRSA, whereas the performance of MC and LCF IRSA deteriorate.

To the best of our knowledge, no existing work has analyzed the effect of MC interference or CF processing in IRSA. Through numerical simulations, we show that inter-cell PC and inter-cell interference result in up to 70% loss in the MC throughput compared to the SC setup. Further, MC IRSA requires a significantly higher training length (about $4-5\times$ compared to SC IRSA), in order to support the same user density and achieve the same throughput. Under the CF architecture, we show that we can achieve more than $14\times$ improvement in the throughput or around $9\times$ reduction in the pilot length of CCF IRSA compared to a massive MIMO SC setup at high loads. We also study the densification trends in MC IRSA and CF IRSA: for CCF IRSA and HCF IRSA, densification always improves the performance; for LCF IRSA and MC IRSA, densification does not help at loads near the inflection loads, i.e., it is better not to densify and to operate with a massive MIMO SC setup.

Structure of this Chapter

We now study IRSA in the multi-cell setup in Section 6.2. The goal of this section is to contrast it with the SC setup and discuss the shortcomings of ignoring inter-cell effects.

Thus, we analyze the performance of the cell at the center of a multi-cell setup, as seen in Fig. 6.1. Hence, the notation for the MC setup will be established on a per-cell basis, and the notation for the necessary signals would be associated with an index denoting the cell to which the user belongs to as well as the cell whose BS has received the corresponding signal. Later, we study IRSA in the cell-free setup under three different schemes in Section 6.3. The goal of this section is to contrast CF with small cell systems and massive MIMO single cell systems. For an apt comparison among them, we keep a fixed spatial antenna density across all the configurations; also, we evaluate the network throughput, i.e., the performance of the entire system, as seen in Fig. 6.2. Further, there is no cell boundary, and thus, the notation for each signal is associated with the receiver AP index and no index for the association of a user with an AP is present.

Notation

The symbols a, $\mathbf{a}$, $\mathbf{A}$, $[\mathbf{A}]_{i,:}$, $[\mathbf{A}]_{:,j}$, $\mathbf{0}_N$, $\mathbf{1}_N$, and $\mathbf{I}_N$ denote a scalar, a vector, a matrix, the ith row of $\mathbf{A}$, the jth column of $\mathbf{A}$, all-zero vector of length N, all ones vector of length N, and an identity matrix of size $N \times N$, respectively. $[\mathbf{a}]_{\mathcal{S}}$ and $[\mathbf{A}]_{:,\mathcal{S}}$ denote the elements of $\mathbf{a}$ and the columns of $\mathbf{A}$ indexed by the set $\mathcal{S}$ respectively. $\mathrm{diag}(\mathbf{a})$ is a diagonal matrix with diagonal entries given by $\mathbf{a}$. The set of real and complex matrices of size $N \times M$ are denoted as $\mathbb{R}^{N\times M}$ and $\mathbb{C}^{N\times M}$. $\mathcal{N}(\mathbf{a}, \mathbf{A})$ and $\mathcal{CN}(\mathbf{a}, \mathbf{A})$ denote the real and complex Gaussian distribution, respectively, with mean $\mathbf{a}$ and covariance $\mathbf{A}$. $[N]$ denotes the set $\{1, 2, \ldots, N\}$. $|\cdot|$, $\|\cdot\|$, $[\cdot]^T$, $[\cdot]^*$, $[\cdot]^H$, $\mathbb{E}[\cdot]$, and $\mathbb{E}_{\mathbf{a}}[\cdot]$ denote the magnitude (or cardinality of a set), ℓ_2 norm, transpose, conjugate, hermitian, expectation, and the expectation conditioned on $\mathbf{a}$, respectively. The superscript $\mathtt{p}$ is used as a descriptive superscript in association with a symbol that is related to the *pilots*. All the other superscripts (or subscripts) that have not been defined as above are indices.

6.2 Multi-Cell IRSA

6.2.1 System Model

We consider an uplink MC system with Q cells, where each cell has an N-antenna BS located at its center as seen in Fig. 6.1. We refer to the BS at the center of the qth cell as the qth BS. Every cell has M single antenna users arbitrarily deployed within the cell who wish to communicate with their own BS.[1] The time-frequency resource is divided into T RBs. These T RBs are common to all the cells, and thus, a total of QM users contend over the T RBs. Each user randomly accesses a subset of the available RBs according to the IRSA protocol, and transmits packet replicas in the chosen RBs. Each replica comprises of a header containing pilot symbols for channel estimation, and a payload containing data and decoding error detection symbols.

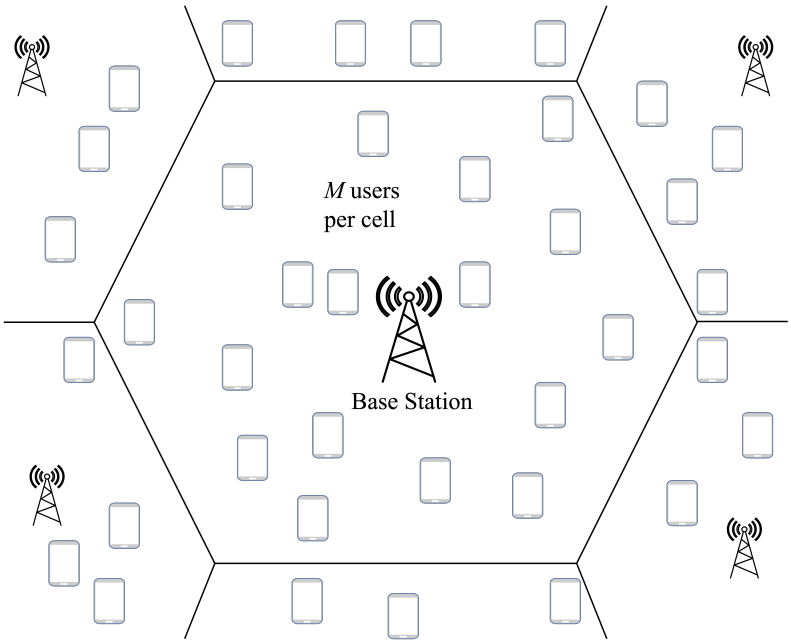


Figure 6.1: A typical uplink MC system with Q cells.

The access of the RBs by the users can be represented by an access pattern matrix $\mathbf{G} = [\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_Q] \in \{0,1\}^{T \times QM}$. Here $\mathbf{G}_j \in \{0,1\}^{T \times M}$ represents the access pattern matrix of the users in the jth cell, and $g_{tji} = [\mathbf{G}_j]_{ti}$ is the access coefficient such that $g_{tji} = 1$ if the ith user in the jth cell transmits in the tth RB, and $g_{tji} = 0$ otherwise. The ith user

[1]For apt comparison with SC IRSA, we consider M users per cell in the MC setup, which will later help us compare the performance of an IRSA system with M users with and without inter-cell effects.

in the jth cell samples its repetition factor d_{ji} from a preset probability distribution. It then chooses d_{ji} RBs from the T RBs uniformly at random for transmission. The access pattern matrix is known at the BS, which is made possible by using pseudo-random matrices generated from a seed that is available at the BS and the users [11]. This can be done in an offline fashion.

The received signal at any BS in the tth RB is a superposition of the packets transmitted by the users who choose to transmit in the tth RB, across all cells. In the pilot phase, the ith user in the jth cell transmits a pilot $\mathbf{p}_{ji} \in \mathbb{C}^{\tau}$ in all the RBs that it has chosen to transmit in, where τ denotes the length of the pilot sequence. The received pilot signal at the qth BS in the tth RB, denoted by $\mathbf{Y}^{\mathtt{p}}_{tq} \in \mathbb{C}^{N\times\tau}$, is

$$\mathbf{Y}^{\mathtt{p}}_{tq} = \sum\nolimits_{j=1}^{Q} \sum\nolimits_{i=1}^{M} g_{tji}\mathbf{h}^{q}_{tji}\mathbf{p}^{H}_{ji} + \mathbf{N}^{\mathtt{p}}_{tq}, \tag{6.1}$$

where $\mathbf{N}^{\mathtt{p}}_{tq} \in \mathbb{C}^{N\times\tau}$ is the additive complex white Gaussian noise at the qth BS with $[\mathbf{N}^{\mathtt{p}}_{tq}]_{nr} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0)\ \forall\ n \in [N],\ r \in [\tau]$ and $t \in [T]$, and N_0 is the noise variance. Here, $\mathbf{h}^{q}_{tji} \in \mathbb{C}^{N}$ is the uplink channel vector between the ith user in the jth cell and the qth BS on the tth RB. The fading is modeled as block-fading, quasi-static and Rayleigh distributed. The uplink channel is distributed as $\mathbf{h}^{q}_{tji} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_N, \beta^{q}_{ji}\sigma^2_{\mathtt{h}}\mathbf{I}_N),\ \forall\, t \in [T],\ i \in [M]$ and $j \in [Q]$, where $\sigma^2_{\mathtt{h}}$ is the fading variance, and β^{q}_{ji} is the path loss coefficient between the ith user in the jth cell and the qth BS.

In the data phase, the received data signal at the qth BS in the tth RB is denoted by $\mathbf{y}_{tq} \in \mathbb{C}^{N}$ and is given by

$$\mathbf{y}_{tq} = \sum\nolimits_{j=1}^{Q} \sum\nolimits_{i=1}^{M} g_{tji}\mathbf{h}^{q}_{tji}x_{ji} + \mathbf{n}_{tq}, \tag{6.2}$$

where x_{ji} is a data symbol with $\mathbb{E}[x_{ji}] = 0$ and $\mathbb{E}[|x_{ji}|^2] = p_{ji}$, i.e., with transmit power p_{ji}, and $\mathbf{n}_{tq} \in \mathbb{C}^{N}$ is the complex additive white Gaussian noise at the BS, with $[\mathbf{n}_{tq}]_n \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0),\ \forall\ n \in [N]$ and $t \in [T]$.

6.2.1.1 Decoding Process in MC IRSA

The decoding of a packet is abstracted into an signal to interference plus noise ratio (SINR) threshold model. Here, if the SINR of a packet in a given RB in any decoding iteration exceeds a threshold γ_{th}, then the packet can be decoded correctly [8, 15].

We now describe the performance evaluation of IRSA via the SINR threshold model. In each cell, the BS computes channel estimates and the SINRs of all users in all RBs. If it finds a user with SINR $\geq \gamma_{\text{th}}$ in some RB, it marks that user's packet as decoded, and performs SIC from all RBs in which the same user has transmitted a replica. This process of estimation and decoding is carried out iteratively. Decoding stops when no more users are decoded in two successive iterations. The throughput is calculated as the number of correctly decoded packets divided by the number of RBs.

Power Control

To ensure fairness among users within each cell, each user performs path loss inversion with respect to the BS in its own cell [142]. That is, the ith user in the jth cell transmits its symbol x_{ji} at a power p_{ji}, i.e., $\mathbb{E}[|x_{ji}|^2] = p_{ji}$, according to $p_{ji} = P/\beta_{ji}^{j}$, where P is a design parameter. The same power control policy is used in the pilot phase where the transmit power of the ith user in the jth cell is $p_{ji}^{\mathtt{p}} = P^{\mathtt{p}}/\beta_{ji}^{j}$, and $P^{\mathtt{p}} \geq P$ is a design parameter, with $\|\mathbf{p}_{ji}\|^2 = \tau p_{ji}^{\mathtt{p}}$. This ensures a uniform SNR at the BS across all users, with the pilot SNR being $P^{\mathtt{p}}\sigma_{\mathtt{h}}^2/N_0$ and the data SNR being $\rho_0 = P\sigma_{\mathtt{h}}^2/N_0$. This ensures the power disparity between cell edge users and users located near the BS is reduced, thus ensuring fairness [142].

6.2.2 Channel Estimation

Channel estimation is performed based on the received pilot signal in each cell. The signals and the channel estimates are indexed by the decoding iteration k, since they are recomputed in every decoding iteration of the SIC-based decoder. We denote the set of users in the jth cell who have not yet been decoded up to the kth decoding iteration by $\mathcal{S}_{kj}$. For some $m \in \mathcal{S}_{kj}$, let $\mathcal{S}_{kj}^{m} \triangleq \mathcal{S}_{kj} \setminus \{m\}$, with $\mathcal{S}_{1j} = [M]$. Let the set of all cell indices

be denoted by $\mathcal{Q} \triangleq \{1, 2, \ldots, Q\}$, and let $\mathcal{Q}^q \triangleq \mathcal{Q} \setminus \{q\}$. The received pilot signal at the qth BS in the tth RB in the kth decoding iteration is given by

$$\mathbf{Y}_{tq}^{\mathtt{p}k} = \sum\nolimits_{i \in \mathcal{S}_{kq}} g_{tqi} \mathbf{h}_{tqi}^{q} \mathbf{p}_{qi}^{H} + \sum\nolimits_{j \in \mathcal{Q}^q} \sum\nolimits_{i \in \mathcal{S}_{1j}} g_{tji} \mathbf{h}_{tji}^{q} \mathbf{p}_{ji}^{H} + \mathbf{N}_{tq}^{\mathtt{p}}, \tag{6.3}$$

where the first term contains signals from users within the qth cell who have not yet been decoded up to the kth decoding iteration, i.e., $\forall i \in \mathcal{S}_{kq}$. The second term contains signals from all users outside the qth cell, i.e., from every $i \in \mathcal{S}_{1j}, \forall j \in \mathcal{Q}^q$. We note that there is no coordination among BSs, and thus, all the users outside the qth cell do not get decoded by the qth BS, and they permanently interfere with the decoding of users in other cells, across all the decoding iterations.

Let $\mathcal{G}_{tq} \triangleq \{i \in \mathcal{S}_{1q} | g_{tqi} = 1\}$ denote the set of users within the qth cell who have transmitted in the tth RB, with $M_{tq} = |\mathcal{G}_{tq}|$. We denote the set of users in the qth cell who have transmitted on the tth RB but have not yet been decoded up to the kth decoding iteration by $\mathcal{M}_{tq}^{qk} \triangleq \mathcal{G}_{tq} \cap \mathcal{S}_{kq}$, with $M_{tq}^{qk} \triangleq |\mathcal{M}_{tq}^{qk}|$. Let $\mathbf{H}_{tj}^{q} \triangleq [\mathbf{h}_{tj1}^{q}, \mathbf{h}_{tj2}^{q}, \ldots \mathbf{h}_{tjM}^{q}]$ contain the uplink channels between all the users in the jth cell and the qth BS in the tth RB, with $\mathbf{H}_{tq}^{qk} \triangleq [\mathbf{H}_{tq}^{q}]_{:, \mathcal{M}_{tq}^{qk}}$ and $\mathbf{H}_{tj}^{qk} \triangleq [\mathbf{H}_{tj}^{q}]_{:, \mathcal{G}_{tj}}, \forall j \in \mathcal{Q}^q$. Let $\mathbf{P}_j \triangleq [\mathbf{p}_{j1}, \mathbf{p}_{j2}, \ldots, \mathbf{p}_{jM}]$ contain the pilots of all users within the jth cell, with $\mathbf{P}_{tq}^{qk} \triangleq [\mathbf{P}_q]_{:, \mathcal{M}_{tq}^{qk}}$ and $\mathbf{P}_{tj}^{qk} \triangleq [\mathbf{P}_j]_{:, \mathcal{G}_{tj}}, \forall j \in \mathcal{Q}^q$. Let $\mathbf{B}_j^q \triangleq \sigma_{\mathtt{h}}^2 \mathrm{diag}(\beta_{j1}^q, \beta_{j2}^q, \ldots, \beta_{jM}^q)$ contain the path loss coefficients between the users within the jth cell and the qth BS, with $\mathbf{B}_{tq}^{qk} \triangleq [\mathbf{B}_q^q]_{:, \mathcal{M}_{tq}^{qk}}$ and $\mathbf{B}_{tj}^{qk} \triangleq [\mathbf{B}_j^q]_{:, \mathcal{G}_{tj}}, \forall j \in \mathcal{Q}^q$. Thus, the received pilot signal from (6.3) can be written as

$$\mathbf{Y}_{tq}^{\mathtt{p}k} = \bar{\mathbf{H}}_{tq}^{qk} \bar{\mathbf{P}}_{tq}^{kH} + \mathbf{N}_{tq}^{\mathtt{p}} = \mathbf{H}_{tq}^{qk} \mathbf{P}_{tq}^{qkH} + \sum\nolimits_{j \in \mathcal{Q}^q} \mathbf{H}_{tj}^{qk} \mathbf{P}_{tj}^{qkH} + \mathbf{N}_{tq}^{\mathtt{p}},$$

where $\bar{\mathbf{H}}_{tq}^{qk} \triangleq [\mathbf{H}_{tq}^{qk}, \mathbf{H}_{t1}^{qk}, \ldots, \mathbf{H}_{tq-1}^{qk}, \mathbf{H}_{tq+1}^{qk}, \ldots, \mathbf{H}_{tQ}^{qk}] \in \mathbb{C}^{N \times \bar{M}_{tq}^{qk}}$, with $\bar{M}_{tq}^{qk} \triangleq M_{tq}^{qk} + \sum_{j \in \mathcal{Q}^q} M_{tj}$, and $\bar{\mathbf{P}}_{tq}^{k} \triangleq [\mathbf{P}_{tq}^{qk}, \mathbf{P}_{t1}^{qk}, \ldots, \mathbf{P}_{tq-1}^{qk}, \mathbf{P}_{tq+1}^{qk}, \ldots, \mathbf{P}_{tQ}^{qk}] \in \mathbb{C}^{\tau \times \bar{M}_{tq}^{qk}}$. We define $\bar{\mathbf{B}}_{tq}^{qk} \triangleq [\mathbf{B}_{tq}^{qk}, \mathbf{B}_{t1}^{qk}, \ldots, \mathbf{B}_{tq-1}^{qk}, \mathbf{B}_{tq+1}^{qk}, \ldots, \mathbf{B}_{tQ}^{qk}] \in \mathbb{C}^{\bar{M}_{tq}^{qk} \times \bar{M}_{tq}^{qk}}$ to derive the channel estimates. Let $\bar{\mathbf{C}}_t^{qk} \triangleq \bar{\mathbf{P}}_{tq}^{k} \bar{\mathbf{B}}_{tq}^{qk} (\bar{\mathbf{P}}_{tq}^{kH} \bar{\mathbf{P}}_{tq}^{k} \bar{\mathbf{B}}_{tq}^{qk} + N_0 \mathbf{I}_{\bar{M}_{tq}^{qk}})^{-1}$; be split as $\bar{\mathbf{C}}_t^{qk} = [\mathbf{C}_{tq}^{qk}, \mathbf{C}_{t1}^{qk}, \ldots, \mathbf{C}_{tq-1}^{qk}, \mathbf{C}_{tq+1}^{qk}, \ldots, \mathbf{C}_{tQ}^{qk}]$, and $\mathbf{c}_{tji}^{qk} \triangleq [\mathbf{C}_{tj}^{qk}]_{:,i}$.

Lemma 7. *The minimum mean squared error (MMSE) channel estimate* $\hat{\bar{\mathbf{H}}}_{tq}^{qk}$ *of* $\bar{\mathbf{H}}_{tq}^{qk}$ *in the* t*th RB in the* k*th decoding iteration at the* q*th BS can be calculated as*

$$\hat{\bar{\mathbf{H}}}_{tq}^{qk} = \mathbf{Y}_{tq}^{\mathrm{p}k}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk}(\bar{\mathbf{P}}_{tq}^{kH}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk} + N_0\mathbf{I}_{\bar{M}_{tq}^{qk}})^{-1}. \tag{6.4}$$

Further, the estimation error $\tilde{\mathbf{h}}_{tji}^{qk} \triangleq \hat{\mathbf{h}}_{tji}^{qk} - \mathbf{h}_{tji}^{q}$ *is distributed as* $\tilde{\mathbf{h}}_{tji}^{qk} \sim \mathcal{CN}(\mathbf{0}_N, \delta_{tji}^{qk}\mathbf{I}_N)$*, where* δ_{tji}^{qk} *is calculated as*

$$\delta_{tji}^{qk} = \beta_{ji}^{q}\sigma_h^2 \left(\frac{\begin{array}{l} N_0\|\mathbf{c}_{tji}^{qk}\|^2 + \sum_{n\in\mathcal{S}_{kj}^{i}} |\mathbf{p}_{qn}^{H}\mathbf{c}_{tji}^{qk}|^2 g_{tqn}\beta_{qn}^{q}\sigma_h^2 \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}} |\mathbf{p}_{ln}^{H}\mathbf{c}_{tji}^{qk}|^2 g_{tln}\beta_{ln}^{q}\sigma_h^2 \end{array}}{\begin{array}{l} N_0\|\mathbf{c}_{tji}^{qk}\|^2 + \sum_{n\in\mathcal{S}_{kj}} |\mathbf{p}_{qn}^{H}\mathbf{c}_{tji}^{qk}|^2 g_{tqn}\beta_{qn}^{q}\sigma_h^2 \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}} |\mathbf{p}_{ln}^{H}\mathbf{c}_{tji}^{qk}|^2 g_{tln}\beta_{ln}^{q}\sigma_h^2 \end{array}} \right).$$

Proof. See Sec. 6.6.1. □

Remark 1: The estimate $\hat{\bar{\mathbf{H}}}_{tq}^{qk}$ can also be calculated as $\hat{\bar{\mathbf{H}}}_{tq}^{qk} = \mathbf{Y}_{tq}^{\mathrm{p}k}(\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk}\bar{\mathbf{P}}_{tq}^{kH} + N_0\mathbf{I}_\tau)^{-1}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk}$, (a $\tau \times \tau$ inverse.) Lemma 7 is applicable for any choice of (possibly non-orthogonal) pilots. We now discuss the case where pilots are reused by users within and across cells.

6.2.2.1 Pilot Reuse

Channel estimation is done based on a pilot codebook $\{\boldsymbol{\phi}_i\}_{i=1}^{\tau}$ of τ orthogonal pilots [142], with each $\boldsymbol{\phi}_i \in \mathbb{C}^\tau$, such that $\boldsymbol{\phi}_i^H\boldsymbol{\phi}_j = 0, \forall i \neq j$, and $\|\boldsymbol{\phi}_i\|^2 = \tau P^{\mathrm{p}}$. Here P^{p} is the pilot power, and the pilot codebook is the same across all cells. Each user uses a pilot from this codebook, and thus, many users share the same pilot sequence, possibly, both within the cell and out of the cell, leading to pilot contamination. Since $\tau < M$, both intra-cell PC and inter-cell PC occur.

Let $\mathcal{P}_{ji}$ denote the set of users that reuse the pilot of the ith user in the jth cell. With this codebook, the channel estimate is distributed as $\hat{\mathbf{h}}_{tji}^{qk} \sim \mathcal{CN}(\mathbf{0}_N, \varsigma_{tji}^{qk}\mathbf{I}_N)$, where

$\varsigma_{tji}^{qk} = \dfrac{\tau P^{\mathtt{p}} g_{tji} \beta_{ji}^{q2} \sigma_{\mathtt{h}}^4}{\left(N_0 + \sum_{n \in \mathcal{S}_{kj} \cap \mathcal{P}_{ji}} \tau P^{\mathtt{p}} g_{tqn} \beta_{qn}^{q} \sigma_{\mathtt{h}}^2 + \sum_{l \in \mathcal{Q}^q} \sum_{n \in \mathcal{S}_{1j} \cap \mathcal{P}_{li}} \tau P^{\mathtt{p}} g_{tln} \beta_{ln}^{q} \sigma_{\mathtt{h}}^2\right)}$, and the estimation error variance is calculated as $\delta_{tji}^{qk} = \beta_{ji}^{q} \sigma_{\mathtt{h}}^2 - \varsigma_{tji}^{qk}$.

6.2.3 Data Decoding in MC IRSA

Let ρ_{tqm}^{k} denote the SINR of the mth user in the qth cell at the qth BS in the tth RB in the kth decoding iteration. Similar to (6.2), the received data signal at the qth BS in the tth RB in the kth decoding iteration is given by

$$\mathbf{y}_{tq}^{k} = \sum\nolimits_{i \in \mathcal{S}_{kq}} g_{tqi} \mathbf{h}_{tqi}^{q} x_{qi} + \sum\nolimits_{j \in \mathcal{Q}^q} \sum\nolimits_{i \in \mathcal{S}_{1j}} g_{tji} \mathbf{h}_{tji}^{q} x_{ji} + \mathbf{n}_{tq}. \tag{6.5}$$

We use a generic combining vector $\mathbf{a}_{tqm}^{k} \in \mathbb{C}^N$ to obtain the post-combined data signal $\tilde{y}_{tqm}^{k} = \mathbf{a}_{tqm}^{kH} \mathbf{y}_{tq}^{k}$ as

$$\begin{aligned} \tilde{y}_{tqm}^{k} = {} & g_{tqm} x_{qm} \mathbf{a}_{tqm}^{kH} \hat{\mathbf{h}}_{tqm}^{qk} - g_{tqm} x_{qm} \mathbf{a}_{tqm}^{kH} \tilde{\mathbf{h}}_{tqm}^{qk} + \sum\nolimits_{i \in \mathcal{S}_{kq}^{m}} g_{tqi} x_{qi} \mathbf{a}_{tqm}^{kH} \mathbf{h}_{tqi}^{q} \\ & + \sum\nolimits_{j \in \mathcal{Q}^q} \sum\nolimits_{i \in \mathcal{S}_{1j}} g_{tji} x_{ji} \mathbf{a}_{tqm}^{kH} \mathbf{h}_{tji}^{q} + \mathbf{a}_{tqm}^{kH} \mathbf{n}_{tq}, \end{aligned} \tag{6.6}$$

where $\tilde{\mathbf{h}}_{tqm}^{qk}$ is as defined in Lemma 7. This combined signal, used to decode the mth user in the qth cell, is composed of five terms. The first term $g_{tqm} x_{qm} \mathbf{a}_{tqm}^{kH} \hat{\mathbf{h}}_{tqm}^{qk}$ is the useful signal component of the mth user; the term $g_{tqm} x_{qm} \mathbf{a}_{tqm}^{kH} \tilde{\mathbf{h}}_{tqm}^{qk}$ arises due to the estimation error $\tilde{\mathbf{h}}_{tqm}^{k}$; the term $\sum_{i \in \mathcal{S}_{kq}^{m}} g_{tqi} x_{qi} \mathbf{a}_{tqm}^{kH} \mathbf{h}_{tqi}^{q}$ represents the intra-cell interference from the users within the qth cell who have transmitted in the tth RB and have not yet been decoded up to the kth decoding iteration; the term $\sum_{j \in \mathcal{Q}^q} \sum_{i \in \mathcal{S}_{1j}} g_{tji} x_{ji} \mathbf{a}_{tqm}^{kH} \mathbf{h}_{tji}^{q}$ models the inter-cell interference from users outside the qth cell; and the last term $\mathbf{a}_{tqm}^{kH} \mathbf{n}_{tq}$ is the additive noise. We now present the SINR for all the users.

Theorem 6.1 ▶ SINR in Multi-Cell IRSA.

The signal to interference plus noise ratio (SINR) achieved by the mth user within the qth cell at the qth BS in the tth RB and the kth decoding iteration can be

written as

$$\rho_{tqm}^{k} = \frac{\mathtt{Gain}_{tqm}^{k}}{N_0 + \mathtt{InCI}_{tqm}^{k} + \mathtt{Est}_{tqm}^{k} + \mathtt{ICI}_{tqm}^{k}}, \forall m \in \mathcal{S}_{kq}, \tag{6.7}$$

where

$$\begin{aligned}
\mathtt{Gain}_{tqm}^{k} &\triangleq p_{qm} g_{tqm} |\mathbf{a}_{tqm}^{kH} \hat{\mathbf{h}}_{tqm}^{qk}|^2 / \|\mathbf{a}_{tqm}^{k}\|^2, \\
\mathtt{InCI}_{tqm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{S}_{kq}^{m}} p_{qi} g_{tqi} |\mathbf{a}_{tqm}^{kH} \hat{\mathbf{h}}_{tqi}^{qk}|^2 / \|\mathbf{a}_{tqm}^{k}\|^2, \\
\mathtt{Est}_{tqm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{S}_{kq}} p_{qi} g_{tqi} \delta_{tqi}^{qk} + \sum_{j \in \mathcal{Q}^q} \sum_{i \in \mathcal{S}_{1j}} p_{ji} g_{tji} \delta_{tji}^{qk}, \\
\mathtt{ICI}_{tqm}^{k} &\triangleq \textstyle\sum_{j \in \mathcal{Q}^q} \sum_{i \in \mathcal{S}_{1j}} p_{ji} g_{tji} |\mathbf{a}_{tqm}^{kH} \hat{\mathbf{h}}_{tji}^{qk}|^2 / \|\mathbf{a}_{tqm}^{k}\|^2.
\end{aligned}$$

The channel estimates $\hat{\mathbf{h}}_{tji}^{qk}$ and the error variances δ_{tji}^{qk} in the above expressions are obtained from Lemma 7.

Proof. See Sec. 6.6.2. □

Remark 2: The SINR derived in Theorem 6.1 holds for any choice of the combining vector $\mathbf{a}_{tqm}^{k}$, the pilots, and the power control policy. The first M_{tq}^{qk} columns of the combining matrix $\mathbf{A}_{tq}^{k} \in \mathbb{C}^{N \times \bar{M}_{tq}^{qk}}$ is used at the qth BS to decode the M_{tq}^{qk} users within the qth cell who have not yet been decoded up to the kth decoding iteration in the tth RB. The SINR in (6.7) is maximized by multi-cell MMSE combining [49], under which the optimal combining matrix can be evaluated as

$$\begin{aligned}
\mathbf{A}_{tq}^{k} &= \hat{\bar{\mathbf{H}}}_{tq}^{qk} \bar{\mathbf{D}}_{tq,\mathtt{p1}}^{qk} (\bar{e}_{tq}^{k} \mathbf{I}_{\bar{M}_{tq}^{qk}} + \hat{\bar{\mathbf{H}}}_{tq}^{qH} \hat{\bar{\mathbf{H}}}_{tq}^{qk} \bar{\mathbf{D}}_{tq,\mathtt{p1}}^{qk}),^{-1} \\
&= (\bar{e}_{tq}^{k} \mathbf{I}_N + \hat{\bar{\mathbf{H}}}_{tq}^{qk} \bar{\mathbf{D}}_{tq,\mathtt{p1}}^{qk} \hat{\bar{\mathbf{H}}}_{tq}^{qH})^{-1} \hat{\bar{\mathbf{H}}}_{tq}^{qk} \bar{\mathbf{D}}_{tq,\mathtt{p1}}^{qk},
\end{aligned}$$

where $\bar{e}_{tq}^{k} \triangleq N_0 + \sum_{i \in \mathcal{S}_{kq}} p_{qi} g_{tqi} \delta_{tqi}^{qk} + \sum_{j \in \mathcal{Q}^q} \sum_{i \in \mathcal{S}_{1j}} p_{ji} g_{tji} \delta_{tji}^{qk}$, $\mathbf{D}_{j,\mathtt{p1}} \triangleq \mathrm{diag}(p_{j1}, p_{j2}, \ldots, p_{jM})$ contains the power coefficients of the users within the jth cell, $\mathbf{D}_{tj,\mathtt{p1}}^{qk} \triangleq [\mathbf{D}_{j,\mathtt{p1}}]_{:,\mathcal{G}_{tj}}, \forall j \in \mathcal{Q}^q$, $\mathbf{D}_{tq,\mathtt{p1}}^{qk} \triangleq [\mathbf{D}_{q,\mathtt{p1}}]_{:,\mathcal{M}_{tq}^{qk}}$, and $\bar{\mathbf{D}}_{tq,\mathtt{p1}}^{qk} \triangleq [\mathbf{D}_{tq,\mathtt{p1}}^{qk}, \mathbf{D}_{t1,\mathtt{p1}}^{qk}, \ldots, \mathbf{D}_{tq-1,\mathtt{p1}}^{qk}, \mathbf{D}_{tq+1,\mathtt{p1}}^{qk}, \ldots, \mathbf{D}_{tQ,\mathtt{p1}}^{qk}] \in \mathbb{C}^{\bar{M}_{tq}^{qk} \times \bar{M}_{tq}^{qk}}$. We note that the above MC processing outperforms the application of SC

processing applied to the MC setup [49].

Algorithm 6: Performance Evaluation of MC IRSA in the qth cell

Input: $\tau, N, T, M, N_0, \{\mathbf{G}_j\}_{j=1}^{Q}, \{\mathbf{P}_j\}_{j=1}^{Q}, \{\mathbf{Y}_{tq}^{\mathsf{p}}\}_{t=1}^{T}, \{\mathbf{y}_{tq}\}_{t=1}^{T}, k_{\max}$

Initialize: $\mathcal{S}_{1q} = [M]$, $\mathcal{G}_{tj} = \{i \in [M] | g_{tji} = 1\}$, $\bar{\mathcal{S}}_q = \{\}$

for $k = 1, 2, \ldots, k_{\max}$ **do**

  for $t = 1, 2, \ldots, T$ **do**

    Find $M_{tq}^{qk} = |\mathcal{G}_{tq} \cap \mathcal{S}_{kq}|$, $\mathbf{P}_{tq}^{qk} = [\mathbf{P}_q]_{:,\mathcal{G}_{tq} \cap \mathcal{S}_{kq}}$, $\{\mathbf{P}_{tj}^{qk} = [\mathbf{P}_j]_{:,\mathcal{G}_{tj}}\}_{j \neq q}$, $\mathbf{Y}_{tq}^{\mathsf{p}k}, \mathbf{y}_{tq}^{k}$

    Compute $\hat{\mathbf{h}}_{tji}^{qk}, \ \forall i \in \mathcal{S}_{kq}$ via Lemma 7

    Evaluate the SINR $\rho_{tqi}^{k}, \ \forall i \in \mathcal{S}_{kq}$ via Theorem 6.1

    If $\rho_{tqi}^{k} \geq \gamma_{\text{th}}$, remove user i from $\mathcal{S}_{kq}$ and perform IC in all RBs where $g_{tqi} = 1$

    Add user i to set of decoded users: $\bar{\mathcal{S}}_q = \bar{\mathcal{S}}_q \cup \{i\}$

  end

end

Output: Set of users decoded in the qth cell: $\bar{\mathcal{S}}_q$

We now evaluate the performance of MC IRSA in the qth cell using Algorithm 6. The algorithm is run till no more users are decoded in two successive iterations (or up to a maximum of $k_{\max}$ iterations). The algorithm outputs $\bar{\mathcal{S}}_q$, which is the set of users decoded in the qth cell, i.e., at the qth BS. Thus, the packet loss rate (PLR) at the qth BS can be computed as $\mathsf{PLR}_q \triangleq (1 - |\bar{\mathcal{S}}_q|/M)$, and the throughput at the qth BS can be computed as $\mathcal{T}_q \triangleq M(1 - \mathsf{PLR}_q)/T$.

6.2.3.1 Deterministic Equivalent of the SINR

We now present simple and interpretable expressions for the SINR in the massive MIMO (large N) regime, and with maximal ratio combining (MRC), i.e., $\mathbf{a}_{tqm}^{k} = \hat{\mathbf{h}}_{tqm}^{qk}$ [49].

Lemma 8. *As the number of antennas N gets large, the SINR with maximal ratio*

combining converges almost surely to

$$\overline{\rho}_{tqm}^{k} = \frac{\mathtt{Sig}_{tqm}^{k}}{\epsilon_{tqm}^{k}(N_0 + \mathtt{IntNC}_{tqm}^{k}) + \mathtt{IntC}_{tqm}^{k}}, \tag{6.8}$$

where $\mathtt{Sig}_{tqm}^{k}$ *is the desired signal,* $\mathtt{IntNC}_{tqm}^{k}$ *represents the non-coherent interference, and* $\mathtt{IntC}_{tqm}^{k}$ *represents the coherent interference. These can be evaluated as*

$$\begin{aligned}
\epsilon_{tqm}^{k} &= \begin{pmatrix} N_0\|\mathbf{c}_{tqm}^{qk}\|^2 + \sum_{n\in\mathcal{S}_{kj}} |\mathbf{p}_{qn}^{H}\mathbf{c}_{tqm}^{qk}|^2 g_{tqn}\beta_{qn}^{q}\sigma_{\mathtt{h}}^{2} \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}} |\mathbf{p}_{ln}^{H}\mathbf{c}_{tqm}^{qk}|^2 g_{tln}\beta_{ln}^{q}\sigma_{\mathtt{h}}^{2} \end{pmatrix}, \\
\mathtt{Sig}_{tqm}^{k} &= N p_{qm} g_{tqm} (\epsilon_{tqm}^{k})^2, \\
\mathtt{IntNC}_{tqm}^{k} &= \begin{pmatrix} p_{qm} g_{tqm}\delta_{tqm}^{qk} + \sum_{n\in\mathcal{S}_{kj}} p_{qn} g_{tqn}\beta_{qn}^{q}\sigma_{\mathtt{h}}^{2} \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}} p_{ln} g_{tln}\beta_{ln}^{q}\sigma_{\mathtt{h}}^{2} \end{pmatrix}, \\
\mathtt{IntC}_{tqm}^{k} &= N \begin{pmatrix} \sum_{n\in\mathcal{S}_{kj}} |\mathbf{p}_{qn}^{H}\mathbf{c}_{tqm}^{qk}|^2 p_{qn} g_{tqn}\beta_{qn}^{q2}\sigma_{\mathtt{h}}^{4} \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}} |\mathbf{p}_{ln}^{H}\mathbf{c}_{tqm}^{qk}|^2 p_{ln} g_{tln}\beta_{ln}^{q2}\sigma_{\mathtt{h}}^{4} \end{pmatrix}.
\end{aligned}$$

Proof. See Sec. 6.6.3. □

Remark 3: $\mathtt{IntNC}_{tqm}^{k}$ represents the non-coherent interference that arises due to estimation errors, intra-cell interference, and inter-cell interference. $\mathtt{IntC}_{tqm}^{k}$ is the coherent interference that arises due to intra-cell PC and inter-cell PC. The former does not scale with the number of antennas N, whereas the latter scales linearly with N. Both inter-cell PC and inter-cell interference degrade the performance of the system [100], and thus it is vital to account for both while analyzing the performance of IRSA.

6.3 Cell-Free IRSA

The term *cell-free* is used to imply that no cell boundaries exist between the users and the APs. Depending on the specific scheme used, the decoding of users can happen separately at each AP or by a subset of the APs or jointly at the CPU. With distributed processing, decoding happens separately at each AP, and any user can be decoded by any

AP if received at a strong enough power. With coordinated processing, decoding happens at the CPU, and the CPU jointly serves the users within the network, and there is no assignment of users to cells. In this work, as opposed to a *network-centric CF architecture*, where the APs are allocated disjoint sets of users to serve, we adopt the *user-centric CF architecture*, where each user could be decoded by multiple APs.[2]

6.3.1 System Model

We consider a system with M users who are located arbitrarily across the network.[3] In order to serve these users, Q APs are arbitrarily placed across the network, wherein each AP is equipped with N antennas. This amounts to a total QN number of antennas across the entire network.[4] As seen in Fig. 6.2, the APs are connected via fronthaul links to the CPU, which facilitates the AP coordination. The users contend to communicate with the Q APs across T time-frequency RBs. Each user samples their repetition factor d_i from a predefined distribution and transmits packet replicas in d_i RBs chosen uniformly at random from the T RBs. The access of the RBs is governed by $\mathbf{G}$, which is termed the access pattern matrix (APM). Here, $g_{ti} \triangleq [\mathbf{G}]_{ti} = 1$ if the ith user transmits their packet on the tth RB, and 0 otherwise. By using a common seed at the AP and the users, the APM can be generated at the AP and thus, we can assume that the AP has knowledge of $\mathbf{G}$.

The APs receive pilot and data signals as superposition of the signals from all the users. The received pilot signal at the qth AP in the tth RB, denoted by $\mathbf{Y}^{\mathtt{p}}_{tq}$ is

$$\mathbf{Y}^{\mathtt{p}}_{tq} = \textstyle\sum_{i=1}^{M} g_{ti}\mathbf{h}^{q}_{ti}\mathbf{p}_i^H + \mathbf{N}^{\mathtt{p}}_{tq}, \tag{6.9}$$

[2]In practice, clustering needs to be implemented to ensure subsets of APs jointly decode any user. This also reduces the complexity of decoding at the APs. However, to establish a baseline for CF IRSA, we do not perform any clustering in this chapter, and we leave it for future work.

[3]For apt comparison with IRSA with small cells and IRSA in the massive MIMO setup, where the goal is to study the effect on the entire system, we consider M users to be located across the entire network in the CF setup.

[4]One major deviation of the CF architecture considered in this chapter is the relation between Q, M, and N. Typical CF mMIMO architectures are for ultra-dense networks where the typical assumption $QN \gg M$, whereas in our work, we make no such assumption.

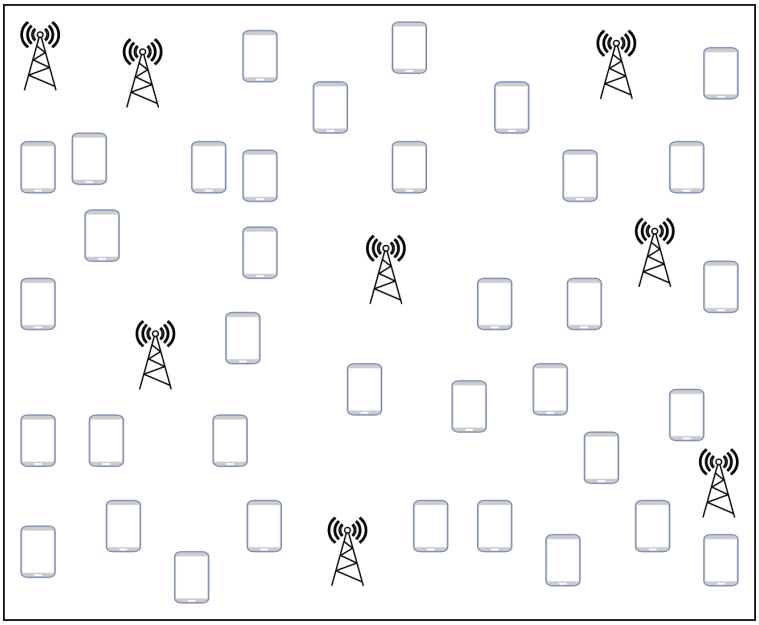


Figure 6.2: A typical uplink CF system with Q APs.

where $\mathbf{h}_{ti}^{q} \in \mathbb{C}^{N}$ is the N-length channel vector between the ith user and the qth AP on the tth RB, $\mathbf{p}_i \in \mathbb{C}^{\tau}$ is the τ-length pilot signal of the ith user, and $\mathbf{N}_{tq}^{\mathtt{p}} \in \mathbb{C}^{N \times \tau}$ is the complex additive white Gaussian noise at the qth AP in the tth RB. Here, $[\mathbf{N}_{tq}^{\mathtt{p}}]_{nr} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0)$ $\forall\ n \in [N],\ r \in [\tau]$ and $t \in [T]$, where N_0 is the noise variance. The fading is modeled as block-fading, quasi-static and Rayleigh distributed: $\mathbf{h}_{ti}^{q} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}_N, \beta_i^q \sigma_{\mathtt{h}}^2 \mathbf{I}_N),\ \forall\ t \in [T],\ i \in [M]$, where $\sigma_{\mathtt{h}}^2$ is the fading variance, and β_i^q is the path loss coefficient between the ith user and the qth AP.

We now write out the data signal corresponding to a single transmitted symbol. The received data signal in the data phase at the qth AP in the tth RB, denoted by $\mathbf{y}_{tq}$, is

$$\mathbf{y}_{tq} = \textstyle\sum_{i=1}^{M} g_{ti} \mathbf{h}_{ti}^{q} x_i + \mathbf{n}_{tq}, \tag{6.10}$$

where x_i is the data symbol of the ith user with $\mathbb{E}[x_i] = 0$ and $\mathbb{E}[|x_i|^2] = p_i$, i.e., with transmit power p_i, and $\mathbf{n}_{tq} \in \mathbb{C}^{N}$ is the complex additive white Gaussian noise at the qth AP, with $[\mathbf{n}_{tq}]_n \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0)$, $\forall\ n \in [N]$ and $t \in [T]$.

6.3.1.1 SINR Threshold Model

Similar to the MC and SC setups, we now use the signal to interference plus noise ratio (SINR) threshold model to abstract the decoding of users' packets. Here, if the SINR of a packet in a given RB in any decoding iteration exceeds a threshold γ_{th}, then the packet can be decoded correctly [8, 15]. Depending on the specific CF architecture under consideration, the decoding process may vary. In the next subsection, we describe the decoding process for each scheme separately.

6.3.2 Local-Processing in Cell-Free IRSA: LCF IRSA

The notation introduced earlier is now used to analyze IRSA in the cell-free scenario with local AP processing as depicted in Fig. 6.3. We use the acronym LCF IRSA to denote the above: Local Cell-Free processing in IRSA. In LCF, each AP performs channel estimation and data decoding for as many users as possible. Then, the APs forward only the data symbols of the successfully decoded users to the CPU on the fronthaul links. The CPU is not responsible for decoding any user. LCF resembles a multi-cell setup but without cell boundaries; LCF is similar to a cooperative MIMO setup [143].

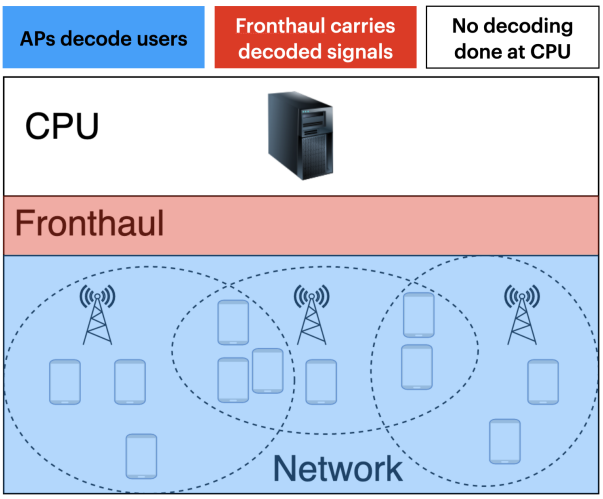


Figure 6.3: Setup of LCF IRSA.

The access pattern structure of IRSA adds a layer to the decoding of users in LCF IRSA. In SC IRSA, we saw that users in singleton slots could be decoded even if they

were far away from the BS and two collided users who were both near to the BS need not be decodable due to low SINR. In MC IRSA, users closer towards the cell edge could not be decoded due to inter-cell interference from adjacent cell's edge users. In LCF IRSA, these cell boundaries are removed. In slots where multiple users collide, users may not be decodable at one AP, but they could be decodable at a different AP (either in a singleton slot or after other users have been decoded). Thus, we allow each AP to decode as many users as possible, even possibly users which are geographically located far away.

LCF IRSA is somewhat similar to a network-centric implementation of coordinated multi-point [21], in which multiple BSs coordinate to decode users. In LCF IRSA, however, there is no coordination or any data exchange amongst the APs. Note that LCF IRSA somewhat resembles *Level 1* receiver cooperation that is considered in many existing works [21, 133]. A major difference is that each user is decoded only by the nearest AP in Level 1 CF systems, whereas in CCF IRSA, due to the access pattern structure of IRSA, users can get decoded at far away APs as well.

6.3.2.1 Decoding Process in LCF IRSA

We now describe the decoding process in LCF IRSA. Firstly, each AP computes the channel estimates and the SINRs of all users in all RBs. If an AP successfully decodes a user in some RB, it performs SIC from all the RBs in which that user has transmitted packets in. This decoding is abstracted into the SINR model as seen before: if the AP finds a user with SINR $\geq \gamma_{\text{th}}$ in some RB, it marks that user's packet as decoded. This process of estimation and data decoding is carried out iteratively. Decoding stops when no more users are decoded in two successive iterations. The throughput of the network is calculated as the total number of correctly decoded unique packets across all the APs divided by the number of RBs. The above definition includes "unique" since users packets could be decoded at multiple APs.

It is possible that some users near to one AP could be decoded at other APs, but we do not consider any inter-AP coordination in LCF IRSA, and thus, the same users could get decoded at multiple APs (depending on the decodability of the users). In order to

reduce complexity, we can allocate the users to each AP via clustering. Here, the qth AP attempts to decode only $\mathcal{A}_q$, which is the set of users allocated to the qth AP. The notation defined previously is still applicable, we simply initialize the decoding process with $\mathcal{S}_{1q} = \mathcal{A}_q$.

