This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Near-Optimal Large-MIMO Detection Using Randomized MCMC and Randomized Search Algorithms

Ashok Kumar, Suresh Chandrasekaran, A. Chockalingam, and B. Sundar Rajan
Department of ECE, Indian Institute of Science, Bangalore 560012, INDIA

Abstract— **Low-complexity near-optimal detection of signals in MIMO systems with large number (tens) of antennas is getting increased attention. In this paper, first, we propose a variant of Markov chain Monte Carlo (MCMC) algorithm which i) alleviates the stalling problem encountered in conventional MCMC algorithm at high SNRs, and ii) achieves near-optimal performance for large number of antennas (e.g., $16\times16, 32\times32, 64\times64$ MIMO) with 4-QAM. We call this proposed algorithm as *randomized MCMC (R-MCMC)* algorithm. Second, we propose another algorithm based on a random selection approach to choose candidate vectors to be tested in a local neighborhood search. This algorithm, which we call as *randomized search (RS)* algorithm, also achieves near-optimal performance for large number of antennas with 4-QAM. The complexities of the proposed R-MCMC and RS algorithms are quadratic/sub-quadratic in number of transmit antennas, which are attractive for detection in large-MIMO systems. We also propose *message passing* aided R-MCMC and RS algorithms, which are shown to perform well for higher-order QAM.**

Keywords – *Markov chain Monte Carlo, random search, message passing, large-MIMO systems, near-optimal low-complexity detection.*

I. Introduction

Multiple-input multiple-output (MIMO) systems with tens of antennas in communication terminals, referred to as large-MIMO systems, are getting increased attention due to their high spectral efficiencies [1],[2]. Achieving near-optimal performance at low-complexities in such large systems has been a topic of recent research [3]-[12]. Near-maximum likelihood (ML) performance achieving detection algorithms using local neighborhood search techniques for MIMO systems with large number ($>$ 10) of antennas have been reported recently in the literature; e.g., likelihood ascent search (LAS) [3],[4],[5],[6] and reactive tabu search (RTS) [7],[8] algorithms. Near-maximum aposteriori probability (MAP) performance achieving large-MIMO detection algorithms based on probabilistic data association (PDA) [10] and belief propagation (BP) [11],[12] have also been reported. In addition, detection performance in 50×50 and 20×20 MIMO systems using Gibbs sampling and message passing techniques have been reported in [16] and [17], respectively.

An interesting class of low-complexity algorithms reported in the context of CDMA and MIMO detection is based on Markov chain Monte Carlo (MCMC) simulation techniques [13],[14],[15]. In MCMC methods, statistical inferences are developed by simulating the underlying processes through Markov chains. By doing so, it becomes possible to reduce exponential detection complexity to linear/polynomial complexities. An issue with MCMC based detection, however, is the *stalling problem*, due to which performance degrades at high SNRs [14]. Stalling problem arises because transitions from some states to other states in a Markov chain can occur with very low probability [14].

Recently, in [16], a MCMC algorithm which used Gibbs sampling with a temperature parameter α is presented. The value of α is fixed in all the iterations with the property that after the Markov chain is mixed, the probability of encountering the optimal solution is only polynomially small (not exponentially small). Though this algorithm scales well for large n_t, it's performance is highly dependent on the fixed temperature value α, the optimum value of which varies with SNR and n_t. This makes the algorithm less attractive. In addition, the algorithm faces the stalling problem at high SNRs. In this context, our first new contribution in this paper is that we propose a variant of MCMC algorithm which does not use temperature parameter (so optimization over α is not needed). The key idea is not to use the conventional Gibbs sampling always, but, in addition, probabilistically use a randomized sampling as well (details given in Section III). We call this proposed algorithm as *randomized MCMC (R-MCMC)* algorithm. Simulation results show that the proposed R-MCMC algorithm i) alleviates the stalling problem, and ii) achieves near-ML performance for large number of antennas (e.g., $16 \times 16, 32 \times 32, 64 \times 64$ MIMO) with 4-QAM.

Our second contribution is that we propose another algorithm based on *randomized search*. The key idea in this algorithm is that it adopts a random selection approach to choose the set of candidate vectors to be tested in the local neighborhood search. This algorithm, which we refer to as the *randomized search (RS)* algorithm (details given in Section IV), also achieves near-optimal performance in large-MIMO systems for 4-QAM. The average per-bit complexities of R-MCMC and RS algorithms are $O(n_t^2)$ and $O(n_t^{1.4})$, respectively.

Our final contribution is to further improve the performance of the R-MCMC and RS algorithms in *higher-order QAM* using improved priors/initial vector from message passing techniques (details given in Section V). The proposed message passing aided R-MCMC and RS are shown to achieve close to ML performance in large-MIMO systems with higher-order QAM (e.g., 12×12 MIMO with 16-QAM).

II. System Model

Consider a V-BLAST MIMO system with n_t transmit and n_r receive antennas. The transmitted symbols take values from a modulation alphabet $\mathbb{B}$. Let $\mathbf{x}_c \in \mathbb{B}^{n_t}$ denote the transmitted vector. Let $\mathbf{H}_c \in \mathbb{C}^{n_r \times n_t}$ denote the channel gain matrix, whose entries are assumed to be i.i.d. Gaussian with zero mean and unit variance. The received vector $\mathbf{y}_c$ is

$$\mathbf{y}_c = \mathbf{H}_c\mathbf{x}_c + \mathbf{n}_c, \tag{1}$$

where $\mathbf{n}_c$ is the noise vector whose entries are modeled as i.i.d. $\mathbb{C}\mathcal{N}(0, \sigma^2)$. In the rest of the paper, we work with the real-valued system model corresponding to (1),

$$\mathbf{y}_r \quad = \quad \mathbf{H}_r \mathbf{x}_r + \mathbf{n}_r, \tag{2}$$

where

$$\mathbf{H}_r = \begin{bmatrix} \Re(\mathbf{H}_c) & -\Im(\mathbf{H}_c) \\ \Im(\mathbf{H}_c) & \Re(\mathbf{H}_c) \end{bmatrix}, \quad \mathbf