6.3.2.2 Performance Analysis of LCF IRSA

In this subsection, we evaluate the performance of LCF IRSA. Channel estimation is performed based on the received pilot signal at each AP. Since the data signals are processed iteratively for each user, the received pilot and data signals are indexed by the decoding iteration. Let k denote the current decoding iteration index, and $\mathcal{S}_{kq}$ denote the set of users not yet decoded up to the kth decoding iteration at the qth AP, with $\mathcal{S}_{1q} = [M]$ and $\mathcal{S}_{kq}^{i} \triangleq \mathcal{S}_{kq} \setminus \{i\}$. The received pilot signal at the qth AP in the tth RB in the kth decoding iteration is given by

$$\mathbf{Y}_{tq}^{\mathtt{p}k} = \textstyle\sum_{i \in \mathcal{S}_{kq}} g_{ti} \mathbf{h}_{ti}^{q} \mathbf{p}_{i}^{H} + \mathbf{N}_{tq}^{\mathtt{p}}, \tag{6.11}$$

where the first term contains signals from users who have not yet been decoded up to the kth decoding iteration at the qth AP, i.e., $\forall i \in \mathcal{S}_{kq}$. Similarly, we can find the received data signal at the qth AP in the tth RB in the kth decoding iteration as

$$\mathbf{y}_{tq}^{k} = \textstyle\sum_{i \in \mathcal{S}_{kq}} g_{ti} \mathbf{h}_{ti}^{q} x_i + \mathbf{n}_{tq}. \tag{6.12}$$

Let $\mathcal{G}_t \triangleq \{i \in [M] | g_{ti} = 1\}$ be the set of users who have transmitted in the tth RB. Let $M_{tq}^{k} \triangleq |\mathcal{G}_t \cap \mathcal{S}_{kq}|$ be the number of users who have transmitted in the tth RB and have not been decoded in the first $k-1$ iterations at the qth AP. Let us stack the channels of the M_{tq}^{k} users as the columns of the matrix $\mathbf{H}_{tq}^{k} \in \mathbb{C}^{N \times M_{tq}^{k}}$, and let $\mathbf{P}_{tq}^{k} \in \mathbb{C}^{\tau \times M_{tq}^{k}}$ denote a matrix that contains the pilot sequences of the M_{tq}^{k} users as its columns. Let $\mathbf{B}_{tq}^{k} \triangleq \sigma_{\mathtt{h}}^{2} \mathrm{diag}(\beta_{i_1}^{q}, \beta_{i_2}^{q}, \ldots, \beta_{i_{M_{tq}^{k}}}^{q})$ be a diagonal matrix that contains the path loss coefficients of the M_{tq}^{k} users, with $\mathcal{G}_t \cap \mathcal{S}_{kq} \triangleq \{i_1, i_2, \ldots, i_{M_{tq}^{k}}\}$. Hence, the received pilot signal from (6.11) can be written as $\mathbf{Y}_{tq}^{\mathtt{p}k} = \mathbf{H}_{tq}^{k} \mathbf{P}_{tq}^{kH} + \mathbf{N}_{tq}^{\mathtt{p}}$. The estimate of $\mathbf{H}_{tq}^{k}$ from the above is

presented in Theorem 6.2.

Let ρ_{tqm}^{k} denote the SINR of the mth user in the tth RB in the kth decoding iteration at the qth AP. Using a combining vector $\mathbf{a}_{tm}^{qk}$, the qth AP obtains the post-combined data signal $\tilde{y}_{tm}^{qk}$ as $\tilde{y}_{tm}^{qk} \triangleq \mathbf{a}_{tm}^{qkH}\mathbf{y}_{tq}^{k}$, where $\mathbf{y}_{tq}^{k}$ is from (6.12). Thus, we obtain the post-combined data signal as

$$\tilde{y}_{tm}^{qk} = \mathbf{a}_{tm}^{qkH}\hat{\mathbf{h}}_{tm}^{qk}g_{tm}x_m - \mathbf{a}_{tm}^{qkH}\tilde{\mathbf{h}}_{tm}^{qk}g_{tm}x_m + \mathbf{a}_{tm}^{qkH}\textstyle\sum_{i\in\mathcal{S}_{kq}^{m}} g_{ti}\mathbf{h}_{ti}^{q}x_i + \mathbf{a}_{tm}^{qkH}\mathbf{n}_{tq}. \tag{6.13}$$

Here the first term on the RHS, $\mathbf{a}_{tm}^{qkH}\hat{\mathbf{h}}_{tm}^{qk}g_{tm}x_m$, denotes the useful signal component and it contains the channel estimate $\hat{\mathbf{h}}_{tm}^{qk}$ of the mth user; the second term $\mathbf{a}_{tm}^{qkH}\tilde{\mathbf{h}}_{tm}^{qk}g_{tm}x_m$ contains $\tilde{\mathbf{h}}_{tm}^{qk}$, the estimation error of the mth user; the third term $\mathbf{a}_{tm}^{qkH}\sum_{i\in\mathcal{S}_{kq}^{m}} g_{ti}\mathbf{h}_{ti}^{q}x_i$ is the multi-user interference faced by the mth user due to the other users in the entire network who have not yet been decoded up to the kth decoding iteration at the qth AP; and the fourth term $\mathbf{a}_{tm}^{qkH}\mathbf{n}_{tq}$ is the additive noise component. We need to compute the SINR from the above post-combined data signal. For this purpose, let $\mathbf{C}_{t}^{qk} \triangleq \mathbf{P}_{tq}^{k}\mathbf{B}_{tq}^{k}(\mathbf{P}_{tq}^{kH}\mathbf{P}_{tq}^{k}\mathbf{B}_{tq}^{k} + N_0\mathbf{I}_{M_{tq}^{k}})^{-1}$, and $\mathbf{c}_{ti}^{qk} \triangleq [\mathbf{C}_{t}^{qk}]_{:,i}$.

We now present the channel estimates of the users and the SINR achieved by the users at the qth AP in LCF IRSA in the following theorem.

Theorem 6.2 ▶ Performance Analysis of LCF IRSA.

The minimum mean squared error (MMSE) channel estimate $\hat{\mathbf{H}}_{tq}^{k}$ of $\mathbf{H}_{tq}^{k}$ in the tth RB in the kth decoding iteration at the qth AP can be calculated as

$$\hat{\mathbf{H}}_{tq}^{k} = \mathbf{Y}_{tq}^{\mathrm{p}k}(\mathbf{P}_{tq}^{k}\mathbf{B}_{tq}^{k}\mathbf{P}_{tq}^{kH} + N_0\mathbf{I}_{\tau})^{-1}\mathbf{P}_{tq}^{k}\mathbf{B}_{tq}^{k}. \tag{6.14}$$

Specifically, the estimate of the channel $\mathbf{h}_{ti}^{q}$ of the ith user to the qth AP is calculated as $\hat{\mathbf{h}}_{ti}^{qk} = [\hat{\mathbf{H}}_{tq}^{k}]_{:,i}$. Further, the estimation error $\tilde{\mathbf{h}}_{ti}^{qk} \triangleq \hat{\mathbf{h}}_{ti}^{qk} - \mathbf{h}_{ti}^{q}$ is distributed as

$\tilde{\mathbf{h}}_{ti}^{qk} \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^{qk}\mathbf{I}_N)$, where δ_{ti}^{qk} is calculated as

$$\delta_{ti}^{qk} = \beta_i^q \sigma_{\mathtt{h}}^2 \left(\frac{N_0 \|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n \in \mathcal{S}_{kq}^i} |\mathbf{p}_n^H \mathbf{c}_{ti}^{qk}|^2 g_{tn} \beta_n^q \sigma_{\mathtt{h}}^2}{N_0 \|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n \in \mathcal{S}_{kq}} |\mathbf{p}_n^H \mathbf{c}_{ti}^{qk}|^2 g_{tn} \beta_n^q \sigma_{\mathtt{h}}^2} \right). \tag{6.15}$$

The signal to interference plus noise ratio (SINR) achieved by the mth user at the qth AP in the tth RB and the kth decoding iteration can be written as

$$\rho_{tqm}^k = \frac{\mathtt{Gain}_{tqm}^k}{N_0 + \mathtt{Est}_{tqm}^k + \mathtt{MUI}_{tqm}^k}, \forall m \in \mathcal{S}_{kq}, \tag{6.16}$$

where

$$\begin{aligned} \mathtt{Gain}_{tqm}^k &\triangleq p_m g_{tm} |\mathbf{a}_{tm}^{qkH} \hat{\mathbf{h}}_{tm}^{qk}|^2 / \|\mathbf{a}_{tm}^{qk}\|^2, \\ \mathtt{Est}_{tqm}^k &\triangleq \textstyle\sum_{i \in \mathcal{S}_{kq}} p_i g_{ti} \delta_{ti}^{qk}, \\ \mathtt{MUI}_{tqm}^k &\triangleq \textstyle\sum_{i \in \mathcal{S}_{kq}^m} p_i g_{ti} |\mathbf{a}_{tm}^{qkH} \hat{\mathbf{h}}_{ti}^{qk}|^2 / \|\mathbf{a}_{tm}^{qk}\|^2. \end{aligned}$$

Proof. See Sec. 6.6.4. □

Remark 4: The MMSE channel estimate $\hat{\mathbf{H}}_{tq}^k$ of $\mathbf{H}_{tq}^k$ can be computed as

$$\hat{\mathbf{H}}_{tq}^k = \mathbf{Y}_{tq}^{\mathtt{p}k} (\mathbf{P}_{tq}^k \mathbf{B}_{tq}^k \mathbf{P}_{tq}^{kH} + N_0 \mathbf{I}_\tau)^{-1} \mathbf{P}_{tq}^k \mathbf{B}_{tq}^k, \tag{6.17a}$$

$$\stackrel{(a)}{=} \mathbf{Y}_{tq}^{\mathtt{p}k} \mathbf{P}_{tq}^k \mathbf{B}_{tq}^k (\mathbf{P}_{tq}^{kH} \mathbf{P}_{tq}^k \mathbf{B}_{tq}^k + N_0 \mathbf{I}_{M_{tq}^k})^{-1}, \tag{6.17b}$$

where (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$. Here, the estimate can be calculated via an inverse of either a $\tau \times \tau$ matrix or an $M_{tq}^k \times M_{tq}^k$ matrix as required.

Remark 5: The results derived in Theorem 6.2 holds for any choice of the combining vector $\mathbf{a}_{tqm}^k$, the pilots $\{\mathbf{p}_m\}$, and the power control policy $\{p_m\}$. The combining vector that maximizes the SINR in (6.7) is the MMSE combiner, which can be found as $\mathbf{a}_{tm}^{qk} = [\mathbf{A}_{tq}^k]_{:,m}$, where

$$\mathbf{A}_{tq}^k = \hat{\mathbf{H}}_{tq}^k \mathbf{D}_{t,\mathtt{p1}}^{qk} (e_{tq}^k \mathbf{I}_{M_{tq}^k} + \hat{\mathbf{H}}_{tq}^{kH} \hat{\mathbf{H}}_{tq}^k \mathbf{D}_{t,\mathtt{p1}}^{qk})^{-1}, \tag{6.18a}$$

$$= (e_{tq}^{k}\mathbf{I}_N + \hat{\mathbf{H}}_{tq}^{k}\mathbf{D}_{t,\mathtt{p1}}^{qk}\hat{\mathbf{H}}_{tq}^{kH})^{-1}\hat{\mathbf{H}}_{tq}^{k}\mathbf{D}_{t,\mathtt{p1}}^{qk}, \tag{6.18b}$$

where $\mathbf{D}_{t,\mathtt{p1}}^{qk} \triangleq \mathrm{diag}(p_{i_1}, p_{i_2}, \ldots, p_{i_{M_{tq}^k}})$ contains the power coefficients of the M_{tq}^k users, and $e_{tq}^k \triangleq N_0 + \sum_{i\in\mathcal{S}_{kq}} p_i g_{ti}\delta_{ti}^{qk}$.

Algorithm 7: Performance Evaluation of LCF IRSA at the qth AP

Input: $\tau, N, T, M, N_0, \mathbf{G}, \mathbf{P}, \{\mathbf{Y}_{tq}^{\mathtt{p}}\}_{t=1}^{T}, \{\mathbf{y}_{tq}\}_{t=1}^{T}, k_{\max}$

Initialize: $\mathcal{S}_{1q} = [M]$, $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$, $\bar{\mathcal{S}}_q = \{\}$

for $k = 1, 2, \ldots, k_{\max}$ **do**

  for $t = 1, 2, \ldots, T$ **do**

    Find $M_{tq}^k = |\mathcal{G}_t \cap \mathcal{S}_{kq}|$, $\mathbf{P}_{tq}^k = [\mathbf{P}]_{:,\mathcal{G}_t\cap\mathcal{S}_{kq}}$, $\mathbf{Y}_{tq}^{\mathtt{p}k}, \mathbf{y}_{tq}^k$

    Compute $\hat{\mathbf{h}}_{ti}^{qk},\ \forall i \in \mathcal{S}_{kq}$ via Theorem 6.2

    Evaluate the SINR $\rho_{tqi}^k,\ \forall i \in \mathcal{S}_{kq}$ via Theorem 6.2

    If $\rho_{tqi}^k \geq \gamma_{\mathrm{th}}$, remove user i from $\mathcal{S}_{kq}$ and perform IC in all RBs where $g_{ti} = 1$

    Add user i to set of decoded users: $\bar{\mathcal{S}}_q = \bar{\mathcal{S}}_q \cup \{i\}$

  end

end

Output: Set of users decoded at the qth AP: $\bar{\mathcal{S}}_q$

We now evaluate the performance of LCF IRSA at the qth AP using Algorithm 7. The algorithm is run till no more users are decoded in two successive iterations (or up to a maximum of $k_{\max}$ iterations). The algorithm outputs $\bar{\mathcal{S}}_q$, which is the set of users decoded at the qth AP. To compute the set of users decoded at all the APs, we construct the set $\bar{\mathcal{S}} \triangleq \cup_{q=1}^{Q}\bar{\mathcal{S}}_q$. Thus, the packet loss rate (PLR) of the network can be computed as $\mathsf{PLR} \triangleq (1 - |\bar{\mathcal{S}}|/M)$, and the throughput can be computed as $\mathcal{T} \triangleq M(1-\mathsf{PLR})/T$.

6.3.3 Centralized-Processing in Cell-Free IRSA: CCF IRSA

We now analyze IRSA in the cell-free scenario with fully centralized CPU processing as depicted in Fig. 6.4. We use the acronym CCF IRSA to denote the above: Centralized

Cell-Free processing in IRSA. In CCF IRSA, each AP blindly forwards its received signals to the CPU, on the fronthaul, which is responsible for performing data decoding. The qth AP forwards the received pilot $\{\mathbf{Y}^{\mathsf{p}}_{tq}, t \in [T]\}$ from (6.9) and the received data $\{\mathbf{y}_{tq}, t \in [T]\}$ from (6.10) to the CPU, which attempts to decode users who are transmitting their data packets in an IRSA fashion. Note that CCF IRSA resembles *Level 4* receiver cooperation that is considered in many existing works [21, 133].

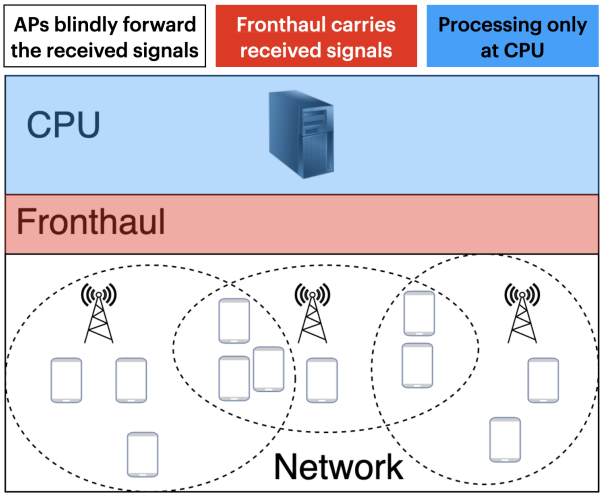


Figure 6.4: Setup of CCF IRSA.

In the previous subsection, we saw that the access pattern structure of IRSA adds a layer to the decoding of users in LCF IRSA. For CCF IRSA, the gain is primarily because the effect of path loss is not pronounced. The access pattern structure affects the decoding of CCF IRSA in the exact same way as SC IRSA: in singleton slots, users can be decoded even if they are far away and in slots with collisions, users need not be decodable even if they are nearby.

One major difference between the SC architecture and the CCF architecture is the presence of a virtual distributed antenna array (DAA) in the system. The CCF architecture can be viewed as a single cell mMIMO system with the CPU as the BS and the N antennas of each of the Q APs together acting as a virtual DAA with QN antennas. In a conventional mMIMO SC setup, the users signals are affected by higher path loss since the QN antennas are collocated at the BS. Further, the antenna array gain and

interference suppression capability of a SC mMIMO setup with QN antennas helps the BS decode more users. Compared to the SC mMIMO setup, the DAA setup in CCF performs better. This is because each user is closer to the nearest AP,and thus, the effect of path loss is not as pronounced. Since the distributed antenna setup in CCF contains QN antennas, the array gain and interference suppression capabilities carry over from the SC mMIMO setup. Thus, CCF is expected to perform better than a SC mMIMO setup, as we will observe in 6.4.2.

6.3.3.1 Decoding Process in CCF IRSA

We now describe the performance evaluation of CCF IRSA via the SINR threshold model. Each AP forwards the received pilot and data signals from all RBs to the CPU. The CPU computes the channel estimates of all the users and the corresponding SINR in all RBs. If the CPU successfully decodes a user in some RB, it performs SIC from all the RBs in which that user has transmitted packets in. This decoding is abstracted into the SINR model as seen before: if the CPU finds a user with SINR $\geq \gamma_{\text{th}}$ in some RB, it marks that user's packet as decoded. This process of channel estimation and data decoding is carried out iteratively at the CPU. Decoding stops when no more users are decoded in two successive iterations. The throughput of the network is calculated as the total number of correctly decoded packets divided by the number of RBs.

6.3.3.2 Performance Analysis of CCF IRSA

In this subsection, we evaluate the performance of CCF IRSA. In the channel estimation phase, the CPU computes channel estimates of all the users to the qth AP similar to (6.14) and then finally combines them to obtain an effective CPU channel estimate. Let k denote the current CPU decoding iteration index, and $\mathcal{S}_k$ denote the set of users not yet decoded at the CPU after $k-1$ decoding iterations, with $\mathcal{S}_1 = [M]$, and $\mathcal{S}_k^i \triangleq \mathcal{S}_k \setminus \{i\}$. At the CPU, the received pilot signal sent by the qth AP in the tth RB in the kth decoding

iteration is given by

$$\mathbf{Y}_{tq}^{\mathtt{p}k} = \sum_{i\in\mathcal{S}_k} g_{ti}\mathbf{h}_{ti}^{q}\mathbf{p}_i^H + \mathbf{N}_{tq}^{\mathtt{p}}. \tag{6.19}$$

where the first term contains signals from users who have not yet been decoded up to the kth decoding iteration at the CPU, i.e., $\forall i \in \mathcal{S}_k$. Note that in the above, unlike (6.11), we have dropped the AP index for the set of undecoded users.

We recall that $\mathcal{G}_t \triangleq \{i \in [M] | g_{ti} = 1\}$ is the set of users who have transmitted in the tth RB. Let $\mathcal{M}_t^k \triangleq \mathcal{G}_t \cap \mathcal{S}_k$, and $M_t^k \triangleq |\mathcal{M}_t^k|$ be the number of users yet to be decoded, $\mathbf{P}_t^k$ contain as its columns the pilots of the M_t^k users yet to be decoded, and $\mathbf{B}_{tq}^k \triangleq \sigma_{\mathtt{h}}^2 \mathrm{diag}(\beta_{i_1}^q, \beta_{i_2}^q, \ldots, \beta_{i_{M_t^k}}^q)$ contain the path loss coefficients of the M_t^k users to the qth AP, with $\mathcal{M}_t^k \triangleq \{i_1, i_2, \ldots, i_{M_t^k}\}$. Let us stack the channels of the M_t^k users to the qth AP as the columns of the matrix $\mathbf{H}_{tq}^k \in \mathbb{C}^{N\times M_t^k}$, and let $\mathbf{P}_t^k \in \mathbb{C}^{\tau\times M_t^k}$ denote a matrix that contains the pilot sequences of the M_t^k users as its columns. The channel of the ith user in the tth slot to the CPU can be computed by stacking the user-AP channels as $\mathbf{h}_{ti} = [\mathbf{h}_{ti}^{1H}, \mathbf{h}_{ti}^{2H}, \ldots, \mathbf{h}_{ti}^{QH}]^H \in \mathbb{C}^{QN\times 1}$. The CPU channel estimation error is found as $\tilde{\mathbf{h}}_{ti}^k \triangleq \hat{\mathbf{h}}_{ti}^k - \mathbf{h}_{ti}$. The estimate of $\mathbf{h}_{ti}$ is presented in Theorem 6.3.

Let ρ_{tm}^k denote the SINR of the mth user in the tth RB in the kth decoding iteration at the CPU. The received data signal can be stacked as $\mathbf{y}_t = [\mathbf{y}_{t1}^H, \mathbf{y}_{t2}^H, \ldots, \mathbf{y}_{tQ}^H]$. Similar to (6.19), in the kth decoding iteration at the CPU, the stacked received data signal in the tth RB at the CPU can be expressed as

$$\mathbf{y}_t^k = \sum_{i\in\mathcal{S}_k} g_{ti}\mathbf{h}_{ti}x_i + \mathbf{n}_t, \tag{6.20}$$

where $\mathbf{n}_t \triangleq [\mathbf{n}_{t1}^H, \mathbf{n}_{t2}^H, \ldots, \mathbf{n}_{tQ}^H]^H$. The CPU uses a combining vector $\mathbf{a}_{tm}^k$ to combine the received data signal. Thus, we obtain the post-combined data signal as $\tilde{y}_{tm}^k \triangleq \mathbf{a}_{tm}^{kH}\mathbf{y}_t^k$ which is expanded as

$$\tilde{y}_{tm}^k = \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k g_{tm}x_m - \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^k g_{tm}x_m + \mathbf{a}_{tm}^{kH}\sum_{i\in\mathcal{S}_k^m} g_{ti}\mathbf{h}_{ti}x_i + \mathbf{a}_{tm}^{kH}\mathbf{n}_t. \tag{6.21}$$

The first term on the RHS, $\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}g_{tm}x_m$, denotes the useful signal component and it contains the CPU channel estimate $\hat{\mathbf{h}}_{tm}^{k}$ of the mth user; the second term $\mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^{k}g_{tm}x_m$ contains $\tilde{\mathbf{h}}_{tm}^{k}$, the CPU estimation error of the mth user; the third term $\mathbf{a}_{tm}^{kH}\sum_{i\in\mathcal{S}_k^m}g_{ti}\mathbf{h}_{ti}x_i$ is the multi-user interference faced by the mth user due to the other users in the network who have not yet been decoded up to the kth decoding iteration at the CPU; and the fourth term $\mathbf{a}_{tm}^{kH}\mathbf{n}_t$ is the additive noise component. We need to compute the SINR from the above post-combined data signal. For this purpose, let $\mathbf{C}_t^{qk}\triangleq\mathbf{P}_t^k\mathbf{B}_{tq}^k(\mathbf{P}_t^{kH}\mathbf{P}_t^k\mathbf{B}_{tq}^k+N_0\mathbf{I}_{M_t^k})^{-1}$, and $\mathbf{c}_{ti}^{qk}\triangleq[\mathbf{C}_t^{qk}]_{:,i}$.

We now present the channel estimates of the users' channels $\mathbf{h}_{ti}$ and the SINR achieved by the users at the CPU in CCF IRSA in the following theorem.

Theorem 6.3 ▶ Performance Analysis of CCF IRSA.

The minimum mean squared error (MMSE) channel estimate $\hat{\mathbf{H}}_{tq}^{k}$ of $\mathbf{H}_{tq}^{k}$ in the tth RB in the kth decoding iteration to the qth AP can be calculated as

$$\hat{\mathbf{H}}_{tq}^{k}=\mathbf{Y}_{tq}^{\mathrm{p}k}(\mathbf{P}_t^k\mathbf{B}_{tq}^k\mathbf{P}_t^{kH}+N_0\mathbf{I}_\tau)^{-1}\mathbf{P}_t^k\mathbf{B}_{tq}^k. \tag{6.22}$$

Specifically, the CPU estimate of the channel $\mathbf{h}_{ti}^{q}$ of the ith user to the qth AP is calculated as $\hat{\mathbf{h}}_{ti}^{qk}=[\hat{\mathbf{H}}_{tq}^{k}]_{:,i}$. For the ith user in the tth RB in the kth decoding iteration, the effective QN-length CPU channel estimate is found as

$$\hat{\mathbf{h}}_{ti}^{k}=[\hat{\mathbf{h}}_{ti}^{1kH},\hat{\mathbf{h}}_{ti}^{2kH},\ldots,\hat{\mathbf{h}}_{ti}^{QkH}]^H\in\mathbb{C}^{QN\times 1}. \tag{6.23}$$

Further, the estimation error $\tilde{\mathbf{h}}_{ti}^{qk}\triangleq\hat{\mathbf{h}}_{ti}^{qk}-\mathbf{h}_{ti}^{q}$ is distributed as $\tilde{\mathbf{h}}_{ti}^{qk}\sim\mathcal{CN}(\mathbf{0}_N,\delta_{ti}^{qk}\mathbf{I}_N)$, where δ_{ti}^{qk} is calculated as

$$\delta_{ti}^{qk}=\beta_i^q\sigma_{\mathtt{h}}^2\left(\frac{N_0\|\mathbf{c}_{ti}^{qk}\|^2+\sum_{n\in\mathcal{S}_{kq}^i}|\mathbf{p}_n^H\mathbf{c}_{ti}^{qk}|^2g_{tn}\beta_n^q\sigma_{\mathtt{h}}^2}{N_0\|\mathbf{c}_{ti}^{qk}\|^2+\sum_{n\in\mathcal{S}_{kq}}|\mathbf{p}_n^H\mathbf{c}_{ti}^{qk}|^2g_{tn}\beta_n^q\sigma_{\mathtt{h}}^2}\right). \tag{6.24}$$

The covariance of the channel estimation error at the CPU is

$$\Delta_{ti}^{k} \triangleq \text{blkdiag}(\delta_{ti}^{1k}\mathbf{I}_N, \delta_{ti}^{2k}\mathbf{I}_N, \ldots, \delta_{ti}^{Qk}\mathbf{I}_N). \tag{6.25}$$

The SINR achieved by the mth user at the CPU in the tth RB and the kth decoding iteration can be written as

$$\rho_{tm}^{k} = \frac{\texttt{Gain}_{tm}^{k}}{N_0 + \texttt{Est}_{tm}^{k} + \texttt{MUI}_{tm}^{k}}, \forall m \in \mathcal{S}_k, \tag{6.26}$$

where

$$\begin{aligned}
\texttt{Gain}_{tm}^{k} &\triangleq p_m g_{tm} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^{k}|^2 / \|\mathbf{a}_{tm}^{k}\|^2, \\
\texttt{Est}_{tm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{S}_k} p_i g_{ti} \mathbf{a}_{tm}^{kH} \Delta_{ti}^{k} \mathbf{a}_{tm}^{k} / \|\mathbf{a}_{tm}^{k}\|^2, \\
\texttt{MUI}_{tm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{S}_k^m} p_i g_{ti} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{ti}^{k}|^2 / \|\mathbf{a}_{tm}^{k}\|^2.
\end{aligned}$$

Proof. See Sec. 6.6.5. □

Remark 6: The MMSE channel estimate $\hat{\mathbf{H}}_{tq}^{k}$ in (6.22) can be computed as

$$\hat{\mathbf{H}}_{tq}^{k} = \mathbf{Y}_{tq}^{\mathtt{p}k} (\mathbf{P}_t^k \mathbf{B}_{tq}^k \mathbf{P}_t^{kH} + N_0 \mathbf{I}_\tau)^{-1} \mathbf{P}_t^k \mathbf{B}_{tq}^k, \tag{6.27a}$$

$$\overset{(a)}{=} \mathbf{Y}_{tq}^{\mathtt{p}k} \mathbf{P}_t^k \mathbf{B}_{tq}^k (\mathbf{P}_t^{kH} \mathbf{P}_t^k \mathbf{B}_{tq}^k + N_0 \mathbf{I}_{M_{tq}^k})^{-1}, \tag{6.27b}$$

where (a) follows from $(\mathbf{AB} + \mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA} + \mathbf{I})^{-1}$ [144]. Here, the estimate can be calculated via an inverse of either a $\tau \times \tau$ matrix or an $M_t^k \times M_t^k$ matrix as required.

Remark 7: The results derived in Theorem 6.3 holds for any choice of the combining vector $\mathbf{a}_{tm}^{k}$, the pilots $\{\mathbf{p}_m\}$, and the power control policy $\{p_m\}$. The channel estimates $\hat{\mathbf{h}}_{ti}^{k}$ are stacked into the matrix $\hat{\mathbf{H}}_t^k \in \mathbb{C}^{QN \times M_t^k}$. The combining vector that maximizes the SINR in (6.26) is the MMSE combiner, which can be found as $\mathbf{a}_{tm}^{k} = [\mathbf{A}_t^k]_{:,m}$, where

$$\mathbf{A}_t^k = (\mathbf{L}_t^k + \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p1}}^k \hat{\mathbf{H}}_t^{kH})^{-1} \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p1}}^k, \tag{6.28}$$

where $\mathbf{L}_t^k \triangleq \sum_{i\in\mathcal{S}_k} p_i g_{ti}\Delta_{ti}^k + N_0\mathbf{I}_{QN}$, and $\mathbf{D}_{t,\mathtt{p1}}^k \triangleq \mathrm{diag}(p_{i_1}, p_{i_2}, \ldots, p_{i_{M_t^k}})$ contains the power coefficients of the M_t^k users. Using the Woodbury identity [144], the combiner matrix can also be found via an $M_t^k \times M_t^k$ inverse as

$$\mathbf{A}_t^k = \mathbf{J}_t^k \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p2}}^k (\mathbf{I}_{QN} - (\mathbf{G}_t^k)^{-1}\mathbf{D}_{t,\mathtt{p2}}^k \hat{\mathbf{H}}_t^{kH} \mathbf{J}_t^k \hat{\mathbf{H}}_t^k)\mathbf{D}_{t,\mathtt{p2}}^k, \tag{6.29}$$

where $\mathbf{J}_t^k \triangleq \mathrm{blkdiag}(j_{t1}^k\mathbf{I}_N, j_{t2}^k\mathbf{I}_N, \ldots, j_{tQ}^k\mathbf{I}_N)$, $\mathbf{G}_t^k \triangleq \mathbf{I}_{U_t^k} + \mathbf{D}_{t,\mathtt{p2}}^k \hat{\mathbf{H}}_t^{kH} \mathbf{J}_t^k \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p2}}^k$, $j_{tq}^k \triangleq 1/(N_0 + \sum_{i\in\mathcal{S}_k} p_i g_{ti}\delta_{ti}^{qk})$, and $\mathbf{D}_{t,\mathtt{p2}}^k \triangleq \mathrm{diag}(\sqrt{p_{i_1}}, \sqrt{p_{i_2}}, \ldots, \sqrt{p_{i_{M_t^k}}})$.

Algorithm 8: Performance Evaluation of CCF IRSA at the CPU

Input: $\tau, N, T, M, N_0, \mathbf{G}, \mathbf{P}, \{\mathbf{Y}_{tq}^{\mathtt{p}}\}_{t=1,2,\ldots,T;q=1,2,\ldots,Q}, \{\mathbf{y}_{tq}\}_{t=1,2,\ldots,T;q=1,2,\ldots,Q}, k_{\max}$

Initialize: $\mathcal{S}_1 = [M]$, $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$, $\bar{\mathcal{S}} = \{\}$

Compute received signal at the CPU $\mathbf{y}_t = [\mathbf{y}_{t1}^H, \mathbf{y}_{t2}^H, \ldots, \mathbf{y}_{tQ}^H]$

for $k = 1, 2, \ldots, k_{\max}$ **do**

  for $t = 1, 2, \ldots, T$ **do**

    Find $M_t^k = |\mathcal{G}_t \cap \mathcal{S}_k|$, $\mathbf{P}_t^k = [\mathbf{P}]_{:,\mathcal{G}_t\cap\mathcal{S}_k}$, $\{\mathbf{Y}_{tq}^{\mathtt{p}k}\}_{q=1}^Q, \mathbf{y}_t^k$

    Compute local estimates $\hat{\mathbf{h}}_{ti}^{qk},\ \forall i \in \mathcal{S}_k, \forall q \in \{1, 2, \ldots, Q\}$ via (6.22)

    Stack local estimates to obtain CPU estimate

    $\hat{\mathbf{h}}_{ti}^k = [\hat{\mathbf{h}}_{ti}^{1kH}, \hat{\mathbf{h}}_{ti}^{2kH}, \ldots, \hat{\mathbf{h}}_{ti}^{QkH}]^H,\ \forall i \in \mathcal{S}_k$ as in Theorem 6.3

    Evaluate the SINR $\rho_{ti}^k,\ \forall i \in \mathcal{S}_k$ via Theorem 6.3

    If $\rho_{ti}^k \geq \gamma_{\mathrm{th}}$, remove user i from $\mathcal{S}_k$ and perform IC in all RBs where $g_{ti} = 1$

    Add user i to set of decoded users: $\bar{\mathcal{S}} = \bar{\mathcal{S}} \cup \{i\}$

  end

end

Output: Set of users decoded at the CPU: $\bar{\mathcal{S}}$

We now evaluate the performance of CCF IRSA at the CPU using Algorithm 8. The algorithm is run till no more users are decoded in two successive iterations (or up to a maximum of $k_{\max}$ iterations). The algorithm outputs $\bar{\mathcal{S}}$, which is the set of users decoded at the CPU. Thus, the PLR of the network can be computed as $\mathsf{PLR} \triangleq (1 - |\bar{\mathcal{S}}|/M)$, and the throughput can be computed as $\mathcal{T} \triangleq M(1 - \mathsf{PLR})/T$.

6.3.4 Hybrid-Processing in Cell-Free IRSA: HCF IRSA

We now analyze IRSA in the cell-free scenario with hybrid processing as depicted in Fig. 6.5. We use the acronym HCF IRSA to denote the above: Hybrid Cell-Free processing in IRSA. In HCF IRSA, there are two phases: the first phase is at the APs and the second phase is at the CPU. In the first phase, each AP tries to decode as many users as possible similar to LCF IRSA. Once any AP cannot decode any more users in two successive iterations, it forwards the residual pilot and data signals alongside a list of users it has already decoded to the CPU on the fronthaul. In the second phase, the CPU attempts to decode users using the combined signal made up of the residual signals from all APs. This exploits the distributed array gain and can achieve a much better performance when compared with LCF IRSA. Further, compared to CCF IRSA, HCF IRSA reduces the load on the CPU by offloading the decoding tasks to the APs. Of course, this depends on the regime (load, pilot length, SNR) in which we are operating as we will see in Sec. 6.4.4.

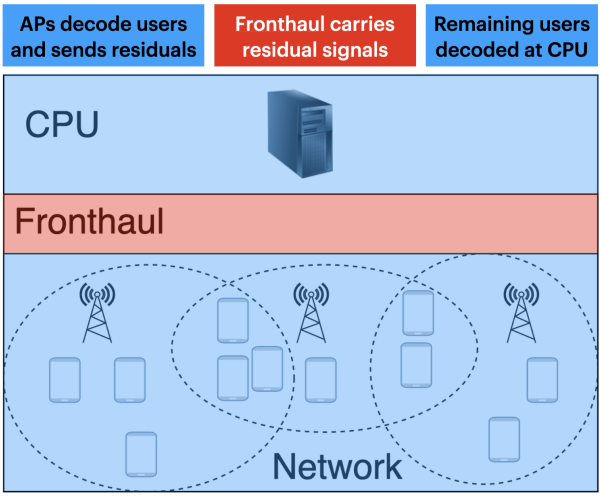


Figure 6.5: Setup of HCF IRSA.

Note that HCF IRSA does not resemble any existing receiver cooperation levels that are considered in existing works [21, 133]. Receiver cooperation designs similar to Level 2 and Level 3 can be considered for CF IRSA in future work. For example, in Level 3 receiver cooperation for CF, the APs send not the original received signal but the local estimates obtained from the post-combined signals.

6.3.4.1 Decoding Process in HCF IRSA

We now describe the performance evaluation of HCF IRSA via the SINR threshold model. HCF IRSA has two phases: phase 1 at the APs and phase 2 at the CPU. In phase 1, the decoding is similar to the decoding process of LCF IRSA: each AP iteratively computes the channel estimates and the SINRs of all users in all RBs, decodes users (using the SINR threshold model) and performs SIC. Decoding stops when no more users are decoded in two successive iterations. The APs forward the residual pilot and data signals to the CPU via the fronthaul links, alongside a list of users already decoded. This marks the end of the first phase.

In phase 2, the CPU uses the residual signals to decode the set of users not decoded at any AP. The process is similar to the decoding process of CCF IRSA: the CPU iteratively computes the channel estimates of all the users yet to be decoded at any AP and the corresponding SINR in all RBs, decodes users (using the SINR threshold model) and performs SIC. Decoding stops when no more users are decoded in two successive iterations. The throughput of the network is calculated as the total number of correctly decoded packets (at both the APs as well as the CPU) divided by the number of RBs.

6.3.4.2 Performance Analysis of HCF IRSA

In this subsection, we evaluate the performance of HCF IRSA.

Phase 1: In Phase 1, as discussed earlier, the APs decode as many users as possible similar to LCF IRSA. This has been discussed in Sec. 6.3.2.2. The same can be repeated here as well, and we skip the details. At the end of Algorithm 7, we obtain the set $\bar{\mathcal{S}}_q$, which is the set of decoded users at the qth AP at the end of the qth AP's decoding process in phase 1.

Phase 2: The set of users that have been decoded across the entire network is given by $\bar{\mathcal{S}} = \bar{\mathcal{S}}_1 \cup \bar{\mathcal{S}}_2 \cup \cdots \cup \bar{\mathcal{S}}_Q$. We now denote $\mathcal{U}_k$ as the set of users in the network not decoded up to the kth CPU decoding iteration, with $\mathcal{U}_1 = [M] \setminus \bar{\mathcal{S}}$ and $\mathcal{U}_k^i \triangleq \mathcal{U}_k \setminus \{i\}$. $\mathcal{U}_1$ contains

only the signals of the users who need to be decoded at the CPU since they have not been decoded at any of the APs.

Let the decoding at the qth AP stop at a decoding iteration k_q, i.e., it cannot decode any more users. Thus, the residual received pilot and data signals sent from the qth AP are denoted by $\mathbf{Z}_{tq}^{\mathrm{p}1} \triangleq \mathbf{Y}_{tq}^{\mathrm{p}k_q}$ and $\mathbf{z}_{tq}^{1} \triangleq \mathbf{y}_{tq}^{k_q}$. These can be expressed at the CPU as

$$\mathbf{Z}_{tq}^{\mathrm{p}1} = \textstyle\sum_{i\in\mathcal{U}_1} g_{ti}\mathbf{h}_{ti}^{q}\mathbf{p}_i^{H} + \sum_{i\in\bar{\mathcal{S}}\setminus\bar{\mathcal{S}}_q} g_{ti}\mathbf{h}_{ti}^{q}\mathbf{p}_i^{H} + \mathbf{N}_{tq}^{\mathrm{p}}, \tag{6.30}$$

$$\mathbf{z}_{tq}^{1} = \textstyle\sum_{i\in\mathcal{U}_1} g_{ti}\mathbf{h}_{ti}^{q}x_i + \sum_{i\in\bar{\mathcal{S}}\setminus\bar{\mathcal{S}}_q} g_{ti}\mathbf{h}_{ti}^{q}x_i + \mathbf{n}_{tq}, \tag{6.31}$$

where $\bar{\mathcal{S}}$ is the set of users who have been decoded at the APs in phase 1, and $\bar{\mathcal{S}} \setminus \bar{\mathcal{S}}_q$ is the set of users who have been decoded at all the other APs in phase 1 except for the qth AP. These components are still present in the residual signal of the qth AP since IC is not performed for all the other users at the qth AP. Thus, $\mathbf{D}_{tq}^{\mathrm{p}} \triangleq \sum_{i\in\bar{\mathcal{S}}\setminus\bar{\mathcal{S}}_q} g_{ti}\mathbf{h}_{ti}^{q}\mathbf{p}_i^{H}$ and $\mathbf{d}_{tq} \triangleq \sum_{i\in\bar{\mathcal{S}}\setminus\bar{\mathcal{S}}_q} g_{ti}\mathbf{h}_{ti}^{q}x_i$ are the interference components due to users decoded at all APs other than the qth AP.

At the CPU, the processed residual pilot signal from the qth AP in the kth CPU decoding iteration is given by

$$\mathbf{Z}_{tq}^{\mathrm{p}k} = \textstyle\sum_{i\in\mathcal{U}_k} g_{ti}\mathbf{h}_{ti}^{q}\mathbf{p}_i^{H} + \mathbf{D}_{tq}^{\mathrm{p}} + \mathbf{N}_{tq}^{\mathrm{p}}. \tag{6.32}$$

We now need to derive the channel estimates at the CPU in phase 2, which are computed based on $\mathbf{Z}_{tq}^{\mathrm{p}k}$. Based on the notation setup above, $\mathcal{U}_k \cup (\bar{\mathcal{S}} \setminus \bar{\mathcal{S}}_q)$ is the set of users who have not yet been decoded up to the kth iteration at the CPU in phase 2 and also the users who have been decoded at all the APs except for the qth AP in phase 1. Thus, $\bar{\mathcal{U}}_{tq}^{k} \triangleq \mathcal{G}_t \cap \mathcal{U}_k \cup (\bar{\mathcal{S}} \setminus \bar{\mathcal{S}}_q)$ is the set of such users who have transmitted in the tth RB and $\bar{U}_{tq}^{k} = |\bar{\mathcal{U}}_{tq}^{k}|$ is the number of users who are yet to be decoded at the CPU up to the kth CPU decoding iteration plus users who transmitted in the tth RB but were not decoded at the qth AP. Note that we account for the pilot contamination due to users decoded at the APs in phase 1 as well.[5]

[5]Similar to the single-cell estimators ignoring inter-cell pilot contamination, we can also ignore the

Let $\bar{\mathbf{P}}_{tq}^{k}$ contain as its columns the pilots of the $\bar{U}_{tq}^{k}$ users, and $\bar{\mathbf{B}}_{tq}^{k} \triangleq \sigma_{\mathtt{h}}^{2}\mathrm{diag}(\beta_{i_1}^{q}, \beta_{i_2}^{q}, \ldots, \beta_{i_{\bar{U}_{tq}^{k}}}^{q})$ contain the path loss coefficients of the $\bar{U}_{tq}^{k}$ users, with $\bar{\mathcal{U}}_{tq}^{k} = \{i_1, i_2, \ldots, i_{\bar{U}_{tq}^{k}}\}$. The subset of the users who are yet to be decoded at the CPU up to the kth CPU decoding iteration is $\mathcal{G}_t \cap \mathcal{U}_k$, for whom the channel estimate matrix is obtained as $\hat{\mathbf{H}}_{tq}^{k} = [\hat{\bar{\mathbf{H}}}_{tq}^{k}]_{:,\mathcal{G}_t \cap \mathcal{U}_k}$, with $U_t^k \triangleq |\mathcal{G}_t \cap \mathcal{U}_k|$. Finally, the channel estimate of the channel $\mathbf{h}_{ti}^{q}$ of the ith user to the qth AP is calculated as $\hat{\mathbf{h}}_{ti}^{qk} = [\hat{\mathbf{H}}_{tq}^{k}]_{:,i}$. The channel of the ith user in the tth slot to the CPU can be computed by stacking the user-AP channels as $\mathbf{h}_{ti} = [\mathbf{h}_{ti}^{1H}, \mathbf{h}_{ti}^{2H}, \ldots, \mathbf{h}_{ti}^{QH}]^H \in \mathbb{C}^{QN\times 1}$. Thus, the CPU channel estimation error is given by $\tilde{\mathbf{h}}_{ti}^{k} \triangleq \hat{\mathbf{h}}_{ti}^{k} - \mathbf{h}_{ti}$. The estimate of $\mathbf{h}_{ti}$ is presented in Theorem 6.4.

Let ρ_{tm}^{k} denote the SINR of the mth user in the tth RB in the kth decoding iteration at the CPU. The received data signal can be stacked as $\mathbf{z}_t = [\mathbf{z}_{t1}^H, \mathbf{z}_{t2}^H, \ldots, \mathbf{z}_{tQ}^H]$. Similar to (6.20), in the kth decoding iteration at the CPU, the stacked received data signal in the tth RB at the CPU can be expressed as

$$\mathbf{z}_t^k = \textstyle\sum_{i\in\mathcal{U}_k} g_{ti}\mathbf{h}_{ti}x_i + \mathbf{d}_t + \mathbf{n}_t. \tag{6.33}$$

where $\mathbf{d}_t \triangleq [\mathbf{d}_{t1}^H, \mathbf{d}_{t2}^H, \ldots, \mathbf{d}_{tQ}^H]^H$, and $\mathbf{n}_t \triangleq [\mathbf{n}_{t1}^H, \mathbf{n}_{t2}^H, \ldots, \mathbf{n}_{tQ}^H]^H$. The CPU uses a combining vector $\mathbf{a}_{tm}^{k}$ to combine the received data signal. Thus, we obtain the post-combined data signal as $\tilde{z}_{tm}^{k} = \mathbf{a}_{tm}^{kH}\mathbf{z}_t^k$ which is expanded as

$$\tilde{z}_{tm}^{k} = \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}g_{tm}x_m - \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^{k}g_{tm}x_m + \mathbf{a}_{tm}^{kH}\textstyle\sum_{i\in\mathcal{U}_k^m} g_{ti}\mathbf{h}_{ti}x_i + \mathbf{a}_{tm}^{kH}\mathbf{d}_t + \mathbf{a}_{tm}^{kH}\mathbf{n}_t. \tag{6.34}$$

The first term on the RHS, $\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^{k}g_{tm}x_m$, denotes the useful signal component and it contains the CPU channel estimate $\hat{\mathbf{h}}_{tm}^{k}$ of the mth user; the second term $\mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^{k}g_{tm}x_m$ contains $\tilde{\mathbf{h}}_{tm}^{k}$, the CPU estimation error of the mth user; the third term $\mathbf{a}_{tm}^{kH}\sum_{i\in\mathcal{U}_k^m} g_{ti}\mathbf{h}_{ti}x_i$ is the multi-user interference faced by the mth user due to the other users in the network who have not yet been decoded up to the kth decoding iteration at the CPU (and who have also not been decoded at any AP in phase 1); the fourth term $\mathbf{a}_{tm}^{kH}\mathbf{d}_t$ is due to the interference from users already decoded at all the APs in phase 1; and the fifth term

same and perform low complexity channel estimation. For more details, see Sec. 6.7.2.

$\mathbf{a}_{tm}^{kH}\mathbf{n}_t$ is the additive noise component. We need to compute the SINR from the above post-combined data signal. For this purpose, let $\bar{\mathbf{C}}_t^{qk} \triangleq \bar{\mathbf{P}}_{tq}^k \bar{\mathbf{B}}_{tq}^k (\bar{\mathbf{P}}_{tq}^{kH} \bar{\mathbf{P}}_{tq}^k \bar{\mathbf{B}}_{tq}^k + N_0 \mathbf{I}_{\bar{U}_{tq}^k})^{-1}$, and $\mathbf{c}_{ti}^{qk} \triangleq [\bar{\mathbf{C}}_t^{qk}]_{:,i}$.

We now present the channel estimates of the users' channels $\mathbf{h}_{ti}$ and the SINR achieved by the users at the CPU in HCF IRSA in the following theorem.

Theorem 6.4 ▶ Performance Analysis of HCF IRSA.

The minimum mean squared error (MMSE) channel estimate $\hat{\bar{\mathbf{H}}}_{tq}^k$ of $\bar{\mathbf{H}}_{tq}^k$ in the tth RB in the kth CPU decoding iteration to the qth AP at the CPU

$$\hat{\bar{\mathbf{H}}}_{tq}^k = \mathbf{Z}_{tq}^{\mathrm{p}k} (\bar{\mathbf{P}}_{tq}^k \bar{\mathbf{B}}_{tq}^k \bar{\mathbf{P}}_{tq}^{kH} + N_0 \mathbf{I}_\tau)^{-1} \bar{\mathbf{P}}_{tq}^k \bar{\mathbf{B}}_{tq}^k. \tag{6.35}$$

The subset of the users who are yet to be decoded at the CPU up to the kth CPU decoding iteration is $\mathcal{G}_t \cap \mathcal{U}_k$, for whom the channel estimate matrix is obtained as $\hat{\mathbf{H}}_{tq}^k \triangleq [\hat{\bar{\mathbf{H}}}_{tq}^k]_{:,\mathcal{G}_t \cap \mathcal{U}_k}$. Finally, the channel estimate of the channel $\mathbf{h}_{ti}^q$ of the ith user to the qth AP is calculated as $\hat{\mathbf{h}}_{ti}^{qk} \triangleq [\hat{\mathbf{H}}_{tq}^k]_{:,i}$. For the ith user in the tth RB in the kth CPU decoding iteration, the effective QN-length CPU channel estimate is found as

$$\hat{\mathbf{h}}_{ti}^k = [\hat{\mathbf{h}}_{ti}^{1kH}, \hat{\mathbf{h}}_{ti}^{2kH}, \ldots, \hat{\mathbf{h}}_{ti}^{QkH}]^H \in \mathbb{C}^{QN \times 1}. \tag{6.36}$$

Further, the estimation error $\tilde{\mathbf{h}}_{ti}^{qk} \triangleq \hat{\mathbf{h}}_{ti}^{qk} - \mathbf{h}_{ti}^q$ is distributed as $\tilde{\mathbf{h}}_{ti}^{qk} \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^{qk} \mathbf{I}_N)$, where δ_{ti}^{qk} is calculated as

$$\delta_{ti}^{qk} = \beta_i^q \sigma_{\mathrm{h}}^2 \left(\frac{N_0 \|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n \in \bar{\mathcal{U}}_{tq}^k \setminus \{i\}} |\mathbf{p}_n^H \mathbf{c}_{ti}^{qk}|^2 g_{tn} \beta_n^q \sigma_{\mathrm{h}}^2}{N_0 \|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n \in \bar{\mathcal{U}}_{tq}^k} |\mathbf{p}_n^H \mathbf{c}_{ti}^{qk}|^2 g_{tn} \beta_n^q \sigma_{\mathrm{h}}^2} \right). \tag{6.37}$$

The covariance of the channel estimation error at the CPU is

$$\Delta_{ti}^k \triangleq \mathrm{blkdiag}(\delta_{ti}^{1k} \mathbf{I}_N, \delta_{ti}^{2k} \mathbf{I}_N, \ldots, \delta_{ti}^{Qk} \mathbf{I}_N). \tag{6.38}$$

The SINR achieved by the mth user at the CPU in the tth RB and the kth decoding

iteration can be written as

$$\rho_{tm}^{k} = \frac{\mathtt{Gain}_{tm}^{k}}{N_0 + \mathtt{Est}_{tm}^{k} + \mathtt{MUI}_{tm}^{k} + \mathtt{Dec}_{tm}^{k}}, \forall m \in \mathcal{U}_k, \tag{6.39}$$

where

$$\begin{aligned}
\mathtt{Gain}_{tm}^{k} &\triangleq p_m g_{tm} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^{k}|^2 / \|\mathbf{a}_{tm}^{k}\|^2, \\
\mathtt{Est}_{tm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{U}_k} p_i g_{ti} \mathbf{a}_{tm}^{kH} \Delta_{ti}^{k} \mathbf{a}_{tm}^{k} / \|\mathbf{a}_{tm}^{k}\|^2, \\
\mathtt{MUI}_{tm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{U}_k^m} p_i g_{ti} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{ti}^{k}|^2 / \|\mathbf{a}_{tm}^{k}\|^2, \\
\mathtt{Dec}_{tm}^{k} &\triangleq \mathbf{a}_{tm}^{kH} \Sigma_{\mathbf{d}_t} \mathbf{a}_{tm}^{k} / \|\mathbf{a}_{tm}^{k}\|^2.
\end{aligned}$$

Here, $\Sigma_{\mathbf{d}_t} \triangleq \text{blkdiag}(\bar{d}_{t1}\mathbf{I}_N, \bar{d}_{t2}\mathbf{I}_N, \ldots, \bar{d}_{tQ}\mathbf{I}_N)$, with $\bar{d}_{tq} \triangleq \sum_{i \in \bar{\mathcal{S}} \setminus \bar{\mathcal{S}}_q} p_i g_{ti} \beta_i^q \sigma_{\mathtt{h}}^2$.

Proof. See Sec. 6.6.6. □

Remark 8: The channel estimates $\hat{\bar{\mathbf{H}}}_{tq}^{k}$ for the users in $\bar{\mathcal{U}}_{tq}^{k}$ in (6.35) can be computed as

$$\hat{\bar{\mathbf{H}}}_{tq}^{k} = \mathbf{Z}_{tq}^{\mathtt{p}k} (\bar{\mathbf{P}}_{tq}^{k} \bar{\mathbf{B}}_{tq}^{k} \bar{\mathbf{P}}_{tq}^{kH} + N_0 \mathbf{I}_\tau)^{-1} \bar{\mathbf{P}}_{tq}^{k} \bar{\mathbf{B}}_{tq}^{k}, \tag{6.40a}$$

$$\overset{(a)}{=} \mathbf{Z}_{tq}^{\mathtt{p}k} \bar{\mathbf{P}}_{tq}^{k} \bar{\mathbf{B}}_{tq}^{k} (\bar{\mathbf{P}}_{tq}^{kH} \bar{\mathbf{P}}_{tq}^{k} \bar{\mathbf{B}}_{tq}^{k} + N_0 \mathbf{I}_{\bar{U}_{tq}^{k}})^{-1}, \tag{6.40b}$$

where (a) follows from $(\mathbf{AB} + \mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA} + \mathbf{I})^{-1}$ [144]. Here, the estimate can be calculated via an inverse of either a $\tau \times \tau$ matrix or an $\bar{U}_{tq}^{k} \times \bar{U}_{tq}^{k}$ matrix as required.

Remark 9: The results derived in Theorem 6.4 holds for any choice of the combining vector $\mathbf{a}_{tm}^{k}$, the pilots $\{\mathbf{p}_m\}$, and the power control policy $\{p_m\}$. The channel estimates $\hat{\mathbf{h}}_{ti}^{k}$ are stacked into the matrix $\hat{\mathbf{H}}_{t}^{k} \in \mathbb{C}^{QN \times U_t^k}$. The combining vector that maximizes the SINR in (6.39) is the MMSE combiner, which can be found as $\mathbf{a}_{tm}^{k} = [\mathbf{A}_t^k]_{:,m}$, where

$$\mathbf{A}_t^k = (\mathbf{L}_t^k + \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p1}}^k \hat{\mathbf{H}}_t^{kH})^{-1} \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p1}}^k,$$

where $\mathbf{L}_t^k \triangleq \Sigma_{\mathbf{d}_t} + \sum_{i \in \mathcal{U}_k} p_i g_{ti} \Delta_{ti}^k + N_0 \mathbf{I}_{QN}$, and $\mathbf{D}_{t,\mathtt{p1}}^k \triangleq \text{diag}(p_{i_1}, p_{i_2}, \ldots, p_{i_{U_t^k}})$ contains the

Algorithm 9: Performance Evaluation of Phase 1 of HCF IRSA at the APs

Input: $\tau, N, T, M, N_0, \mathbf{G}, \mathbf{P}, \{\mathbf{Y}^{\mathtt{p}}_{tq}\}_{t=1,2,\ldots,T;q=1,2,\ldots,Q}, \{\mathbf{y}_{tq}\}_{t=1,2,\ldots,T;q=1,2,\ldots,Q}, k_{\max}$

for $q = 1, 2, \ldots, Q$ **do**

  Initialize: $\mathcal{S}_{1q} = [M]$, $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$, $\bar{\mathcal{S}}_q = \{\}$

  for $k = 1, 2, \ldots, k_{\max}$ **do**

    for $t = 1, 2, \ldots, T$ **do**

      Find $M^k_{tq} = |\mathcal{G}_t \cap \mathcal{S}_{kq}|$, $\mathbf{P}^k_{tq} = [\mathbf{P}]_{:,\mathcal{G}_t \cap \mathcal{S}_{kq}}$, $\mathbf{Y}^{\mathtt{p}k}_{tq}, \mathbf{y}^k_{tq}$

      Compute $\hat{\mathbf{h}}^{qk}_{ti}$, $\forall i \in \mathcal{S}_{kq}$ via Theorem 6.2

      Evaluate the SINR ρ^k_{tqi}, $\forall i \in \mathcal{S}_{kq}$ via Theorem 6.2

      If $\rho^k_{tqi} \geq \gamma_{\text{th}}$, remove user i from $\mathcal{S}_{kq}$ and perform IC in all RBs where $g_{ti} = 1$

      Add user i to set of decoded users: $\bar{\mathcal{S}}_q = \bar{\mathcal{S}}_q \cup \{i\}$

    end

  end

  Output: Set of users decoded at the qth AP $\bar{\mathcal{S}}_q$, residual pilot signal $\mathbf{Z}^{\mathtt{p}1}_{tq} \triangleq \mathbf{Y}^{\mathtt{p}k_{\max}}_{tq}$, and residual data signal $\mathbf{z}^1_{tq} \triangleq \mathbf{y}^{k_{\max}}_{tq}$

end

Algorithm 10: Performance Evaluation of Phase 2 of HCF IRSA at the CPU

Input: $\tau, N, T, M, N_0, \mathbf{G}, \mathbf{P}, \{\mathbf{Z}^{\mathtt{p1}}_{tq}\}_{t=1,2,\ldots,T;q=1,2,\ldots,Q}, \{\mathbf{z}^{1}_{tq}\}_{t=1,2,\ldots,T;q=1,2,\ldots,Q}, k_{\max}$

Compute set of users decoded at all the APs $\bar{\mathcal{S}} = \bar{\mathcal{S}}_1 \cup \bar{\mathcal{S}}_2 \cup \cdots \cup \bar{\mathcal{S}}_Q$

Compute residual received data signal at the CPU $\mathbf{z}^1_t = [\mathbf{z}^{1H}_{t1}, \mathbf{z}^{1H}_{t2}, \ldots, \mathbf{z}^{1H}_{tQ}]$

Initialize: $\mathcal{U}_1 = [M] \setminus \bar{\mathcal{S}}$, $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$, $\bar{\mathcal{U}} = \{\}$

for $k = 1, 2, \ldots, k_{\max}$ **do**

  for $t = 1, 2, \ldots, T$ **do**

    Find $\bar{\mathcal{U}}^k_{tq} = \mathcal{G}_t \cap \mathcal{U}_k \cup (\bar{\mathcal{S}} \setminus \bar{\mathcal{S}}_q)$, $\bar{U}^k_{tq} = |\bar{\mathcal{U}}^k_{tq}|$,
    $\mathbf{P}^k_{tq} = [\mathbf{P}]_{:,\bar{\mathcal{U}}^k_{tq}}, \mathbf{Z}^{\mathtt{p}k}_{tq}, \forall q \in \{1, 2, \ldots, Q\}$, and $\mathbf{z}^k_t$

    Compute local estimates $\hat{\mathbf{h}}^{qk}_{ti},\ \forall i \in \mathcal{U}_k, \forall q \in \{1, 2, \ldots, Q\}$ via (6.35)

    Stack local estimates to obtain CPU estimate
    $\hat{\mathbf{h}}^k_{ti} = [\hat{\mathbf{h}}^{1kH}_{ti}, \hat{\mathbf{h}}^{2kH}_{ti}, \ldots, \hat{\mathbf{h}}^{QkH}_{ti}]^H,\ \forall i \in \mathcal{U}_k$ as in Theorem 6.4

    Evaluate the SINR $\rho^k_{ti},\ \forall i \in \mathcal{U}_k$ via Theorem 6.4

    If $\rho^k_{ti} \geq \gamma_{\text{th}}$, remove user i from $\mathcal{U}_k$ and perform IC in all RBs where $g_{ti} = 1$

    Add user i to set of decoded users: $\bar{\mathcal{U}} = \bar{\mathcal{U}} \cup \{i\}$

  end

end

Output: Set of users decoded at the CPU: $\bar{\mathcal{U}}$

power coefficients of the U_t^k users. Alternately, using the Woodbury identity [144], the combiner matrix can also be found via a $U_t^k \times U_t^k$ inverse as

$$\mathbf{A}_t^k = \mathbf{J}_t^k \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p2}}^k (\mathbf{I}_{QN} - (\mathbf{G}_t^k)^{-1} \mathbf{D}_{t,\mathtt{p2}}^k \hat{\mathbf{H}}_t^{kH} \mathbf{J}_t^k \hat{\mathbf{H}}_t^k) \mathbf{D}_{t,\mathtt{p2}}^k,$$

where $\mathbf{J}_t^k \triangleq \text{blkdiag}(j_{t1}^k \mathbf{I}_N, j_{t2}^k \mathbf{I}_N, \ldots, j_{tQ}^k \mathbf{I}_N)$, $\mathbf{G}_t^k \triangleq \mathbf{I}_{U_t^k} + \mathbf{D}_{t,\mathtt{p2}}^k \hat{\mathbf{H}}_t^{kH} \mathbf{J}_t^k \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p2}}^k$, $j_{tq}^k \triangleq 1/(N_0 + \bar{d}_{tq} + \sum_{i \in \mathcal{U}_k} p_i g_{ti} \delta_{ti}^{qk})$, and $\mathbf{D}_{t,\mathtt{p2}}^k \triangleq \text{diag}(\sqrt{p_{i_1}}, \sqrt{p_{i_2}}, \ldots, \sqrt{p_{i_{U_t^k}}})$.

We now evaluate the performance of HCF IRSA at the APs using Algorithm 9 and at the CPU using Algorithm 10. Algorithm 9 is run similar to LCF IRSA. Algorithm 10 at the CPU is run till no more users are decoded in two successive iterations (or up to a maximum of $k_{\max}$ iterations). The algorithm outputs $\bar{\mathcal{S}}$, which is the set of users decoded at all the APs in phase 1, and $\bar{\mathcal{U}}$, which is the set of users decoded at the CPU, in phase 2. Thus, the PLR of the network can be computed as $\mathsf{PLR} \triangleq (1 - |\bar{\mathcal{S}} \cup \bar{\mathcal{U}}|/M)$, and the throughput can be computed as $\mathcal{T} \triangleq M(1 - \mathsf{PLR})/T$.

6.4 Numerical Results

6.4.1 MC IRSA

In this section, we evaluate the throughput of MC IRSA via Monte Carlo simulations and provide insights into the impact of various system parameters on the performance of the system. In each simulation, we generate independent realizations of the user locations, the access pattern matrix, and the channels. The throughput in each run is calculated as described in Sec. 6.2.1.1, and the effective system throughput is calculated by averaging over the runs. We consider a set of $Q = 9$ square cells, stacked in a 3×3 grid, and report the performance of the center cell [10]. Each cell has M users spread uniformly at random across an area of 250×250 m^2, with the BS at the center [49].[6]

The results in this section are for $T = 50$ RBs, $N_s = 10^3$ Monte Carlo runs, $\sigma_{\mathtt{h}}^2 = 1$,

[6]Due to path loss inversion, the area of the cell does not significantly affect the throughput, but affects the *area spectral efficiency*, which we do not analyze here.

SINR threshold $\gamma_{\text{th}} = 10$. The number of users contending for the T RBs in each cell is computed based on the load L as $M = \lfloor LT \rceil$. The path loss is calculated as β_{ji}^{q} (dB) $= -37.6 \log_{10}(d_{ji}^{q}/10\text{m})$, where d_{ji}^{q} is the distance of the ith user in the jth cell from the qth BS [49]. The pilot sequences are chosen as the columns of the $\tau \times \tau$ discrete Fourier transform matrix normalized to have column norm $\sqrt{\tau P^{\text{p}}}$. The soliton distribution [72] with $d_{\max} = 8$ maximum repetitions is used to generate the repetition factor d_{ji}, for the ith user in the jth cell, whose access vector is formed by uniformly randomly choosing d_{ji} RBs from T RBs without replacement [7].[7] The access pattern matrix is formed by stacking the access vectors of all the users. The power level is set to $P = P^{\text{p}} = 10\,\text{dBm}$ [49] and N_0 is chosen such that the data and pilot SNR are 10 dB, unless otherwise stated.

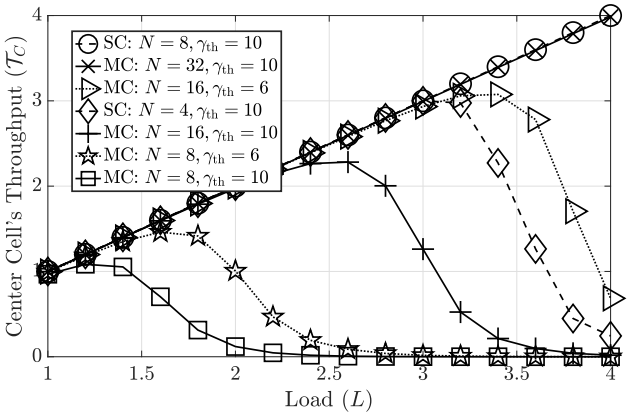


Figure 6.6: MC IRSA: Effect of load L with $\tau = M$.

In Fig. 6.6, we show the effect of the load L on the center cell's throughput $\mathcal{T}_C$. All the curves increase linearly till a peak, which is the desired region of operation, and then drop quickly to zero as the system becomes interference limited. All the users' packets are successfully decoded in the linear region of increase, and at high L, beyond the peak, the throughputs drop to zero. For $N = 8, \gamma_{\text{th}} = 10$, we see a 70% drop in the peak throughput from $\mathcal{T}_C = 4$ at $L = 4$ for SC to $\mathcal{T}_C = 1.2$ at $L = 1.2$ for MC. This is because users face a high degree of inter-cell interference in the MC setup, unlike the SC setup, especially at high L. In the SC setup, the peak throughput reduces from $\mathcal{T}_C = 4$ for

[7]The soliton distribution has been shown to achieve 96% of the throughput that can be achieved with the optimal repetition distribution [8]. For the optimal repetition distribution, see Chapter 7.

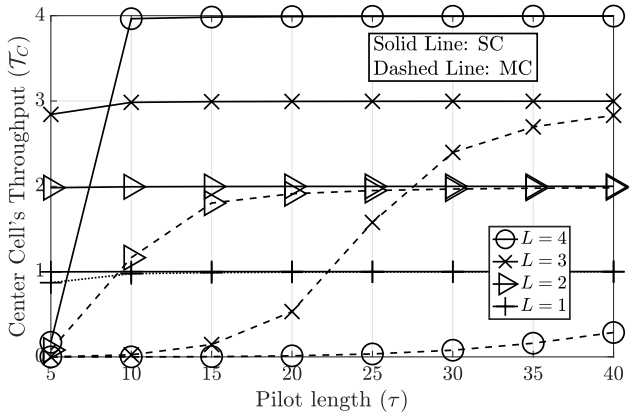


Figure 6.7: MC IRSA: Impact of pilot length τ with $N = 32$.

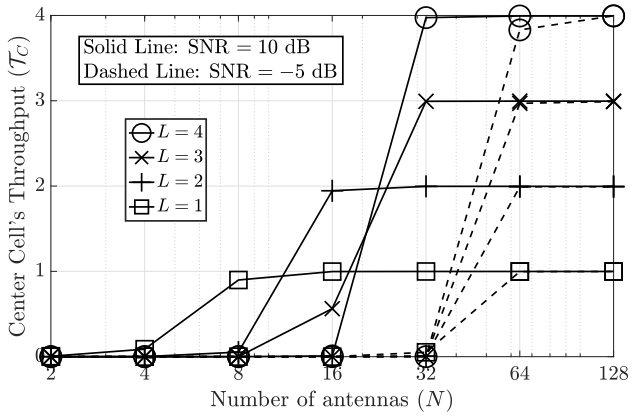


Figure 6.8: MC IRSA: Effect of number of antennas N.

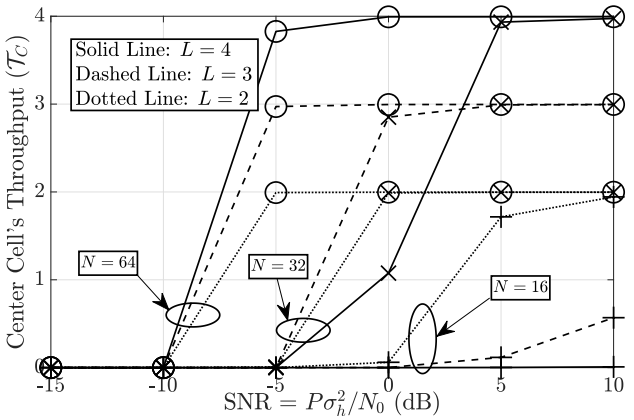


Figure 6.9: MC IRSA: Impact of SNR.

$N = 8$ to $\mathcal{T}_C = 3$ at $L = 3$ for $N = 4$. This trend is similar to the MC setup for which the peak throughputs are $\mathcal{T}_C = 4, 2.6, 1.2$ for $N = 32, 16, 8$, respectively. This is because the system's interference suppression ability with MMSE combining reduces as we decrease N [49]. This holds true with $\gamma_{\text{th}} = 6$ also, which corresponds to a lower SINR threshold, and consequently higher $\mathcal{T}_C$. To summarize, at high L, there is a high degree of inter-cell interference which SC processing does not account for, resulting in a substantial drop in performance.

Fig. 6.7 studies the impact of the pilot length τ. The performance of SC IRSA at all L is optimal (note that the throughput is upper bounded by L) for $\tau > 10$. In MC IRSA, nearly optimal throughputs are achieved for $L = 1, 2, 3$ at $\tau = 10, 30, 40$, respectively. The throughput for $L = 4$ does not improve much with τ. At high L, the impact of inter-cell interference is severe, as expected. Increasing τ implies that each cell has a higher number of orthogonal pilots, and hence can help in reducing intra-cell PC, but the system is still impacted by inter-cell PC and inter-cell interference. Thus, we see that MC IRSA requires significantly higher τ (at least $4 - 5\times$) to overcome inter-cell PC and inter-cell interference to achieve the same performance as that of SC IRSA.

In Fig. 6.8, we study the effect of N for $L = 1, 2, 3, 4$, with SNR $= 10, -5$ dB and $\tau = M$. Nearly optimal throughputs for $L = 1, 2, 3, 4$ can be achieved with $N = 8, 16, 32, 32$ for SNR $= 10$ dB, and with $N = 64$ for SNR $= -5$ dB. The system performance improves because of the array gain and higher interference suppression ability at high N. This aids in reducing not only intra-cell interference, but also inter-cell interference. However, as discussed in *Remark 3*, the SINRs of the users have a coherent interference component that scales with N. Thus, while an increase in N helps reducing intra-cell interference and inter-cell interference, and improves the system performance, it does not reduce intra-cell PC and inter-cell PC. Similar observations about N can be made where we study the impact of SNR in Fig. 6.9. At very low SNR, the system is noise limited, and increasing N does not help increase the throughput, which is at zero. For $N = 16$, the throughput is always zero and nearly zero for $L = 4$ and $L = 3$, respectively. Optimal throughputs are obtained at higher SNRs for $N = 32$ and 64. Since boosting transmit powers of the users

scales both the signal and interference components equally, the SINR does not increase, and therefore the system performance saturates with SNR. To summarize, increasing τ, N, and the SNR can judiciously help reduce the impact of intra-cell PC and inter-cell PC, as well as intra-cell interference and inter-cell interference.

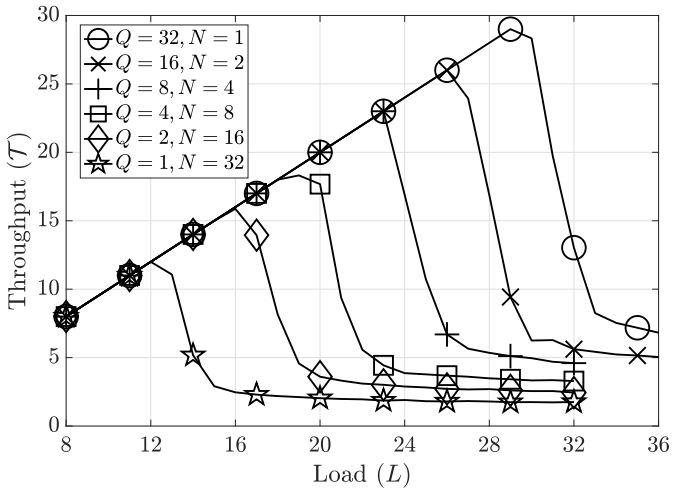
6.4.2 CCF IRSA

So far we have studied the performance of the central cell in MC IRSA via the *center cell's throughput*. We now study the performance of the network in CF IRSA via the *system throughput*. In this section, we evaluate the performance of CCF IRSA via Monte Carlo simulations and then, we provide insights into the impact of varied system parameters on the performance of the system. In each simulation, we generate independent realizations of the user locations, the access pattern matrix, and the channels. The throughput in each run is calculated as described in Sec. 6.3.3.2, and the effective system throughput is calculated by averaging over the runs. The results in this section are for $T = 50$ RBs, $N_s = 10^3$ Monte Carlo runs, $\sigma_{\mathtt{h}}^2 = 1$, SINR threshold $\gamma_{\text{th}} = 10$, plane size $d_{\text{p}} = 1$km, and $\tau = 10$ symbols [48]. The number of users contending for the T RBs is computed based on the load L as $M = \lfloor LT \rceil$. The pilot sequences are chosen as τ-length pilot sequences made up of random QPSK symbols, normalized to have column norm $\sqrt{\tau P_{\mathtt{P}}}$. The soliton distribution [72] with $d_{\max} = 4$ maximum repetitions is used to generate the repetition factor d_i, for the ith user, the access vector is formed by uniformly randomly choosing d_i RBs from T RBs without replacement [7]. The access pattern matrix is formed by stacking the access vectors of all the users.

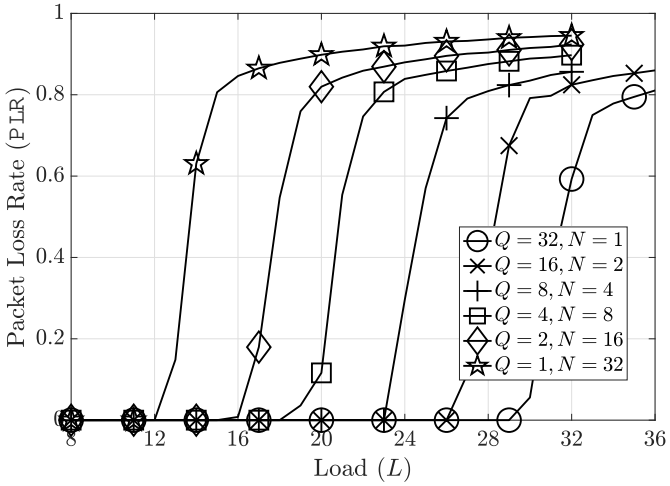
To generate the AP locations and make a fair comparison with SC IRSA, we divide the plane of size $d_p \times d_p$ into a total of Q virtual square cells of size $d_c \times d_c$ with $d_c = d_p/\sqrt{Q}$ [116]. We place the APs at the locations where the BSs would have been placed in a multi-cell setup, i.e., the centre of each cell, and then remove the cell boundaries to make up the CF system [21]. The path loss at the qth AP is calculated as β_i^q (dB) $= -37.6 \log_{10}(d_i^q/10\text{m}) + \mathtt{Shad.}$, where d_i^q is the distance of the ith user from the qth AP, and $\mathtt{Shad.} \sim \mathcal{N}(0, 16)$ is the log-normal shadowing component [21]. The received SNR at

the center of the plane of a user situated at the edge of the plane is termed as the *edge SNR*, and is denoted by ρ_{e}. The power levels of all users is chosen such that the signal from a user at the edge of the plane, i.e., at a distance $d_p/\sqrt{2}$ from the center of the plane, is received at ρ_{e}. This ensures that all users' signals are received at an SINR that at least ρ_{e} on average, in *singleton* RBs. The power level is set such that $\rho_{\mathrm{e}} = 10\,\mathrm{dB}$ [49], unless otherwise stated. We consider typical communication settings over a 100kHz bandwidth with a noise figure of 7 dB in the receiver hardware and calculate the noise power N_0 based on this, according to standard models [49]. The pilot power $P^{\mathtt{P}}$ is chosen to be twice the data power P [49].

In Fig. 6.10, we study the effect of the system load L on the performance of CCF IRSA. In Fig. 6.10(a), we plot the throughput for a fixed $QN = 32$ and varying Q in order to study the effects of densification on CCF IRSA (and the PLR in Fig. 6.10(b)). Here densification refers to increase in the number of cells with the same fixed network size, thereby having smaller cells in order to combat fading. The throughput under each configuration initially increases linearly with the load L since more users are decoded as and when they are added to the system. Once they hit a peak at the inflection load L^* for the corresponding configuration, the throughput starts dropping due to interference limitation. The throughput at high L beyond the inflection load saturates at a constant value since users close to the BS always get decoded due to high received powers. In the linear throughput regime, i.e., when $L \leq L^*$, the corresponding PLR is zero and becomes a non-zero value and close to unity at high load beyond L^*. We observe peak throughputs of $\mathcal{T} = 12$, 16, 19, 23, 26, and 29 for $(Q, N) = (1, 32)$, $(2, 16)$, $(4, 8)$, $(8, 4)$, $(16, 2)$, and $(32, 1)$. The inflection load of IRSA increases from $L^* = 12$ for SC IRSA to $L^* = 29$ for CCF IRSA, which is a 142% increase in the load. Specifically, at a load of $L = 29$, SC IRSA with a mMIMO setup achieves a throughput of $\mathcal{T} = 2$ packets per RB and CCF IRSA achieves a throughput of $\mathcal{T} = 29$ packets per RB, which is over a $14\times$ improvement in the throughput. This is because CCF IRSA exploits the MDG offered due to the distributed nature of the antennas. As we increase Q (keeping QN fixed), even though the N at each AP reduces, since we perform CPU only processing in CCF IRSA, the



(a) Throughput in CCF IRSA.



(b) Packet loss rate in CCF IRSA.

Figure 6.10: CCF IRSA: Effect of system load L.

distances between the users and the nearest AP reduces. Users are closer to the decoders in CCF IRSA and as such, the effect of path loss is compensated by the MDG. Thus, CCF IRSA outperforms SC IRSA due to the MDG of CF mMIMO systems.

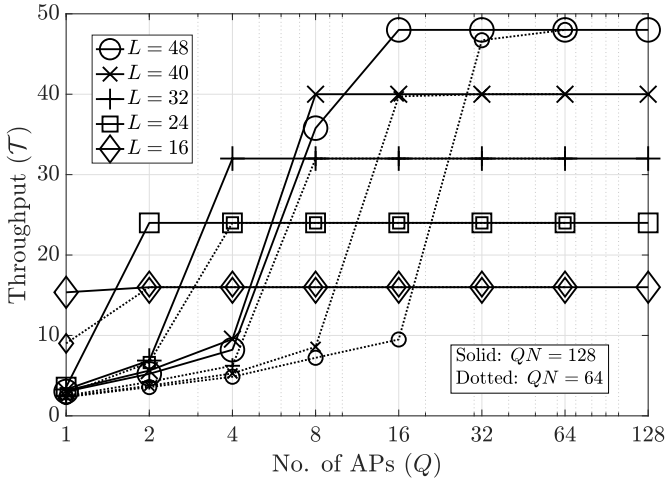


Figure 6.11: CCF IRSA: Impact of Q.

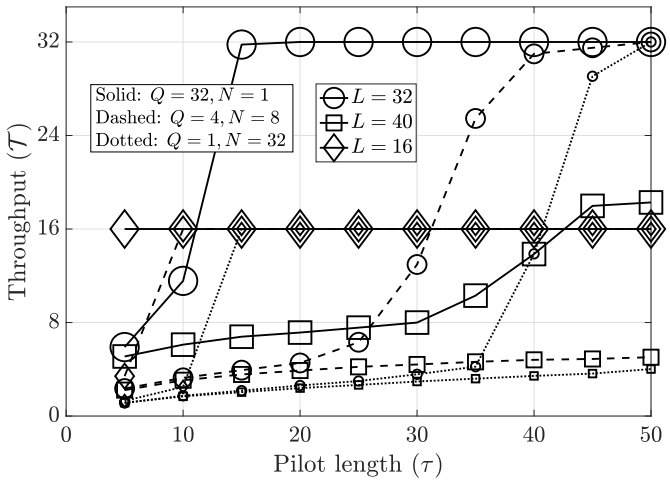
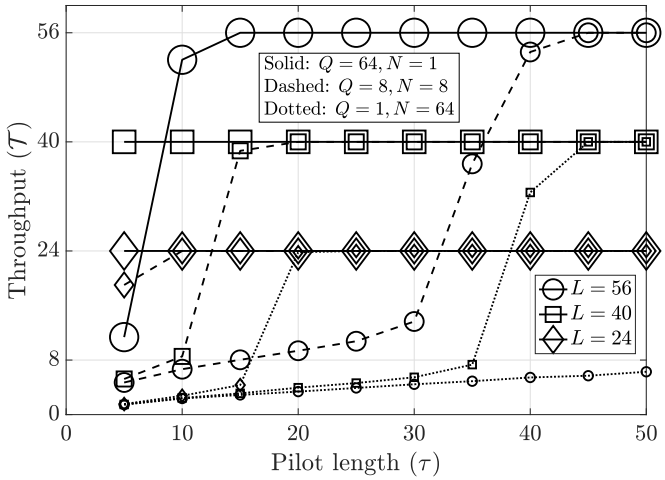
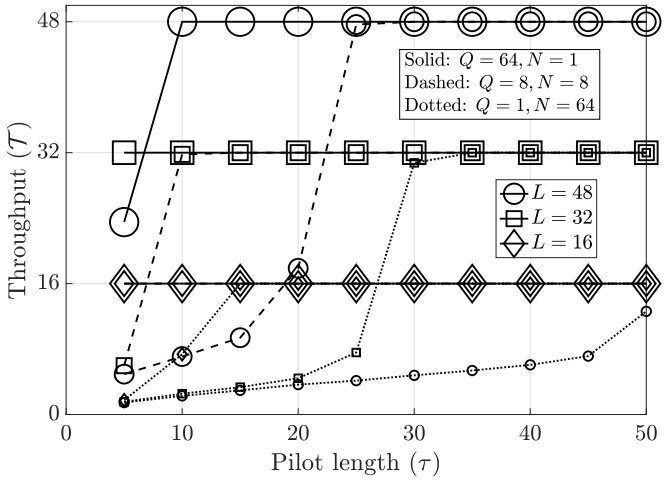


Figure 6.12: CCF IRSA: Effect of pilot length τ with $QN = 32$.

In Fig. 6.11, we study the impact of Q on the throughput of CCF IRSA for $QN = 128$ and $QN = 64$. The curves in the previous figure were for $QN = 32$. Here we study it for higher QN and higher L. We observe that all the curves improve with an increase in Q and achieve the optimal throughputs. With $QN = 128$, the optimal throughput of



(a) For $L = 56$,40, and 24.



(b) For $L = 48$,32, and 16.

Figure 6.13: CCF IRSA: Effect of pilot length τ with $QN = 64$.

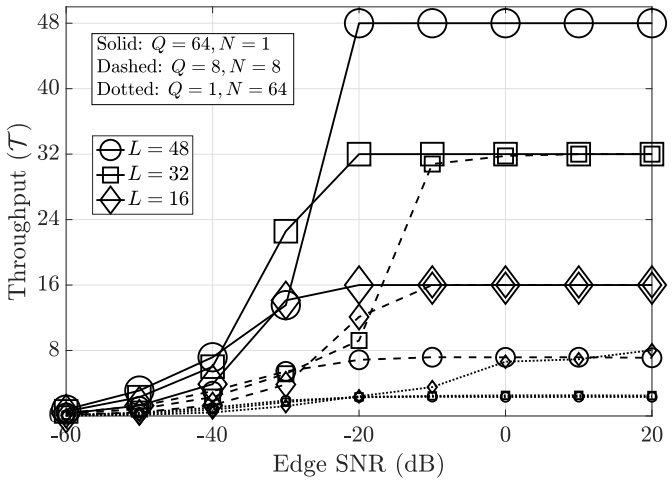


Figure 6.14: CCF IRSA: Impact of edge SNR.

$\mathcal{T} = L$ are achieved at $Q = 2$, 2, 4, 8, and 16 for $L = 16$, 24, 32, 40, and 48, respectively. With $QN = 64$, the optimal throughput of $\mathcal{T} = L$ are achieved at $Q = 2$, 4, 8, 16, and 64 for $L = 16$, 24, 32, 40, and 48, respectively. For $L = 32$, with $Q = 8$, having $N = 8$ antennas per AP yields a low throughput of $\mathcal{T} \approx 10$, whereas having $N = 16$ antennas per AP yields the optimal throughput of $\mathcal{T} = L = 32$. Similar observations can be made for $L = 48$ with $Q = 16$. This shows the behaviour of doubling the antennas per AP close to the inflection load in CCF IRSA. The observation about densification in CF systems is in agreement with existing works [22]. CCF IRSA yields optimal peak throughputs of $\mathcal{T} = L$ for every L. Comparing this with SC IRSA, we obtain improvements of 1.6×, 6×, 8×, 10×, and 12× in the throughput for $L = 16, 24, 32, 40$, and 48. Thus, for higher QN, we achieve even higher gains in both the throughputs at high loads as well as higher inflection loads.

In Fig. 6.12 and Fig. 6.13, we study the effect of the pilot length on the performance of CCF IRSA with $QN = 32$ and $QN = 64$ respectively. As observed before, densification highly improves the performance of CCF IRSA. In both the figures, we observe that all the curves improve with an increase in Q, but only a few achieve the optimal throughputs. Majority of the configurations in both figures perform poorly at $\tau = 5$, but the performance of all of them improve significantly with an increase in τ. For $QN = 32$

and $L = 32$, we observe that the peak throughput is achieved at $\tau = 15$ for CCF IRSA whereas SC IRSA achieves a throughput of 4, which is a $8\times$ improvement in the performance due to MDG in CCF IRSA. We also observe that the curves for $L = 40$ with $QN = 32$ never achieve the optimal throughputs, which is because $L = 40$ is always beyond the inflection load for $QN = 32$ (and $L = 32$ and 16 are well below the inflection load). When we increase QN from 32 to 64, we see that the optimal throughputs can be achieved. The increase in the antennas helps improve interference suppression. At $\tau = 5$, CCF IRSA achieves more than a $10\times$ improvement in the throughput for $L = 40$. Also, for $L = 40$, CCF IRSA achieves the optimal throughput with just $\tau = 5$, whereas SC mMIMO achieves the same with $\tau = 45$, which is a $9\times$ reduction in the pilot length. Further, we observe that for a few configurations, the performance is poor at for a wide range of τ, but at high τ, we achieve the optimal performance: with $Q = 8, N = 8$ and $L = 56, 48$, with $Q = 1, N = 64$ and $L = 40, 32$, with $Q = 4, N = 8$ and $L = 32$. For low L, the optimal throughputs are achieved by lower pilot lengths. This demonstrates the impact of the length of the pilot sequences in improving the performance of the system.

Finally, in Fig. 6.14, we study the impact of the edge SNR on the performance of the system for varied loads. As observed before, densification highly improves the performance of CCF IRSA. All the curves initially marginally improve with the SNR at very low SNRs ($< -40\,\text{dB}$), significantly improve at moderate SNRs (between $-40\,\text{dB}$ and $-20\,\text{dB}$), and then saturate at high SNRs ($> -20\,\text{dB}$). This saturation occurs at the peak optimal performance for $Q = 64, N = 1$ at all loads, and at low values for $Q = 1, N = 64$ at all loads. With $Q = 8, N = 8$, we observe that $L = 32$ and 16 achieve the peak throughputs at $0\,\text{dB}$ and $-10\,\text{dB}$ SNRs, whereas the performance of $L = 48$ saturates near $\mathcal{T} = 7$. This is because $L = 48$ is higher than the inflection load for the corresponding configuration. This plot also shows that devices could transmit with lower powers in CCF IRSA and still be decodable at the CPU compared to SC IRSA which does not even achieve the optimal throughputs for these high loads.

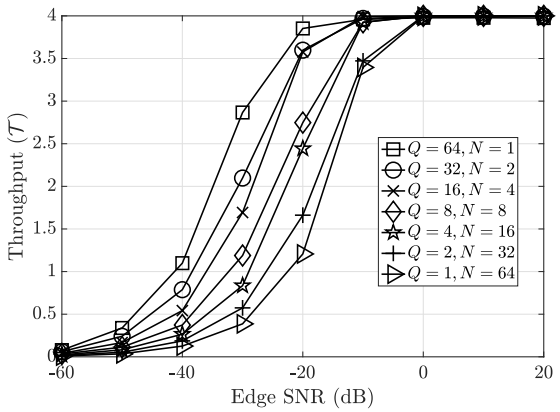
In summary, CCF IRSA massively improves upon the performance of SC IRSA, mainly due to the MDG of CF mMIMO systems. A fully densified network operates with the

best performance and can even achieve the peak optimal throughputs up to $L = 56$ for $QN = 64$ and even beyond. The inflection load massively improves with densification, pilot length, and the edge SNR.

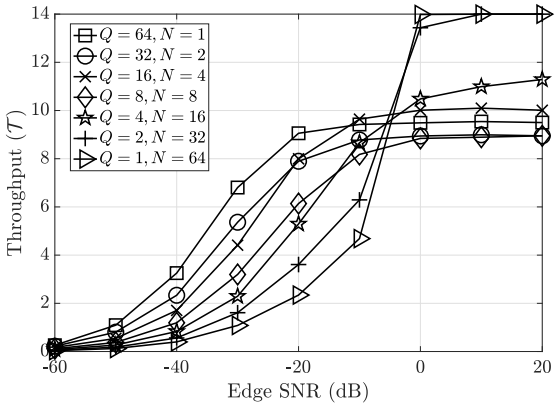
6.4.3 LCF IRSA

In this section, we study the impact of system parameters on the performance of LCF IRSA. The system parameters are identical to the previous subsection. Existing works handle interference-limitation of communication systems by using the divide-and-conquer approach, where the network is split up into small cells [22]. This reduces the effect of path loss and improves the performance. This is the exact behaviour we observed with CCF IRSA, and what we will also next observe in HCF IRSA. However, this behaviour is not observed in either MC or LCF IRSA systems as we will see below.

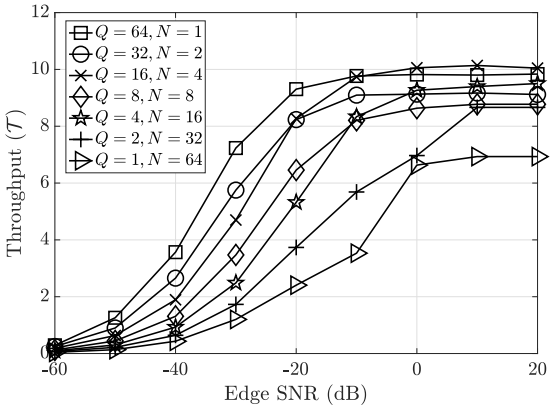
We first study the effect of densification on the performance of LCF IRSA in Fig. 6.15, keeping the antenna density QN fixed and $\tau = 10$. In Figs. 6.15(a), 6.15(b), 6.15(c), we study the effect of the edge SNR on LCF IRSA with $L = 4$, 14, and 16, respectively. As observed in SC IRSA, improving the edge SNR of the system improves the performance up to the point where the system either performs optimally or where the performance saturates. $L = 4$ is well below the inflection load L^*, and thus, the performance is optimal at high SNR. At low $L < L^*$, the system performance is dominated by the path loss, and thus, the performance improves with densification. $L = 16$ is beyond the inflection load L^*, and thus, the performance saturates at high SNR. At high $L > L^*$, the system's performance is already saturated, and thus, densifying helps only decode a few more users, thereby improving the performance only slightly. At both low load of $L = 4$ and high load $L = 16$, densification helps improve the performance unilaterally across all edge SNRs, i.e., SC IRSA performs the poorest and densification in LCF IRSA improves the performance for every SNR. $L = 14$ is close to the inflection load L^*, and thus, the performance is optimal at high SNR under certain configurations and the performance saturates at a suboptimal value under other configurations. For SNR $< -10\,\text{dB}$, any densification only improves the throughput. In this noise-limited regime, densifying the



(a) $L = 4$.

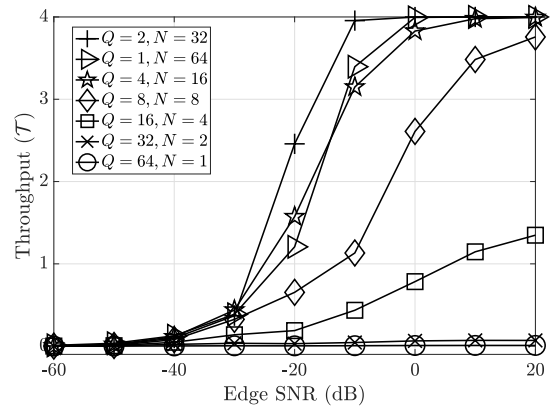


(b) $L = 14$.

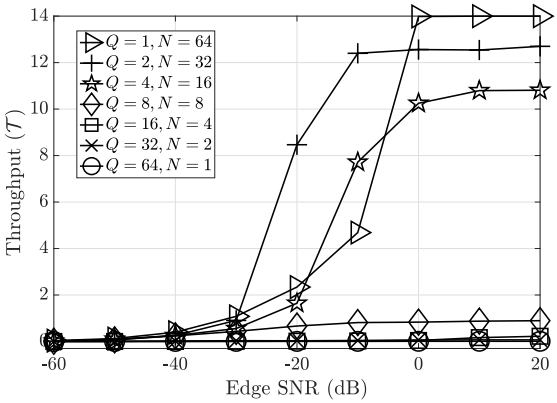


(c) $L = 16$.

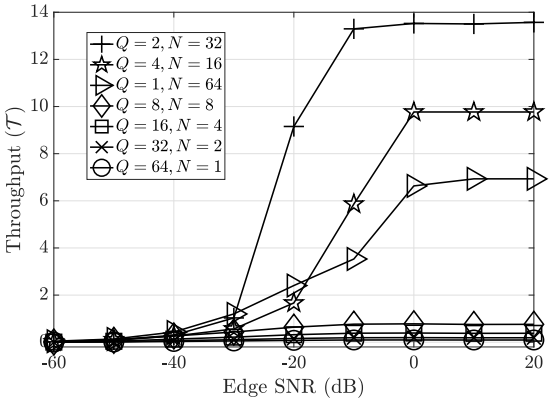
Figure 6.15: LCF IRSA: Effect of densification.



(a) $L = 4$.



(b) $L = 14$.



(c) $L = 16$.

Figure 6.16: MC IRSA: Effect of densification.

network pushes users closer to APs, thereby increasing the throughput. However, an inverse trend is observed for SNR $> -10\,\text{dB}$. In this regime, it is better to operate the system as a SC mMIMO system with 64 antennas rather than a small cell setup with 64 cells with 1 antenna each.

We observe similar trends for MC IRSA in Fig. 6.16, for which the simulation settings are identical to that of the previous figure. Here we compute the system throughput of MC IRSA and do not perform any path loss inversion. For a low load of $L = 4$, we observe that $Q = 2, N = 32$ performs the best, and it performs slightly better than $Q = 1, N = 64$, which is SC IRSA. All the other configurations, as the system is made more dense with smaller cells, perform poorly. Specifically for MC IRSA, even when the cells are made small, there are users out-of-cell who could be decoded at a different cell's BS, but will not be decoded. However, in LCF IRSA, these users get decoded as well since there are no cell boundaries. Other than this, the MC setup performs similar to the LCF setup, and the trends are similar: this is because LCF IRSA is similar to MC IRSA when each AP is replaced by a BS and the cell boundaries are removed, enabling each AP to decode users in singleton slots even from adjacent cells. For $L = 14$, $Q = 1, N = 64$ performs the best at edge SNR $\geq 0\,\text{dB}$, whereas for edge SNR $< 0\,\text{dB}$, $Q = 2, N = 32$ performs the best. This trend is similar to LCF IRSA. For $L = 16$, $Q = 2, N = 32$ performs the best, $Q = 4, N = 16$ is the next best configuration, $Q = 1, N = 64$ is the next best configuration, and the other configurations perform poorly. Similar to what was observed in LCF IRSA, for $L = 4$ and $L = 16$, both of which are not near to the inflection load, densification helps improve the performance of MC IRSA; albeit this happens only initially, and the most dense network performs very poorly. For $L = 14$, using a SC mMIMO system would help us even achieve the optimal throughput at high SNR.

Even though the dense network in MC or LCF IRSA has many small cells which can help overcome the effect of path loss, the system is now operating with fewer antennas, with which the system has a lower interference suppression ability. Thus, both MC and LCF IRSA perform poor with higher densification. This observation about densification in MC systems is different when compared with the observations in existing works [22].

This is because the main metric of interest in usual cellular communications is the spectral efficiency (SE) or sum-rate-type metrics and the main metric of interest in our work is the packet throughput of the system, i.e., the connectivity of the system. The SE metric does not talk about the connectivity of the system and captures the achievable rate of the system. It only measures the number of bits that can be successfully decoded. The metric we consider is captures "outage" events where packets transmitted at a given (fixed) rate are successfully decoded. Thus, in systems where the connectivity or outage is the main metric of interest, densification need not always help improve the performance of the system. Thus, in MC and LCF systems, densification helps improve the performance of IRSA at loads below or higher than the inflection load; at loads close to the inflection load, an SC mMIMO system performs better than MC and LCF systems.

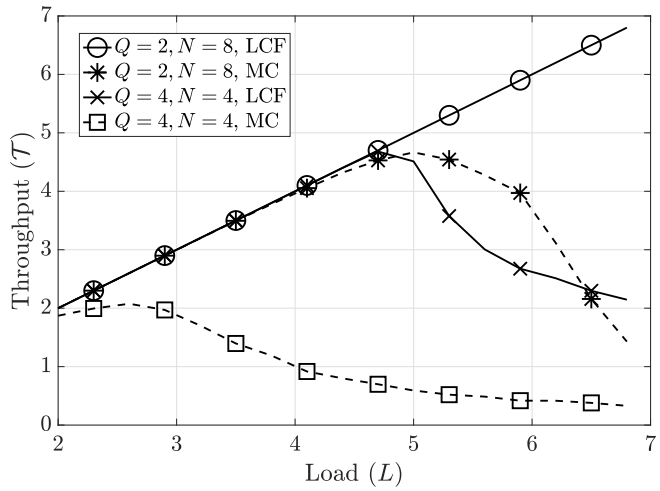
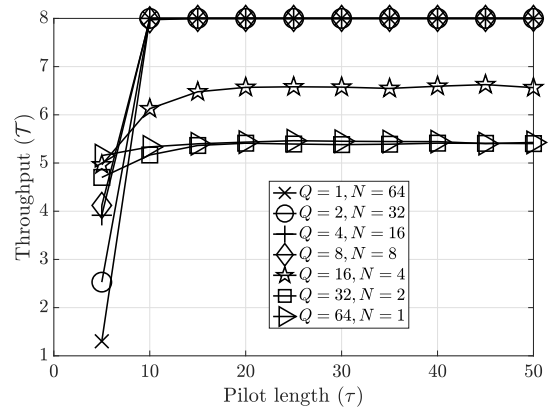


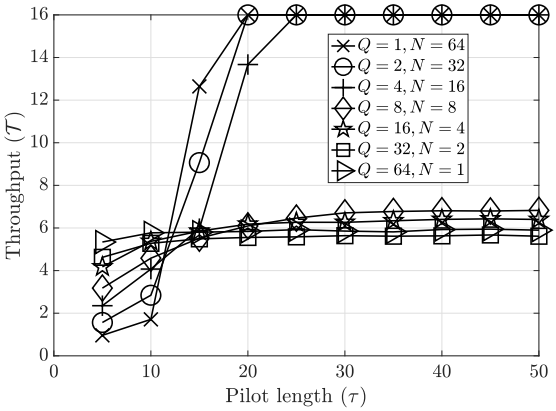
Figure 6.17: MC IRSA vs LCF IRSA.

In Fig. 6.17, we compare MC IRSA and LCF IRSA with identical parameter settings. With $Q = 4, N = 4$, LCF IRSA achieves an inflection load of $L^* = 4.7$, whereas MC IRSA achieves an inflection load of $L^* = 2.6$. With $Q = 2, N = 8$, LCF IRSA is still in the linear throughput regime, whereas MC IRSA achieves an inflection load of $L^* = 5$. Further, we still observe the same densification trends as before for both MC and LCF IRSA: densification does not help in improving the throughput. Also, by removing the cell boundaries and allowing the BSs or APs to decode as many users as possible, the

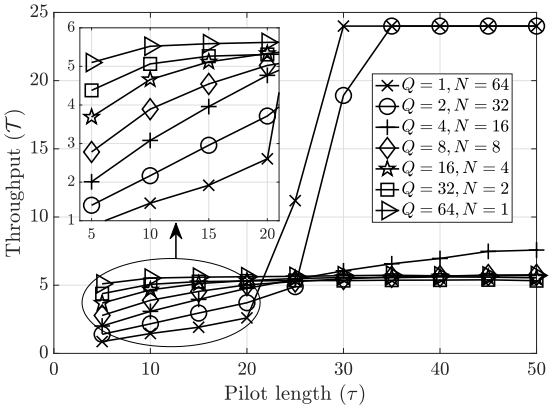
performance of the system improves.



(a) $L = 8$.



(b) $L = 16$.



(c) $L = 24$.

Figure 6.18: LCF IRSA: Impact of pilot length τ.

In Fig. 6.18, we study the effect of the pilot length τ on the performance of LCF IRSA for $L = 8$,16, and 24. Under all the configurations, the throughput either increases or saturates with the pilot length τ. For $L = 8$, the throughput is optimal when densifying from $Q = 1, N = 64$ to $Q = 8, N = 8$, and suboptimal otherwise, for $\tau \geq 10$. For $L = 16$, the throughput is optimal when densifying from $Q = 1, N = 64$ to $Q = 4, N = 16$, and suboptimal otherwise, for $\tau \geq 25$. For $L = 24$, the throughput is optimal when densifying from $Q = 1, N = 64$ to $Q = 2, N = 32$, and suboptimal otherwise, for $\tau \geq 35$. These observations are in agreement with the previous results, and they show that densifying is not always the solution to improve the performance of communication systems. The inflection pilot lengths for the SC IRSA setup in each of the figures are $\tau^* = 10$, 20, and 30, i.e., for all $\tau < \tau^*$, the throughput is poor and for all $\tau \geq \tau^*$, the throughput is optimal. In fact in each of the figures here, this inflection pilot length demarcates the regions when densification helps: for $\tau < \tau^*$, densifying unilaterally improves the system throughput; for $\tau \geq \tau^*$, densifying unilaterally reduces the system throughput. Similar to what we observed in Chapter 3, we can identify these inflection loads, inflection pilot lengths, inflection SNRs, and inflection number of antennas under any system configuration, and then use it to operate the system in the required regimes of interest.

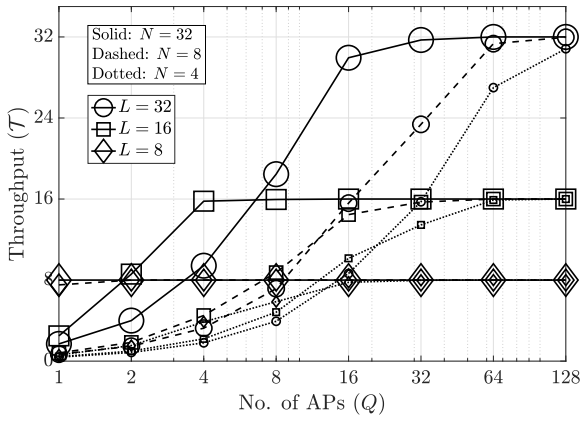


Figure 6.19: LCF IRSA: Effect of Q.

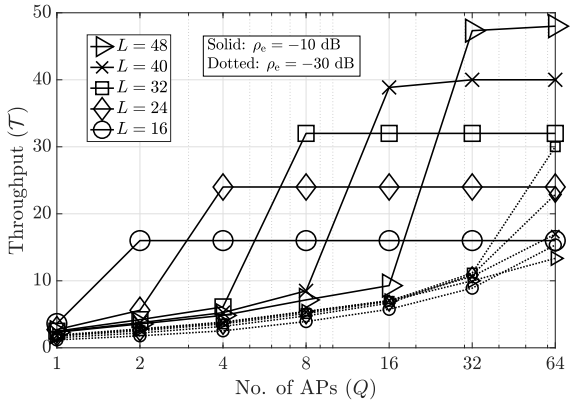
In Fig. 6.19, we study the impact of increasing the antenna density QN on the performance of LCF IRSA. For this, we plot the throughput versus the number of APs Q

and keep the number of antennas per AP N fixed. All of the curves improve with densification and reach optimal throughputs of $\mathcal{T} = L$ with high enough Q ($N = 4, L = 32$ is nearly optimal at $Q = 128$). These trends match the densification strategy in existing works, albeit we are increasing the antenna density, making the comparison unfair from an energy efficiency/power consumed perspective. $L = 8$ achieves the optimal throughput with $N = 4$, 8, 32 antennas at $Q = 16$, 2, and 1 AP(s), respectively. $L = 16$ achieves the optimal throughput with $N = 4$, 8, 32 antennas at $Q = 64$, 32, and 4 APs, respectively. $L = 32$ achieves the optimal throughput with $N = 4$, 8, 32 antennas at $Q > 128$, $Q = 128$, and 32 APs, respectively. Thus, increasing the antenna density can help overcome the limitations of LCF IRSA.

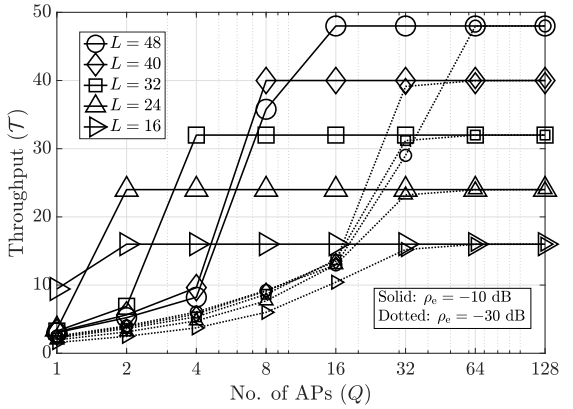
6.4.4 HCF IRSA

In this section, we study the impact of system parameters on the performance of HCF IRSA. The system parameters are identical to the previous subsection. In Fig. 6.20, we study the impact of densification on the performance of HCF IRSA for edge SNR $\rho_{\mathrm{e}} = -10\,\mathrm{dB}$ and $-30\,\mathrm{dB}$, and antenna density $QN = 64$ and 128. For $QN = 64$, $\rho_{\mathrm{e}} = -10\,\mathrm{dB}$, the peak performances are achieved for all L with densification; for $\rho_{\mathrm{e}} = -30\,\mathrm{dB}$, the peak performances are nearly obtained with densification only for $L = 16, 24$, and 32. For $QN = 64$, $\rho_{\mathrm{e}} = -10\,\mathrm{dB}$, the optimal throughputs with $L = 16$, 24, 32, 40, and 48 are achieved at $Q = 2$, 4, 8, 32, and 64 APs. Thus, for $QN = 64$, $\rho_{\mathrm{e}} = -10\mathrm{dB}$, HCF IRSA performs $4\times$, $6\times$, $8\times$, $10\times$, and $12\times$ for $L = 16$, 24, 32, 40, and 48, compared to SC IRSA. We see similar trends for $QN = 128$. For $QN = 128, \rho_{\mathrm{e}} = -30\,\mathrm{dB}$, with densification, the system performs optimally at high Q for all loads, whereas for $QN = 64, \rho_{\mathrm{e}} = -30\,\mathrm{dB}$, it did not. Thus, at low SNRs, increasing the antenna density (by doubling the number of antennas) helps improve the throughput.

In Fig. 6.21, we study the impact of SNR on the performance of HCF IRSA. Here, we compare only the fully densified HCF IRSA (with $Q = 128, N = 1$) with mMIMO SC IRSA (with $Q = 1, N = 128$). HCF IRSA achieves an inflection SNR of $\rho_{\mathrm{e}}^* = -30\,\mathrm{dB}$, i.e., for $\rho_{\mathrm{e}} < \rho_{\mathrm{e}}^*$, the system performs poorly and for $\rho_{\mathrm{e}} \geq \rho_{\mathrm{e}}^*$, the system performs optimally.



(a) $QN = 64$.



(b) $QN = 128$.

Figure 6.20: HCF IRSA: Effect of densification.

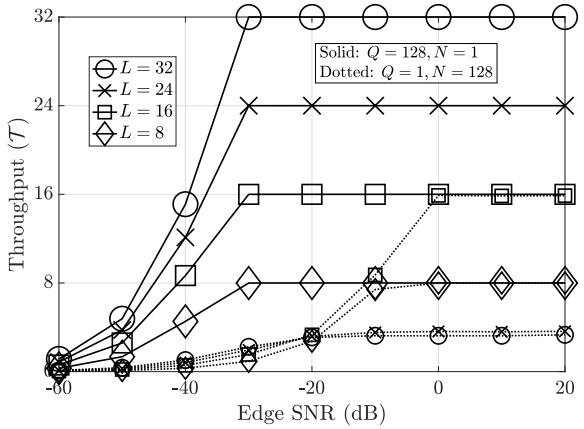
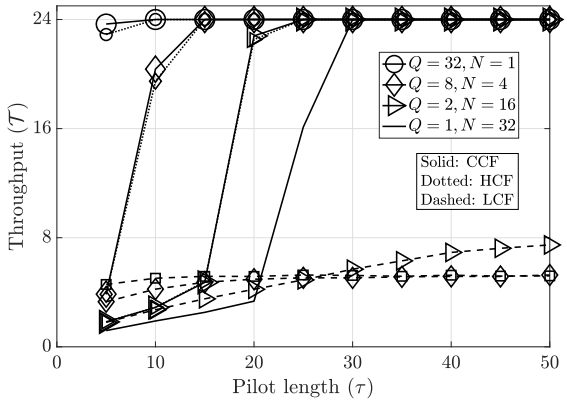
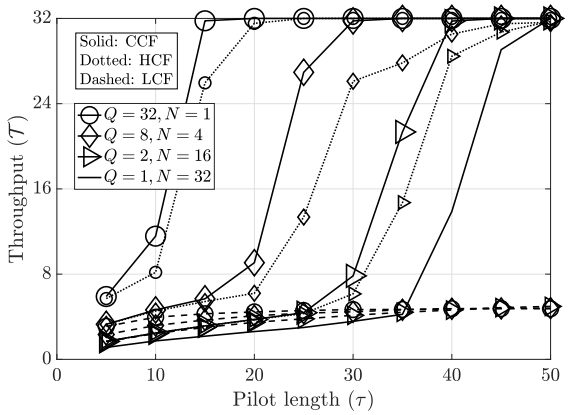


Figure 6.21: HCF IRSA: Impact of SNR.

SC IRSA achieves the optimal performance beyond 0 dB only for $L = 8$ and $L = 16$; for $L = 24$ and $L = 32$ the system throughput saturates at very low values. In fact, the trends in HCF are similar to the trends in CCF IRSA, as we will see next.



(a) $L = 24$.



(b) $L = 32$.

Figure 6.22: CCF vs LCF vs HCF vs SC IRSA.

In Fig. 6.22, we compare all the schemes presented in this chapter for $L = 24$ and $L = 32$.[8] Firstly, we observe that HCF performs closely to CCF IRSA in both the subfigures. The gap between CCF and HCF increases from $L = 24$ to $L = 32$, which indicates that the interference from decoded users in HCF IRSA increases with the load L.[9] For $L = 24$, with $\tau \leq 20$, SC IRSA performs the poorest and CCF IRSA with

[8]We have already compared MC and LCF IRSA previously, and thus, we skip MC IRSA in this plot.
[9]We note that for lower L, CCF and HCF perform identically. We have not included the plots for the

$Q = 32, N = 1$ performs the best; HCF is very close to CCF and LCF performs poorly. For $L = 24$, with $\tau > 20$, SC IRSA performs better than LCF IRSA and also achieves the optimal throughput at $\tau = 30$, same as CCF and HCF IRSA. For $L = 32$, with $\tau \leq 35$, SC IRSA performs the poorest and CCF IRSA with $Q = 32, N = 1$ performs the best; HCF is very close to CCF and LCF performs poorly at all τ. For $L = 32$, with $\tau > 35$, SC IRSA performs better than LCF IRSA and also achieves the optimal throughput at $\tau = 50$, same as CCF and HCF IRSA. In both figures, it is evident that densification helps improve the performance of CCF IRSA and HCF IRSA, but not LCF IRSA, which performs the poorest at all pilot lengths. Thus, depending on the regime of interest, SC IRSA performs better than LCF IRSA. Further, the CCF scheme has the best performance amongst the proposed schemes.

6.5 Summary

In this chapter, we first studied the effect of MC interference, namely inter-cell PC and inter-cell interference, on the performance of IRSA. Firstly, we derived the channel estimates and the SINR in MC IRSA, accounting for path loss, MIMO fading, intra-cell PC, and intra-cell interference. We saw that MC IRSA had a significant degradation in performance compared to SC IRSA, even resulting in up to 70% loss of throughput in certain regimes. Recuperating this loss requires at least $4-5\times$ larger pilot length in MC IRSA to yield the same performance as that of SC IRSA. Increasing τ, N, and SNR helped improve the performance of MC IRSA. These results underscore the importance of accounting for multiuser interference in analyzing IRSA in multi-cell settings. Future work could include design of optimal pilot sequences to reduce PC and density evolution [7] to obtain the asymptotic throughput.

We then analyzed IRSA in the CF setup, accounting for path loss, MIMO fading, and channel estimation errors. Specifically, we studied three CF schemes for IRSA: LCF IRSA, CCF IRSA, and HCF IRSA, which have different levels of partial signal processing

same.

at the APs and the CPU. We provided insights into the effect of system parameters such as number of antennas, number of APs, pilot length, and SNR on the performance of all three schemes in CF IRSA. We showed that we can achieve more than 14× improvement in the throughput or a 9× reduction in the pilot length of CCF IRSA compared to a massive MIMO SC setup at high loads. We also studied the BS and AP densification trends in MC IRSA and LCF IRSA, respectively, where we observe an inverse behaviour in the throughput compared to CCF IRSA and HCF IRSA. For CCF IRSA and HCF IRSA, densification always improves the performance. For LCF IRSA and MC IRSA, densification does not help at loads near the inflection loads: it is better to not densify and to operate with a massive MIMO SC setup. Future work could include calculating the load on the CPU and involve user-AP clustering to reduce the decoding complexity. We can also design optimal distributions for the MC and CF setups.

6.6 Proofs

6.6.1 Proof of Lemma 7: Channel Estimation

The minimum mean squared error (MMSE) channel estimate $\hat{\bar{\mathbf{H}}}_{tq}^{qk}$ of the channel matrix $\bar{\mathbf{H}}_{tq}^{qk}$ in the tth RB in the kth decoding iteration at the qth BS can be calculated as

$$\hat{\bar{\mathbf{H}}}_{tq}^{qk} = \mathbf{Y}_{tq}^{\mathtt{p}k}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk}(\bar{\mathbf{P}}_{tq}^{kH}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk} + N_0\mathbf{I}_{\bar{M}_{tq}^{qk}})^{-1}. \tag{6.41}$$

6.6.1.1 Channel estimation

The received signal is first vectorized as

$$\overline{\mathbf{y}}_{tq}^{k} \triangleq \mathrm{vec}(\mathbf{Y}_{tq}^{\mathtt{p}k}) = (\bar{\mathbf{P}}_{tq}^{k*} \otimes \mathbf{I}_N)\mathbf{h}_{tq}^{k} + \overline{\mathbf{n}}_{tq}, \tag{6.42}$$

where $\mathbf{h}_{tq}^{k} \triangleq \mathrm{vec}(\bar{\mathbf{H}}_{tq}^{qk})$, $\overline{\mathbf{n}}_{tq} \triangleq \mathrm{vec}(\mathbf{N}_{tq}^{\mathtt{p}})$, and $\otimes$ is the Kronecker product. The MMSE estimate is $\hat{\mathbf{h}}_{tq}^{k} \triangleq \mathbb{E}_{\mathbf{z}}[\mathbf{h}_{tq}^{k}]$, where $\mathbf{z} = \overline{\mathbf{y}}_{tq}^{k}$. The estimation error $\tilde{\mathbf{h}}_{tq}^{k} \triangleq \hat{\mathbf{h}}_{tq}^{k} - \mathbf{h}_{tq}^{k}$ is uncorrelated with the estimate and with $\mathbf{z}$. The conditional statistics of a Gaussian random vector $\mathbf{x}$

are

$$\mathbb{E}_{\mathbf{z}}\left[\mathbf{x}\right] = \mathbb{E}\left[\mathbf{x}\right] + \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}\left(\mathbf{z} - \mathbb{E}\left[\mathbf{z}\right]\right), \tag{6.43}$$

$$\mathbf{K}_{\mathbf{xx|z}} = \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}\mathbf{K}_{\mathbf{zx}}. \tag{6.44}$$

Here, $\mathbf{K}_{\mathbf{xx}}$, $\mathbf{K}_{\mathbf{xx|z}}$, and $\mathbf{K}_{\mathbf{xz}}$ are the unconditional covariance of $\mathbf{x}$, the conditional covariance of $\mathbf{x}$ conditioned on $\mathbf{z}$, and the cross-covariance of $\mathbf{x}$ & $\mathbf{z}$ respectively. From (6.43), the MMSE estimate $\hat{\mathbf{h}}_{tq}^{k}$ of the channel can be evaluated as

$$\hat{\mathbf{h}}_{tq}^{k} = \mathbb{E}\left[\mathbf{h}_{tq}^{k}\right] + \mathbb{E}\left[\mathbf{h}_{tq}^{k}\overline{\mathbf{y}}_{tq}^{kH}\right]\mathbb{E}[\overline{\mathbf{y}}_{tq}^{k}\overline{\mathbf{y}}_{tq}^{kH}]^{-1}(\overline{\mathbf{y}}_{tq}^{k} - \mathbb{E}\left[\overline{\mathbf{y}}_{tq}^{k}\right]).$$

The terms in the above expression can be calculated as

$$\begin{aligned}
\mathbb{E}\left[\mathbf{h}_{tq}^{k}\overline{\mathbf{y}}_{tq}^{kH}\right] &= \bar{\mathbf{B}}_{tq}^{qk}\bar{\mathbf{P}}_{tq}^{kT} \otimes \mathbf{I}_N, \\
\mathbb{E}[\overline{\mathbf{y}}_{tq}^{k}\overline{\mathbf{y}}_{tq}^{kH}] &= (\bar{\mathbf{P}}_{tq}^{k*}\bar{\mathbf{B}}_{tq}^{qk}\bar{\mathbf{P}}_{tq}^{kT} + N_0\mathbf{I}_\tau) \otimes \mathbf{I}_N, \\
\hat{\mathbf{h}}_{tq}^{k} &= (\bar{\mathbf{B}}_{tq}^{qk}\bar{\mathbf{P}}_{tq}^{kT}(\bar{\mathbf{P}}_{tq}^{k*}\bar{\mathbf{B}}_{tq}^{qk}\bar{\mathbf{P}}_{tq}^{kT} + N_0\mathbf{I}_\tau)^{-1} \otimes \mathbf{I}_N)\overline{\mathbf{y}}_{tq}^{k},
\end{aligned}$$

and thus, the MMSE estimate $\hat{\bar{\mathbf{H}}}_{tq}^{qk}$ of $\bar{\mathbf{H}}_{tq}^{qk}$ is

$$\hat{\bar{\mathbf{H}}}_{tq}^{qk} = \mathbf{Y}_{tq}^{\mathtt{p}k}(\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk}\bar{\mathbf{P}}_{tq}^{kH} + N_0\mathbf{I}_\tau)^{-1}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk} \tag{6.45}$$

$$\overset{(a)}{=} \mathbf{Y}_{tq}^{\mathtt{p}k}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk}(\bar{\mathbf{P}}_{tq}^{kH}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk} + N_0\mathbf{I}_{\bar{M}_{tq}^{qk}})^{-1}, \tag{6.46}$$

where (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$.

6.6.1.2 Error variance

The conditional covariance of $\mathbf{h}_{tji}^{q}$ is calculated conditioned on the knowledge of $\mathbf{z} = \hat{\mathbf{h}}_{tji}^{qk}$. Let $\bar{\mathbf{C}}_{t}^{qk} \triangleq \bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk}(\bar{\mathbf{P}}_{tq}^{kH}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{qk} + N_0\mathbf{I}_{\bar{M}_{tq}^{qk}})^{-1}$ be split as $\bar{\mathbf{C}}_{t}^{qk} = [\mathbf{C}_{tq}^{qk}, \mathbf{C}_{t1}^{qk}, \ldots, \mathbf{C}_{tq-1}^{qk}, \mathbf{C}_{tq+1}^{qk}, \ldots, \mathbf{C}_{tQ}^{qk}]$, and $\mathbf{c}_{tji}^{qk} \triangleq [\mathbf{C}_{tj}^{qk}]_{:,i}$. Thus, we can evaluate

$$\mathbf{K}_{\mathbf{h}_{tji}^{q}\mathbf{h}_{tji}^{q}} = \mathbb{E}[\mathbf{h}_{tji}^{q}\mathbf{h}_{tji}^{qH}] = \beta_{ji}^{q}\sigma_{\mathtt{h}}^{2}\mathbf{I}_N,$$

$$\mathbf{K}_{\mathbf{h}^q_{tji}\mathbf{z}} = \mathbb{E}[\mathbf{h}^q_{tji}\hat{\mathbf{h}}^{qkH}_{tji}] = \mathbf{p}^H_{ji}\mathbf{c}^{qk}_{tji}g_{tji}\beta^q_{ji}\sigma^2_{\mathtt{h}}\mathbf{I}_N,$$
$$\mathbf{K}_{\mathbf{zz}} = \begin{pmatrix} N_0\|\mathbf{c}^{qk}_{tji}\|^2 + \sum_{n\in\mathcal{S}_{kj}}|\mathbf{p}^H_{qn}\mathbf{c}^{qk}_{tji}|^2 g_{tqn}\beta^q_{qn}\sigma^2_{\mathtt{h}} \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}}|\mathbf{p}^H_{ln}\mathbf{c}^{qk}_{tji}|^2 g_{tln}\beta^q_{ln}\sigma^2_{\mathtt{h}} \end{pmatrix}\mathbf{I}_N.$$

Thus, the conditional covariance is

$$\mathbf{K}_{\mathbf{h}^q_{tji}\mathbf{h}^q_{tji}|\mathbf{z}} = \mathbf{K}_{\mathbf{h}^q_{tji}\mathbf{h}^q_{tji}} - \mathbf{K}_{\mathbf{h}^q_{tji}\mathbf{z}}\mathbf{K}^{-1}_{\mathbf{zz}}\mathbf{K}_{\mathbf{z}\mathbf{h}^q_{tji}} \triangleq \delta^{qk}_{tji}\mathbf{I}_N,$$

where δ^{qk}_{tji} is calculated as

$$\delta^{qk}_{tji} = \beta^q_{ji}\sigma^2_{\mathtt{h}}\left(\frac{N_0\|\mathbf{c}^{qk}_{tji}\|^2 + \sum_{n\in\mathcal{S}^i_{kj}}|\mathbf{p}^H_{qn}\mathbf{c}^{qk}_{tji}|^2 g_{tqn}\beta^q_{qn}\sigma^2_{\mathtt{h}} + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}}|\mathbf{p}^H_{ln}\mathbf{c}^{qk}_{tji}|^2 g_{tln}\beta^q_{ln}\sigma^2_{\mathtt{h}}}{N_0\|\mathbf{c}^{qk}_{tji}\|^2 + \sum_{n\in\mathcal{S}_{kj}}|\mathbf{p}^H_{qn}\mathbf{c}^{qk}_{tji}|^2 g_{tqn}\beta^q_{qn}\sigma^2_{\mathtt{h}} + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}}|\mathbf{p}^H_{ln}\mathbf{c}^{qk}_{tji}|^2 g_{tln}\beta^q_{ln}\sigma^2_{\mathtt{h}}}\right).$$

The conditional autocorrelation follows as

$$\mathbb{E}_{\mathbf{z}}[\mathbf{h}^q_{tji}\mathbf{h}^{qH}_{tji}] = \mathbf{K}_{\mathbf{h}^q_{tji}\mathbf{h}^q_{tji}|\mathbf{z}} + \mathbb{E}_{\mathbf{z}}[\mathbf{h}^q_{tji}]\mathbb{E}_{\mathbf{z}}[\mathbf{h}^q_{tji}]^H = \delta^{qk}_{tji}\mathbf{I}_N + \hat{\mathbf{h}}^{qk}_{tji}\hat{\mathbf{h}}^{qkH}_{tji}.$$

The unconditional and conditional means of the estimation error are $\mathbb{E}[\tilde{\mathbf{h}}^{qk}_{tji}] = \mathbb{E}[\hat{\mathbf{h}}^{qk}_{tji} - \mathbf{h}^q_{tji}] = 0$ and $\mathbb{E}_{\mathbf{z}}[\tilde{\mathbf{h}}^{qk}_{tji}] = \mathbb{E}_{\mathbf{z}}[\hat{\mathbf{h}}^{qk}_{tji} - \mathbf{h}^q_{tji}] = \hat{\mathbf{h}}^{qk}_{tji} - \hat{\mathbf{h}}^{qk}_{tji} = 0$. The conditional autocovariance of the error therefore simplifies as

$$\mathbf{K}_{\tilde{\mathbf{h}}^{qk}_{tji}\tilde{\mathbf{h}}^{qk}_{tji}|\mathbf{z}} = \mathbb{E}_{\mathbf{z}}[\tilde{\mathbf{h}}^{qk}_{tji}\tilde{\mathbf{h}}^{qkH}_{tji}] = \mathbb{E}_{\mathbf{z}}[\mathbf{h}^q_{tji}\mathbf{h}^{qH}_{tji}] - \hat{\mathbf{h}}^{qk}_{tji}\hat{\mathbf{h}}^{qkH}_{tji} = \delta^{qk}_{tji}\mathbf{I}_N,$$

and thus, δ^{qk}_{tji} is also the variance of the estimation error.

6.6.2 Proof of Theorem 6.1: SINR Computation

In order to calculate the SINR, we first evaluate the power of the received signal, which is calculated conditioned on the knowledge of the channel estimates $\mathbf{z} \triangleq \text{vec}(\hat{\tilde{\mathbf{H}}}^{qk}_{tq})$ as $\mathbb{E}_{\mathbf{z}}[|\tilde{y}^k_{tqm}|^2] = \mathbb{E}_{\mathbf{z}}[|\sum_{i=1}^5 T_i|^2]$. Since noise is uncorrelated with data, $\mathbb{E}_{\mathbf{z}}[T_1T_5^H] = \mathbb{E}_{\mathbf{z}}[T_2T_5^H] = \mathbb{E}_{\mathbf{z}}[T_3T_5^H] = \mathbb{E}_{\mathbf{z}}[T_4T_5^H] = 0$. Since MMSE estimates are uncorrelated with their errors [49], $\mathbb{E}_{\mathbf{z}}[T_1T_2^H] = 0$. Finding the other components requires $\mathbb{E}_{\mathbf{z}}[x_{ji}x_{jl}]$ for $i \neq l$ which can

be found as $\mathbb{E}_{\mathbf{z}}[x_{ji}x_{jl}] = \mathbb{E}_{\mathbf{z}}[x_{ji}]\mathbb{E}_{\mathbf{z}}[x_{jl}] = 0$. Thus, all the five terms are uncorrelated and the power in the received signal is just a sum of the powers of the individual components $\mathbb{E}_{\mathbf{z}}[|\tilde{y}^k_{tqm}|^2] = \sum_{i=1}^{5} \mathbb{E}_{\mathbf{z}}[|T_i|^2]$. We now compute the powers of each of the components. The useful signal power is

$$\mathbb{E}_{\mathbf{z}}[|T_1|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}^{kH}_{tqm}\hat{\mathbf{h}}^{qk}_{tqm}g_{tqm}x_{qm}|^2] = p_{qm}g^2_{tqm}|\mathbf{a}^{kH}_{tqm}\hat{\mathbf{h}}^{qk}_{tqm}|^2.$$

The desired gain is written as

$$\mathtt{Gain}^k_{tqm} \triangleq \frac{\mathbb{E}_{\mathbf{z}}[|T_1|^2]}{\|\mathbf{a}^k_{tqm}\|^2} = p_{qm}g_{tqm}\frac{|\mathbf{a}^{kH}_{tqm}\hat{\mathbf{h}}^{qk}_{tqm}|^2}{\|\mathbf{a}^k_{tqm}\|^2}. \tag{6.47}$$

The power of the estimation error is expressed as

$$\mathbb{E}_{\mathbf{z}}[|T_2|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}^{kH}_{tqm}\tilde{\mathbf{h}}^{qk}_{tqm}g_{tqm}x_{qm}|^2] = p_{qm}g^2_{tqm}\delta^{qk}_{tqm}\|\mathbf{a}^k_{tqm}\|^2.$$

Next, the power of the intra-cell interference term T_3 is

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}}[|T_3|^2] &= \mathbb{E}_{\mathbf{z}}[|\mathbf{a}^{kH}_{tqm}\textstyle\sum_{i\in\mathcal{S}^m_{kq}} g_{tqi}\mathbf{h}^q_{tqi}x_{qi}|^2] \\
&= \textstyle\sum_{i\in\mathcal{S}^m_{kq}} p_{qi}g^2_{tqi}\mathbf{a}^{kH}_{tqm}\mathbb{E}_{\mathbf{z}}[\mathbf{h}^q_{tqi}\mathbf{h}^{qH}_{tqi}]\mathbf{a}^k_{tqm} \\
&= \textstyle\sum_{i\in\mathcal{S}^m_{kq}} p_{qi}g^2_{tqi}\mathbf{a}^{kH}_{tqm}(\delta^{qk}_{tqi}\mathbf{I}_N + \hat{\mathbf{h}}^{qk}_{tqi}\hat{\mathbf{h}}^{qkH}_{tqi})\mathbf{a}^k_{tqm} \\
&= \textstyle\sum_{i\in\mathcal{S}^m_{kq}} p_{qi}g^2_{tqi}(\|\mathbf{a}^k_{tqm}\|^2\delta^{qk}_{tqi} + |\mathbf{a}^{kH}_{tqm}\hat{\mathbf{h}}^{qk}_{tqi}|^2).
\end{aligned}$$

Then, the power of the inter-cell interference term T_4 is

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}}[|T_4|^2] &= \mathbb{E}_{\mathbf{z}}[|\mathbf{a}^{kH}_{tqm}\textstyle\sum_{j\in\mathcal{Q}^q}\sum_{i\in\mathcal{S}_{1j}} g_{tji}\mathbf{h}^q_{tji}x_{ji}|^2] \\
&= \textstyle\sum_{j\in\mathcal{Q}^q}\sum_{i\in\mathcal{S}_{1j}} p_{ji}g^2_{tji}\mathbf{a}^{kH}_{tqm}\mathbb{E}_{\mathbf{z}}[\mathbf{h}^q_{tji}\mathbf{h}^{qH}_{tji}]\mathbf{a}^k_{tqm} \\
&= \textstyle\sum_{j\in\mathcal{Q}^q}\sum_{i\in\mathcal{S}_{1j}} p_{ji}g^2_{tji}\mathbf{a}^{kH}_{tqm}(\delta^{qk}_{tji}\mathbf{I}_N + \hat{\mathbf{h}}^{qk}_{tji}\hat{\mathbf{h}}^{qkH}_{tji})\mathbf{a}^k_{tqm} \\
&= \textstyle\sum_{j\in\mathcal{Q}^q}\sum_{i\in\mathcal{S}_{1j}} p_{ji}g^2_{tji}(\|\mathbf{a}^k_{tqm}\|^2\delta^{qk}_{tji} + |\mathbf{a}^{kH}_{tqm}\hat{\mathbf{h}}^{qk}_{tji}|^2).
\end{aligned}$$

Let $P_{\mathtt{int}} = \mathbb{E}_{\mathbf{z}}[|T_2|^2] + \mathbb{E}_{\mathbf{z}}[|T_3|^2] + \mathbb{E}_{\mathbf{z}}[|T_4|^2]$ represent the joint contribution of estimation

errors and multi-user interference components of the other users (both within the qth cell and outside the qth cell). Since g_{tji} is binary, its powers are dropped. We now split $P_{\texttt{int}}/\|\mathbf{a}_{tqm}^k\|^2$ into the sum of the estimation error component $\texttt{Est}_{tqm}^k$, intra-cell interference $\texttt{InCI}_{tqm}^k$ and inter-cell interference $\texttt{ICI}_{tqm}^k$ as follows

$$\begin{aligned}
\texttt{Est}_{tqm}^k &= \textstyle\sum_{i\in\mathcal{S}_{kq}} p_{qi} g_{tqi} \delta_{tqi}^{qk} + \sum_{j\in\mathcal{Q}^q} \sum_{i\in\mathcal{S}_{1j}} p_{ji} g_{tji} \delta_{tji}^{qk}, \\
\texttt{InCI}_{tqm}^k &= \textstyle\sum_{i\in\mathcal{S}_{kq}^m} p_{qi} g_{tqi} |\mathbf{a}_{tqm}^{kH} \hat{\mathbf{h}}_{tqi}^{qk}|^2 / \|\mathbf{a}_{tqm}^k\|^2, \\
\texttt{ICI}_{tqm}^k &= \textstyle\sum_{j\in\mathcal{Q}^q} \sum_{i\in\mathcal{S}_{1j}} p_{ji} g_{tji} |\mathbf{a}_{tqm}^{kH} \hat{\mathbf{h}}_{tji}^{qk}|^2 / \|\mathbf{a}_{tqm}^k\|^2.
\end{aligned}$$

The noise power is calculated as

$$\mathbb{E}_{\mathbf{z}}[|T_5|^2] = \mathbb{E}_{\mathbf{z}}[|\mathbf{a}_{tqm}^k \mathbf{n}_{tq}|^2] = N_0 \|\mathbf{a}_{tqm}^k\|^2. \tag{6.48}$$

A meaningful SINR expression can be written out by dividing the useful gain from (6.47) by the sum of the interference and the noise powers (from $P_{\texttt{int}}$ and (6.48)) [49]. Note that the interference component is comprised of the estimation error term and the signal powers of other users who have also transmitted in the same RB (from both in-cell and out-of-cell users). SINR can thus be evaluated as in (6.7) for all users. The SINR can be calculated by plugging in the channel estimates as detailed in Theorem 6.1.

6.6.3 Proof of Lemma 8: Massive MIMO Equivalent

As the number of antennas gets large, both $\|\hat{\mathbf{h}}_{tqm}^{qk}\|^2$ and $|\hat{\mathbf{h}}_{tqm}^{qkH} \hat{\mathbf{h}}_{tji}^{qk}|^2$ converge almost surely (a.s.) to their deterministic equivalents [106]. Evaluating the deterministic equivalents as in [106] and plugging into the SINR expression in place of the original terms, we can find an approximation to the SINR in the high antenna regime. As N gets large, the SINR with maximal ratio combining converges almost surely ($\rho_{tqm}^k \xrightarrow{\text{a.s.}} \overline{\rho}_{tqm}^k$) to

$$\overline{\rho}_{tqm}^k = \frac{\texttt{Sig}_{tqm}^k}{\epsilon_{tqm}^k (N_0 + \texttt{IntNC}_{tqm}^k) + \texttt{IntC}_{tqm}^k},$$

where $\mathtt{Sig}_{tqm}^{k}$ is the desired signal, $\mathtt{IntNC}_{tqm}^{k}$ represents the non-coherent interference, and $\mathtt{IntC}_{tqm}^{k}$ represents the coherent interference. These can be evaluated as

$$\begin{aligned}
\epsilon_{tqm}^{k} &= \left(\begin{array}{l} N_0\|\mathbf{c}_{tqm}^{qk}\|^2 + \sum_{n\in\mathcal{S}_{kj}} |\mathbf{p}_{qn}^H \mathbf{c}_{tqm}^{qk}|^2 g_{tqn}\beta_{qn}^{q}\sigma_{\mathtt{h}}^2 \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}} |\mathbf{p}_{ln}^H \mathbf{c}_{tqm}^{qk}|^2 g_{tln}\beta_{ln}^{q}\sigma_{\mathtt{h}}^2 \end{array} \right), \\
\mathtt{Sig}_{tqm}^{k} &= N p_{qm} g_{tqm} (\epsilon_{tqm}^{k})^2, \\
\mathtt{IntNC}_{tqm}^{k} &= \left(\begin{array}{l} p_{qm} g_{tqm} \delta_{tqm}^{qk} + \sum_{n\in\mathcal{S}_{kj}} p_{qn} g_{tqn}\beta_{qn}^{q}\sigma_{\mathtt{h}}^2 \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}} p_{ln} g_{tln}\beta_{ln}^{q}\sigma_{\mathtt{h}}^2 \end{array} \right), \\
\mathtt{IntC}_{tqm}^{k} &= N \left(\begin{array}{l} \sum_{n\in\mathcal{S}_{kj}} |\mathbf{p}_{qn}^H \mathbf{c}_{tqm}^{qk}|^2 p_{qn} g_{tqn}\beta_{qn}^{q2}\sigma_{\mathtt{h}}^4 \\ + \sum_{l\in\mathcal{Q}^q}\sum_{n\in\mathcal{S}_{1j}} |\mathbf{p}_{ln}^H \mathbf{c}_{tqm}^{qk}|^2 p_{ln} g_{tln}\beta_{ln}^{q2}\sigma_{\mathtt{h}}^4 \end{array} \right).
\end{aligned}$$

Here, δ_{tqm}^{qk} and $\mathbf{c}_{tqm}^{qk}$ are obtained from Lemma 7 and Theorem 6.1, respectively, for the three estimation schemes. The above expressions are obtained by setting $\mathbf{a}_{tqm}^{k} = \hat{\mathbf{h}}_{tqm}^{qk}$ [49] and replacing each of the terms involving $\hat{\mathbf{h}}_{tji}^{qk}$ in (6.7) with their respective deterministic equivalents.

6.6.4 Proof of Theorem 6.2: LCF IRSA

6.6.4.1 Channel Estimation

Let k denote the current decoding iteration index, and $\mathcal{S}_{kq}$ denote the set of users not yet decoded up to the kth decoding iteration at the qth AP, with $\mathcal{S}_{1q} = [M]$ and $\mathcal{S}_{kq}^{i} \triangleq \mathcal{S}_{kq}\backslash\{i\}$. As seen before, the received pilot signal at the qth AP in the tth RB in the kth decoding iteration is

$$\mathbf{Y}_{tq}^{\mathtt{p}k} = \sum_{i\in\mathcal{S}_{kq}} g_{ti}\mathbf{h}_{ti}^{q}\mathbf{p}_{i}^{H} + \mathbf{N}_{tq}^{\mathtt{p}}, \tag{6.49}$$

where the first term contains signals from users who have not yet been decoded up to the kth decoding iteration at the qth AP, i.e., $\forall i \in \mathcal{S}_{kq}$.

We now derive the MMSE channel estimates at the qth AP in each RB. Let $\mathcal{G}_t = \{i \in [M] | g_{ti} = 1\}$ be the set of users who have transmitted in the tth RB. Let $M_{tq}^{k} = |\mathcal{G}_t \cap \mathcal{S}_{kq}|$

be the number of users who have transmitted in the tth RB and have not been decoded in the first $k-1$ iterations at the qth AP. Let us stack the channels of the M_{tq}^k users as the columns of the matrix $\mathbf{H}_{tq}^k \in \mathbb{C}^{N \times M_{tq}^k}$, and let $\mathbf{P}_{tq}^k \in \mathbb{C}^{\tau \times M_{tq}^k}$ denote a matrix that contains the pilot sequences of the M_{tq}^k users as its columns. Let $\mathbf{B}_{tq}^k \triangleq \sigma_{\mathtt{h}}^2 \text{diag}(\beta_{i_1}^q, \beta_{i_2}^q, \ldots, \beta_{i_{M_{tq}^k}}^q)$ be a diagonal matrix that contains the path loss coefficients of the M_{tq}^k users, with $\mathcal{G}_t \cap \mathcal{S}_{kq} = \{i_1, i_2, \ldots, i_{M_{tq}^k}\}$. Hence, the received pilot signal from (6.49) can be written as $\mathbf{Y}_{tq}^{\mathtt{p}k} = \mathbf{H}_{tq}^k \mathbf{P}_{tq}^{kH} + \mathbf{N}_{tq}^{\mathtt{p}}$.

The MMSE channel estimate $\hat{\mathbf{H}}_{tq}^k$ of $\mathbf{H}_{tq}^k$ can be computed similar to Chapter 3 (see Section 3.8.1) as

$$\hat{\mathbf{H}}_{tq}^k = \mathbf{Y}_{tq}^{\mathtt{p}k} (\mathbf{P}_{tq}^k \mathbf{B}_{tq}^k \mathbf{P}_{tq}^{kH} + N_0 \mathbf{I}_\tau)^{-1} \mathbf{P}_{tq}^k \mathbf{B}_{tq}^k, \tag{6.50a}$$

$$\overset{(a)}{=} \mathbf{Y}_{tq}^{\mathtt{p}k} \mathbf{P}_{tq}^k \mathbf{B}_{tq}^k (\mathbf{P}_{tq}^{kH} \mathbf{P}_{tq}^k \mathbf{B}_{tq}^k + N_0 \mathbf{I}_{M_{tq}^k})^{-1}, \tag{6.50b}$$

where (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$. Here, the estimate can be calculated via an inverse of either a $\tau \times \tau$ matrix or an $M_{tq}^k \times M_{tq}^k$ matrix as required. Specifically, the estimate of the channel $\mathbf{h}_{ti}^q$ of the ith user at the qth AP is calculated as $\hat{\mathbf{h}}_{ti}^{qk} = [\hat{\mathbf{H}}_{tq}^k]_{:,i}$.

Let $\mathbf{C}_t^{qk} \triangleq \mathbf{P}_{tq}^k \mathbf{B}_{tq}^k (\mathbf{P}_{tq}^{kH} \mathbf{P}_{tq}^k \mathbf{B}_{tq}^k + N_0 \mathbf{I}_{M_{tq}^k})^{-1}$, and $\mathbf{c}_{ti}^{qk} \triangleq [\mathbf{C}_t^{qk}]_{:,i}$. Further, the estimation error $\tilde{\mathbf{h}}_{ti}^{qk} \triangleq \hat{\mathbf{h}}_{ti}^{qk} - \mathbf{h}_{ti}^q$ is distributed as $\tilde{\mathbf{h}}_{ti}^{qk} \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^{qk} \mathbf{I}_N)$, where δ_{ti}^{qk} is calculated as

$$\delta_{ti}^{qk} = \beta_i^q \sigma_{\mathtt{h}}^2 \left(\frac{N_0 \|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n \in \mathcal{S}_{kq}^i} |\mathbf{p}_n^H \mathbf{c}_{ti}^{qk}|^2 g_{tn} \beta_n^q \sigma_{\mathtt{h}}^2}{N_0 \|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n \in \mathcal{S}_{kq}} |\mathbf{p}_n^H \mathbf{c}_{ti}^{qk}|^2 g_{tn} \beta_n^q \sigma_{\mathtt{h}}^2} \right). \tag{6.51}$$

6.6.4.2 Data Decoding

Similar to (6.49), we can find the received data signal at the qth AP in the tth RB in the kth decoding iteration as

$$\mathbf{y}_{tq}^k = \sum_{i \in \mathcal{S}_{kq}} g_{ti} \mathbf{h}_{ti}^q x_i + \mathbf{n}_{tq}. \tag{6.52}$$

Let ρ_{tqm}^k denote the SINR of the mth user in the tth RB in the kth decoding iteration at the qth AP. Using a combining vector $\mathbf{a}_{tm}^{qk}$, the qth AP obtains the post-combined data

signal $\tilde{y}_{tm}^{qk}$ as $\tilde{y}_{tm}^{qk} = \mathbf{a}_{tm}^{qkH}\mathbf{y}_{tq}^{k}$, where $\mathbf{y}_{tq}^{k}$ is from (6.52). Thus, we obtain

$$\tilde{y}_{tm}^{qk} = \mathbf{a}_{tm}^{qkH}\hat{\mathbf{h}}_{tm}^{qk}g_{tm}x_m - \mathbf{a}_{tm}^{qkH}\tilde{\mathbf{h}}_{tm}^{qk}g_{tm}x_m + \mathbf{a}_{tm}^{qkH}\sum_{i\in\mathcal{S}_{kq}^{m}} g_{ti}\mathbf{h}_{ti}^{q}x_i + \mathbf{a}_{tm}^{qkH}\mathbf{n}_{tq}. \tag{6.53}$$

Here the first term on the RHS, $\mathbf{a}_{tm}^{qkH}\hat{\mathbf{h}}_{tm}^{qk}g_{tm}x_m$, denotes the useful signal component and it contains the channel estimate $\hat{\mathbf{h}}_{tm}^{qk}$ of the mth user; the second term $\mathbf{a}_{tm}^{qkH}\tilde{\mathbf{h}}_{tm}^{qk}g_{tm}x_m$ contains $\tilde{\mathbf{h}}_{tm}^{qk}$, the estimation error of the mth user; the third term $\mathbf{a}_{tm}^{qkH}\sum_{i\in\mathcal{S}_{kq}^{m}} g_{ti}\mathbf{h}_{ti}^{q}x_i$ is the multi-user interference faced by the mth user due to the other users in the entire network who have not yet been decoded up to the kth decoding iteration at the qth AP; and the fourth term $\mathbf{a}_{tm}^{qkH}\mathbf{n}_{tq}$ is the additive noise component.

In order to evaluate the SINR, we first calculate the power of the received signal, which is calculated conditioned on the knowledge of the estimates. Similar to Chapter 3, all the four terms are uncorrelated and the power in the received signal is just a sum of the powers of the individual components. A meaningful SINR expression can be written out by dividing the useful signal power from by the sum of the interference and the noise powers [49]. Thus, the signal to interference plus noise ratio (SINR) achieved by the mth user at the qth AP in the tth RB and the kth decoding iteration can be calculated similar to Chapter 3 (see Section 3.8.2) and can be written as

$$\rho_{tqm}^{k} = \frac{\mathtt{Gain}_{tqm}^{k}}{N_0 + \mathtt{Est}_{tqm}^{k} + \mathtt{MUI}_{tqm}^{k}}, \forall m \in \mathcal{S}_{kq}, \tag{6.54}$$

where

$$\begin{aligned}
\mathtt{Gain}_{tqm}^{k} &\triangleq p_m g_{tm}|\mathbf{a}_{tm}^{qkH}\hat{\mathbf{h}}_{tm}^{qk}|^2/\|\mathbf{a}_{tm}^{qk}\|^2,\\
\mathtt{Est}_{tqm}^{k} &\triangleq \textstyle\sum_{i\in\mathcal{S}_{kq}} p_i g_{ti}\delta_{ti}^{qk},\\
\mathtt{MUI}_{tqm}^{k} &\triangleq \textstyle\sum_{i\in\mathcal{S}_{kq}^{m}} p_i g_{ti}|\mathbf{a}_{tm}^{qkH}\hat{\mathbf{h}}_{ti}^{qk}|^2/\|\mathbf{a}_{tm}^{qk}\|^2.
\end{aligned}$$

6.6.5 Proof of Theorem 6.3: CCF IRSA

6.6.5.1 Channel Estimation

In the channel estimation phase, the CPU computes channel estimates $\hat{\mathbf{h}}_{ti}^{qk}$ of all the users similar to (6.17a) and then finally obtains an effective CPU channel estimate.

Let $\mathcal{S}_k$ denote the set of users not yet decoded at the CPU after $k-1$ decoding iterations, with $\mathcal{S}_1 = [M]$, and $\mathcal{S}_k^i \triangleq \mathcal{S}_k \setminus \{i\}$. At the CPU, the received pilot signal of the qth AP in the kth decoding iteration is given by

$$\mathbf{Y}_{tq}^{\mathtt{p}k} = \textstyle\sum_{i\in\mathcal{S}_k} g_{ti}\mathbf{h}_{ti}^{q}\mathbf{p}_i^H + \mathbf{N}_{tq}^{\mathtt{p}}. \tag{6.55}$$

The channel estimates can be obtained similar to LCF IRSA as

$$\hat{\mathbf{H}}_{tq}^{k} = \mathbf{Y}_{tq}^{\mathtt{p}k}(\mathbf{P}_t^k\mathbf{B}_{tq}^k\mathbf{P}_t^{kH} + N_0\mathbf{I}_\tau)^{-1}\mathbf{P}_t^k\mathbf{B}_{tq}^k, \tag{6.56a}$$

$$\overset{(a)}{=} \mathbf{Y}_{tq}^{\mathtt{p}k}\mathbf{P}_t^k\mathbf{B}_{tq}^k(\mathbf{P}_t^{kH}\mathbf{P}_t^k\mathbf{B}_{tq}^k + N_0\mathbf{I}_{M_t^k})^{-1}, \tag{6.56b}$$

Here, $\mathcal{M}_t^k = \mathcal{G}_t \cap \mathcal{S}_k$, and $M_t^k = |\mathcal{M}_t^k|$ is the number of users yet to be decoded, $\mathbf{P}_t^k$ contains as its columns the pilots of the M_t^k users yet to be decoded, and $\mathbf{B}_{tq}^k \triangleq \sigma_{\mathtt{h}}^2\mathrm{diag}(\beta_{i_1}^q, \beta_{i_2}^q, \ldots, \beta_{i_{M_t^k}}^q)$ contains the path loss coefficients of the M_t^k users, with $\mathcal{M}_t^k = \{i_1, i_2, \ldots, i_{M_t^k}\}$. Here, the estimate can be calculated via an inverse of either a $\tau\times\tau$ matrix or an $M_t^k \times M_t^k$ matrix as required: (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$ [144]. Specifically, the CPU estimate of the channel $\mathbf{h}_{ti}^q$ of the ith user to the qth AP is calculated as $\hat{\mathbf{h}}_{ti}^{qk} = [\hat{\mathbf{H}}_{tq}^k]_{:,i}$.

For the ith user, the effective channel and the channel estimate at the CPU is found by stacking the channels and the channel estimates as

$$\mathbf{h}_{ti} = [\mathbf{h}_{ti}^{1H}, \mathbf{h}_{ti}^{2H}, \ldots, \mathbf{h}_{ti}^{QH}]^H \in \mathbb{C}^{QN\times 1}, \tag{6.57}$$

$$\hat{\mathbf{h}}_{ti}^{k} = [\hat{\mathbf{h}}_{ti}^{1kH}, \hat{\mathbf{h}}_{ti}^{2kH}, \ldots, \hat{\mathbf{h}}_{ti}^{QkH}]^H \in \mathbb{C}^{QN\times 1}. \tag{6.58}$$

Combining the channel estimates as above helps in exploiting the distributed array gain due to the inherent locations of the APs.

Let $\mathbf{C}_t^{qk} \triangleq \mathbf{P}_t^k\mathbf{B}_{tq}^k(\mathbf{P}_t^{kH}\mathbf{P}_t^k\mathbf{B}_{tq}^k + N_0\mathbf{I}_{M_t^k})^{-1}$, and $\mathbf{c}_{ti}^{qk} \triangleq [\mathbf{C}_t^{qk}]_{:,i}$. The estimation error

$\tilde{\mathbf{h}}_{ti}^{qk} \triangleq \hat{\mathbf{h}}_{ti}^{qk} - \mathbf{h}_{ti}^{q}$ is distributed as $\tilde{\mathbf{h}}_{ti}^{qk} \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^{qk}\mathbf{I}_N)$, where δ_{ti}^{qk} can be calculated similar to LCF IRSA as

$$\delta_{ti}^{qk} = \beta_i^q \sigma_{\mathtt{h}}^2 \left(\frac{N_0 \|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n \in \mathcal{S}_k^i} |\mathbf{p}_n^H \mathbf{c}_{ti}^{qk}|^2 g_{tn} \beta_n^q \sigma_{\mathtt{h}}^2}{N_0 \|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n \in \mathcal{S}_k} |\mathbf{p}_n^H \mathbf{c}_{ti}^{qk}|^2 g_{tn} \beta_n^q \sigma_{\mathtt{h}}^2} \right). \tag{6.59}$$

The channel estimation error at the CPU is given by $\tilde{\mathbf{h}}_{ti}^{k} \triangleq \hat{\mathbf{h}}_{ti}^{k} - \mathbf{h}_{ti}$. If $\mathbf{C} = \text{blkdiag}(\mathbf{A}, \mathbf{B})$ and $\mathbb{E}[\mathbf{A}\mathbf{B}^H] = 0$, then covariance($\mathbf{C}$) = blkdiag(covariance($\mathbf{A}$),covariance($\mathbf{B}$)) [105, 144]. Thus, the covariance of the estimation error at the CPU is $\Delta_{ti}^{k} \triangleq \text{blkdiag}(\delta_{ti}^{1k}\mathbf{I}_N, \delta_{ti}^{2k}\mathbf{I}_N, \ldots, \delta_{ti}^{Qk}\mathbf{I}_N)$.

6.6.5.2 Data Decoding

Let ρ_{tm}^{k} denote the SINR of the mth user in the tth RB in the kth decoding iteration at the CPU. The received data signal can be stacked as $\mathbf{y}_t = [\mathbf{y}_{t1}^H, \mathbf{y}_{t2}^H, \ldots, \mathbf{y}_{tQ}^H]$. Similar to (6.55), in the kth decoding iteration at the CPU, the received data signal can be expressed as

$$\mathbf{y}_t^k = \sum_{i \in \mathcal{S}_k} g_{ti} \mathbf{h}_{ti} x_i + \mathbf{n}_t, \tag{6.60}$$

where $\mathbf{n}_t \triangleq [\mathbf{n}_{t1}^H, \mathbf{n}_{t2}^H, \ldots, \mathbf{n}_{tQ}^H]^H$. The CPU uses a combining vector $\mathbf{a}_{tm}^k$ to combine the received data signal. Thus, we obtain the post-combined data signal as $\tilde{y}_{tm}^k = \mathbf{a}_{tm}^{kH} \mathbf{y}_t^k$ which is expanded as

$$\tilde{y}_{tm}^k = \mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k g_{tm} x_m - \mathbf{a}_{tm}^{kH} \tilde{\mathbf{h}}_{tm}^k g_{tm} x_m + \mathbf{a}_{tm}^{kH} \sum_{i \in \mathcal{S}_k^m} g_{ti} \mathbf{h}_{ti} x_i + \mathbf{a}_{tm}^{kH} \mathbf{n}_t. \tag{6.61}$$

The first term on the RHS, $\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^k g_{tm} x_m$, denotes the useful signal component and it contains the CPU channel estimate $\hat{\mathbf{h}}_{tm}^k$ of the mth user; the second term $\mathbf{a}_{tm}^{kH} \tilde{\mathbf{h}}_{tm}^k g_{tm} x_m$ contains $\tilde{\mathbf{h}}_{tm}^k$, the CPU estimation error of the mth user; the third term $\mathbf{a}_{tm}^{kH} \sum_{i \in \mathcal{S}_k^m} g_{ti} \mathbf{h}_{ti} x_i$ is the multi-user interference faced by the mth user due to the other users in the network who have not yet been decoded up to the kth decoding iteration at the CPU; and the fourth term $\mathbf{a}_{tm}^{kH} \mathbf{n}_t$ is the additive noise component. We need to compute the SINR from

the above post-combined data signal. All the terms in the above equation are pairwise uncorrelated. The structure of (6.61) resembles the structure of the post-combined signal in Chapter 3, but with QN-length signals instead of N-length signals. We can derive the SINR similar to Chapter 3. The SINR achieved by the mth user at the CPU in the tth RB and the kth decoding iteration can thus be calculated as

$$\rho_{tm}^{k} = \frac{\mathtt{Gain}_{tm}^{k}}{N_0 + \mathtt{Est}_{tm}^{k} + \mathtt{MUI}_{tm}^{k}}, \forall m \in \mathcal{S}_k, \tag{6.62}$$

where

$$\begin{aligned}
\mathtt{Gain}_{tm}^{k} &\triangleq p_m g_{tm} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^{k}|^2 / \|\mathbf{a}_{tm}^{k}\|^2, \\
\mathtt{Est}_{tm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{S}_k} p_i g_{ti} \mathbf{a}_{tm}^{kH} \Delta_{ti}^{k} \mathbf{a}_{tm}^{k} / \|\mathbf{a}_{tm}^{k}\|^2, \\
\mathtt{MUI}_{tm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{S}_k^m} p_i g_{ti} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{ti}^{k}|^2 / \|\mathbf{a}_{tm}^{k}\|^2.
\end{aligned}$$

6.6.6 Proof of Theorem 6.4: HCF IRSA

6.6.6.1 Channel Estimation

In the channel estimation process in phase 2 of HCF IRSA, the CPU computes channel estimates $\hat{\mathbf{h}}_{ti}^{qk}$ of all the users similar to (6.27a) and then finally obtains an effective CPU channel estimate.

Let $\bar{\mathcal{S}}_q$ denote the set of decoded users at the qth AP at the end of the APs decoding process in the first phase. The set of users that have been decoded across the entire network is given by $\bar{\mathcal{S}} = \bar{\mathcal{S}}_1 \cup \bar{\mathcal{S}}_2 \cup \cdots \cup \bar{\mathcal{S}}_Q$. We now denote $\mathcal{U}_k$ as the set of users not decoded up to the kth CPU decoding iteration, with $\mathcal{U}_1 = [M] \setminus \bar{\mathcal{S}}$ and $\mathcal{U}_k^i \triangleq \mathcal{U}_k \setminus \{i\}$. $\mathcal{U}_1$ contains only the signals of the users who need to be decoded at the CPU since they have not been decoded at any of the APs.

At the CPU, the processed residual pilot signal from the qth AP in the kth CPU decoding iteration is given by

$$\mathbf{Z}_{tq}^{\mathtt{p}k} = \textstyle\sum_{i \in \mathcal{U}_k} g_{ti} \mathbf{h}_{ti}^{q} \mathbf{p}_i^{H} + \mathbf{D}_{tq}^{\mathtt{p}} + \mathbf{N}_{tq}^{\mathtt{p}}. \tag{6.63}$$

We now need to derive the channel estimates at the CPU in phase 2, which are computed based on $\mathbf{Z}_{tq}^{\mathtt{p}k}$. Based on the notation setup in Sec. 6.3.4.2, the channel estimates can be obtained similar to CCF IRSA (6.23) as

$$\hat{\bar{\mathbf{H}}}_{tq}^{k} = \mathbf{Z}_{tq}^{\mathtt{p}k}(\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{k}\bar{\mathbf{P}}_{tq}^{kH} + N_0\mathbf{I}_{\tau})^{-1}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{k}, \tag{6.64a}$$

$$\overset{(a)}{=} \mathbf{Z}_{tq}^{\mathtt{p}k}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{k}(\bar{\mathbf{P}}_{tq}^{kH}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{k} + N_0\mathbf{I}_{\bar{U}_{tq}^{k}})^{-1}, \tag{6.64b}$$

Here, the estimate can be calculated via an inverse of either a $\tau \times \tau$ matrix or an $\bar{U}_{tq}^{k} \times \bar{U}_{tq}^{k}$ matrix as required: (a) follows from $(\mathbf{AB}+\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{BA}+\mathbf{I})^{-1}$ [144]. The subset of the users who are yet to be decoded at the CPU up to the kth CPU decoding iteration is $\mathcal{G}_t \cap \mathcal{U}_k$, for whom the channel estimate matrix is obtained as $\hat{\mathbf{H}}_{tq}^{k} = [\hat{\bar{\mathbf{H}}}_{tq}^{k}]_{:,\mathcal{G}_t\cap\mathcal{U}_k}$. Finally, the channel estimate of the channel $\mathbf{h}_{ti}^{q}$ of the ith user to the qth AP is calculated as $\hat{\mathbf{h}}_{ti}^{qk} = [\hat{\mathbf{H}}_{tq}^{k}]_{:,i}$. For the ith user, the effective channel and the channel estimate at the CPU is found by stacking the channels and the channel estimates as

$$\mathbf{h}_{ti} = [\mathbf{h}_{ti}^{1H}, \mathbf{h}_{ti}^{2H}, \ldots, \mathbf{h}_{ti}^{QH}]^{H} \in \mathbb{C}^{QN\times 1}, \tag{6.65}$$

$$\hat{\mathbf{h}}_{ti}^{k} = [\hat{\mathbf{h}}_{ti}^{1kH}, \hat{\mathbf{h}}_{ti}^{2kH}, \ldots, \hat{\mathbf{h}}_{ti}^{QkH}]^{H} \in \mathbb{C}^{QN\times 1}. \tag{6.66}$$

Combining the channel estimates as above helps in exploiting the distributed array gain due to the inherent locations of the APs.

Let $\bar{\mathbf{C}}_{t}^{qk} \triangleq \bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{k}(\bar{\mathbf{P}}_{tq}^{kH}\bar{\mathbf{P}}_{tq}^{k}\bar{\mathbf{B}}_{tq}^{k} + N_0\mathbf{I}_{\bar{U}_{tq}^{k}})^{-1}$, and $\mathbf{c}_{ti}^{qk} \triangleq [\bar{\mathbf{C}}_{t}^{qk}]_{:,i}$. The estimation error $\tilde{\mathbf{h}}_{ti}^{qk} \triangleq \hat{\mathbf{h}}_{ti}^{qk} - \mathbf{h}_{ti}^{q}$ is distributed as $\tilde{\mathbf{h}}_{ti}^{qk} \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^{qk}\mathbf{I}_N)$, where δ_{ti}^{qk} can be calculated similar to CCF IRSA as

$$\delta_{ti}^{qk} = \beta_{i}^{q}\sigma_{\mathtt{h}}^{2}\left(\frac{N_0\|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n\in\bar{\mathcal{U}}_{tq}^{k}\setminus\{i\}}|\mathbf{p}_{n}^{H}\mathbf{c}_{ti}^{qk}|^2 g_{tn}\beta_{n}^{q}\sigma_{\mathtt{h}}^{2}}{N_0\|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n\in\bar{\mathcal{U}}_{tq}^{k}}|\mathbf{p}_{n}^{H}\mathbf{c}_{ti}^{qk}|^2 g_{tn}\beta_{n}^{q}\sigma_{\mathtt{h}}^{2}}\right).$$

The channel estimation error at the CPU is given by $\tilde{\mathbf{h}}_{ti}^{k} \triangleq \hat{\mathbf{h}}_{ti}^{k} - \mathbf{h}_{ti}$. If $\mathbf{C} = \text{blkdiag}(\mathbf{A}, \mathbf{B})$ and $\mathbb{E}[\mathbf{AB}^{H}] = 0$, then covariance$(\mathbf{C})$ = blkdiag(covariance$(\mathbf{A})$,covariance$(\mathbf{B})$) [105, 144]. Thus, the covariance of the estimation error at the CPU is $\Delta_{ti}^{k} \triangleq$ blkdiag$(\delta_{ti}^{1k}\mathbf{I}_N, \delta_{ti}^{2k}\mathbf{I}_N, \ldots, \delta_{ti}^{Qk}\mathbf{I}_N)$.

6.6.6.2 Data Decoding

Let ρ_{tm}^{k} denote the SINR of the mth user in the tth RB in the kth decoding iteration at the CPU. The received data signal can be stacked as $\mathbf{z}_t = [\mathbf{z}_{t1}^H, \mathbf{z}_{t2}^H, \ldots, \mathbf{z}_{tQ}^H]$. Similar to (6.63), in the kth decoding iteration at the CPU, the received data signal can be expressed as

$$\mathbf{z}_t^k = \textstyle\sum_{i\in\mathcal{U}_k} g_{ti}\mathbf{h}_{ti}x_i + \mathbf{d}_t + \mathbf{n}_t. \tag{6.67}$$

where $\mathbf{d}_t \triangleq [\mathbf{d}_{t1}^H, \mathbf{d}_{t2}^H, \ldots, \mathbf{d}_{tQ}^H]^H$, and $\mathbf{n}_t \triangleq [\mathbf{n}_{t1}^H, \mathbf{n}_{t2}^H, \ldots, \mathbf{n}_{tQ}^H]^H$. The CPU uses a combining vector $\mathbf{a}_{tm}^k$ to combine the received data signal. Thus, we obtain the post-combined data signal as $\tilde{z}_{tm}^k = \mathbf{a}_{tm}^{kH}\mathbf{z}_t^k$ which is expanded as

$$\tilde{z}_{tm}^k = \mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k g_{tm}x_m - \mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^k g_{tm}x_m + \mathbf{a}_{tm}^{kH}\textstyle\sum_{i\in\mathcal{U}_k^m} g_{ti}\mathbf{h}_{ti}x_i + \mathbf{a}_{tm}^{kH}\mathbf{d}_t + \mathbf{a}_{tm}^{kH}\mathbf{n}_t. \tag{6.68}$$

The first term on the RHS, $\mathbf{a}_{tm}^{kH}\hat{\mathbf{h}}_{tm}^k g_{tm}x_m$, denotes the useful signal component and it contains the CPU channel estimate $\hat{\mathbf{h}}_{tm}^k$ of the mth user; the second term $\mathbf{a}_{tm}^{kH}\tilde{\mathbf{h}}_{tm}^k g_{tm}x_m$ contains $\tilde{\mathbf{h}}_{tm}^k$, the CPU estimation error of the mth user; the third term $\mathbf{a}_{tm}^{kH}\sum_{i\in\mathcal{U}_k^m} g_{ti}\mathbf{h}_{ti}x_i$ is the multi-user interference faced by the mth user due to the other users in the network who have not yet been decoded up to the kth decoding iteration at the CPU (and who have also not been decoded at any AP in phase 1); the fourth term $\mathbf{a}_{tm}^{kH}\mathbf{d}_t$ is due to the interference from users already decoded at all the APs in phase 1; and the fifth term $\mathbf{a}_{tm}^{kH}\mathbf{n}_t$ is the additive noise component. We need to compute the SINR from the above post-combined data signal. All the terms in the above equation are pairwise uncorrelated. The structure of (6.68) resembles the structure of the post-combined signal in Chapter 3, but with QN-length signals instead of N-length signals. We can derive the SINR similar to Chapter 3. For this, the covariance of the interference due to decoded users in phase 1 reduces to $\Sigma_{\mathbf{d}_t} \triangleq \text{blkdiag}(\bar{d}_{t1}\mathbf{I}_N, \bar{d}_{t2}\mathbf{I}_N, \ldots, \bar{d}_{tQ}\mathbf{I}_N)$, where $\bar{d}_{tq} \triangleq \sum_{i\in\bar{\mathcal{S}}\setminus\bar{\mathcal{S}}_q} p_i g_{ti}\beta_i^q\sigma_{\mathrm{h}}^2$. The SINR achieved by the mth user at the CPU in the tth RB and the kth CPU decoding

iteration can thus be calculated as

$$\rho_{tm}^{k} = \frac{\mathtt{Gain}_{tm}^{k}}{N_0 + \mathtt{Est}_{tm}^{k} + \mathtt{MUI}_{tm}^{k} + \mathtt{Dec}_{tm}^{k}}, \forall m \in \mathcal{U}_k, \tag{6.69}$$

where

$$\begin{aligned}
\mathtt{Gain}_{tm}^{k} &\triangleq p_m g_{tm} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{tm}^{k}|^2 / \|\mathbf{a}_{tm}^{k}\|^2, \\
\mathtt{Est}_{tm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{U}_k} p_i g_{ti} \mathbf{a}_{tm}^{kH} \Delta_{ti}^{k} \mathbf{a}_{tm}^{k} / \|\mathbf{a}_{tm}^{k}\|^2, \\
\mathtt{MUI}_{tm}^{k} &\triangleq \textstyle\sum_{i \in \mathcal{U}_k^m} p_i g_{ti} |\mathbf{a}_{tm}^{kH} \hat{\mathbf{h}}_{ti}^{k}|^2 / \|\mathbf{a}_{tm}^{k}\|^2, \\
\mathtt{Dec}_{tm}^{k} &\triangleq \mathbf{a}_{tm}^{kH} \Sigma_{\mathbf{d}_t} \mathbf{a}_{tm}^{k} / \|\mathbf{a}_{tm}^{k}\|^2.
\end{aligned}$$

6.7 Appendix

6.7.1 HCF IRSA with SIC

There can be an additional SIC phase at the CPU to cancel out the interference of decoded users from the residual signal of each of the Q APs at the CPU. Let the decoding at the qth AP stop at a decoding iteration k_q, i.e., it cannot decode any more users. Thus, the residual received pilot and data signals at the qth AP are $\mathbf{Y}_{tq}^{\mathtt{p}k_q}$ and $\mathbf{y}_{tq}^{k_q}$. The additional SIC phase follows as

$$\mathbf{Z}_{tq}^{\mathtt{p}1} = \mathbf{Y}_{tq}^{\mathtt{p}k_q} - \textstyle\sum_{i \in \bar{\mathcal{S}} \setminus \bar{\mathcal{S}}_q} g_{ti} \mathbf{h}_{ti}^{q} \mathbf{p}_i^{H}, \tag{6.70}$$

$$\mathbf{z}_{tq}^{1} = \mathbf{y}_{tq}^{k_q} - \textstyle\sum_{i \in \bar{\mathcal{S}} \setminus \bar{\mathcal{S}}_q} g_{ti} \mathbf{h}_{ti}^{q} x_i, \tag{6.71}$$

where the processed residual pilot and data signals are denoted by $\mathbf{Z}_{tq}^{\mathtt{p}1}$ and $\mathbf{z}_{tq}^{1}$, respectively.

6.7.2 Channel estimation ignoring PC from decoded users

In the channel estimation process described previously, we can ignore the contamination due to decoded users, similar to the single-cell MMSE channel estimator that ignores the pilot contamination due to pilots from other cells. Then, the effective channel estimate

for all the users becomes

$$\hat{\mathbf{H}}_{tq}^{k} = \mathbf{Z}_{tq}^{\mathtt{p}k}(\mathbf{P}_{t}^{k}\mathbf{B}_{tq}^{k}\mathbf{P}_{t}^{kH} + N_0\mathbf{I}_{\tau})^{-1}\mathbf{P}_{tq}^{k}\mathbf{B}_{tq}^{k}, \tag{6.72a}$$

$$\overset{(a)}{=} \mathbf{Z}_{tq}^{\mathtt{p}k}\mathbf{P}_{t}^{k}\mathbf{B}_{tq}^{k}(\mathbf{P}_{t}^{kH}\mathbf{P}_{t}^{k}\mathbf{B}_{tq}^{k} + N_0\mathbf{I}_{U_t^k})^{-1}, \tag{6.72b}$$

Here, $\mathcal{U}_t^k = \mathcal{G}_t \cap \mathcal{U}_k$, and $U_t^k = |\mathcal{U}_t^k|$ is the number of users yet to be decoded, $\mathbf{P}_t^k$ contains as its columns the pilots of the U_t^k users yet to be decoded, and $\mathbf{B}_{tq}^k \triangleq \sigma_{\mathtt{h}}^2\mathrm{diag}(\beta_{i_1}^q, \beta_{i_2}^q, \ldots, \beta_{i_{U_t^k}}^q)$ contains the path loss coefficients of the U_t^k users, with $\mathcal{U}_t^k = \{i_1, i_2, \ldots, i_{U_t^k}\}$. Specifically, the estimate of the channel $\mathbf{h}_{ti}^q$ of the ith user to the qth AP is calculated as $\hat{\mathbf{h}}_{ti}^{qk} = [\hat{\mathbf{H}}_{tq}^k]_{:,i}$.

Let $\mathbf{C}_t^{qk} \triangleq \mathbf{P}_t^k\mathbf{B}_{tq}^k(\mathbf{P}_t^{kH}\mathbf{P}_t^k\mathbf{B}_{tq}^k + N_0\mathbf{I}_{U_t^k})^{-1}$, and $\mathbf{c}_{ti}^{qk} \triangleq [\mathbf{C}_t^{qk}]_{:,i}$. The estimation error $\tilde{\mathbf{h}}_{ti}^{qk} \triangleq \hat{\mathbf{h}}_{ti}^{qk} - \mathbf{h}_{ti}^q$ is distributed as $\tilde{\mathbf{h}}_{ti}^{qk} \sim \mathcal{CN}(\mathbf{0}_N, \delta_{ti}^{qk}\mathbf{I}_N)$, where δ_{ti}^{qk} is calculated as

$$\delta_{ti}^{qk} = \beta_i^q\sigma_{\mathtt{h}}^2\left(\frac{N_0\|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n\in\mathcal{U}_k^i}|\mathbf{p}_n^H\mathbf{c}_{ti}^{qk}|^2 g_{tn}\beta_n^q\sigma_{\mathtt{h}}^2}{N_0\|\mathbf{c}_{ti}^{qk}\|^2 + \sum_{n\in\mathcal{U}_k}|\mathbf{p}_n^H\mathbf{c}_{ti}^{qk}|^2 g_{tn}\beta_n^q\sigma_{\mathtt{h}}^2}\right). \tag{6.73}$$

The effective channel estimate at the CPU for the ith user is formed by stacking the channel estimates as

$$\hat{\mathbf{h}}_{ti}^k = [\hat{\mathbf{h}}_{ti}^{1kH}, \hat{\mathbf{h}}_{ti}^{2kH}, \ldots, \hat{\mathbf{h}}_{ti}^{QkH}]^H \in \mathbb{C}^{QN\times 1}. \tag{6.74}$$

Thus, the channel estimation error at the CPU is given by $\tilde{\mathbf{h}}_{ti}^k \triangleq \hat{\mathbf{h}}_{ti}^k - \mathbf{h}_{ti}$, where $\mathbf{h}_{ti} = [\mathbf{h}_{ti}^{1H}, \mathbf{h}_{ti}^{2H}, \ldots, \mathbf{h}_{ti}^{QH}]^H \in \mathbb{C}^{QN\times 1}$. The covariance of the estimation error at the CPU is $\Delta_{ti}^k \triangleq \mathrm{blkdiag}(\delta_{ti}^{1k}\mathbf{I}_N, \delta_{ti}^{2k}\mathbf{I}_N, \ldots, \delta_{ti}^{Qk}\mathbf{I}_N)$.

The channel estimates $\hat{\mathbf{h}}_{ti}^k$ are stacked into the matrix $\hat{\mathbf{H}}_t^k \in \mathbb{C}^{QN\times U_t^k}$. The combining vector that maximizes the SINR in (6.39) is the MMSE combiner, which can be found as $\mathbf{a}_{tm}^k = [\mathbf{A}_t^k]_{:,m}$, where

$$\mathbf{A}_t^k = (\mathbf{L}_t^k + \hat{\mathbf{H}}_t^k\mathbf{D}_{t,\mathtt{p1}}^k\hat{\mathbf{H}}_t^{kH})^{-1}\hat{\mathbf{H}}_t^k\mathbf{D}_{t,\mathtt{p1}}^k,$$

where $\mathbf{L}_t^k \triangleq \sum_{i\in\mathcal{U}_k} p_i g_{ti}\Delta_{ti}^k + N_0\mathbf{I}_{QN}$, and $\mathbf{D}_{t,\mathtt{p1}}^k \triangleq \mathrm{diag}(p_{i_1}, p_{i_2}, \ldots, p_{i_{U_t^k}})$ contains the power

coefficients of the U_t^k users. Alternately, using the Woodbury identity [144], the combiner matrix can also be found via a $U_t^k \times U_t^k$ inverse as

$$\mathbf{A}_t^k = \mathbf{J}_t^k \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p2}}^k (\mathbf{I}_{QN} - (\mathbf{G}_t^k)^{-1} \mathbf{D}_{t,\mathtt{p2}}^k \hat{\mathbf{H}}_t^{kH} \mathbf{J}_t^k \hat{\mathbf{H}}_t^k) \mathbf{D}_{t,\mathtt{p2}}^k,$$

where $\mathbf{J}_t^k \triangleq \text{blkdiag}(j_{t1}^k \mathbf{I}_N, j_{t2}^k \mathbf{I}_N, \ldots, j_{tQ}^k \mathbf{I}_N)$, $\mathbf{G}_t^k \triangleq \mathbf{I}_{U_t^k} + \mathbf{D}_{t,\mathtt{p2}}^k \hat{\mathbf{H}}_t^{kH} \mathbf{J}_t^k \hat{\mathbf{H}}_t^k \mathbf{D}_{t,\mathtt{p2}}^k$, $j_{tq}^k \triangleq 1/(N_0 + \sum_{i \in \mathcal{U}_k} p_i g_{ti} \delta_{ti}^{qk})$, and $\mathbf{D}_{t,\mathtt{p2}}^k \triangleq \text{diag}(\sqrt{p_{i_1}}, \sqrt{p_{i_2}}, \ldots, \sqrt{p_{i_{U_t^k}}})$.

In this case, $\mathtt{Dec}_{tm}^k$, the extra interference due to decoded users, does not arise in (6.39). The rest of the SINR-based decoding process continues as usual.

Part 5

Chapter 7

Optimal Repetition Distributions

7.1 Introduction

In this chapter, we optimize the repetition distributions in IRSA via the *Differential Evolution* algorithm (DEA). The objective here is to optimize the IRSA system by aptly choosing the repetition distribution under the system constraints, by solving an optimization problem. We use DEA to obtain the throughput optimal and the energy efficiency optimal repetition distributions for the IRSA protocol.

Energy efficiency (EE) is a fundamental aspect of both cellular and machine-type communications [28]. With the advent of massive MIMO communications, many works have predicted a huge increase in the spectral efficiency (SE) [145] and promised an exponential increase in the throughputs [27]. But this comes at the expense of an increase in the power consumed and by extension, an adverse impact on the EE of the system under consideration. EE becomes more valuable in mMTC since the low-power devices deployed in mMTC scenarios are expected to be IoT devices which last for several years [29]. These devices need to be energy efficient and consume as low power as possible while maintaining high throughputs [30]. Thus, EE is a fundamental metric of mMTC systems, and in particular, in the IRSA protocol as well.

For IRSA, the inflection load L^* is the fundamental limit of the system beyond which the system performs poorly. It indicates both the peak throughput and the peak load at

which the system can be operated. This throughput is indicative of the packet success rate, and it does not account for packet length, bandwidth, or other system parameters [7]. As seen in Chapter 3, the throughput can be increased via improving the pilot length τ and the number of antennas N. This comes at the cost of lower SE and higher power consumed at the BS, respectively. Thus, there is a trade-off between the EE and the system throughput [146], and understanding this trade-off for IRSA is important.

7.1.1 Existing Distributions

The first paper on IRSA [7] has optimized the repetition distributions for IRSA under the collision channel and proposed a few distributions which have a peak inflection load of 0.965. The *truncated soliton* distribution has been shown to be the throughput optimal distribution for IRSA with a collision channel [72]. Recall that the truncated soliton distribution is defined as

$$\phi_d = \begin{cases} \dfrac{1-a}{2z}, & d = 2, \\ \dfrac{1}{d(d-1)z}, & 3 \leq d \leq d_{\max}, \\ 0 & \text{otherwise}, \end{cases} \tag{7.1}$$

where ϕ_d is the probability that a user has a repetition factor d, $d_{\max}$ is the maximum value that the repetition factor can take, $a \in (0, 1)$ is a convergence parameter, and $z = 1 - a/2 - 1/d_{\max}$ is a normalization constant. The authors in [17] have optimized the repetition distributions for IRSA with pure fading channels in the SISO case and have demonstrated that the achievable peak inflection loads with the optimized distributions exceed unity. The authors in [8] have shown that the soliton distribution is nearly optimal for IRSA with path-loss-only-channels, but they also show better performing distributions. The authors in [74] have claimed that CRDSA, i.e., a 2-regular distribution is the most energy efficient distribution for IRSA.

All of the above papers have optimized the repetition distributions of IRSA under different assumptions. There is no guarantee that those distributions will be optimal for

IRSA in the general case, i.e., with pilot contamination, channel estimation errors, multiple antennas, etc. Thus, one of the goals of this chapter is to find repetition distributions that perform better than the soliton distribution or other existing distributions in the general case.

7.1.2 Contributions

We summarize the contributions of this chapter below.

1. Firstly, we optimize the repetition distributions of IRSA with the throughput and the energy efficiency objectives.

2. Next, we study the optimal repetition distributions under three cases: first case with the K-collision channel; second case with the fading channel under the assumption of perfect CSI, MIMO, and MRC; and third case with the fading channel accounting for channel estimation errors, MIMO, pilot contamination, and MMSE combining.

3. Via extensive numerical simulations, we study the effect of various system parameters such as the maximum repetition factor $d_{\max}$, the average repetition factor $\bar{d}$, the number of antennas N, the pilot length τ on the repetition distributions, the inflection load, and the peak energy efficiency.

We demonstrate that, in general, the 2-regular distribution is the most energy efficient distribution for IRSA at high number of antennas and high pilot lengths. The 2-regular distribution is nearly throughput optimal at high number of antennas. In other regimes of interest (other than high number of antennas and high pilot lengths), we obtain varied distributions that are not the 2-regular distribution. In fact, near the inflection loads, we can optimize the repetition distributions to obtain higher EEs. Compared to the best existing distributions, we show that our optimized distributions can achieve up to 58% increase in the inflection load and up to 49% increase in the peak EE. The obtained optimal distributions can be used to operate mMTC at the peak throughputs as well as the peak EEs.

Remark 1: The results presented in this chapter are based on DEA and density evolution, which depends on the frame length T via the system load $L = M/T$ only asymptotically, i.e., with a fixed L and $M, T \to \infty$. Thus, similar to existing works, the presented distributions are optimal for IRSA on a frame-by-frame basis where the frame has a load L and not for a fixed T.

Remark 2: Density evolution is applicable for IRSA without path loss. Hence, in this chapter, we assume users perform path loss inversion. We have observed that the optimal distributions obtained in this chapter can be applied even when users do not perform path loss based inversion, and they perform much better than existing distributions. Thus, in practice, the distributions presented in this chapter can be used to obtain near-optimal performance in the general case.

Remark 3: The distributions output by DEA are not unique, i.e., there are several distributions that could achieve the same optimal loss rate or throughput or EE. We only present a few optimal distributions and not all of them.

7.2 An Overview of the Differential Evolution Algorithm

Genetic algorithms (GAs) are typically used for global optimization over continuous spaces. In GAs, a population of candidates to an optimization problem is evolved toward the optimal solutions. Direct search approaches like GAs are used when the cost function is non-linear and non-differentiable. The template of GAs is as follows:

- Generate a random initial population within the parameter space.
- Evaluate the "fitness" of each individual, where the fitness is in terms of the objective function of the optimization problem.
- Select the fittest individuals as "parents" of the next generation.
- Via "mutation" and "crossover" of the parents, obtain new individuals.

- Replace least-fit individuals by new individuals in each generation.

Differential evolution algorithm (DEA) is a GA that has been widely used for multi-dimensional multi-objective constrained optimization [26]. Typical applications include signal processing, filter design, data clustering, and optimal control. The advantages of using DEA are

- DEA is a direct search method [26]. There is no requirement of computing gradients for the algorithm and it can handle non-differentiable functions as well.
- DEA has been shown to possess good convergence properties [147].
- The number of control parameters in DEA are only three in number and can be easily tuned [148].
- DEA uses a vector population where the perturbation of the vectors can be done independently of each other [149]. This makes the algorithm parallelizable.

7.2.1 DEA for Optimizing IRSA

For an IRSA system, the optimal distribution can be obtained as a solution to the optimization problem defined as follows:

$$\mathbb{P}_1: \quad \min_{\phi_2,\phi_3,\ldots,\phi_{d_{\max}}} \quad \mathcal{L}(\{\phi_2,\phi_3,\ldots,\phi_{d_{\max}}\}) \tag{7.2}$$

$$\text{subject to} \quad 0 \le \phi_d \le 1, \quad 2 \le d \le d_{\max}, \tag{7.3}$$

$$\sum_{d=2}^{d_{\max}} \phi_d = 1. \tag{7.4}$$

Here, $\mathcal{L}(\cdot)$ is the loss function which needs to be minimized, which can be the packet loss rate, and $\{\phi_2,\phi_3,\ldots,\phi_{d_{\max}}\}$ is the node-perspective user degree distribution, where ϕ_d represents the probability a user will pick a repetition factor d. In the above problem, we can alternatively maximize the throughput or the EE of the system as well. Note that the above problem is a stochastic optimization problem since it is dependent on

the distributions of the repetition factors and not the random instances of the users' instantaneous channels or repetition factors, which are random variables.

We now setup some notation and present the DEA. Let $\mathcal{L} : \mathbb{R}^N \to \mathbb{R}$ be a loss function that needs to be minimized. Let M be the number of population vectors in any generation and N be the dimension of each vector in the population. For the optimal distribution problem, we have that $N = d_{\max} - 1$. We denote the population in the gth generation by $\mathcal{P}^g = \{\mathbf{x}_1^g, \mathbf{x}_2^g, \ldots, \mathbf{x}_M^g\}$, with $\mathbf{x}_i^g \in \mathbb{R}^N$. Each $\mathbf{x}_i^g$ contains non-negative entries that add up to one. There are three steps in each generation of DEA, which are as follows:

1. *Mutation step:* In the first step, new *mutated* vectors are generated by adding the weighted difference between two vectors. Firstly, for each vector $\mathbf{x}_i^g$ in the population, we choose three other distinct random vectors in the population $\bar{\mathbf{x}}_1^g, \bar{\mathbf{x}}_2^g, \bar{\mathbf{x}}_3^g \in \mathcal{P}^g$. Then we add a scaled difference of two of the vectors to the third and generate a mutant vector $\mathbf{v}_i^g$ as $\mathbf{v}_i^g = \bar{\mathbf{x}}_1^g + F(\bar{\mathbf{x}}_2^g - \bar{\mathbf{x}}_3^g)$. Here, F is a mutant scale factor. DEA derives its namesake from this step, where we use the differential of two vectors to generate the mutated vector.

2. *Crossover step:* The entries of the mutated vector $\mathbf{v}_i^g$ are now mixed and crossed-over with the entries of the original vector $\mathbf{x}_i^g$, and the *trial* vector $\mathbf{u}_i^g$ is generated. For each entry of $\mathbf{u}_i^g$, with a probability C_r, the entry $\mathbf{u}_i^g[j]$ is chosen to be $\mathbf{v}_i^g[j]$ and with a probability $1 - C_r$, the entry $\mathbf{u}_i^g[j]$ is chosen to be $\mathbf{x}_i^g[j]$, where C_r is the crossover probability. Crossover is performed in order to increase the diversity of the population vectors. We can also ensure that at least one component for the trial vector $\mathbf{u}_i^g$ is obtained from the mutant. This can be done by mandatorily crossing over some entry with index $j = R_i$, where R_i is sampled uniformly from the set of integers $\{1, 2, \ldots, N\}$.

3. *Selection step:* In the final step, we select the population vectors for the next generation. If the trial vector yields a lower loss function value than the original vector, i.e., if $\mathcal{L}(\mathbf{u}_i^g) < \mathcal{L}(\mathbf{x}_i^g)$, then the trial vector $\mathbf{u}_i^g$ replaces the original vector $\mathbf{x}_i^g$ in the next generation.

The above steps are run iteratively until some termination criteria has been reached, and the candidate with the least loss in the final generation is output as the optimal candidate. We present the DEA in Algorithm 11.

Remark 4: We first generate a random initial population $\mathcal{P}^0 = \{\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots, \mathbf{x}_M^0\}$. Here the entries of each vector $\mathbf{x}_i^0$ is chosen uniformly at random from the entire parameter space, i.e., $\mathbf{x}_i^0[j] \sim \mathcal{U}[0,1]$, $1 \leq j \leq N$ [148]. The parameter space $[0,1]$ ensures that the entries are valid probability entries. Finally, the entries of the initial vectors can be normalized to ensure that their sum is unity. To ensure the entries of the trial vector add up to one, we can normalize each trial vector after the crossover step. The terminating criteria $\texttt{TC}(\cdot)$ for DEA could be a maximum number of iterations or the standard deviation in the loss functions across the candidates, and c_ε is an appropriate threshold.

Remark 5: The maximization of the throughput or EE can be performed by considering the function $\mathcal{C} = -\mathcal{L}$ instead. In this case, we select the candidate vectors which have the highest throughput or EE in the selection step.

Remark 6: The computational complexity of the algorithm is $\mathcal{O}(MNG_{\max})$, where $G_{\max}$ is the maximum number of generations that DEA is run for [26]. To make it parallelizable, we can run DEA on each entry of the population on N parallel computers, which results in a complexity of $\mathcal{O}(MG_{\max})$.

Remark 7: A few good rules of thumb for DEA [149] are as follows:

1. Choose the number of candidate vectors $M \in [5N, 10N]$ to ensure a wide range of candidate vectors that can adequately sample from the entire space. If there are vectors which are suspected to be close to the optima, they can be fed into the initial population.

2. Choose mutation scale factor $F \in [0,2]$. It can be initialized with $F = 0.5$, and if the problem converges prematurely, then increase either M or F.

3. A large crossover probability C_r close to unity can speed up convergence. A low C_r implies only a few parameters are changed in every iteration and a high C_r implies a huge variation in the entries of the population vectors.

Algorithm 11: Differential Evolution Algorithm

Input: $N, M, F, C_r, c_\varepsilon, \mathcal{L}(\cdot), \texttt{TC}(\cdot)$

Initialize: $\mathcal{P}^0 = \{\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots, \mathbf{x}_M^0\}$, $g = 0$

while $\texttt{TC}(\mathcal{P}^g) > c_\varepsilon$ **do**

  for $i = 1, 2, \ldots, M$ **do**

    Choose three distinct vectors $\bar{\mathbf{x}}_1^g, \bar{\mathbf{x}}_2^g, \bar{\mathbf{x}}_3^g \in \mathcal{P}^g$

    Mutation Step:

$$\mathbf{v}_i^g = \bar{\mathbf{x}}_1^g + F(\bar{\mathbf{x}}_2^g - \bar{\mathbf{x}}_3^g)$$

    Sample index for mandatory crossover $R_i \sim \mathcal{U}\{1, 2, \ldots, N\}$

    for $j = 1, 2, \ldots, N$ **do**

      Crossover Step:

$$\mathbf{u}_i^g[j] = \begin{cases} \mathbf{v}_i^g[j], & j = R_i \text{ or with probability } C_r \\ \mathbf{x}_i^g[j], & j \neq R_i \text{ and with probability } 1 - C_r \end{cases}$$

    end

    Selection Step:

$$\mathbf{x}_i^{g+1} = \begin{cases} \mathbf{u}_i^g, & \text{if } \mathcal{L}(\mathbf{u}_i^g) < \mathcal{L}(\mathbf{x}_i^g) \\ \mathbf{x}_i^g, & \text{if } \mathcal{L}(\mathbf{u}_i^g) \geq \mathcal{L}(\mathbf{x}_i^g) \end{cases}$$

  end

  $g = g + 1$

end

Output: Optimal candidate index $\bar{i} = \arg\min_{1 \leq i \leq M}(\mathcal{L}(\mathbf{x}_i^g))$ and Optimal candidate $\mathbf{x}_{\bar{i}}^g$

7.3 Throughput Maximization

The theoretical throughput of IRSA can be calculated from the density evolution (DE) process presented in Chapter 3. For this purpose, we now quickly recap the DE process. For this, we recall that p_i is the probability that an edge carries a failure message from an RB node to a user node in the ith iteration, and that q_i is the probability that an edge carries a failure message from a user node to an RB node in the ith iteration. The failure probability q_i is calculated using the edge-perspective user degree distribution as

$$q_i = \sum\nolimits_{d=2}^{d_{\max}} \lambda_d q_i^{(d)} = \sum\nolimits_{d=2}^{d_{\max}} \lambda_d p_{i-1}^{d-1} = \lambda(p_{i-1}). \tag{7.5}$$

The failure probability p_i is calculated using the edge-perspective RB degree distribution as

$$p_i = 1 - e^{-L\bar{d}q_i} \sum_{r=1}^{\infty} \theta_r \frac{(L\bar{d}q_i)^{r-1}}{(r-1)!} \triangleq f(q_i), \tag{7.6}$$

where θ_r denotes the probability that the reference packet gets decoded in the current decoding iteration starting from degree r using only intra-RB SIC [8]. Thus, $q_i = \lambda(p_{i-1})$ and $p_i = f(q_i)$ are calculated alternately as functions of each other as seen in (7.5) and (7.6). The procedure can be initialized with either $q_0 = 1$ or $p_0 = f(1)$. The failure probability at the end of decoding is $p_\infty = \lim_{i\to\infty} p_i$ and $(p_\infty)^d$ is the probability that a packet transmitted from a user with repetition factor d does not get decoded at the receiver. Therefore, the asymptotic packet loss rate function ($\texttt{PLR}(\cdot)$),[1] which is the fraction of packets that are not decoded at the BS, is calculated as

$$\texttt{PLR}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\}) = \phi(p_\infty) = \sum\nolimits_{d=2}^{d_{\max}} \phi_d (p_\infty)^d. \tag{7.7}$$

[1]For the purpose of this chapter, we condense (7.5) and (7.6) into a PLR function.

The asymptotic throughput of the system can now be obtained from the asymptotic PLR as

$$\texttt{Thpt}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\}) = L(1 - \texttt{PLR}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\})). \tag{7.8}$$

The iterations $p_i = f(\lambda(p_{i-1}))$ converge asymptotically to $p_\infty = 0$ if the system load $L < L^*$ [7]. Here, L^* is called the *inflection load* of the system: for any $L \geq L^*$, the system becomes interference limited and $\texttt{PLR}(\cdot)$ does not converge to 0 as L increases. Thus, for $L < L^*$, $p_\infty = 0$ and therefore the asymptotic $\texttt{PLR}(\cdot) = 0$, and $\texttt{Thpt}(\cdot) = L$. For $L \geq L^*$, the throughput decreases monotonically with L.

The throughput optimization problem now reduces as follows.

$$\mathbb{P}_2 : \quad \max_{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}} \quad \texttt{Thpt}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\}) \tag{7.9}$$

$$\text{subject to} \quad 0 \leq \phi_d \leq 1, \quad 2 \leq d \leq d_{\max}, \tag{7.10}$$

$$\sum_{d=2}^{d_{\max}} \phi_d = 1. \tag{7.11}$$

Here, the constraints ensure that the candidate solutions are indeed probability vectors. The optimal distribution yields the maximum throughput $\mathcal{T}_{\max}$, which occurs at $L = L^*$. Since $\mathcal{T} = L$ for all L^* and $\mathcal{T} = \mathcal{T}_{\max}$ at $L = L^*$, we analyze only the inflection load since its value is exactly equal to the peak throughput.

Remark 8: Alternately, some existing works maximize the load L with a target PLR [17]. In this case, the optimization objective in problem $\mathbb{P}_1$ is just simply L and an additional target loss rate constraint is added: $\texttt{PLR}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\}) \leq \mathsf{PLR}_{\text{tgt}}$.

7.4 Numerical Results for Throughput Maximization

We now optimize the repetition distributions in order to maximize the throughput of IRSA. For the results presented in this chapter, we operate the system at a load of $L = L^*$ in order to operate the system at the peak throughput and the reported distributions are

optimized at the inflection load. The DEA parameters are $F = 0.9$, $M = 10N$, and $C_r = 0.9$ [149]. We choose the terminating criteria $\texttt{TC}(\cdot)$ as the standard deviation of the throughputs of all the population vectors in any generation, and terminate the DEA generations when $\texttt{TC}(\cdot) < c_\varepsilon = 10^{-6}$ [149].

7.4.1 K-Collision Channel

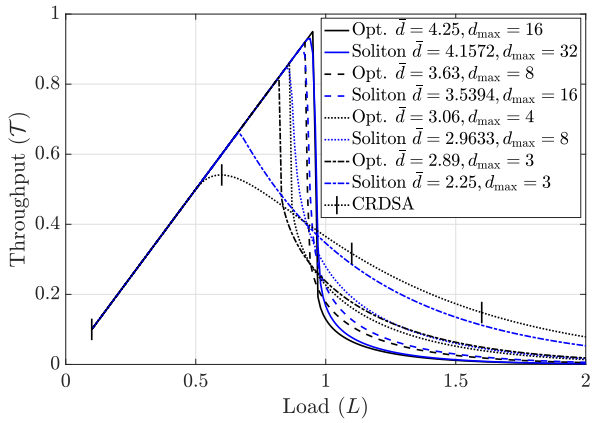
Table 7.1: Throughput Optimal Distributions for IRSA with the K-Collision Channel.

K	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*
1	3	$0.11x^2 + 0.89x^3$	2.89	0.82
	4	$0.47x^2 + 0.53x^4$	3.06	0.86
	8	$0.47x^2 + 0.31x^3 + 0.22x^8$	3.65	0.92
	16	$0.37x^2 + 0.41x^3 + 0.07x^4 + 0.05x^8 + 0.1x^{16}$	4.25	0.95
2	3	$0.71x^2 + 0.29x^3$	2.29	1.71
	4	$0.82x^2 + 0.18x^4$	2.36	1.74
	8	$0.86x^2 + 0.01x^3 + 0.13x^8$	2.79	1.85
	16	$0.86x^2 + 0.02x^6 + 0.02x^7 + 0.05x^8 + 0.05x^{16}$	3.18	1.89
3	3	$0.94x^2 + 0.06x^3$	2.05	2.57
	4	$0.92x^2 + 0.08x^4$	2.16	2.58
	8	$0.91x^2 + 0.09x^8$	2.54	2.65
	16	$0.93x^2 + 0.02x^{13} + 0.04x^{15} + 0.01x^{16}$	2.88	2.75

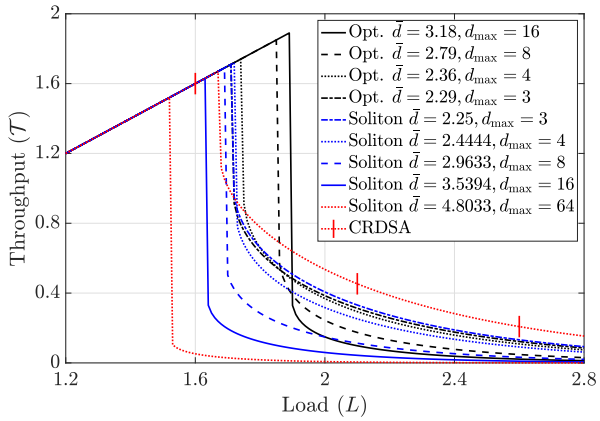
For the K-collision channel, up to K collisions can be decoded perfectly, i.e., $\theta_r = 1$, $1 \leq r \leq K$. We present the throughput optimal distributions for the K collision channel in Table 7.1.

We now present numerical results for the K-Collision channel in Fig. 7.1. We compare the optimized distributions in Table 7.1 with the truncated Soliton distribution [72], with the appropriate maximum repetition factor as presented in the plots.

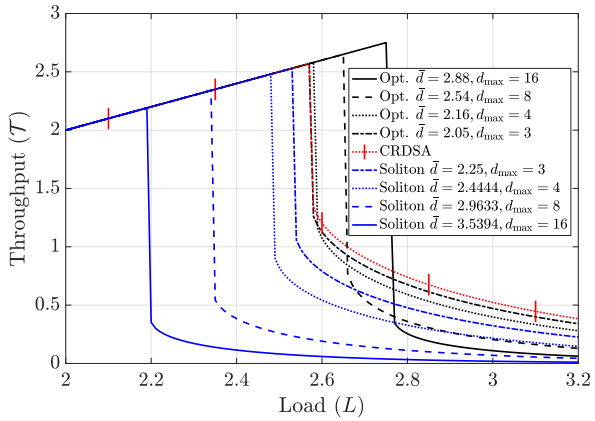
For $K = 1$, we have the usual collision channel. In Fig. 7.1(a), we observe that CRDSA has the least peak performance at $\mathcal{T} = 0.55$, and the optimized distribution has the best performance with $\bar{d} = 4.25, d_{\max} = 16$. The performance of this coincides with the soliton distribution with almost the same average repetition factor of $\bar{d} = 4.1572$, but



(a) $K = 1$.



(b) $K = 2$.



(c) $K = 3$.

Figure 7.1: Optimized Performance for the K-Collision Channel.

double the maximum repetition factor at $d_{\max} = 16$. Similar trends are seen for lower values of $d_{\max}$. The results from [72] indicate that the truncated soliton distribution is an unique optimal distribution for the 1-collision channel, with infinite maximum repetition factor. But practically, with finite maximum repetition factor, we can find other optimized distributions that perform similar to the truncated soliton distribution.

For $K = 2$, in Fig. 7.1(b), we see that the optimized distribution with $\bar{d} = 3.18, d_{\max} = 16$ maximizes the throughput at $L^* \approx 1.9$. The soliton distribution with $\bar{d} = 4.8033, d_{\max} = 64$ is no longer the optimal distribution, and in fact, it is the lowest curve, i.e., it is the distribution that achieves the lowest peak performance with an inflection load of $L^* \approx 1.6$. As we decrease the maximum repetition factor from $d_{\max} = 16$ to $8, 4$, and 3, we observe that the peak throughputs reduce. The trends for the soliton distribution is exactly the opposite and the peak throughputs (and the corresponding inflection laods) increase with a decrease in $d_{\max}$. This shows that directly using the soliton distribution which has been optimized for the 1-Collision channel need not be optimal for any other channel. Further, CRDSA achieves the lowest inflection load $L^* = 1.5$, which shows that irregularly transmitting as in IRSA is beneficial for the K-Collision channel.

For $K = 3$, in Fig. 7.1(b), the optimized distribution with $\bar{d} = 2.88, d_{\max} = 16$ maximizes the throughput at $L^* \approx 2.75$. The trends from the previous plot as we decrease the maximum repetition factor from $d_{\max} = 16$ to $8, 4$, and 3 are observed for $K = 3$ as well,. The soliton distribution again turns out to achieve the lowest inflection load $L^* = 2.2$, and here it is far lower than CRDSA as well. In fact, CRDSA performs better than all the soliton distribution curves, and is inferior to all the numerically optimized distributions.

7.4.2 Fading Channel with MIMO, Perfect CSI, and Maximal Ratio Combining

In this section, we optimize the repetition distributions for the fading channel with MIMO, perfect CSI, and MRC. The optimal distributions for the SISO case with perfect CSI is presented in [17]. Further, for the success probability θ_r, we observed in Chapter 3, that

the Normal approximation performed the best. Hence, in this section, we use the Normal approximation to find the throughput optimal distributions.[2]

Table 7.2: Throughput Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with the Normal Approximation and $N = 2$.

Parameters	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*
$\rho_0 = 1, \gamma_{\text{th}} = 1$	3	$0.03x^2 + 0.97x^3$	2.97	1.78
	4	$0.4x^2 + 0.09x^3 + 0.51x^4$	3.1	1.85
	8	$0.26x^2 + 0.6x^3 + 0.14x^8$	3.37	1.9
$\rho_0 = 5, \gamma_{\text{th}} = 5$	3	x^3	3	0.71
	4	x^4	4	0.85
	8	$0.41x^3 + 0.38x^4 + 0.21x^8$	4.6	0.91
$\rho_0 = 10, \gamma_{\text{th}} = 10$	3	x^3	3	0.62
	4	$0.03x^2 + 0.97x^4$	3.94	0.75
	8	$0.5x^2 + 0.21x^3 + 0.29x^8$	4.74	0.8
$\rho_0 = 10, \gamma_{\text{th}} = 5$	3	x^3	3	1.05
	4	$0.33x^2 + 0.67x^4$	3.35	1.11
	8	$0.2x^2 + 0.6x^3 + 0.2x^8$	3.71	1.17

We present the optimal distributions for the fading channel in Table 7.2, and we compare the effect of varying the SNR ρ_0 and the capture threshold γ_{th} on the inflection load L^*. As expected, as $d_{\max}$ is increased, the optimized distributions perform marginally better. Thus, the disparity that is offered by a higher repetition factor helps improve the performance. Increasing $\rho_0 = \gamma_{\text{th}}$ from 1 to 5 to 10 reduces the inflection load L^* as expected. This is because even though the SNR improves, the capture threshold also increases and thus the number of users who cross the SINR threshold reduces. Further for $\rho_0 = 10$, with a reduction in γ_{th} from 10 to 5, L^* improves. These observations are also supported by the observations in Chapter 3.

We present the optimal distributions for the fading channel in Table 7.3, with similar settings as the previous table but with $N = 8$ instead. The inflection loads increase from $N = 2$ to $N = 8$ and the distributions require a higher $d_{\max}$ to achieve marginally higher inflection loads. Here the inflection load has increased from $L^* = 1.9$ with $N = 2$ to

[2]For the other approximations, see Sec. 7.8.1.

Table 7.3: Throughput Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with the Normal Approximation and $N = 8$.

Parameters	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*
$\rho_0 = 1, \gamma_{\text{th}} = 1$	3	x^2	2	7.36
	8	$0.97x^2 + 0.03x^8$	2.14	7.38
	16	$0.96x^2 + 0.01x^{14} + 0.01x^{15} + 0.02x^{16}$	2.55	7.53
$\rho_0 = 5, \gamma_{\text{th}} = 5$	3	$0.37x^2 + 0.63x^3$	2.62	2.12
	4	$0.68x^2 + 0.32x^4$	2.64	2.24
	8	$0.77x^2 + 0.01x^5 + 0.06x^7 + 0.16x^8$	3.22	2.35
$\rho_0 = 10, \gamma_{\text{th}} = 10$	3	$0.11x^2 + 0.89x^3$	2.88	1.51
	8	$0.54x^2 + 0.24x^3 + 0.22x^8$	3.56	1.69
	16	$0.52x^2 + 0.16x^3 + 0.18x^4 + 0.02x^{11}$ $+0.01x^{14} + 0.01x^{15} + 0.1x^{16}$	4.3	1.74
$\rho_0 = 10, \gamma_{\text{th}} = 5$	3	$0.41x^2 + 0.59x^3$	2.59	2.21
	4	$0.69x^2 + 0.31x^4$	2.62	2.29
	8	$0.79x^2 + 0.21x^8$	3.21	2.45

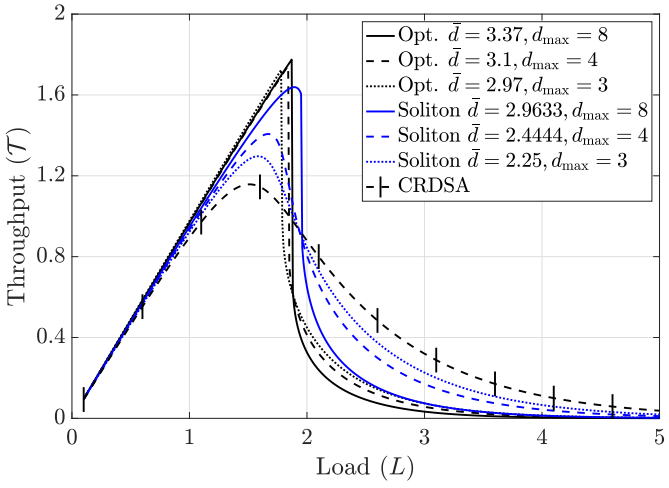


Figure 7.2: Comparison between optimized performance and soliton performance for $N = 2, \rho_0 = 1, \gamma_{\text{th}} = 1$, for the fading channel with MIMO, MRC, and the Normal approximation.

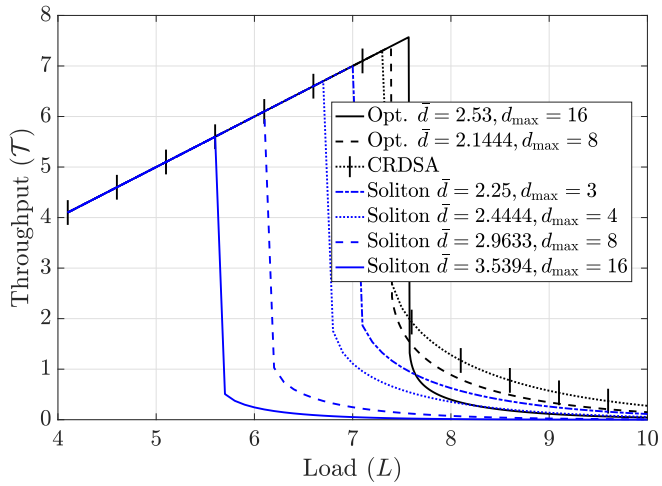


Figure 7.3: Comparison between optimized performance and soliton performance for $N = 8, \rho_0 = 1, \gamma_{\text{th}} = 1$, for the fading channel with MIMO, MRC, and the Normal approximation.

Table 7.4: Throughput Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with $\rho_0 = 1, \gamma_{\text{th}} = 1$, and the Normal Approximation.

N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*
2	3	$0.03x^2 + 0.97x^3$	2.97	1.78
	4	$0.4x^2 + 0.09x^3 + 0.51x^4$	3.1	1.85
	8	$0.26x^2 + 0.6x^3 + 0.14x^8$	3.37	1.9
4	3	$0.86x^2 + 0.14x^3$	2.14	3.94
	4	$0.89x^2 + 0.11x^4$	2.22	3.97
	8	$0.91x^2 + 0.01x^6 + 0.08x^8$	2.5	4.11
8	3	x^2	2	7.36
	8	$0.97x^2 + 0.03x^8$	2.14	7.38
	16	$0.96x^2 + 0.01x^{14} + 0.01x^{15} + 0.02x^{16}$	2.55	7.53
16	3	x^2	2	13.31
	16	$0.98x^2 + 0.01x^9 + 0.01x^{16}$	2.15	13.35
	24	$0.97x^2 + 0.01x^{16} + 0.01x^{23} + 0.01x^{24}$	2.35	13.37

Table 7.5: Throughput Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with $\rho_0 = 10, \gamma_{\text{th}} = 10$, and the Normal Approximation.

N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*
2	3	x^3	3	0.62
	4	$0.03x^2 + 0.97x^4$	3.94	0.75
	8	$0.5x^2 + 0.21x^3 + 0.29x^8$	4.74	0.8
4	3	x^3	3	1.11
	4	$0.27x^3 + 0.73x^4$	3.72	1.15
	8	$0.11x^2 + 0.72x^3 + 0.17x^8$	3.75	1.2
8	3	$0.11x^2 + 0.89x^3$	2.88	1.51
	8	$0.54x^2 + 0.24x^3 + 0.22x^8$	3.56	1.69
	16	$0.52x^2 + 0.16x^3 + 0.18x^4 + 0.02x^{11}$ $+0.01x^{14} + 0.01x^{15} + 0.1x^{16}$	4.3	1.74
16	3	$0.43x^2 + 0.57x^3$	2.57	2.18
	16	$0.51x^2 + 0.36x^3 + 0.01x^4 + 0.03x^8$ $+0.01x^{11} + 0.04x^{13} + 0.03x^{15} + 0.01x^{16}$	3.67	2.42
	24	$0.5x^2 + 0.32x^3 + 0.08x^5 + 0.02x^{11}$ $+0.02x^{17} + 0.01x^{20} + 0.02x^{22} + 0.03x^{24}$	4.11	2.46

$L^* = 7.53$ with $N = 8$ due to array gain. The distributions themselves change a lot and no obvious trends in the optimized distributions are observed with variations in ρ_0 and γ_{th}.

In Fig. 7.2, we compare the performance of different repetition distributions for the fading channel with MIMO, MRC, and the Normal approximation, and with $N = 2, \rho_0 = 1, \gamma_{\text{th}} = 1$. All the optimized distributions perform better than the Soliton distribution, and CRDSA has the lowest peak performance with $\mathcal{T} = 1.15$ at $L = 1.5$. We see that considering a realistic fading assumption on the channel allows for power disparity amongst users which leads to all the throughputs exceeding unity. Further, with $d_{\max} = 8$, the optimized distribution performs the best, albeit with a higher $\bar{d}$ when compared to the soliton distribution. Thus, even for the fading channel, the soliton distribution is not good enough and we can find distributions that perform better.

In Fig. 7.3, we analyze the same as the previous figure, but with $N = 8, \rho_0 = 1, \gamma_{\text{th}} = 1$. The performance highly improves compared to the previous figure due to the increase in the diversity gain offered by the number of antennas. The soliton distribution with $\bar{d} = 3.5394, d_{\max} = 16$ is no longer the optimal distribution, and in fact, it is the lowest curve, i.e., it is the distribution that achieves the lowest peak performance with an inflection load of $L^* \approx 5.5$. As we decrease the maximum repetition factor from $d_{\max} = 16$ to $8, 4$, and 3, we observe that the peak throughputs for the soliton distribution increases. CRDSA also performs better than all the soliton distributions. The optimized distribution with $\bar{d} = 2.53, d_{\max} = 16$ achieves the highest throughput and it performs better than CRDSA.

We now present the optimal distributions for the fading channel, where we study the effect of the number of antennas on the inflection load L^*. In Table 7.4, we present the result with $\rho_0 = 1, \gamma_{\text{th}} = 1$, whereas in Table 7.5, we present the result with $\rho_0 = 10, \gamma_{\text{th}} = 10$. For $\rho_0 = 1, \gamma_{\text{th}} = 1$, the maximum inflection loads are $L^* = 1.9$, 4.11, 7.36, and 13.37 for $N = 2$, 4, 8, and 16. For $\rho_0 = 10, \gamma_{\text{th}} = 10$, the maximum inflection loads are $L^* = 0.8$, 1.2, 1.74, and 2.46 for $N = 2$, 4, 8, and 16. With an increase in the number of antennas, the diversity gain increases for MRC, and thus, the decodability of users improves which leads to a significant increase in the inflection load. As N is increased,

the optimal distributions for $\rho_0 = 1, \gamma_{\text{th}} = 1$ become closer to that of CRDSA. Recall that a D-regular distribution for DRRSA is $\phi(x) = x^D$. In fact, for $N = 2$, the optimal distribution is closer to 3-regular DRRSA, and when N is increased to 16, the optimal distribution is indeed the 2-regular DRRSA, which is the same as CRDSA. This behaviour is very different for $\rho_0 = 10, \gamma_{\text{th}} = 10$, for which as N is increased, the optimal distributions for $\rho_0 = 1, \gamma_{\text{th}} = 1$ remain as irregular distributions. For $\rho_0 = 10, \gamma_{\text{th}} = 10$, the optimal distributions become less sparse compared to the distributions for $\rho_0 = 1, \gamma_{\text{th}} = 1$ for each N. Here, *sparse* distributions are distributions which have most entries to be zero, whereas *dense* distributions are distributions which have most entries as non-zero entries. Thus, we can summarize that for higher SNRs and higher modulation and coding schemes (MCSs) which require a higher γ_{th}, the optimized distributions are indeed irregular in nature, and would require us to use more dense distributions as seen for $\rho_0 = 10, \gamma_{\text{th}} = 10$, and $N = 16$.

7.4.3 Fading Channel with MIMO, Channel Estimation Errors, Pilot Contamination, and MMSE Combining

In the general case, i.e., with MIMO, fading, channel estimation, pilot contamination, and MMSE combining, we cannot compute the success probability θ_r in closed form as observed in Chapter 3. Hence, we resort to empirical calculation of θ_r for the general case, and then optimize the throughput of IRSA using the same.

We now present the throughput-optimized distributions for IRSA with the fading channel, MIMO, channel estimation errors, pilot contamination, and MMSE combining in Table 7.6 for $\tau = 10, \rho_0 = 1$, and $\gamma_{\text{th}} = 1$; in Table 7.7 for $\tau = 10, \rho_0 = 10$, and $\gamma_{\text{th}} = 10$. For $N = 2$, the optimal distribution is 3-regular for $d_{\max} = 3$, whereas for $d_{\max} = 4$, it is 4-regular. For $d_{\max} = 8$, it is more dense but with the same $\bar{d}$ as the previous distribution. As we increase N to 4, 8, and then 16, we see that the distributions become closer to a 2-regular distribution.

We now present the throughput-optimized distributions for different pilot lengths in IRSA with the fading Channel, MIMO, channel estimation errors, pilot contamination,

Table 7.6: Throughput Optimal Distributions for IRSA with the Fading Channel, MIMO, Channel Estimation Errors, Pilot Contamination, and MMSE Combining, with $\tau = 10, \rho_0 = 1$, and $\gamma_{\text{th}} = 1$.

N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*
2	3	x^3	3	1.64
	4	$0.06x^3 + 0.94x^4$	3.94	1.76
	8	$0.16x^2 + 0.18x^3 + 0.55x^5 + 0.11x^8$	3.94	1.82
4	3	$0.93x^2 + 0.07x^3$	2.06	5.11
	4	$0.92x^2 + 0.08x^4$	2.16	5.15
	8	$0.91x^2 + 0.06x^6 + 0.03x^8$	2.25	5.18
8	3	x^2	2	8.1
	8	$0.99x^2 + 0.01x^8$	2.1	8.15
	16	$0.97x^2 + 0.01x^{14} + 0.01x^{15} + 0.01x^{16}$	2.2	8.18
16	3	x^2	2	10.8
	16	$0.98x^2 + 0.01x^{14} + 0.01x^{16}$	2.03	10.84
	24	$0.97x^2 + 0.01x^3 + 0.01x^{19} + 0.01x^{24}$	2.35	10.86

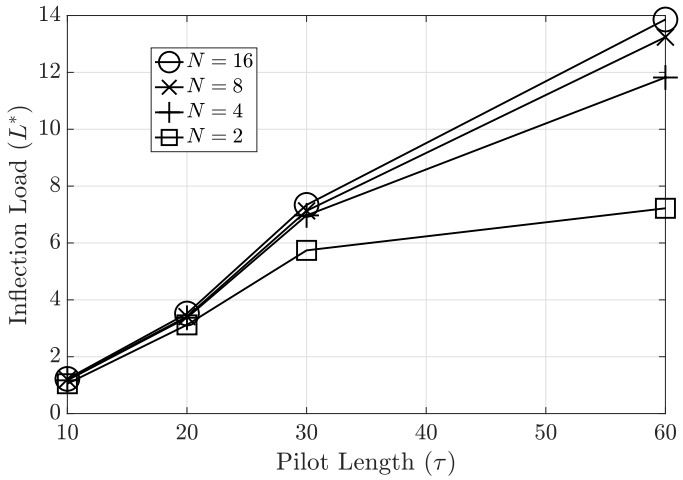


Figure 7.4: Impact of the pilot length on the inflection load L^*, for the fading channel with MIMO, MMSE combining, pilot contamination, and channel estimation errors.

Table 7.7: Throughput Optimal Distributions for IRSA with the Fading Channel, MIMO, Channel Estimation Errors, Pilot Contamination, and MMSE Combining, with $\tau = 10, \rho_0 = 10$, and $\gamma_{\text{th}} = 10$.

N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*
2	3	x^3	3	0.89
	4	$0.01x^2 + 0.02x^3 + 0.97x^4$	3.97	1.01
	8	$0.43x^3 + 0.39x^4 + 0.18x^8$	4.3	1.05
4	3	$0.6x^2 + 0.4x^3$	2.4	2.93
	4	$0.85x^2 + 0.02x^3 + 0.13x^4$	2.27	3.05
	8	$0.89x^2 + 0.02x^3 + 0.09x^8$	2.53	3.12
8	3	$0.99x^2 + 0.01x^3$	2.01	5.7
	8	$0.98x^2 + 0.01x^5 + 0.01x^8$	2.06	5.74
	16	$0.97x^2 + 0.01x^6 + 0.01x^{10} + 0.01x^{16}$	2.21	5.78
16	3	$0.97x^2 + 0.03x^3$	2.03	7.2
	16	$0.97x^2 + 0.01x^{12} + 0.01x^{13} + 0.01x^{16}$	2.22	7.24
	24	$0.96x^2 + 0.01x^5 + 0.01x^7$ $+0.01x^{19} + 0.01x^{24}$	2.35	7.27

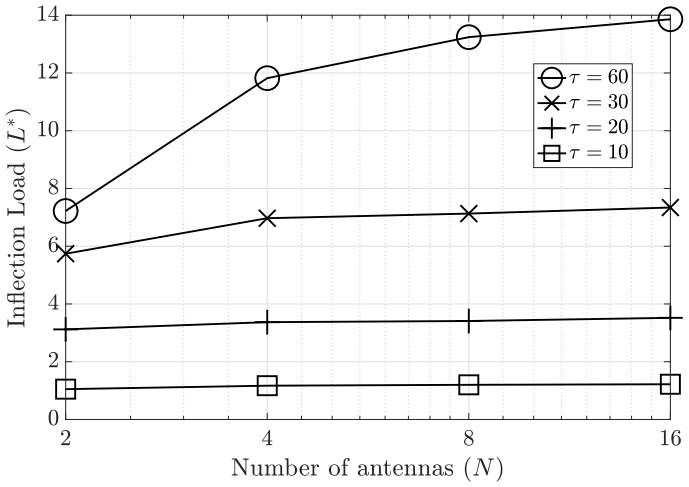


Figure 7.5: Impact of the number of antennas on the inflection load L^*, for the fading channel with MIMO, MMSE combining, pilot contamination, and channel estimation errors.

Table 7.8: Throughput Optimal Distributions for IRSA with the Fading Channel, MIMO, Channel Estimation Errors, Pilot Contamination, and MMSE Combining, with $d_{\max} = 8, \rho_0 = 10$, and $\gamma_{\text{th}} = 10$.

N	τ	$\phi(x)$	$\bar{d}$	L^*
2	10	$0.43x^3 + 0.39x^4 + 0.18x^8$	4.3	1.05
	20	$0.59x^2 + 0.25x^3 + 0.16x^8$	4.06	1.15
	30	$0.71x^3 + 0.1x^4 + 0.18x^8$	3.97	1.2
	60	$0.81x^3 + 0.01x^4 + 0.18x^8$	3.91	1.22
4	10	$0.89x^2 + 0.02x^3 + 0.09x^8$	2.53	3.12
	20	$0.93x^2 + 0.01x^3 + 0.06x^8$	2.4	3.37
	30	$0.89x^2 + 0.04x^3 + 0.07x^8$	2.46	3.41
	60	$0.93x^2 + 0.07x^8$	2.42	3.52
8	10	$0.98x^2 + 0.01x^5 + 0.01x^8$	2.06	5.74
	20	$0.97x^2 + 0.01x^3 + 0.02x^8$	2.17	6.97
	30	$0.98x^2 + 0.01x^7 + 0.01x^8$	2.11	7.13
	60	$0.99x^2 + 0.01x^8$	2.17	7.34
16	10	$0.98x^2 + 0.01x^6 + 0.01x^8$	2.1	7.22
	20	$0.99x^2 + 0.01x^8$	2.17	11.82
	30	$0.99x^2 + 0.01x^8$	2.17	13.24
	60	x^2	2	13.86

and MMSE combining in Table 7.8 for $d_{\max} = 8, \rho_0 = 10$, and $\gamma_{\text{th}} = 10$. For each N, it is observed that the inflection loads monotonically increase with an increase in the pilot length τ. For $N = 2$, the optimized distributions are varied and differ from each other as τ is increased from 10 to 20 to 30 to 60. For a similar increase in τ, the optimized distributions vary only slightly for $N = 4$ and vary negligibly for $N = 8$ and $N = 16$. This is because the system is already performing at the peak for low pilot lengths at higher N. Thus, optimizing the distribution does not help much here. We can also see a reduction in $\bar{d}$ with an increase in τ for $N = 2$ and $N = 4$, which shows that fewer packets could be transmitted in order to obtain higher inflection loads.

In Fig. 7.4, we study the impact of the pilot length on the inflection load L^*. For every N, the inflection load L^* increases with τ. At low pilot lengths, there is a very minute difference between the inflection loads for different N. Beyond $\tau = 30$, we can see a very significant increase in the inflection load. This is because the pilot lengths are now sufficient enough to ensure that channel estimation errors reduce (i.e., the variance of the channel estimation errors have reduced enough) and improve the inflection loads. At $\tau = 10$, nearly all the curves are at $L^* = 1.5$. For $N = 16$, the inflection load highly improves from $L^* = 1.5$ at $\tau = 10$ to $L^* = 14$ at $\tau = 60$, which is an 833.33% increase in the inflection load. Here, even though $N = 16$ is enough to suppress interference, the system is limited by the channel estimation errors.

Fig. 7.5 studies the effect of the number of antennas on the inflection load for varied pilot lengths. As observed in the previous plot as well, for $\tau = 10$, the inflection load L^* stays fixed as we increase the number of antennas. This is because the system is limited by channel estimation errors. For $\tau = 20$, the inflection load only slightly increases from $N = 2$ to $N = 4$, and then stays fixed. For $\tau = 30$, the inflection load increases from approximately $L^* = 6$ to 7.5. Finally, for $\tau = 60$, the inflection load increases from $L^* = 7.5$ for $N = 2$ to 14 for $N = 16$, which is an 86.67% increase in the inflection load. In contrast to the perfect CSI case with MRC, where the impact of the number of antennas was incremental on the inflection load (see Fig. 7.12), here the impact is pretty significant. Thus, increasing both the pilot lengths and the number of antennas is

beneficial towards obtaining higher inflection loads.

7.5 Energy Efficiency Maximization

In this section, we focus on the EE of IRSA. We first define the EE metric and then later, obtain the corresponding optimal distributions in IRSA. In (7.8), we computed the throughput of IRSA, $\texttt{Thpt}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\})$ (packets per slot or RB), which we now use to compute the EE of IRSA, denoted by $\texttt{EE}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\})$. The spectral efficiency $\texttt{SE}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\})$ in IRSA can be computed in bps/Hz as

$$\texttt{SE}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\}) = (1 - \tau/\tau_c)\, \texttt{Thpt}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\}) \times \log_2(1 + \gamma_{\text{th}}), \quad (7.12)$$

where τ is the pilot length, τ_c is the length of the packet of any user, and γ_{th} is the SINR decoding threshold. Using the above, the EE can be conventionally calculated in bits/Joule as

$$\texttt{EE}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\}) = \frac{B\ \texttt{SE}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\})}{P_{\text{C}}}. \quad (7.13)$$

Here, B is the bandwidth, and P_{C} is the total power consumed, which is dependent on other factors such as circuit power at the BS. We use a well known realistic model for P_{C} as seen in [49, 150] as

$$P_{\text{C}} = P_{\text{Fix}} + P_{\text{TC}} + P_{\text{CE}} + P_{\text{CD}} + P_{\text{BH}} + P_{\text{SP}}, \quad (7.14)$$

where the terms are as follows: P_{Fix} is a fixed power required for control signaling and includes the load-independent power of backhaul infrastructure and baseband processors, P_{TC} accounts for the power consumed by the transceiver chains, P_{CE} is the power required for for the channel estimation process, P_{CD} accounts for the channel encoding and decoding powers, P_{BH} is the load-dependent backhaul signaling power, and P_{SP} is the signal processing power at the BS.

We now define the above terms specifically for IRSA. Firstly, recall that the number

of users transmitting per slot for an IRSA system with load L and with average repetition factor $\bar{d}$ is $L\bar{d}$. The transceiver power is calculated as $P_{\text{TC}} = NP_{\text{BS}} + L\bar{d}P_{\text{Users}} + P_{\text{LO}}$, where P_{BS} is the power required to run the circuit components per antenna at the BS, P_{Users} is the power required at each of the single-antenna users, and P_{LO} is the power required by the local oscillator at each BS (for a cell-free setup with a distributed antenna array, P_{LO} grows with N). The channel estimation process plays a huge rule in MIMO communications and the channel estimation power P_{CE} is dependent on the number of multiplication operations performed as detailed in Chapter 3. If the computational efficiency of the BS is denoted by η_C, then the channel estimation power P_{CE} is calculated as

$$P_{\text{CE}} = \frac{3B}{\tau_c \eta_C} N \cdot \begin{cases} s(L\bar{d}\tau + (L\bar{d})^2), & \text{MSBL Estimator,} \\ L\bar{d}\tau + (L\bar{d})^2, & \text{MMSE Estimator,} \\ L\bar{d}\tau + L\bar{d}, & \text{LCMMSE Estimator.} \end{cases} \tag{7.15}$$

Here, s is the average number of iterations the MSBL algorithm is run for. The encoding/decoding power contribution P_{CD} is calculated from the encoding power P_{Enc} and the decoding power P_{Dec} as $P_{\text{CD}} = (P_{\text{Enc}} + P_{\text{Dec}}) B\, \mathtt{SE}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\})$ according to [49]. Unlike [49], we model the encoding/decoding power contribution not just from the fraction of packets of the decoded users but from all the users. Hence, we model it as $P_{\text{CD}} = (P_{\text{Enc}} + P_{\text{Dec}}) B\, (1 - \tau/\tau_c)\, L \log_2(1 + \gamma_{\text{th}})$. The backhaul power P_{BH} is calculated from the backhaul traffic power P_{BT} as $P_{\text{BH}} = P_{\text{BT}} B\, \mathtt{SE}(\{\phi_2, \phi_3, \ldots, \phi_{d_{\max}}\})$. The signal processing power P_{SP} is a sum of the transmission/reception power $P_{\text{SP-TR}}$ and the combining (MMSE combining or MRC) power $P_{\text{SP-C}}$. These can be calculated as

$$P_{\text{SP-TR}} = \frac{3B\tau}{\tau_c \eta_C} L\bar{d}N, \tag{7.16}$$

$$P_{\text{SP-C}} = \begin{cases} \dfrac{3B}{\tau_c \eta_C} \left(\dfrac{3N(N+1)L\bar{d}}{2} + \dfrac{N^3 + 6N}{3} \right), & \text{MMSE Combining,} \\ \dfrac{7B}{\tau_c \eta_C} N, & \text{MR Combining.} \end{cases} \tag{7.17}$$

For the multi-cell and the cell-free setups, the above have to be changed as detailed in [49].

The EE optimization problem now becomes

$$\mathbb{P}_3: \quad \max_{\phi_2,\phi_3,\ldots,\phi_{d_{\max}}} \quad \texttt{EE}(\{\phi_2,\phi_3,\ldots,\phi_{d_{\max}}\}) \tag{7.18}$$

$$\text{subject to} \quad 0 \leq \phi_d \leq 1, \quad 2 \leq d \leq d_{\max}, \tag{7.19}$$

$$\sum_{d=2}^{d_{\max}} \phi_d = 1, \tag{7.20}$$

$$L = L_{\text{ref}}. \tag{7.21}$$

Here, the first two constraints ensure that the candidate solutions are indeed probability vectors. The third constraint ensures that the distributions are optimized at a reference load of L_{ref}. We use the DEA to numerically obtain the optimal distributions in the next section.

Remark 9: Existing works have considered simpler models to compute the EE. The simplest model for the EE is perhaps the ratio of the SE to the total power consumed [30]. For IRSA, this model can be written as

$$\texttt{EE}(\{\phi_2,\phi_3,\ldots,\phi_{d_{\max}}\}) = \frac{B\ \texttt{SE}(\{\phi_2,\phi_3,\ldots,\phi_{d_{\max}}\})}{P_{\text{Fix}} + NP_{\text{BS}} + L\bar{d}P_{\text{Users}}}. \tag{7.22}$$

The power consumed in the denominator of this simplistic model does not account for the SE. Even though we use the EE model from (7.13) to optimize the distributions, we will use the model from (7.22) in the next section purely to obtain insights into the dependence of the EE of IRSA on the system parameters such as N, L, $\bar{d}$, and τ.

Remark 10: The trivial power allocation for maximizing the EE in typical communications is to let the transmit power to be zero [30], since this minimizes the denominator of the EE. The equivalent of this for the energy efficient distribution design would be to use the trivial distribution $\phi(x) = x^2$ since it minimizes $\bar{d}$ in the denominator of the EE.

7.6 Numerical Results for Energy Efficiency Maximization

We now optimize the repetition distributions in order to maximize the EE of IRSA. For the results presented in this chapter, we operate the system with a reference load of $L_{\text{ref}} = L^*$ in order to operate the system at the peak throughput and the reported distributions are optimized at the inflection load to maximize the EE. The maximum EE obtained from the DEA is denoted by $\mathcal{E}_{\max}$ and is measured in kilobits per Joule [49]. The DEA parameters are $F = 0.9$, $M = 10N$, and $C_r = 0.9$ [149]. We choose the terminating criteria $\texttt{TC}(\cdot)$ as the standard deviation of the EEs of all the population vectors in any generation, and terminate the DEA generations when $\texttt{TC}(\cdot) < c_\varepsilon = 10^{-3}$ [149]. For the power consumption model, we consider the following set of parameters as in [49]: $\tau_c = 100$ packet length, $P_{\text{Fix}} = 10\,\text{W}$, $P_{\text{LO}} = 0.2\,\text{W}$, $P_{\text{BS}} = 0.4\,\text{W}$, $P_{\text{Users}} = 0.2\,\text{W}$, $P_{\text{Enc}} = 0.1\,\text{W/(Gbit/s)}$, $P_{\text{Dec}} = 0.8\,\text{W/(Gbit/s)}$, $P_{\text{BT}} = 0.25\,\text{W/(Gbit/s)}$, and $\eta_C = 75\,\text{Gflops/W}$. We use $B = 100\,\text{kHz}$ as the bandwidth, similar to the settings used by standard IoT-type devices in NB-IoT [3]. Since we consider low bandwidths of 100 kHz compared to 20 MHz in conventional cellular communications, the achievable spectral efficiencies (and hence energy efficiencies) are a magnitude of order smaller in IRSA.

7.6.1 K-Collision Channel

In Table 7.9, we present the EE optimal distributions for IRSA with the K-collision channel with varied maximum repetition factors. For $K = 1$, as we increase $d_{\max}$ (or alternately increase $\bar{d}$), the inflection load L^* increases from 0.82 to 0.95. The peak EE $\mathcal{E}_{\max}$ increases from 23.01 to 25.34 kilobits per Joule. For $K = 2$, as we increase $d_{\max}$, the inflection load L^* increases from 1.71 to 1.89, and $\mathcal{E}_{\max}$ increases from 46.77 to 49.36 kilobits per Joule. For $K = 3$, as we increase $d_{\max}$, the inflection load L^* increases from 2.57 to 2.75, and $\mathcal{E}_{\max}$ increases from 67.1 to 70.09 kilobits per Joule. We observe that we obtain higher peak EEs for higher K which is a result of improved throughputs since up to K collisions can be resolved simultaneously. In the above table, we observe that

Table 7.9: Energy Efficiency Optimal Distributions for IRSA with the K-Collision Channel.

K	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*	$\mathcal{E}_{\max}$
1	3	$0.24x^2 + 0.76x^3$	2.76	0.82	23.1
	4	$0.54x^2 + 0.46x^4$	2.91	0.86	24.12
	8	$0.47x^2 + 0.31x^3 + 0.09x^6 + 0.12x^7 + 0.01x^8$	3.27	0.92	25.04
	16	$0.47x^2 + 0.31x^3 + 0.06x^4 + 0.02x^5 + 0.02x^6$ $+0.05x^8 + 0.04x^9 + 0.01x^{10} + 0.01x^{12} + 0.01x^{16}$	3.47	0.95	25.34
2	3	$0.89x^2 + 0.11x^3$	2.11	1.71	46.77
	4	$0.96x^2 + 0.04x^4$	2.08	1.74	47.81
	8	$0.9x^2 + 0.01x^5 + 0.08x^7 + 0.01x^8$	2.49	1.85	48.78
	16	$0.9x^2 + 0.01x^4 + 0.01x^5 + 0.04x^7$ $+0.02x^8 + 0.01x^9 + 0.01x^{16}$	2.58	1.89	49.36
3	3	x^2	2	2.57	67.1
	4	$0.98x^2 + 0.02x^4$	2.04	2.58	68.81
	8	$0.91x^2 + 0.09x^8$	2.54	2.65	69.21
	16	$0.94x^2 + 0.04x^{12} + 0.01x^{14} + 0.01x^{16}$	2.66	2.75	70.09

increasing $d_{\max}$ increases $\bar{d}$ which increases the inflection load L^* as well as the energy efficiency. This is because in the expression for the energy efficiency in (7.22), in this regime, any increase in $\bar{d}$ increases the SE which dominates the corresponding small increase in the denominator.

The authors in [74] have claimed that CRDSA, i.e., a 2-regular distribution is the most energy efficient distribution for IRSA under every scenario, even under the collision channel. In fact, for the rest of this chapter, we show that it is not universally true. Our results above show that we can indeed obtain more efficient distributions than the 2-regular distribution. In several cases, a 2-regular distribution can be the most energy efficient distribution (e.g., at high pilot lengths and high number of antennas), but in general, it is not the EE optimal distribution.

Table 7.10: Energy Efficiency Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with the Normal Approximation and $N = 2$.

Parameters	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*	$\mathcal{E}_{\max}$
$\rho_0 = 1, \gamma_{\text{th}} = 1$	3	$0.07x^2 + 0.93x^3$	2.92	1.78	12.91
	4	$0.25x^2 + 0.49x^3 + 0.26x^4$	3.01	1.85	12.94
	8	$0.32x^2 + 0.56x^3 + 0.12x^8$	3.29	1.9	13.01
$\rho_0 = 5, \gamma_{\text{th}} = 5$	3	x^3	3	0.71	12.4
	4	x^4	4	0.85	16.74
	8	$0.4x^3 + 0.23x^4 + 0.08x^5 + 0.23x^8$	4.36	0.91	17.15
$\rho_0 = 10, \gamma_{\text{th}} = 10$	3	x^3	3	0.62	14.35
	4	$0.03x^2 + 0.97x^4$	3.94	0.75	19.78
	8	$0.48x^3 + 0.24x^4 + 0.28x^8$	4.63	0.8	20.41
$\rho_0 = 10, \gamma_{\text{th}} = 5$	3	x^3	3	1.05	20.97
	4	$0.23x^2 + 0.29x^3 + 0.48x^4$	3.25	1.11	21.68
	8	$0.15x^2 + 0.69x^3 + 0.16x^8$	3.63	1.17	22.45

Table 7.11: Energy Efficiency Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with the Normal Approximation and $N = 8$.

Parameters	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*	$\mathcal{E}_{\max}$
$\rho_0 = 1, \gamma_{\text{th}} = 1$	3	x^2	2	7.36	40.53
	8	x^2	2	7.36	40.53
	16	$0.99x^2 + 0.01x^{16}$	2.14	7.42	40.94
$\rho_0 = 5, \gamma_{\text{th}} = 5$	3	$0.68x^2 + 0.32x^3$	2.32	2.12	33.99
	4	$0.68x^2 + 0.32x^4$	2.64	2.24	35.08
	8	$0.79x^2 + 0.19x^5 + 0.01x^7 + 0.01x^8$	2.81	2.35	36.36
$\rho_0 = 10, \gamma_{\text{th}} = 10$	3	$0.46x^2 + 0.54x^3$	2.54	1.51	32.97
	8	$0.56x^2 + 0.22x^3 + 0.02x^7 + 0.2x^8$	3.52	1.69	35.77
	16	$0.5x^2 + 0.32x^3 + 0.02x^5 + 0.05x^7$ $+0.06x^{10} + 0.01x^{11} + 0.03x^{12} + 0.01x^{16}$	3.64	1.74	36.21
$\rho_0 = 10, \gamma_{\text{th}} = 5$	3	$0.67x^2 + 0.33x^3$	2.33	2.21	35.48
	4	$0.8x^2 + 0.2x^8$	3.2	2.29	36.14
	8	$0.78x^2 + 0.03x^3 + 0.19x^8$	3.17	2.45	38.17

Table 7.12: Energy Efficiency Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with $\rho_0 = 1, \gamma_{\text{th}} = 1$, and the Normal Approximation.

N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*	$\mathcal{E}_{\max}$
2	3	$0.07x^2 + 0.93x^3$	2.92	1.78	12.91
	4	$0.25x^2 + 0.49x^3 + 0.26x^4$	3.01	1.85	12.94
	8	$0.32x^2 + 0.56x^3 + 0.12x^8$	3.29	1.9	13.01
4	3	x^2	2	3.94	26.26
	4	$0.98x^2 + 0.01x^4$	2.04	3.97	26.52
	8	$0.93x^2 + 0.07x^8$	2.42	4.11	26.52
8	3	x^2	2	7.36	40.53
	8	x^2	2	7.36	40.53
	16	$0.99x^2 + 0.01x^{16}$	2.14	7.42	40.94
16	3	x^2	2	13.31	54.6
	16	$0.99x^2 + 0.01x^{16}$	2.14	13.35	54.94
	24	$0.98x^2 + 0.01x^4 + 0.01x^{24}$	2.24	13.37	55.28

7.6.2 Fading Channel with MIMO, Perfect CSI, and Maximal Ratio Combining

In Table 7.10, we present the EE optimal distributions for the fading channel with perfect CSI, MIMO, and MRC, with the Normal approximation and $N = 2$. Here, most of the EE optimal distributions are close to the 3- or 4-regular distributions and not the 2-regular distribution. For higher $d_{\max}$, we obtain slightly denser distributions for every combination of ρ_0 and γ_{th}. An important observation is that we obtain higher EEs for $\rho_0 = 10, \gamma_{\text{th}} = 10$ compared to $\rho_0 = 5, \gamma_{\text{th}} = 5$, whereas the inflection loads reduce for $\rho_0 = 10, \gamma_{\text{th}} = 10$ compared to $\rho_0 = 5, \gamma_{\text{th}} = 5$. A similar observation is made for $\rho_0 = 5, \gamma_{\text{th}} = 5$ compared to $\rho_0 = 1, \gamma_{\text{th}} = 1$, both of which have equal $\rho_0^{-1}\gamma_{\text{th}} = 1$. Even though the decodability of the users, which roughly depends upon the equal $\rho_0^{-1}\gamma_{\text{th}}$, the EE increases because the SE increases due to increase in γ_{th}. From $\rho_0 = 5, \gamma_{\text{th}} = 5$ to $\rho_0 = 10, \gamma_{\text{th}} = 5$, there is an increase in the SNR and the corresponding inflection loads and EEs both improve. In contrast, from $\rho_0 = 10, \gamma_{\text{th}} = 5$ to $\rho_0 = 10, \gamma_{\text{th}} = 10$, there is an increase in the SINR threshold and the corresponding inflection loads and EEs both reduce. This is because

Table 7.13: Energy Efficiency Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with $\rho_0 = 10, \gamma_{\text{th}} = 10$, and the Normal Approximation.

N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*	$\mathcal{E}_{\max}$
2	3	x^3	3	0.62	14.35
	4	$0.03x^2 + 0.97x^4$	3.94	0.75	19.78
	8	$0.48x^3 + 0.24x^4 + 0.28x^8$	4.63	0.8	20.41
4	3	$0.34x^2 + 0.66x^3$	2.66	1.11	27.63
	4	$0.36x^2 + 0.39x^3 + 0.25x^4$	2.88	1.15	28.7
	8	$0.37x^2 + 0.45x^3 + 0.18x^8$	3.53	1.25	35.52
8	3	$0.46x^2 + 0.54x^3$	2.54	1.51	32.97
	8	$0.56x^2 + 0.22x^3 + 0.02x^7 + 0.2x^8$	3.52	1.69	35.77
	16	$0.5x^2 + 0.32x^3 + 0.02x^5 + 0.05x^7$ $+0.06x^{10} + 0.01x^{11} + 0.03x^{12} + 0.01x^{16}$	3.64	1.74	36.21
16	3	$0.68x^2 + 0.32x^3$	2.32	2.18	38.04
	16	$0.71x^2 + 0.05x^3 + 0.03x^4 + 0.07x^5$ $+0.04x^6 + 0.02x^8 + 0.06x^{15} + 0.02x^{16}$	3.66	2.45	41.56
	24	$0.74x^2 + 0.06x^4 + 0.1x^6 + 0.03x^8$ $+0.02x^{12} + 0.02x^{15} + 0.01x^{19} + 0.02x^{24}$	3.77	2.46	41.58

γ_{th} is unilaterally increased, leading to fewer users getting decoded.

In Table 7.11 we present the EE optimal distributions for the fading channel with perfect CSI, MIMO, and MRC, with the Normal approximation and $N = 8$. In contrast to the previous table, we observe different trends. When the decodability at the BS improves due to increase in N from 2 to 8, the increase from $\rho_0 = 1, \gamma_{\text{th}} = 1$ to $\rho_0 = 5, \gamma_{\text{th}} = 5$ to $\rho_0 = 10, \gamma_{\text{th}} = 10$ monotonically reduces both the inflection loads as well as the peak EEs. This is because we are now operating in the regime where the higher number of antennas N consumes more power and the corresponding decodability does not increase a lot to counteract the effect of N. The peak EEs have, however, increased from the previous table since an increase in N helps the decodability of MRC and also improves the inflection loads. Finally, for $\rho_0 = 1, \gamma_{\text{th}} = 1$, the 2-regular distribution is optimal, but for the other sets of parameters, we obtain other distributions.

For both the above tables, we observe that increasing $\bar{d}$ in each configuration, increases the inflection load L^* as well as the EE. This is because in the expression for the EE in (7.22), in this regime, any increase in $\bar{d}$ increases the SE which dominates the corresponding small increase in the denominator. Now, increasing N here across the tables does not decrease the EE because the corresponding increase in the inflection load is high enough to dominate the 4-fold increase in N. This increase in N improves the decoding capability due to array gain and interference suppression. We can also observe that the average repetition factors of the optimal distributions at the inflection loads reduce across the tables when we increase N, since the denominator of the EE reduces.

In Table 7.12, we present the EE optimal distributions for the fading channel with MIMO, MRC, $\rho_0 = 1$, and $\gamma_{\text{th}} = 1$. As we increase the number of antennas N from 2 to 4 to 8 to 16, the maximum inflection loads increase from 1.9 to 4.11 to 7.42 to 13.37, and the maximum peak EEs increase from 13.01 to 26.52 to 40.94 to 55.28 kilobits per Joule. The optimal distributions as N increases becomes closer to the 2-regular distribution. Further, the inflection loads significantly increase with N. The peak EEs also significantly increase since the decodability with MRC highly improves with N which is also sufficient enough to overcome the extra power consumed due to higher N. The success probability θ_r,

under $\rho_0 = 1$ & $\gamma_{\text{th}} = 1$, is lower for every r for $N = 2$ compared to $N = 8$, which yields lower decodability for $N = 2$ compared to $N = 8$. In order to compensate for the lower decoding capability, the distribution is optimized to achieve distributions that are variedly different from $\phi(x) = x^2$. For $N = 4, 8, 16$, we observe that since θ_r is better, the inflection loads are higher, but the optimal distribution reduces to $\phi(x) = x^2$ since the effect of N and $\bar{d}$ dominates the SE.

In Table 7.13, we present the EE optimal distributions for the fading channel with MIMO, MRC, $\rho_0 = 10$, and $\gamma_{\text{th}} = 10$. As we increase the number of antennas N from 2 to 4 to 8 to 16, the maximum inflection loads increase from 0.8 to 1.25 to 1.74 to 2.46, which is a very small increase compared to the previous table. The maximum peak EEs increase from 20.41 to 35.52 to 36.21 to 41.58 kilobits per Joule. For $N = 2$ and 4, the peak EEs were higher in the previous table, but for $N = 8$ and 16, the peak EEs are higher in this table. This is because, even though the decodability stays roughly the same, the increase in the SNR is not sufficient enough to counteract the increase in the decoding threshold, at higher N. This is supported by a very small increase in inflection load which does not depend on the power consumed. This is the opposite for lower N. We also observe that the distributions are more dense compared to the previous table.

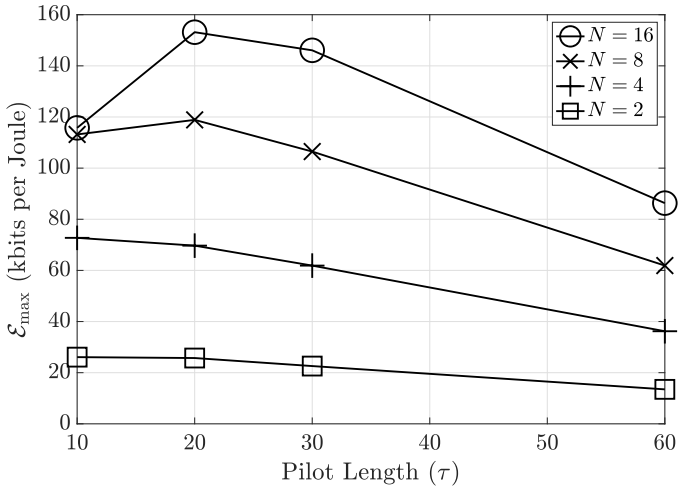
In both the above tables, for each N, we observe that increasing $\bar{d}$, increases the inflection load L^* as well as the EE. This is because in the expression for the EE in (7.22), in this regime, any increase in $\bar{d}$ increases the SE which dominates the corresponding small increase in the denominator. Increasing the number of antennas N has a higher effect on increasing the SE rather than the denominator of the EE, and thus, the EE improves. In Table 7.13, due to a higher $\gamma_{\text{th}} = 10$, the inflection loads and the EE do not significantly increase with N as it increased for $\gamma_{\text{th}} = 1$. However, we observe that the distributions become more dense with increase in N to compensate for the lower decodability due to higher γ_{th}. We also note that, in this regime, a good increase in $\bar{d}$ improves the SE and thus, even though the denominator of the EE reduces, the overall effect of the SE dominates the system and hence the EE improves.

7.6.3 Fading Channel with MIMO, Channel Estimation Errors, Pilot Contamination, and MMSE Combining

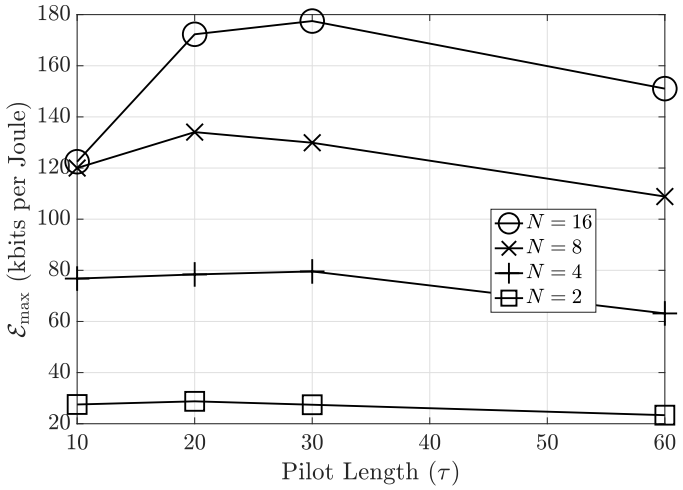
Table 7.14: Energy Efficiency Optimal Distributions for IRSA with the Fading Channel, MIMO, Channel Estimation Errors, Pilot Contamination, and MMSE Combining, with $\tau = 10, \rho_0 = 1$, and $\gamma_{\text{th}} = 1$.

N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*	$\mathcal{E}_{\max}$
2	3	x^3	3	1.64	9.91
	4	$0.07x^2 + 0.11x^3 + 0.82x^4$	3.75	1.76	11.96
	8	$0.01x^2 + 0.32x^3 + 0.64x^4 + 0.03x^8$	3.78	1.82	12.74
4	3	x^2	2	5.11	33.12
	4	x^2	2	5.11	33.12
	8	$0.99x^2 + 0.01x^8$	2.06	5.18	33.37
8	3	x^2	2	8.1	43.81
	8	x^2	2	8.1	43.81
	16	$0.99x^2 + 0.01x^{16}$	2.14	8.18	43.92
16	3	x^2	2	10.8	46.46
	16	x^2	2	10.8	46.46
	24	$0.97x^2 + 0.01x^3 + 0.01x^{19} + 0.01x^{24}$	2.4	10.86	46.83

In Table 7.14 we present the EE optimal distributions for the fading channel with MIMO, MMSE combining, pilot contamination, and channel estimation errors, with $\tau = 10, \rho_0 = 1$, and $\gamma_{\text{th}} = 1$. It can be seen that the 2-regular distribution is optimal or nearly optimal for $N \geq 4$ and distributions with higher $\bar{d}$ are optimal for $N = 2$. In Table 7.15 we present the EE optimal distributions for the fading channel with MIMO, MMSE combining, pilot contamination, and channel estimation errors, with $\tau = 10, \rho_0 = 10$, and $\gamma_{\text{th}} = 10$. It can be seen that the 2-regular distribution is optimal or nearly optimal for $N \geq 8$ and distributions with higher $\bar{d}$ are optimal for $N = 2$ and 4. In both the tables, the inflection loads highly increase with the number of antennas due to improved suppression of interfering users due to MMSE combining. The performance improvement is drastic since MMSE combining is better than MRC, at the cost of higher complexity. This improvement is also seen in the EEs: MMSE combining consumes higher power compared to MRC.



(a) $\tau_c = 100$.



(b) $\tau_c = 200$.

Figure 7.6: Effect of the pilot length on the peak energy efficiency $\mathcal{E}_{\max}$ with varied τ_c, for the fading channel with MIMO, MMSE combining, pilot contamination, and channel estimation errors.

Table 7.15: Energy Efficiency Optimal Distributions for IRSA with the Fading Channel, MIMO, Channel Estimation Errors, Pilot Contamination, and MMSE Combining, with $\tau = 10, \rho_0 = 10$, and $\gamma_{\text{th}} = 10$.

N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*	$\mathcal{E}_{\max}$
2	3	x^3	3	0.89	20.02
	4	x^4	4	1.01	25.76
	8	$0.4x^3 + 0.45x^4 + 0.15x^8$	4.2	1.05	26.08
4	3	x^2	2	2.98	71.31
	4	$0.98x^2 + 0.02x^4$	2.04	3.05	71.61
	8	$0.94x^2 + 0.06x^8$	2.36	3.12	72.73
8	3	x^2	2	5.7	113.18
	8	x^2	2	5.7	113.18
	16	$0.99x^2 + 0.01x^{16}$	2.14	5.78	113.45
16	3	x^2	2	7.2	115.75
	16	x^2	2	7.2	115.75
	24	$0.98x^2 + 0.01x^{18} + 0.01x^{24}$	2.38	7.27	116.06

The huge increase in EE is because of high SE improvement due to MMSE combining, but the effect on the distribution is different. In both the tables, we observe that with low L^* (e.g., with $N = 2$ and $N = 4$), increasing $d_{\max}$ to 8 optimizes the distribution to a variedly different distribution and also increases the EEs. This is not observed with high L^* (e.g., with $N = 8$ and $N = 16$), where the distributions only slightly vary. This is because the extra power consumed due to higher number of replicas and operating at higher L is not dominant enough to improve the SE but would actually reduce $\bar{d}$ in the denominator of the EE in (7.22). The peak EEs are much higher with MMSE combining than MRC due to improved interference suppression. When going from $\rho_0 = 1, \gamma_{\text{th}} = 1$ to $\rho_0 = 10, \gamma_{\text{th}} = 10$, the peak EEs reduce for MRC as we saw earlier. Unlike what was observed with MRC, for MMSE combining, the peak EEs improve when going from $\rho_0 = 1, \gamma_{\text{th}} = 1$ to $\rho_0 = 10, \gamma_{\text{th}} = 10$. For MRC, the effect of improved decodability is eclipsed by the higher amount of power consumed due to higher N. However, for MMSE combining, the amount of power consumed due to more computations and high N is

Table 7.16: Energy Efficiency Optimal Distributions for IRSA with the Fading Channel, MIMO, Channel Estimation Errors, Pilot Contamination, and MMSE Combining, with $d_{\max} = 8, \rho_0 = 10$, and $\gamma_{\text{th}} = 10$.

N	τ	$\phi(x)$	$\bar{d}$	L^*	$\mathcal{E}_{\max}$ for $\tau_c = 100$	$\mathcal{E}_{\max}$ for $\tau_c = 200$
2	10	$0.4x^3 + 0.45x^4 + 0.15x^8$	4.2	1.05	26.08	27.53
	20	$0.41x^3 + 0.5x^4 + 0.09x^8$	3.95	1.15	25.73	28.76
	30	$0.52x^3 + 0.38x^4 + 0.1x^8$	3.88	1.2	22.59	27.43
	60	$0.64x^3 + 0.26x^4 + 0.1x^8$	3.76	1.22	13.48	23.35
4	10	$0.94x^2 + 0.06x^8$	2.36	3.12	72.73	76.77
	20	$0.96x^2 + 0.04x^8$	2.24	3.37	69.69	78.39
	30	$0.97x^2 + 0.03x^8$	2.18	3.41	61.88	79.56
	60	$0.96x^2 + 0.04x^8$	2.24	3.52	36.21	63.14
8	10	x^2	2	5.7	113.18	120
	20	x^2	2	6.97	118.87	134.05
	30	x^2	2	7.13	106.48	129.89
	60	x^2	2	7.34	61.89	108.85
16	10	x^2	2	7.22	115.75	122.47
	20	x^2	2	11.82	153.16	172.31
	30	x^2	2	13.24	146.07	177.48
	60	x^2	2	13.86	86.31	151.05

eclipsed by the decodability of MMSE. Thus, MRC and MMSE combining have different trends in the peak EEs.

In Table 7.16, we present the EE optimal distributions for IRSA in the general case, when the pilot length is varied. For $N = 2$, the 2-regular distribution is not optimal whereas for $N = 4$, it is nearly optimal, and for $N = 8$ and 16, it is optimal. As the pilot length is increased for $N = 2$ and 4, the average repetition factor of the optimized distribution reduces whereas the inflection loads increase. The inflection loads increase due to improved channel estimation. For $N = 4$, the inflection load increases only marginally with an increase in τ, since the system is limited by decodability even if the channel estimates vastly improve. For $N = 16$, the inflection load increases significantly with an increase in τ, since the system is not limited by decodability. Here, the channel estimates vastly improve and thus, the inflection loads highly increase. At low N, the success probability θ_r is lower for every r compared to higher N (e.g., with $N = 8$), which yields lower decodability for $N = 2$ compared to $N = 8$. In order to compensate for the lower decoding capability, the distribution is optimized to obtain distributions that are different from $\phi(x) = x^2$. At high N, the system already performs very well to obtain high L^*, and thus the optimal distribution would just result in the minimizer of the denominator of the EE, which is the 2-regular distribution. We also present the peak EEs for varied τ_c. The optimal distributions do not change when we change τ_c, but the trends in the EE are different, as we will see next.

In Fig. 7.6, we study the impact of the pilot length τ and the number of antennas N on the peak EE of IRSA. The EE is dependent on the packet length τ_c, via both the SE and the power consumed. For $\tau_c = 100$, $\tau = 60$ achieves the lowest peak EE since a majority of the time is spent in channel estimation. This effect is not indicative in either the throughput or the inflection loads since their definitions do not include τ_c. Here, $\tau_c = 10, 20$ achieves the highest peak EE for $N = 2/4$ and $N = 8/16$, respectively. For $\tau_c = 200$ and $N \leq 8$, $\tau = 60$ achieves the lowest peak EE since a majority of the time is spent in channel estimation. For $\tau_c = 200$ and $N = 16$, $\tau = 10$ achieves the lowest peak EE since channel estimates are not good enough to decode users. This inverse

trend is due to increase in τ_c. EE improves due to both increase in SE due to higher L^* and lower $\bar{d}$, but this happens only in the regime where the pilot length is not too high enough to eclipse the length of the data symbols. In both the subfigures, the peak EEs monotonically increase with an increase in N. As discussed earlier, this is due to the dominant effect of the throughput over the power consumed due to higher N. Unlike the trends in the inflection loads seen with the throughput optimal distributions previously, where the inflection loads monotonically increase with τ, here, it is not the case. This is because of the pre-log factor $(1-\tau/\tau_c)$ within the SE used to compute the EE. Thus, there is a specific pilot length at which the EE is maximized at which IRSA can be operated at the peak EE.

7.6.4 Comparison with Existing Distributions

We now compare the optimized distributions presented in this chapter with existing distributions [7], [17], and the soliton distribution in [72] (with $a = 0$ in (7.1)), in Table 7.17. For this, we consider only the general case, i.e., IRSA with MIMO, MMSE combining, channel estimation errors, and pilot contamination. For comparison, we present the peak performance with the 2-regular distribution, i.e., with CRDSA. CRDSA achieves an inflection load of $L^* = 0.55$ and peak EE of $\mathcal{E}_{\max} = 14.92$ kilobits per Joule for $N = 2$ and an inflection load of $L^* = 2.98$ and peak EE of $\mathcal{E}_{\max} = 71.31$ kilobits per Joule for $N = 2$. A high peak EE is obtained due to lower $\bar{d}$ which dominates the denominator of the EE. For each $d_{\max}$, we compare our optimized distribution with the soliton distribution (see (7.1)) and also with other existing distributions (when available). We observe that for every $d_{\max}$, our optimized distributions perform better than existing ones. The only exception is $d_{\max} = 3, N = 4$, for which the 2-regular distribution is indeed optimal. Hence, our optimized distributions perform either same as or better than existing distributions.

For $N = 2$ and $d_{\max} = 3$, our optimized distribution is the 3-regular distribution, which achieves a 51% increase in the inflection load and a 25% increase in the peak EE compared to the Soliton distribution which achieves an inflection load of 0.59. For $N = 2$ and $d_{\max} = 4$, our optimized distribution is the 4-regular distribution, which achieves: a

Table 7.17: Comparison of our energy efficiency optimized distributions with existing distributions under MIMO, MMSE combining, channel estimation errors, and pilot contamination, with $\tau = 10, \rho_0 = 10$, and $\gamma_{\text{th}} = 10$.

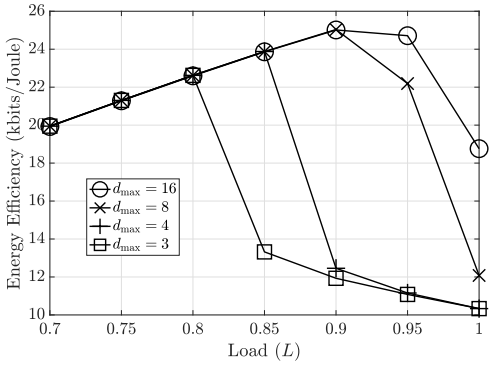
N	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*	% Imp.	$\mathcal{E}_{\max}$	% Imp.
2	2	2-regular: x^2	2	0.55	-	14.92	-
	3	Our: x^3	3	0.89	-	20.02	-
		Soliton	2.25	0.59	51	15.97	25
	4	Our: x^4	4	1.01	-	25.76	-
		[17]: $0.66x^2 + 0.16x^3 + 0.18x^4$	2.52	0.64	58	17.24	49
		[7]: $0.5102x^2 + 0.4898x^4$	2.49	0.63	60	17.08	51
		Soliton	2.44	0.62	63	16.87	53
	8	Our: $0.4x^3 + 0.45x^4 + 0.15x^8$	4.2	1.05	-	26.08	-
		[7]: $0.5x^2 + 0.28x^3 + 0.22x^8$	3.6	0.8	31	21.2	23
		Soliton	2.96	0.71	48	19.09	37
	16	Our: $0.16x^3 + 0.78x^4 + 0.03x^{15} + 0.03x^{16}$	4.53	1.06	-	26.4	-
		Soliton	3.54	0.78	36	20.56	28
		[17]: $0.59x^2 + 0.27x^3 + 0.02x^5 + 0.12x^{16}$	4.01	0.73	45	19.22	37
4	2	2-regular: x^2	2	2.98	-	71.31	-
	3	Our: x^2	2	2.98	-	71.31	-
		Soliton	2.25	2.96	0.7	70.11	1.7
	4	Our: $0.98x^2 + 0.02x^4$	2.04	3.05	-	71.61	-
		Soliton	2.44	2.91	4.8	68.46	4.6
		[17]: $0.66x^2 + 0.16x^3 + 0.18x^4$	2.52	2.89	5.5	67.82	5.6
		[7]: $0.5102x^2 + 0.4898x^4$	2.49	2.84	7.4	67.09	6.7
	8	Our: $0.94x^2 + 0.06x^8$	2.36	3.12	-	72.73	-
		Soliton	2.96	2.75	13	63.71	14
		[7]: $0.5x^2 + 0.28x^3 + 0.22x^8$	3.6	2.51	24	57.4	27
	16	Our: $0.93x^2 + 0.03x^9 + 0.03x^{10} + 0.01x^{16}$	2.59	3.15	-	73.2	-
		Soliton	3.54	2.57	23	58.72	25
		[17]: $0.59x^2 + 0.27x^3 + 0.02x^5 + 0.12x^{16}$	4.01	2.54	24	57.11	28

58% increase in L^* and a 49% increase in $\mathcal{E}_{\max}$ compared to the distribution in [17]. For $N = 2$ and $d_{\max} = 8$, our optimized distribution achieves a 31% increase in the inflection load and a 23% increase in the peak EE compared to the best existing distribution. For $N = 2$ and $d_{\max} = 16$, our optimized distribution achieves a 36% increase in the inflection load and a 28% increase in the peak EE compared to the best existing distribution. For $N = 4$, the percentage improvements in the inflection loads are as low as 0.7% at $d_{\max} = 2$ and as high as 23% at $d_{\max} = 16$ compared to the best existing distributions; the percentage improvements in the peak EEs are as low as 1.7% at $d_{\max} = 2$ and as high as 25% at $d_{\max} = 16$ compared to the best existing distributions. For higher N, as we have seen before, the optimal distributions become closer to the 2-regular distribution. While the percentage improvement in both the inflection load and the peak EE may reduce for higher N, the optimized distributions presented in this chapter always perform better than or the same as the best existing distributions. Finally, to improve the peak EE, we can increase $d_{\max}$ (and consequently $\bar{d}$) as observed in Table 7.17. This is because the improved throughput in the numerator of the EE dominates $\bar{d}$ in the denominator of the EE.

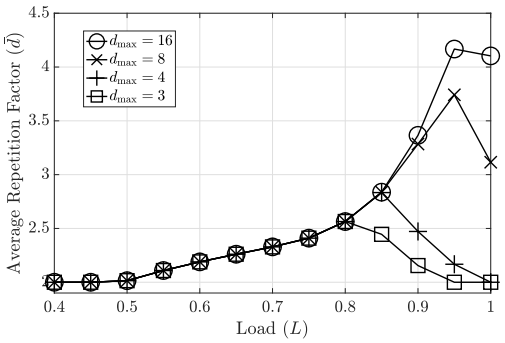
7.6.5 Variation of the Optimal Distributions with Load

So far we have optimized the repetition distributions of IRSA only at the inflection load, i.e., for $L = L^*$, and reported the distributions that maximize the throughput or the EE at the inflection point only. In this section, we now optimize the repetition distributions for each L and report the performance achieved by the optimal distributions at each L. We perform this by setting $L_{\text{ref}} = L$ in (7.21), and optimizing the distribution for every L, possibly even beyond the inflection load. Beyond the inflection load, even though the throughput of $\mathcal{T} = L$ is not achievable, we still assume it is indeed the reference load and optimize the distribution. As we will see, we indeed obtain suboptimal throughputs beyond L^* as a reason, and thus, the main regime of interest for the results presented in this section is $L \leq L^*$. Since reporting the optimized distributions at each L for each of the configurations is laborious, we only present the average repetition factors of the

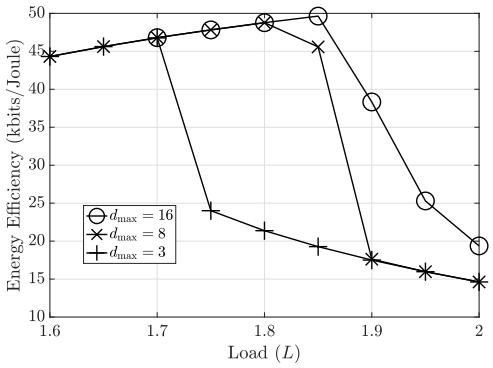
optimized distributions at each load.



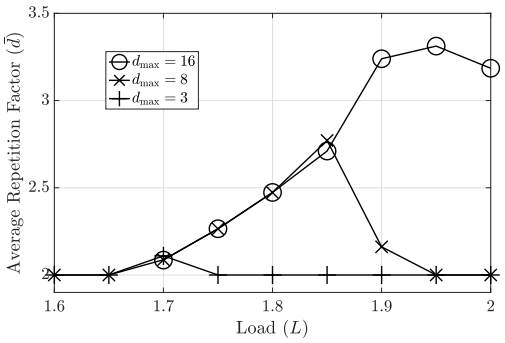
(a) EE achieved by the optimal distribution with $K = 1$.



(b) Average repetition factor of the optimal distribution with $K = 1$.



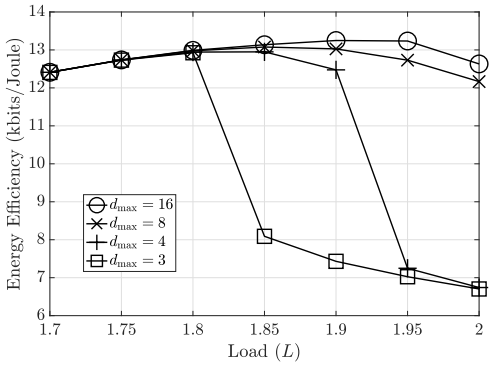
(c) EE achieved by the optimal distribution with $K = 2$.



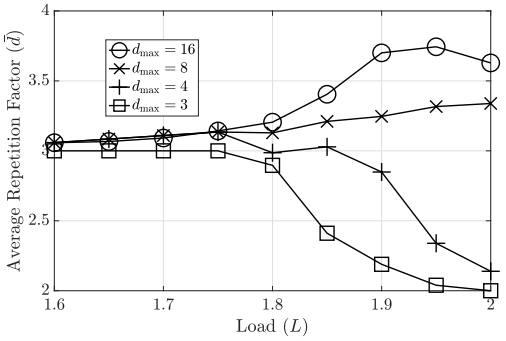
(d) Average repetition factor of the optimal distribution with $K = 2$.

Figure 7.7: Variation of the EE with L, for the K-collision channel.

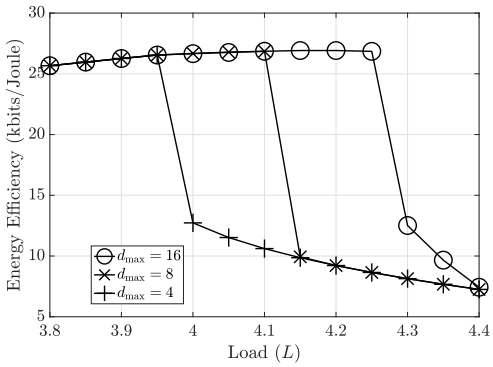
In Fig. 7.7, we present the variation of the EEs of the K-collision channel achieved by the distributions optimized for each L. For $K = 1$, the inflection load occurs at $L^* = 0.8$, 0.85, 0.9, and 0.9 for $d_{\max} = 3$, 4, 8, and 16. For $K = 2$, the inflection load occurs at $L^* = 1.7$, 1.8, and 1.85 for $d_{\max} = 3$, 8, and 16. In each of the cases, up to L^*, the energy efficiency increases monotonically with L, hits a peak and then starts dropping, similar to the trends in throughput observed in Chapter 3. All the distributions at low loads are $\phi(x) = x^2$, since it is the only distribution with $\bar{d} = 2$. At the inflection loads, the peak EE and the optimal distributions for the above were presented before in Table 7.9. The



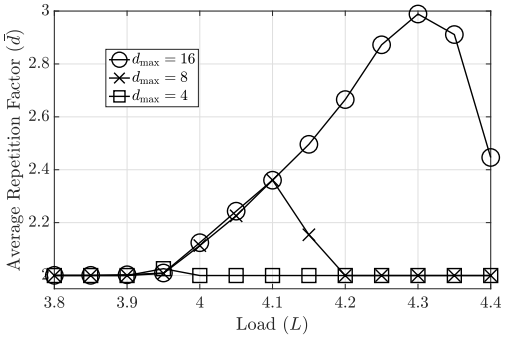
(a) EE achieved by the optimal distribution for $N = 2$.



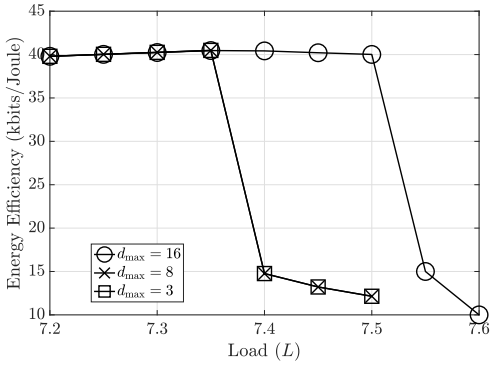
(b) Average repetition factor of the optimal distribution for $N = 2$.



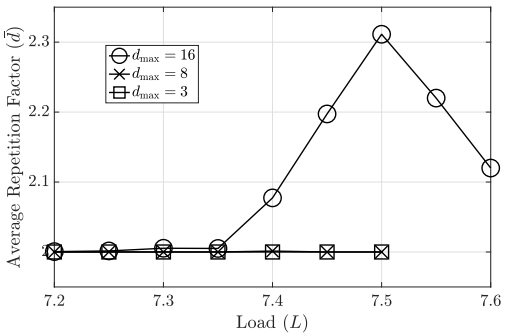
(c) EE achieved by the optimal distribution for $N = 4$.



(d) Average repetition factor of the optimal distribution for $N = 4$.

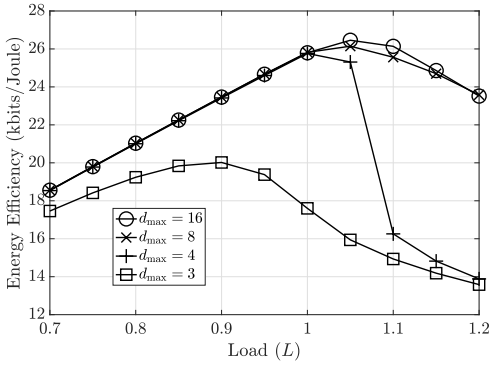


(e) EE achieved by the optimal distribution for $N = 8$.

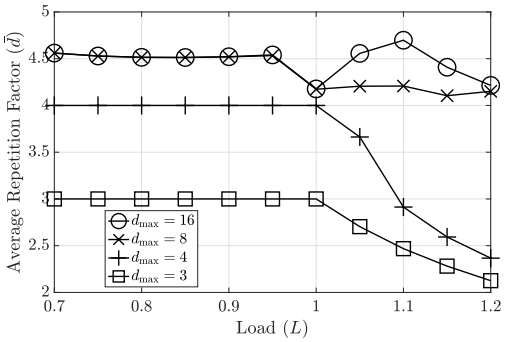


(f) Average repetition factor of the optimal distribution for $N = 8$.

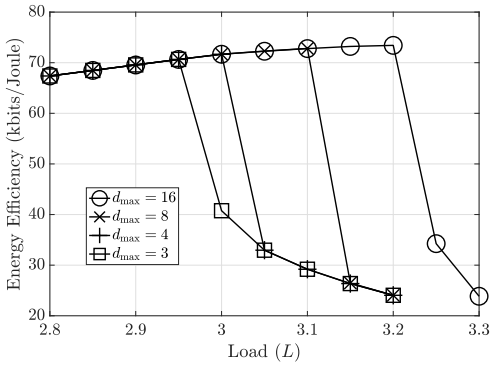
Figure 7.8: Variation of the EE with L, for the fading channel with MIMO, MRC, and the Normal approximation with $\rho_0 = 1, \gamma_{\text{th}} = 1$.



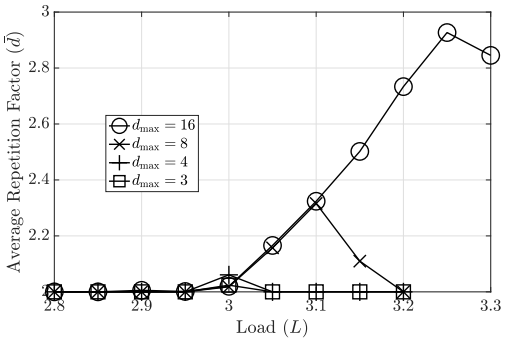
(a) EE achieved by the optimal distribution for $N = 2$.



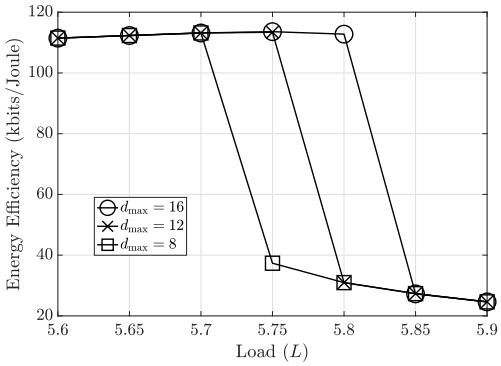
(b) Average repetition factor of the optimal distribution for $N = 2$.



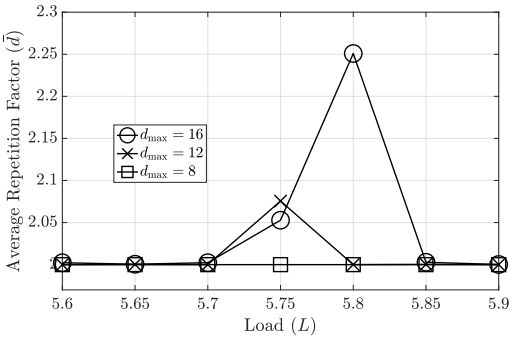
(c) EE achieved by the optimal distribution for $N = 4$.



(d) Average repetition factor of the optimal distribution for $N = 4$.

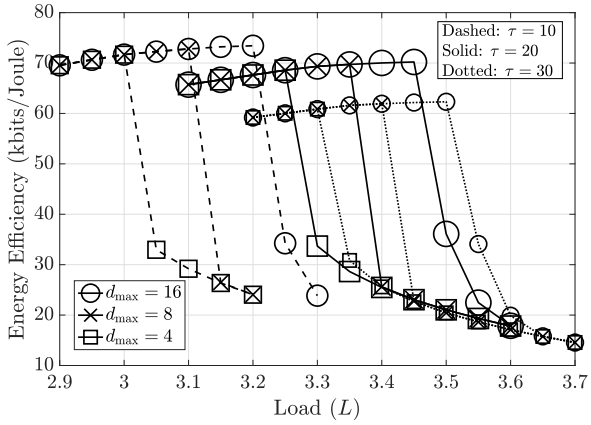


(e) EE achieved by the optimal distribution for $N = 8$.

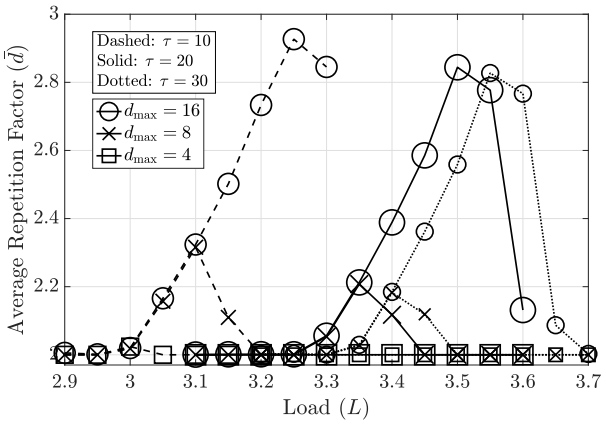


(f) Average repetition factor of the optimal distribution for $N = 8$.

Figure 7.9: Variation of the EE with L, for the fading channel with MIMO, MMSE combining, pilot contamination, and channel estimation errors, with $\tau = 10$, $\rho_0 = 10$, and $\gamma_{\text{th}} = 10$.



(a) EE achieved by the optimal distribution.



(b) Average repetition factor of the optimal distribution.

Figure 7.10: Variation of the EE with L, for the fading channel with MIMO, MMSE combining, pilot contamination, and channel estimation errors, with $N = 4$, $\rho_0 = 10$ and $\gamma_{\text{th}} = 10$.

optimal distributions exhibit an interesting trend as described below. At loads much less than L^*, the distributions remain the same for every $d_{\max}$ (all are $\phi(x) = x^2$). At loads near the inflection load ($0.5 < L < 0.8$ for $K = 1$, and $1.65 < L < 1.8$ for $K = 2$), the distributions are not the 2-regular distributions anymore; they are optimized with higher $\bar{d}$ to obtain higher SEs near the inflection load. For a few of the plotted configurations, the optimized distributions at very low and very high L are the 2-regular distributions. At loads beyond the inflection load, the SE is poor, and thus, the $\bar{d}$ of the optimal distribution starts dropping to increase EE. Thus, we infer that optimizing the repetition distributions near the inflection load helps in improving the inflection load as well as the energy efficiency of the system.

At these low loads of $0.7 < L < 1$ for $K = 1$ (and $1.6 < L < 2$ for $K = 2$) increasing $d_{\max}$ improves the SE which dominates the small increase in L and $\bar{d}$ in the denominator of the EE in (7.22). In fact, we see that for $K = 2$, the rate of increase of the EE is lower than the rate of increase of the EE for $K = 1$. Since $K = 2$ can achieve higher L^* than $K = 1$, any corresponding increase in L for $K = 2$ results in a correspondingly smaller increase in the EE due to higher load of operation. Increasing $d_{\max}$ significantly improves L^* and the EE initially (i.e., from $d_{\max} = 3$ to $d_{\max} = 4$ to $d_{\max} = 8$) and then only marginally (to $d_{\max} = 16$). As we saw before, from $d_{\max} = 3$ to $d_{\max} = 8$ increases the SE significantly but increasing it to $d_{\max} = 16$ does not warrant the increase in $\bar{d}$ to increase the EE. For example, with $K = 1$, increasing $\bar{d}$ to 4.2 at $L = 0.95$ does not increase the EE; in fact there is a small drop in the EE from $L = 0.9$ to $L = 0.95$ with a huge increase in $\bar{d}$ from 3.4 to 4.2. Hence, beyond L^*, both EE and $\bar{d}$ starts reducing.

In Fig. 7.8, we present the variation of the EEs of the fading channel (with MIMO, MRC, and the Normal approximation with $\rho_0 = 1, \gamma_{\text{th}} = 1$) achieved by the distributions optimized for each L. For $N = 2$, the inflection load occurs at $L^* = 1.8$, 1.85, 1.9, and 1.95 for $d_{\max} = 3$, 4, 8, and 16. For $N = 4$, the inflection load occurs at $L^* = 3.95$, 4.1, and 4.25 for $d_{\max} = 4$, 8, and 16. For $N = 8$, the inflection load occurs at $L^* = 7.35$, 7.35, and 7.5 for $d_{\max} = 3$, 8, and 16. At the inflection loads, the peak EE and the optimal distributions for the above were presented before in Table 7.12. Previously we observed

that the energy efficiency increases monotonically with L up to L^*, hits a peak and then starts dropping. The same is true here for $N = 2$, but is not true for $N = 4$ or $N = 8$. For $N = 4$ and $N = 8$, the EEs remain roughly equal with increase in L up to L^*: for $N = 4$, the EEs remain the same; for $N = 8$, the EEs dip only slightly; for $L > L^*$, the EEs drop with L as before. This behaviour is because the system is now in the regime where any increase in L would dominate the denominator of the EE rather than the SE. For $N = 4$ and $N = 8$, the huge increase in $\bar{d}$ has roughly the same impact as the increase in SE, and thus, the EEs do not change significantly. In fact, we see that for $N = 8$, the rate of increase of the EE is lower than the rate of increase of the EE for $N = 4$, which is lower than the rate of increase of the EE for $N = 2$. Since $N = 8$ can achieve higher L^* than $N = 4$, any corresponding increase in L for $N = 8$ results in a correspondingly smaller increase in the EE due to higher load of operation. Further, for $N = 2$, we observe that the optimal distributions for $L < 1.8$ under all the configurations are close to the 3-regular distribution ($\phi(x) = x^3$): for $d_{\max} = 3$, it is exactly the 3-regular distribution and for higher $d_{\max}$ it is still close to the 3-regular distribution.

Beyond $L = 1.8$, $\bar{d}$ drops down to 2 to achieve higher EE, i.e., the 2-regular distribution is obtained. For higher $d_{\max}$, near the inflection loads, the SE increases as $\bar{d}$ increases, and thus we obtain higher EEs for all N. Increasing $d_{\max}$ significantly improves L^* and the EE initially (i.e., from $d_{\max} = 3$ to $d_{\max} = 4$ to $d_{\max} = 8$) and then only marginally (to $d_{\max} = 16$). For $N = 8$, this change is invisible from $d_{\max} = 3$ to $d_{\max} = 8$, but is visible with $d_{\max} = 16$. For $N = 4$ and $N = 8$, all the distributions at low loads are $\phi(x) = x^2$, since it is the only distribution with $\bar{d} = 2$. At loads much less than L^*, the distributions remain the same for every $d_{\max}$ (all are $\phi(x) = x^2$). At loads near the inflection load ($3.95 < L < 4.3$ for $N = 4$, and $7.35 < L < 7.5$ for $N = 8$), the distributions are not the 2-regular distributions; they are optimized with higher $\bar{d}$ to obtain higher SEs near the inflection load. At loads beyond the inflection load, the SE is poor, and thus, the $\bar{d}$ of the optimal distribution starts dropping to increase EE. Thus, for the above configurations, we infer that optimizing the repetition distributions near the inflection load helps in achieving a higher inflection load, but the energy efficiency of the system

remains roughly the same.

In Fig. 7.9, we present the variation of the EEs of the fading channel (with MIMO, MMSE combining, pilot contamination, and channel estimation errors, with $\tau = 10$, $\rho_0 = 10$, and $\gamma_{\text{th}} = 10$) achieved by the distributions optimized for each L. For $N = 2$, the inflection load occurs at $L^* = 0.9$, 1, 1.05, and 1.05 for $d_{\max} = 3$, 4, 8, and 16. For $N = 4$, the inflection load occurs at $L^* = 2.95$, 3, 3.1, and 3.2 for $d_{\max} = 3$, 4, 8, and 16. For $N = 8$, the inflection load occurs at $L^* = 5.7$, 5.75, and 5.8 for $d_{\max} = 8$, 12, and 16. At the inflection loads, the peak EE and the optimal distributions for the above were presented before in Table 7.15. Similar to the results with MRC, the energy efficiency increases monotonically with L up to L^*, hits a peak and then starts dropping for $N = 2$, for $N = 4$ the EE very marginally improves with L, and for $N = 8$ the EEs remain roughly equal with increase in L up to L^*. This is because the system is once again in the regime where any increase in L would dominate the denominator of the EE rather than the SE. For $N = 8$, the increase in $\bar{d}$ has roughly the same impact as the increase in SE, and thus, the EEs do not change significantly. Similar to before, the rate of increase of the EEs with respect to L is lower for higher N. Since we can achieve higher L^* with higher N, any corresponding increase in L for a higher N results in a correspondingly smaller increase in the EE due to higher load of operation.

For $N = 2$, unlike before, we observe that the optimal distributions for $L < 1$ under all the configurations are different from each other. In order to compensate for the lower decoding capability, the distribution is optimized to achieve distributions that are variedly different from $\phi(x) = x^2$. For $d_{\max} = 3$, it is $\phi(x) = x^3$; for $d_{\max} = 4$, it is $\phi(x) = x^4$; and for $d_{\max} = 8, 16$ it is a distribution with higher average repetition factor of $\bar{d} = 4.5$. Beyond $L = 1$, for $d_{\max} = 3/4$, $\bar{d}$ drops down close to 2 to achieve higher EE, i.e., close to the 2-regular distribution. For $d_{\max} = 8, 16$, $\bar{d}$ remains roughly the same, and increases till $L = 1.1$ and then drops down, respectively. For higher $d_{\max}$, near the inflection loads, the SE increases as $\bar{d}$ increases, and thus we obtain higher EEs for all N. For $N = 2$, increasing $d_{\max}$ significantly improves L^* and the EE initially (i.e., from $d_{\max} = 3$ to 4 to 8) and then only marginally (to $d_{\max} = 16$). Similar trends are observed for $N = 4$

and $N = 8$, but the rate of improvement of the EE reduces. For $N = 4$ and $N = 8$, all the distributions at low loads are $\phi(x) = x^2$, since it is the only distribution with $\bar{d} = 2$. At loads much less than L^*, the distributions remain the same for every $d_{\max}$ (all are $\phi(x) = x^2$). At loads near the inflection load ($2.95 < L < 3.2$ for $N = 4$, and $5.7 < L < 5.8$ for $N = 8$), the distributions are not the 2-regular distributions anymore; they are optimized with higher $\bar{d}$ to obtain higher SEs near the inflection load. For $L > L^*$, the SE is poor, and thus, the $\bar{d}$ of the optimal distribution starts dropping to increase EE. Thus, similar to the previous case, for the above configurations, we infer that optimizing the repetition distributions near the inflection load helps in achieving a higher inflection load, but the EE of the system remains roughly the same.

In Fig. 7.10, we study the effect of the pilot length τ on the variation of the optimal EE distributions for each L, for the fading channel with $N = 4$ antennas, MMSE combining, pilot contamination, and channel estimation errors (with $\rho_0 = 10$ and $\gamma_{\text{th}} = 10$). For $\tau = 10$, the inflection load occurs at $L^* = 3$, 3.1, and 3.2 for $d_{\max} = 4$, 8, and 16. For $\tau = 20$, the inflection load occurs at $L^* = 3.25$, 3.35, and 3.45 for $d_{\max} = 4$, 8, and 16. For $\tau = 30$, the inflection load occurs at $L^* = 3.3$, 3.4, and 3.5 for $d_{\max} = 4$, 8, and 16. Each set of curves for a specific τ follow the same observations about the EE as what we have seen earlier in this section; hence, we now talk only about the effect of variation in τ. With increase in τ, we observe that the EE curves and the average repetition factor curves both shift rightward; the peak EE and the peak $\bar{d}$ reduces. Increasing in τ improves the quality of the channel estimates, leading to higher throughputs and inflection loads. However, the change in the inflection loads is not high enough to justify using a higher pilot length. The SE reduces due to the pre-log factor $(1 - \tau/\tau_c)$ and thus, the peak EE reduces with increase in τ as seen in (7.22).

We observe that with loads near L^* (for each τ), increasing $d_{\max}$ from 4 to 8 or 16 optimizes the distribution to a variedly different distribution. For higher $d_{\max}$, near the inflection loads ($3 < L < 3.2$ for $\tau = 10$, $3.25 < L < 3.4$ for $\tau = 20$, and $3.3 < L < 3.5$ for $\tau = 30$), the SE increases as $\bar{d}$ increases, and thus we obtain higher EEs for all τ. For each τ, all the distributions at loads much lower than L^* are $\phi(x) = x^2$, since it is the

only distribution with $\bar{d} = 2$. At loads much less than L^*, the distributions remain the same for every $d_{\max}$ (all are $\phi(x) = x^2$). For $L > L^*$, the SE is poor, and thus, the $\bar{d}$ of the optimal distribution starts dropping to increase EE as L increases. We note that these trends are for $N = 4$ and $\tau_c = 100$, and under these parameters $\tau = 10$ performs the best as observed earlier in Table 7.16. For different N and τ_c, any other τ may be the maximizer of the EE.

7.7 Summary

In this chapter, we optimized the repetition distributions of IRSA using the differential evolution genetic algorithm. Firstly, we optimized the repetition distributions of IRSA with the throughput objective. Next, we found the EE optimal repetition distributions. We studied the effect of the optimal distributions $\phi(x)$ on the inflection load L^* under three cases: first case with the K-collision channel; second case with perfect CSI, MIMO, and MRC; and third case with channel estimation errors, MIMO, pilot contamination, and MMSE combining. We studied the impact of the maximum repetition factor $d_{\max}$, the average repetition factor $\bar{d}$, the number of antennas N, the pilot length τ on the repetition distributions, the inflection load, and also the peak EE $\mathcal{E}_{\max}$. The throughput optimal distributions were close to the 2-regular distribution at high number of antennas. The 2-regular distribution was also the most energy efficient distribution for IRSA at high number of antennas and high pilot lengths. However, it was not optimal in the general case with fewer number of antennas or lower pilot lengths. Compared to the best existing distributions, we showed that our optimized distributions can achieve up to 58% increase in the inflection load and up to 49% increase in the peak EE. The optimal distributions for EE were generally more sparse whereas the optimal distributions for throughput were more dense. The obtained optimal distributions can be used to operate mMTC at the peak throughputs as well as the peak EEs. Future work could include finding the optimal pilot length that maximizes the peak EE.

7.8 Appendix

7.8.1 Other Results

We recall from Chapter 3 that when perfect CSI is available at the BS, and MRC is used for decoding, θ_1 is given by

$$\theta_1 = \Gamma_{\text{inc}}(N, \rho_0^{-1}\gamma_{\text{th}})/\Gamma(N), \tag{7.23}$$

where ρ_0 is the SNR, $\Gamma_{\text{inc}}(s,x) = \int_x^\infty t^{s-1}\, e^{-t}\, \mathrm{d}t$ is the upper incomplete gamma function, and $\Gamma(s)$ is the ordinary gamma function. For $r \geq 2$, with $t_0 \triangleq \gamma_{\text{th}}^{-1} - (N\rho_0)^{-1}$, θ_2 can be calculated as

$$\theta_2 = \mathbb{1}\{t_0 \geq 1\} + (1-(1-t_0)^N)\mathbb{1}\{0 \leq t_0 \leq 1\}. \tag{7.24}$$

For $r \geq 3$, the *Gamma* approximation is:

$$\textit{Gamma:} \quad \theta_r = 1 - \Gamma_{\text{inc}}(r-1, Nt_0)/\Gamma(r-1), \tag{7.25}$$

whereas the *Normal* approximation is

$$\textit{Normal:} \quad \theta_r = 1 - \mathcal{Q}\left(\frac{t_0 - (r-1)\mu_N}{\sqrt{r-1}\sigma_N}\right), \tag{7.26}$$

and the *Deterministic* approximation is

$$\textit{Deterministic:} \quad \theta_r = \mathbb{1}\{r \leq \lfloor N/\gamma_{\text{th}} - \rho_0^{-1} + 1 \rfloor\}. \tag{7.27}$$

With these approximations, we now optimize the performance of IRSA.

In Table 7.18, we present the throughput optimal distributions for the fading channel, and we compare the effect of the approximations on the inflection load L^*. The Gamma and the Normal approximations have similar optimized distributions as well as average repetition factors, but the Normal approximation has higher inflection loads. Further,

Table 7.18: Throughput Optimal Distributions for IRSA with the Fading Channel, MIMO, and Maximal Ratio Combining, with $\rho_0 = 1, \gamma_{\text{th}} = 1$, and $N = 2$.

Approximation	$d_{\max}$	$\phi(x)$	$\bar{d}$	L^*
Gamma	3	$0.05x^2 + 0.95x^3$	2.95	1.69
	4	$0.39x^2 + 0.14x^3 + 0.47x^4$	3.08	1.75
	8	$0.28x^2 + 0.6x^3 + 0.12x^8$	3.33	1.78
Normal	3	$0.03x^2 + 0.97x^3$	2.97	1.78
	4	$0.4x^2 + 0.09x^3 + 0.51x^4$	3.1	1.85
	8	$0.26x^2 + 0.6x^3 + 0.14x^8$	3.37	1.9
Deterministic	3	$0.71x^2 + 0.29x^3$	2.29	1.71
	4	$0.82x^2 + 0.18x^4$	2.36	1.74
	8	$0.86x^2 + 0.02x^3 + 0.12x^8$	2.75	1.84

the Deterministic approximation yields similar inflection loads but very different repetition distributions. This is because the Deterministic approximation is not as good an approximation to θ_r as compared to the other two approximations. Also, we have seen in Chapter 3 that the Normal approximation approximates the inflection load better than the Gamma approximation.

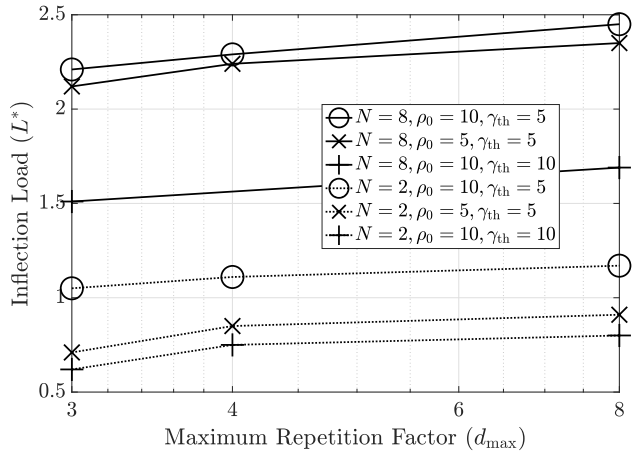


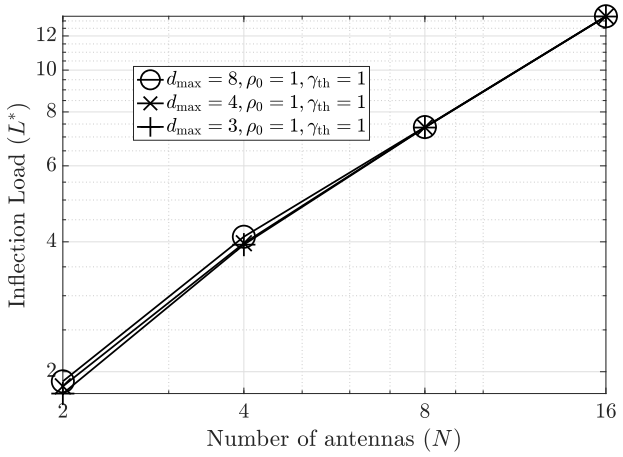
Figure 7.11: Effect of $d_{\max}$ on the inflection load L^*, for the fading channel with MIMO, MRC, and the Normal approximation.

In Fig. 7.11, we study the effect of $d_{\max}$ on the inflection load L^*, for the fading channel with MIMO, MRC, and the Normal approximation. The inflection loads marginally increase with increase in $d_{\max}$ as observed previously. Further, the performance with $N = 8$ is more than $2\times$ that with $N = 2$. The effect of the SNR and the threshold on the inflection load is pretty similar to the previous observations. There is, however, a saturation in the inflection load at high $d_{\max}$, which shows that the throughput does not keep increasing with $d_{\max}$. The marginal increase in the inflection load, at times, need not warrant a significant increase in $d_{\max}$.

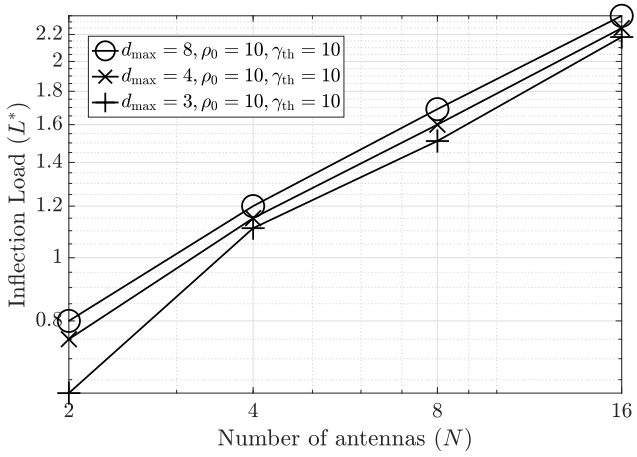
We plot the inflection load versus the number of antennas in Fig. 7.12 for Table 7.4 and Table 7.5. In both the subfigures, the inflection load L^* increases with the number of antennas. This is because the diversity gain increases with the number of antennas for MRC, and thus, the SINRs of the users improve which leads to an increase in the inflection load. Fig. 7.12(a) shows that increasing $d_{\max}$ and trading off a higher $\bar{d}$ for a higher L^* is not highly useful. The inflection load increases only margnally and the reduction in EE due to higher $\bar{d}$ is not helpful. In contrast, in Fig. 7.12(b) we see that increasing $d_{\max}$ and trading off a higher $\bar{d}$ for a higher L^* is indeed useful, albeit only compared to the previous figure. In applications where it is necessary to have higher throughputs and higher inflection loads, increasing $d_{\max}$ is indeed beneficial. The inflection load for every N reduces with an increase in ρ_0 and γ_{th} when the two subfigures are compared.

In Fig. 7.13, we study the effect of the number of antennas on the inflection load for the general case (fading channel with MIMO, MMSE combining, pilot contamination, and channel estimation errors). This is based on Table 7.6 and Table 7.7. As we observed in the perfect CSI case with MIMO, the gap between the inflection loads for varied $d_{\max}$ is very small. In both the subfigures, the inflection load L^* increases with the number of antennas. This is because the interference suppression capability increases with the number of antennas for MMSE, and thus, the SINRs of the users improve which leads to an increase in the inflection load.

In Fig. 7.14, we plot the peak EEs as a function of the number of antennas for both $\rho_0 = 1$, $\gamma_{\text{th}} = 1$ and $\rho_0 = 10$, $\gamma_{\text{th}} = 10$ from Table 7.12 and Table 7.13. The maximum

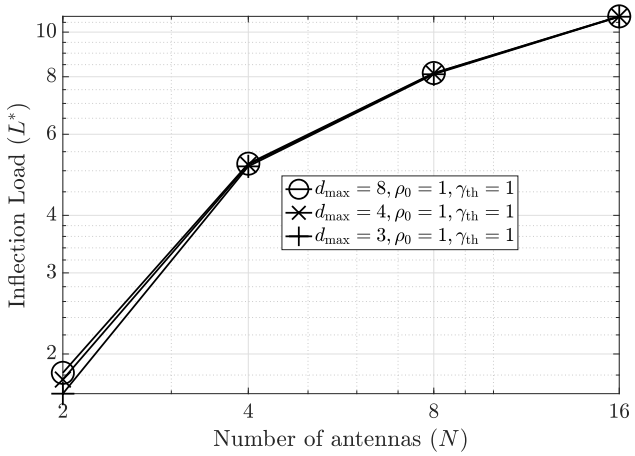


(a) $\rho_0 = 1, \gamma_{\text{th}} = 1$.

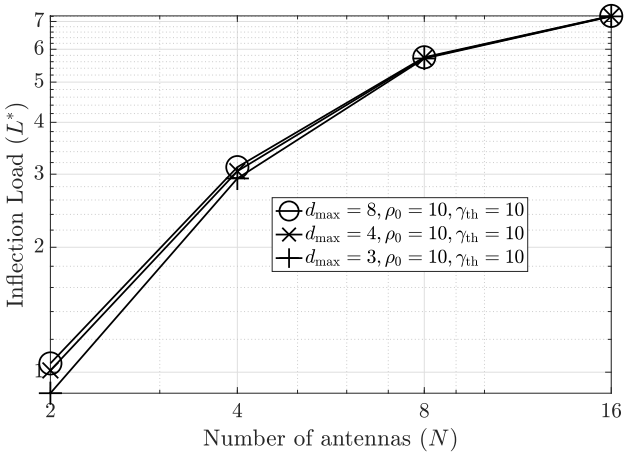


(b) $\rho_0 = 10, \gamma_{\text{th}} = 10$.

Figure 7.12: Effect of the number of antennas on the inflection load L^*, for the fading channel with MIMO, MRC, and the Normal approximation.

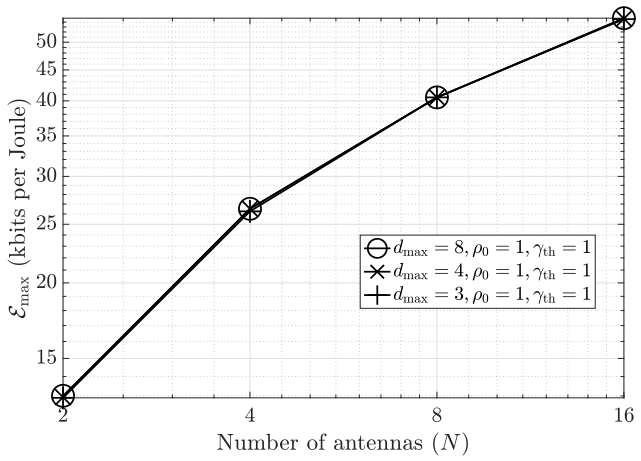


(a) $\rho_0 = 1, \gamma_{\text{th}} = 1$.

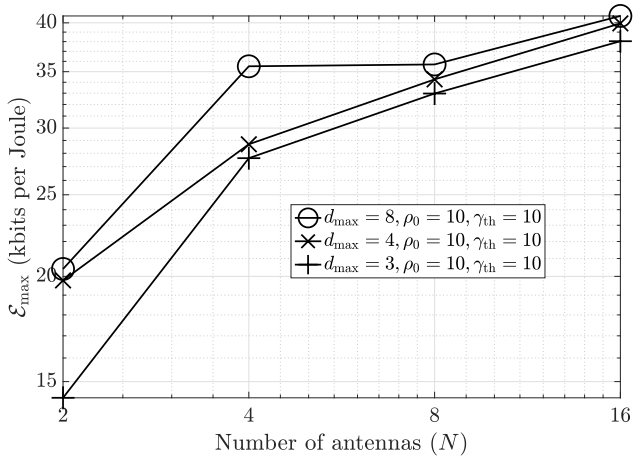


(b) $\rho_0 = 10, \gamma_{\text{th}} = 10$.

Figure 7.13: Effect of the number of antennas on the inflection load L^*, for the fading channel with MIMO, MMSE combining, pilot contamination, channel estimation errors, and $\tau = 10$.



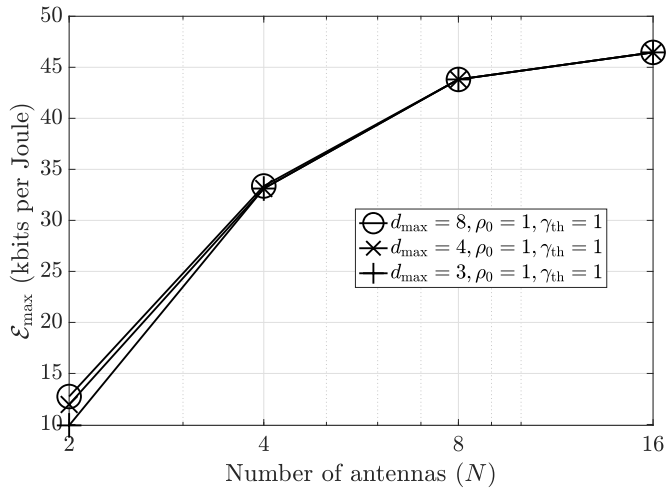
(a) $\rho_0 = 1, \gamma_{\text{th}} = 1$.



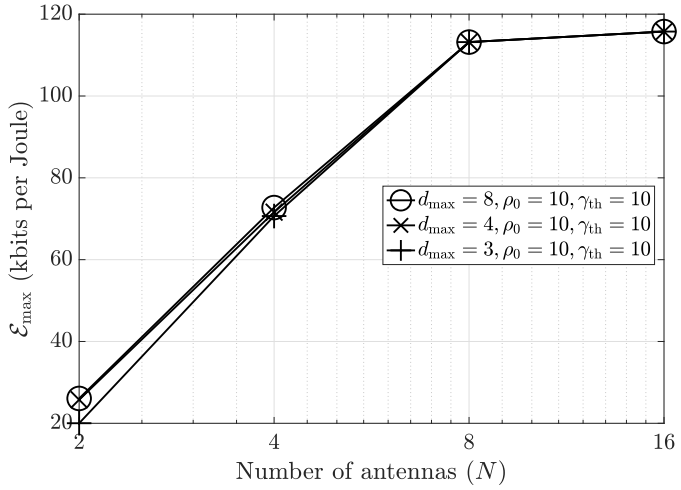
(b) $\rho_0 = 10, \gamma_{\text{th}} = 10$.

Figure 7.14: Effect of the number of antennas on the peak energy efficiency $\mathcal{E}_{\max}$, for the fading channel with MIMO, MRC, and the Normal approximation.

peak EEs increase from 20.41 to 35.52 to 36.21 to 41.58 kilobits per Joule.



(a) $\rho_0 = 1, \gamma_{\text{th}} = 1$.



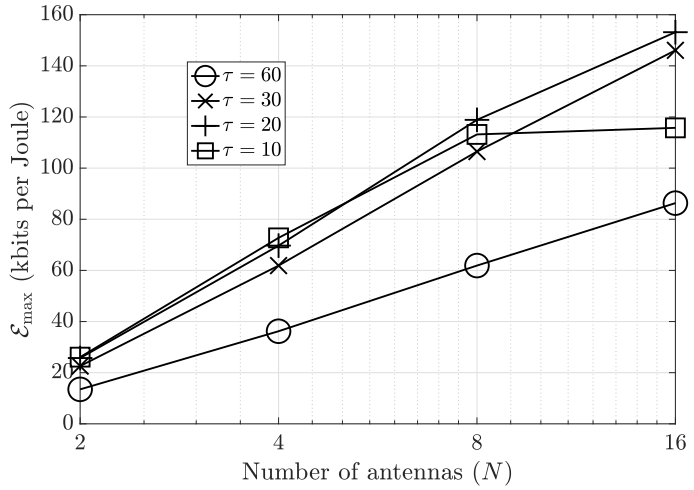
(b) $\rho_0 = 10, \gamma_{\text{th}} = 10$.

Figure 7.15: Effect of the number of antennas on the peak energy efficiency $\mathcal{E}_{\max}$, for the fading channel with MIMO, MMSE combining, pilot contamination, channel estimation errors, and $\tau = 10$.

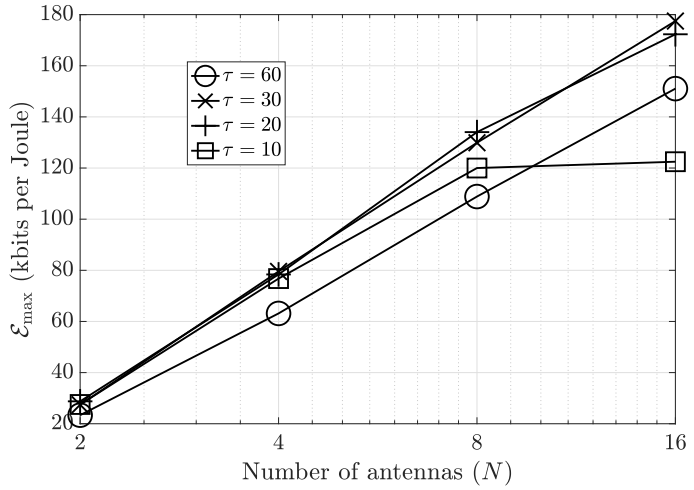
In Fig. 7.15, we plot the peak EEs as a function of the number of antennas. The EEs in both the subfigures marginally improve with an increase in $d_{\max}$. The peak EEs are much higher with MMSE combining than MRC due to improved interference suppression. When going from $\rho_0 = 1, \gamma_{\text{th}} = 1$ to $\rho_0 = 10, \gamma_{\text{th}} = 10$, the peak EEs reduce for MRC

(see Fig. 7.14). Unlike what was observed with MRC, in Fig. 7.15, for MMSE combining, the peak EEs improve when going from $\rho_0 = 1, \gamma_{\text{th}} = 1$ to $\rho_0 = 10, \gamma_{\text{th}} = 10$. For MRC, the effect of improved decodability is eclipsed by the higher amount of power consumed due to higher N. However, for MMSE combining, the amount of power consumed due to more computations and high N is eclipsed by the decodability of MMSE. Thus, MRC and MMSE combining have different trends in the peak EEs.

In Fig. 7.16, we study the effect of the number of antennas N on the peak EE of IRSA. The EE is dependent on the packet length τ_c, via both the SE and the power consumed. For $\tau_c = 100$, $\tau = 60$ achieves the lowest peak EE since a majority of the time is spent in channel estimation. This effect is not indicative in either the throughput or the inflection loads since their definitions do not include τ_c. Here, $\tau_c = 20$ achieves the highest peak EE across all N. For $\tau_c = 200$ and $N \leq 8$, $\tau = 60$ achieves the lowest peak EE since a majority of the time is spent in channel estimation. For $\tau_c = 200$ and $N = 16$, $\tau = 10$ achieves the lowest peak EE since channel estimates are not good enough to decode users. This inverse trend is due to increase in τ_c. EE improves due to both increase in SE due to higher L^* and lower $\bar{d}$, but this happens only in the regime where the pilot length is not too high enough to eclipse the length of the data symbols.



(a) $\tau_c = 100$.



(b) $\tau_c = 200$.

Figure 7.16: Effect of the number of antennas on the peak energy efficiency $\mathcal{E}_{\max}$ with varied packet lengths, for the fading channel with MIMO, MMSE combining, pilot contamination, channel estimation errors, and $\tau = 10$.

Chapter 8

Conclusion

In this thesis, we studied and analyzed the IRSA random access protocol for mMTC applications. We firstly investigated the performance of IRSA accounting for multiple antennas, channel estimation errors, and pilot contamination. We also studied the theoretical performance of IRSA via density evolution. We next developed a UAD algorithm based on sparse Bayesian learning to detect the subset of active users in IRSA. We also analyzed the Cramér-Rao bound on the channel estimation errors in IRSA. After this, we developed an enhanced version of IRSA termed as censored-IRSA, which performs at the peak throughput for overloaded mMTC applications. We then investigated IRSA in the multi-cell and cell-free setups accounting for inter-cell effects and a distributed antenna array setup, respectively. We studied the effects of densification in IRSA systems and the advantages of cell-free IRSA systems. Finally, we found the optimal repetition distributions for IRSA using the differential evolution algorithm for both the throughput and the energy efficiency objectives. We summarize the main contributions of this thesis below.

8.1 Summary of the Thesis

In Chapter 3, we studied the effect of estimated CSI on the throughput of IRSA. We first derived the channel estimates in IRSA with the MSBL algorithm, then we derived the MMSE channel estimates, and finally the low complexity MMSE channel estimates.

We then analyzed the performance of IRSA under all three channel estimation schemes accounting for pilot contamination, channel estimation errors, path loss, and multiple antennas at the BS. The peak achievable throughput significantly reduced: in certain regimes, it resulted in up to 70% loss. We then presented a density evolution based analysis which can be used to compute the asymptotic performance of IRSA, when users perform path loss inversion based power control. This analysis included the derivation of three approximations to the success probability θ_r, and we observed that these approximations match well as the number of antennas at the BS becomes large. Finally, we discussed several new insights into the design of IRSA-based systems: the improvement of the system throughput, the evaluation of the operating load beyond which the system becomes interference limited, and the choice of the decoding threshold γ_{th}. The results underscored the importance of accounting for practical channel estimation in studying the throughput offered by the IRSA protocol.

In Chapter 4, we studied the impact of UAD on the throughput of IRSA. We first proposed a novel Bayesian algorithm to detect the set of active users in IRSA, which exploited the knowledge of the APM, and combined the hyperparameter updates across all RBs to yield an improved UAD performance. We then derived the channel estimates in IRSA accounting for UAD errors. We next derived the Cramér-Rao bound for the channels estimated under the hierarchical Bayesian model used to develop the proposed algorithm. After that, we derived the SINR of all the users accounting for UAD, channel estimation errors, and pilot contamination, and then studied the effect of these errors on the throughput via extensive simulations. We finally discussed many new insights into the design of the IRSA protocol: the complexity of UAD compared to channel estimation, and the improvement of both UAD and throughput with respect to τ, N, SNR, and L.

In Chapter 5, we developed Censored-IRSA (C-IRSA), which overcame the interference limitation of IRSA at high loads. In C-IRSA, users self-censor depending on their CSI based on an adaptive threshold that is periodically broadcast by the BS, and the protocol retains the fully distributed, random access nature of IRSA. Firstly, we derived the MMSE channel estimates and the SINR in C-IRSA accounting for multiple antennas

the BS, fading, path loss, MMSE combining, pilot contamination, and channel estimation errors. We then analyzed the empirical performance of C-IRSA with the SINR threshold model. We next characterized the asymptotic performance of C-IRSA using DE, and also derived closed-form for the success probability θ_r, under the Normal and the Gamma approximations. We also developed two semi-analytic censor functions $i(L, L_{\text{tgt}}, L_0)$ and $h(L, L_{\text{tgt}})$, with which the system throughput was maximized and the active PLR was minimized, respectively. After this, we developed an algorithm to find the active PLR-optimal censor threshold. With this, the PLR of the active users can be driven close to zero, whilst achieving the highest throughput. At high loads, C-IRSA operates at the full capacity compared to IRSA which has near-zero throughputs. Finally, we discussed several insights into the design of C-IRSA: the choice of the target load L_{tgt}, the load L_0, and the optimal censor function.

In Chapter 6, we first studied the effect of MC interference, namely inter-cell PC and inter-cell interference, on the performance of IRSA. Firstly, the channel estimates and the SINR in MC IRSA were derived, accounting for path loss, MIMO fading, intra-cell PC, and intra-cell interference. It was seen that MC IRSA had a significant degradation in performance compared to SC IRSA, even resulting in up to 70% loss of throughput in certain regimes. Recuperating this loss requires at least $4-5\times$ larger pilot length in MC IRSA to yield the same performance as that of SC IRSA. We then analyzed IRSA in the CF setup, accounting for path loss, MIMO fading, and channel estimation errors. Specifically, we studied three CF schemes for IRSA: LCF IRSA, CCF IRSA, and HCF IRSA. We provided insights into the effect of system parameters such as number of antennas, pilot length, and SNR on the throughput of MC and CF IRSA. We showed that we can achieve more than $14\times$ improvement in the throughput of CCF IRSA compared to a massive MIMO SC setup at high loads. We also studied the densification trends in MC IRSA and LCF IRSA, where we observe an inverse behaviour in the throughput compared to CCF IRSA and HCF IRSA. For CCF IRSA and HCF IRSA, densification always improves the performance. For LCF IRSA and MC IRSA, densification does not help at loads near the inflection loads: it is better to not densify and to operate with a

massive MIMO SC setup.

In Chapter 7, we optimized the repetition distributions of IRSA using the differential evolution genetic algorithm. Firstly, we optimized the repetition distributions of IRSA with the throughput and the energy efficiency objectives. We studied the effect of the optimal distributions $\phi(x)$ on the inflection load L^* under three cases: first case with the K-collision channel; second case with perfect CSI, MIMO, and MRC; and third case with channel estimation errors, MIMO, pilot contamination, and MMSE combining. We studied the impact of the maximum repetition factor $d_{\max}$, the average repetition factor $\bar{d}$, the number of antennas N, the pilot length τ on the repetition distributions, the inflection load, and also the peak energy efficiency $\mathcal{E}_{\max}$. The throughput optimal distributions were close to the 2-regular distribution at high number of antennas. The 2-regular distribution was also the most energy efficient distribution for IRSA at high number of antennas and high pilot lengths. However, it is not optimal in the general case with lower number of antennas or lower pilot lengths. Compared to the best existing distributions, we showed that our optimized distributions can achieve up to 58% increase in the inflection load and up to 49% increase in the peak energy efficiency. The optimal distributions for energy efficiency were generally more sparse whereas the optimal distributions for throughput were more dense. The obtained optimal distributions can be used to operate mMTC at the peak throughputs and the peak energy efficiencies.

8.2 Future Work

We list a few interesting research directions which can be pursued as future work.

1. *Asynchronism:* The asynchronism in random access transmissions can be exploited to identify which users are active and estimate their channels. In this case, users' pilot sequences could be thought of as slightly delayed versions of a common synchronization signal. A database of the time shifted pilots available at BS itself can be used to detect the users and estimate their channels. Exploiting the asynchronous nature of random access transmissions to detect active users and estimate

their channels instead of orthogonal/non-orthogonal pilots (as seen in Chapter 4) is an interesting direction for future work.

2. *Fairness in C-IRSA:* In Chapter 5, we observed that C-IRSA is fair for users since censoring is performed based only on the small scale fading states of the users. If a user is shadowed for a long duration of time, then the user can transmit in a different frequency band. However, if there is a scarcity of bands, then that user will never be able to send its data to the BS. Accounting for a proportional fairness mechanism for users with poor CSI can help with this. Another way to incorporate this is to use multiple thresholds instead of a single threshold. Further, accounting for load estimation errors, and also a scheme in which the repetition distribution itself is CSI dependent can improve the performance of C-IRSA.

3. *Pilot design for MC IRSA:* In Chapter 3, we have seen that employing QPSK pilots with pilot reuse improved the throughput of IRSA (in Chapter 3, the UAD performance improved as well). Existing works show that the pilot sequences can be designed in order to reduce PC [119]. This is especially important in the multi-cell setup where inter-cell pilot contamination bogs down the performance of IRSA (see Chapter 6). Designing the pilot sequence sets intelligently can help us overcome the poor performance of MC IRSA at low pilot lengths.

4. *Clustering for CF IRSA:* In CF IRSA, only the APs that are associated to any user can try to decode the user instead of all the APs attempting to decode every user (see Chapter 6). This reduces the complexity of decoding at each AP, and thus, user-AP clustering is an interesting direction for future work.

5. *Pilot length:* The spectral and energy efficiencies of IRSA are maximized by a moderate pilot length. Low pilot lengths would yield poor throughputs, whereas high pilot lengths would leave only a small fraction of the packet for data symbol transmission and would also consume higher power. This was observed in Chapter 3 and Chapter 7. Finding the optimal pilot length can help us operate the system at the maximum energy efficiency or throughput, depending on the metric of interest.

6. *Quantization and synchronization errors:* We assumed full-resolution quantization, and also perfect RB- and frame- level synchronization across users and the BS in this thesis (see Chapter 3). Future work can consider relaxing this assumption, and analyze IRSA with synchronization and quantization errors.

7. *Age of information:* The age-of-information is a metric that can be used to evaluate the latency of decoding the users' packets. This is especially important in massive random access since it can help us terminate users' transmissions especially if they are old packets containing "stale" information. Few recent works in IRSA [51, 79] have considered this metric with simplistic system models. Analyzing this metric in the general case can be an interesting direction for research.

8. *Machine learning:* Machine learning techniques can be used to improve the performance of IRSA for mMTC. When the scale of the problem increases to a million users in mMTC, deep unfolding techniques from machine learning can be leveraged with our algorithm in Chapter 4 to perform UAD. Further, learning techniques can be incorporated to improve the multi-user decoding for mMTC.

Bibliography

[1] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, "Machine-type communications: current status and future perspectives toward 5G systems," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 10–17, 2015.

[2] T. Xia, M. M. Wang, C. Jiang, J. Zhang, L. Wang, and X. You, "Practical machine-type communication for energy internet of things: An introduction," *IEEE Communications Standards Magazine*, vol. 3, no. 1, pp. 48–59, 2019.

[3] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, and R. Schober, "Massive access for 5G and beyond," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 615–637, 2021.

[4] N. H. Mahmood, H. Alves, O. A. López, M. Shehab, D. P. M. Osorio, and M. Latva-Aho, "Six key features of machine type communication in 6G," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.

[5] L. Liu, E. G. Larsson, W. Yu, P. Popovski, C.Stefanović, and E. de Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the internet of things," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 88–99, 2018.

[6] L. Liu and W. Yu, "Massive connectivity with massive MIMO—part I: Device activity detection and channel estimation," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2933–2946, 2018.

[7] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 477–487, February 2011.

[8] E. E. Khaleghi, C. Adjih, A. Alloum, and P. Muhlethaler, "Near-far effect on coded slotted ALOHA," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–7.

[9] Y. Wu, X. Gao, S. Zhou, W. Yang, Y. Polyanskiy, and G. Caire, "Massive access for future wireless communication systems," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 148–156, 2020.

[10] N. Krishnan, R. D. Yates, and N. B. Mandayam, "Uplink linear receivers for multi-cell multiuser MIMO with pilot contamination: Large system analysis," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4360–4373, 2014.

[11] C. R. Srivatsa and C. R. Murthy, "On the impact of channel estimation on the design and analysis of IRSA based systems," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4186–4200, 2022.

[12] C. R. Srivatsa and C. R. Murthy, "Throughput analysis of PDMA/IRSA under practical channel estimation," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2019, pp. 1–5.

[13] A. Fengler, S. Haghighatshoar, P. Jung, and G. Caire, "Non-Bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive MIMO receiver," *IEEE Transactions on Information Theory*, pp. 1–1, 2021.

[14] S. Haghighatshoar, P. Jung, and G. Caire, "Improved scaling law for activity detection in massive MIMO systems," in *2018 IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 381–385.

[15] C. R. Srivatsa and C. R. Murthy, "User activity detection for irregular repetition slotted aloha based MMTC," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3616–3631, 2022.

[16] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3704–3716, 2007.

[17] F. Clazzer, E. Paolini, I. Mambelli, and C. Stefanović, "Irregular repetition slotted ALOHA over the rayleigh block fading channel with capture," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[18] C. R. Srivatsa and C. R. Murthy, "Channel state information based user censoring in irregular repetition slotted aloha," in *2023 IEEE International Conference on Communications (ICC)*, 2023, pp. 1–6.

[19] ——, "Censored-IRSA for massive random access in interference-limited mMTC scenarios," *Manuscript under preparation*, 2023.

[20] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "User-centric cell-free massive MIMO networks: A survey of opportunities, challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 611–652, 2022.

[21] Ö. T. Demir, E. Björnson, L. Sanguinetti *et al.*, *Foundations of user-centric cell-free massive MIMO*. Now Publishers, Inc., 2021, vol. 14, no. 3-4.

[22] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, 2017.

[23] Z. Chen, E. Björnson, and E. G. Larsson, "Dynamic resource allocation in co-located and cell-free massive MIMO," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 1, pp. 209–220, 2020.

[24] C. R. Srivatsa and C. R. Murthy, "Performance analysis of irregular repetition slotted aloha with multi-cell interference," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022, pp. 1–5.

[25] ——, "IRSA for massive random access in multi-cell and cell-free massive MIMO systems," *Manuscript under preparation*, 2023.

[26] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, 1997.

[27] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[28] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1436–1449, 2013.

[29] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[30] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive MIMO systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 7112–7139, 2014.

[31] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[32] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[33] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[34] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 124–130, 2018.

[35] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[36] 3GPP, "Study on scenarios and requirements for next generation access technologies," Technical Specification Group Radio Access Network, Technical Report 38.913, 2016.

[37] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 426–471, 2020.

[38] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[39] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[40] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, "5G evolution: A view on 5G cellular technology beyond 3GPP release 15," *IEEE Access*, vol. 7, pp. 127 639–127 651, 2019.

[41] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network

slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[42] E. de Carvalho, B. Emil, J. H. Sørensen, E. G. Larsson, and P. Popovski, "Random pilot and data access in massive MIMO for machine-type communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 7703–7717, 2017.

[43] S. M. R. Islam, N. Avazov, O. A. Dobre, and K. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Communications Surveys And Tutorials*, vol. 19, no. 2, pp. 721–742, Secondquarter 2017.

[44] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-free non-orthogonal multiple access for IoT: A survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1805–1838, 2020.

[45] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, "A survey of non-orthogonal multiple access for 5G," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2294–2323, 2018.

[46] Y. Tan, J. Zhou, and J. Qin, "Novel channel estimation for non-orthogonal multiple access systems," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1781–1785, Dec 2016.

[47] M. Mohammadkarimi, M. A. Raza, and O. A. Dobre, "Signature-based nonorthogonal massive multiple access for future wireless networks: Uplink massive connectivity for machine-type communications," *IEEE Vehicular Technology Magazine*, vol. 13, no. 4, pp. 40–50, 2018.

[48] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of massive MIMO*. Cambridge University Press, 2016.

[49] E. Björnson, J. Hoydis, and L. Sanguinetti, *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency*. Now Publishers, Inc., 2017, vol. 11, no. 3-4.

[50] I. Hmedoush, "Connectionless transmission in wireless networks (IoT)," Ph.D. dissertation, Inria Paris, 2022.

[51] A. Munari, "Modern random access: An age of information perspective on irregular repetition slotted ALOHA," *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3572–3585, 2021.

[52] S. Jiang, X. Yuan, X. Wang, C. Xu, and W. Yu, "Joint user identification, channel estimation, and signal detection for grant-free NOMA," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6960–6976, 2020.

[53] K. Senel and E. G. Larsson, "Grant-free massive MTC-enabled massive MIMO: A compressive sensing approach," *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6164–6175, 2018.

[54] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Transactions on Information Theory*, vol. 61, no. 12, pp. 6815–6832, Dec 2015.

[55] H. Nikopour and H. Baligh, "Sparse code multiple access," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013, pp. 332–336.

[56] Y. Polyanskiy, "A perspective on massive random-access," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2523–2527.

[57] M. J. Ahmadi and T. M. Duman, "Unsourced random access with a massive MIMO receiver using multiple stages of orthogonal pilots," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 2880–2885.

[58] A. Vem, K. R. Narayanan, J. Cheng, and J.-F. Chamberland, "A user-independent serial interference cancellation based coding scheme for the unsourced random access gaussian channel," in *2017 IEEE Information Theory Workshop (ITW)*, 2017, pp. 121–125.

[59] V. K. Amalladinne, A. Vem, D. K. Soma, K. R. Narayanan, and J.-F. Chamberland, "A coupled compressive sensing scheme for unsourced multiple access," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6628–6632.

[60] S. Chen, B. Ren, Q. Gao, S. Kang, S. Sun, and K. Niu, "Pattern division multiple access-a novel nonorthogonal multiple access for fifth-generation radio networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3185–3196, April 2017.

[61] Y. Jiang, P. Li, Z. Ding, F. Zheng, M. Ma, and X. You, "Joint transmitter and receiver design for pattern division multiple access," *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 885–895, 2019.

[62] Z. Yuan, G. Yu, W. Li, Y. Yuan, X. Wang, and J. Xu, "Multi-user shared access for internet of things," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–5.

[63] L. Valentini, A. Faedi, M. Chiani, and E. Paolini, "Coded random access for 6G: Intra-frame spatial coupling with ACKs," in *2021 IEEE Globecom Workshops (GC Wkshps)*, 2021, pp. 1–6.

[64] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 340–354, 2013.

[65] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 912–926, 2011.

[66] L. Qiao, J. Zhang, Z. Gao, S. Chen, and L. Hanzo, "Compressive sensing based massive access for IoT relying on media modulation aided machine type communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 391–10 396, 2020.

[67] A. Tipping and A. Faul, "Analysis of sparse Bayesian learning," *Advances in neural information processing systems*, vol. 14, pp. 383–389, 2002.

[68] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[69] J. Palmer, B. Rao, and D. Wipf, "Perspectives on sparse Bayesian learning," *Advances in neural information processing systems*, vol. 16, 2003.

[70] Q. He, T. Q. S. Quek, Z. Chen, Q. Zhang, and S. Li, "Compressive channel estimation and multi-user detection in C-RAN with low-complexity methods," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3931–3944, 2018.

[71] S. Verdu *et al.*, *Multiuser detection*. Cambridge university press, 1998.

[72] K. R. Narayanan and H. D. Pfister, "Iterative collision resolution for slotted ALOHA: An optimal uncoordinated transmission policy," in *2012 7th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Aug 2012, pp. 136–139.

[73] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "The multicell multiuser MIMO uplink with very large antenna arrays and a finite-dimensional channel," *IEEE Transactions on Communications*, vol. 61, no. 6, pp. 2350–2361, 2013.

[74] A. Mengali, R. De Gaudenzi, and C. Stefanović, "On the modeling and performance assessment of random access with SIC," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 292–303, 2018.

[75] M. Ghanbarinejad and C. Schlegel, "Irregular repetition slotted ALOHA with multiuser detection," in *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2013, pp. 201–205.

[76] J. Su, G. Ren, B. Zhao, and J. Ding, "Enhancing irregular repetition slotted ALOHA with polarization diversity in LEO satellite networks," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 9, pp. 3907–3923, 2020.

[77] F. Clazzer, C. Kissling, and M. Marchese, "Enhancing contention resolution ALOHA using combining techniques," *IEEE Transactions on Communications*, vol. 66, no. 6, pp. 2576–2587, 2017.

[78] F. Clazzer, F. Lázaro, G. Liva, and M. Marchese, "Detection and combining techniques for asynchronous random access with time diversity," in *SCC 2017; 11th International ITG Conference on Systems, Communications and Coding*. VDE, 2017, pp. 1–6.

[79] S. Saha, V. B. Sukumaran, and C. R. Murthy, "On the minimum average age of information in IRSA for grant-free mMTC," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1441–1455, 2021.

[80] U. Demirhan and T. M. Duman, "Irregular repetition slotted ALOHA with energy harvesting nodes," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4505–4517, 2019.

[81] T. Akyıldız, U. Demirhan, and T. M. Duman, "Energy harvesting irregular repetition ALOHA with replica concatenation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 955–968, 2021.

[82] E. Paolini, L. Valentini, V. Tralli, and M. Chiani, "Irregular repetition slotted ALOHA in an information-theoretic setting," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 3019–3024.

[83] Z. Zhang, K. Niu, J. Dai, and C. Dong, "Polar-slotted ALOHA over slot erasure channel," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 760–771, 2023.

[84] J. Choi and J. Ding, "Network coding for K-repetition in grant-free random access," *IEEE Wireless Communications Letters*, vol. 10, no. 11, pp. 2557–2561, 2021.

[85] L. Zhao, S. Ma, W. Chen, and F. Hu, "Optimal probabilistic repetition for massive

MIMO-aided grant-free short-packet transmissions," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 12 407–12 412, 2022.

[86] S.-L. Shieh and S.-H. Yang, "Enhanced irregular repetition slotted ALOHA under SIC limitation," *IEEE Transactions on Communications*, pp. 1–1, 2022.

[87] J. Haghighat and T. M. Duman, "An energy-efficient feedback-aided irregular repetition slotted ALOHA scheme and its asymptotic performance analysis," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[88] C. Dumas, L. Salaün, I. Hmedoush, C. Adjih, and C. S. Chen, "Design of coded slotted ALOHA with interference cancellation errors," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 12 742–12 757, 2021.

[89] L. Valentini, M. Chiani, and E. Paolini, "Massive grant-free access with massive MIMO and spatially coupled replicas," *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7337–7350, 2022.

[90] Z. Qu, H. Cao, and G. Zhang, "A study of irregular repetition slotted ALOHA over LEO satellite channel with capture effect," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, pp. 1–6.

[91] J. Su, G. Ren, and B. Zhao, "NOMA-based coded slotted ALOHA for machine-type communications," *IEEE Communications Letters*, vol. 25, no. 7, pp. 2435–2439, 2021.

[92] L. Valentini, M. Chiani, and E. Paolini, "A joint PHY and MAC layer design for coded random access with massive MIMO," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 2505–2510.

[93] J. Sun, R. Liu, and E. Paolini, "Detecting the number of active users in IRSA access protocols," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2018, pp. 1972–1976.

[94] M. Guo and M. C. Gursoy, "Joint activity detection and channel estimation in cell-free massive MIMO networks with massive connectivity," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 317–331, 2022.

[95] B. K. Jeong, B. Shim, and K. B. Lee, "MAP-based active user and data detection for massive machine-type communications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8481–8494, 2018.

[96] A. Rajoriya, S. Rukhsana, and R. Budhiraja, "Centralized and decentralized active user detection and channel estimation in mMTC," *IEEE Transactions on Communications*, vol. 70, no. 3, pp. 1759–1776, 2022.

[97] X. Zhang, F. Labeau, L. Hao, and J. Liu, "Joint active user detection and channel estimation via Bayesian learning approaches in MTC communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6222–6226, 2021.

[98] Q. Zou, H. Zhang, D. Cai, and H. Yang, "Message passing based joint channel and user activity estimation for uplink grant-free massive MIMO systems with low-precision ADCs," *IEEE Signal Processing Letters*, vol. 27, pp. 506–510, 2020.

[99] W. Zhang, J. Li, X. Zhang, and S. Zhou, "A joint user activity detection and channel estimation scheme for packet-asynchronous grant-free access," *IEEE Wireless Communications Letters*, vol. 11, no. 2, pp. 338–342, 2022.

[100] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell TDD systems," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2640–2651, 2011.

[101] T. Van Chien, E. Björnson, and E. G. Larsson, "Joint pilot design and uplink power allocation in multi-cell massive MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2000–2015, 2018.

[102] Z. Chen, Y. Feng, C. Feng, L. Liang, Y. Jia, and T. Q. S. Quek, "Analytic distribution design for irregular repetition slotted ALOHA with multi-packet reception," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1360–1365, 2023.

[103] Wang Lin, Xiao Juan, and Guanrong Chen, "Density evolution method and threshold decision for irregular LDPC codes," in *2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No.04EX914)*, vol. 1, 2004, pp. 25–28 Vol.1.

[104] B. Hassibi and B. Hochwald, "How much training is needed in multiple-antenna wireless links?" *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 951–963, 2003.

[105] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[106] R. Couillet and M. Debbah, *Random matrix methods for wireless communications*. Cambridge University Press, 2011.

[107] B. M. Lee and Y. Kim, "Practical determination of impulse sample power boosting factor in impulse postfix OFDM systems," *IEEE Communications Letters*, vol. 13, no. 3, pp. 187–189, 2009.

[108] B. M. Lee, Y. Kim, and H. R. You, "A power boosting factor determination for the impulse sample in impulse postfix OFDM systems," in *2008 IEEE International Symposium on Wireless Communication Systems*, 2008, pp. 376–380.

[109] Y.-C. Huang, S.-L. Shieh, Y.-P. Hsu, and H.-P. Cheng, "Iterative collision resolution for slotted ALOHA with NOMA for heterogeneous devices," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 2948–2961, 2021.

[110] E. Paolini, G. Liva, and M. Chiani, "High throughput random access via codes on graphs: Coded slotted ALOHA," in *2011 IEEE International Conference on Communications (ICC)*, 2011, pp. 1–6.

[111] A. Rotenberg, "A new pseudo-random number generator," *J. ACM*, vol. 7, no. 1, p. 75–77, 1960.

[112] G. Marsaglia, "The structure of linear congruential sequences," in *Applications of number theory to numerical analysis*. Elsevier, 1972, pp. 249–285.

[113] I. Borosh and H. Niederreiter, "Optimal multipliers for pseudo-random number generation by the linear congruential method," *BIT Numerical Mathematics*, vol. 23, no. 1, pp. 65–74, 1983.

[114] M. J. Durst, "Using linear congruential generators for parallel random number generation," in *Proceedings of the 21st conference on Winter simulation*, 1989, pp. 462–466.

[115] Z. Zhang, J. Liu, and L. Keping, "Low-complexity cell search with fast PSS identification in LTE," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1719–1729, 2012.

[116] A. Chowdhury, P. Sasmal, C. R. Murthy, and R. Chopra, "On the performance of distributed antenna array systems with quasi-orthogonal pilots," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3326–3331, 2022.

[117] R.-A. Pitaval, B. M. Popović, P. Wang, and F. Berggren, "Overcoming 5G PRACH capacity shortfall: Supersets of Zadoff–Chu sequences with low-correlation zone," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5673–5688, 2020.

[118] J. H. Sorensen, E. de Carvalho, C. Stefanović, and P. Popovski, "Coded pilot random access for massive MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8035–8046, 2018.

[119] A. Chowdhury, P. Sasmal, and C. R. Murthy, "Comparison of orthogonal vs. union of subspace based pilots for multi-cell massive MIMO systems," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2020, pp. 1–5.

[120] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[121] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[122] G. Tang and A. Nehorai, "Performance analysis for sparse support recovery," *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1383–1399, 2010.

[123] R. Prasad and C. R. Murthy, "Cramér-Rao-type bounds for sparse Bayesian learning," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 622–632, 2013.

[124] L. Valentini, A. Faedi, M. Chiani, and E. Paolini, "Impact of interference subtraction on grant-free multiple access with massive MIMO," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 1318–1323.

[125] M. Mohammadkarimi, O. A. Dobre, and M. Z. Win, "Massive uncoordinated multiple access for beyond 5G," *IEEE Transactions on Wireless Communications*, vol. 21, no. 5, pp. 2969–2986, 2022.

[126] A. Algans, K. Pedersen, and P. Mogensen, "Experimental analysis of the joint statistical properties of azimuth spread, delay spread, and shadow fading," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 3, pp. 523–531, 2002.

[127] T. Jiang, N. Sidiropoulos, and G. Giannakis, "Kalman filtering for power estimation in mobile communications," *IEEE Transactions on Wireless Communications*, vol. 2, no. 1, pp. 151–161, 2003.

[128] A. Dogandzic and B. Zhang, "Dynamic shadow-power estimation for wireless communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2942–2948, 2005.

[129] J. Reig and L. Rubio, "Estimation of the composite fast fading and shadowing distribution using the log-moments in wireless communications," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 3672–3681, 2013.

[130] Y. Hu and G. Leus, "Self-estimation of path-loss exponent in wireless networks and applications," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5091–5102, 2015.

[131] N. Salman, A. H. Kemp, and M. Ghogho, "Low complexity joint estimation of location and path-loss exponent," *IEEE Wireless Communications Letters*, vol. 1, no. 4, pp. 364–367, 2012.

[132] A. Chowdhury, R. Chopra, and C. R. Murthy, "Can dynamic TDD enabled half-duplex cell-free massive MIMO outperform full-duplex cellular massive MIMO?" *IEEE Transactions on Communications*, vol. 70, no. 7, pp. 4867–4883, 2022.

[133] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77–90, 2020.

[134] ——, "Scalable cell-free massive MIMO systems," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4247–4261, 2020.

[135] K. Hosseini, J. Hoydis, S. ten Brink, and M. Debbah, "Massive MIMO and small cells: How to densify heterogeneous networks," in *2013 IEEE International Conference on Communications (ICC)*, 2013, pp. 5442–5447.

[136] U. K. Ganesan, E. Björnson, and E. G. Larsson, "Clustering-based activity detection algorithms for grant-free random access in cell-free massive MIMO," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7520–7530, 2021.

[137] Y. Li, Q. Lin, Y.-F. Liu, B. Ai, and Y.-C. Wu, "Asynchronous activity detection for cell-free massive MIMO: From centralized to distributed algorithms," *IEEE Transactions on Wireless Communications*, vol. 22, no. 4, pp. 2477–2492, 2023.

[138] H. Wang, J. Wang, and J. Fang, "Grant-free massive connectivity in massive MIMO systems: Collocated versus cell-free," *IEEE Wireless Communications Letters*, vol. 10, no. 3, pp. 634–638, 2021.

[139] D. Jiang and Y. Cui, "ML and MAP device activity detections for grant-free massive access in multi-cell networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3893–3908, 2022.

[140] X. Wang, A. Ashikhmin, Z. Dong, and C. Zhai, "Two-stage channel estimation approach for cell-free IoT with massive random access," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1428–1440, 2022.

[141] V. Croisfelt, T. Abrão, and J. C. Marinello, "User-centric perspective in random access cell-free aided by spatial separability," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16 562–16 576, 2022.

[142] F. Mirhosseini, A. Tadaion, and S. M. Razavizadeh, "Spectral efficiency of dense multicell massive MIMO networks in spatially correlated channels," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1307–1316, 2021.

[143] Z. Chen, F. Sohrabi, and W. Yu, "Multi-cell sparse activity detection for massive random access: Massive MIMO versus cooperative MIMO," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4060–4074, 2019.

[144] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[145] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, 2014.

[146] E. Björnson, L. Sanguinetti, J. Hoydis, and M. Debbah, "Optimal design of energy-efficient multi-user MIMO systems: Is massive MIMO the answer?" *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3059–3075, 2015.

[147] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2010.

[148] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[149] D. Zaharie, "Critical values for the control parameters of differential evolution algorithms," in *Proc. of MENDEL 2002, 8th Int. Conf. on Soft Computing*, 2002, pp. 62–67.

[150] A. Pizzo, D. Verenzuela, L. Sanguinetti, and E. Björnson, "Network deployment for maximal energy efficiency in uplink with multislope path loss," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 3, pp. 735–750, 2018.

[151] B. Wang, L. Dai, Y. Zhang, T. Mir, and J. Li, "Dynamic compressive sensing-based multi-user detection for uplink grant-free NOMA," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2320–2323, 2016.

[152] S. Haghighatshoar and G. Caire, "Multiple measurement vectors problem: A decoupling property and its applications," *arXiv preprint arXiv:1810.13421*, 2018.

[153] S. S. Thoota and C. R. Murthy, "Massive MIMO-OFDM systems with low resolution ADCs: Cramér-Rao bound, sparse channel estimation, and soft symbol decoding," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4835–4850, 2022.

[154] B. Li, J. Zheng, and Y. Gao, "Compressed sensing based multiuser detection of grant-free NOMA with dynamic user activity," *IEEE Communications Letters*, vol. 26, no. 1, pp. 143–147, 2022.

[155] D. Gore and A. Paulraj, "MIMO antenna subset selection with space-time coding," *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2580–2588, 2002.

[156] S. Kandukuri and S. Boyd, "Optimal power control in interference-limited fading wireless channels with outage-probability specifications," *IEEE Transactions on Wireless Communications*, vol. 1, no. 1, pp. 46–55, 2002.

[157] G. Durisi, T. Koch, J. Östman, Y. Polyanskiy, and W. Yang, "Short-packet communications over multiple-antenna rayleigh-fading channels," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 618–629, 2016.

[158] T. Nonaka, T. Fujii, O. Takyu, and M. Ohta, "Adapting the number of replicas in the E-IRSA system using the power control," in *2020 International Conference on Information Networking (ICOIN)*, 2020, pp. 787–792.

[159] M. Peng, J. Zeng, X. Su, and B. Liu, "Power allocation in PDMA systems with imperfect channel state information," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–6.

[160] L. Ramesh, C. R. Murthy, and H. Tyagi, "Sample-measurement tradeoff in support recovery under a subgaussian prior," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 2709–2713.

[161] Y. Feng, Z. Chen, Z. Tian, M. Wang, and Y. Jia, "Performance analysis for irregular repetition slotted ALOHA with short packet transmissions," in *2022 IEEE/CIC International Conference on Communications in China (ICCC)*, 2022, pp. 914–919.

[162] J. Ding and J. Choi, "SIC aided K-repetition for mission-critical MTC in cell-free massive MIMO," in *2021 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2021, pp. 167–173.

[163] S. M. Kay and S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-hall Englewood Cliffs, NJ, 1993, vol. 1.

[164] 3GPP, "NR; Physical channels and modulation," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.211, 09 2023,

version 17.6.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3213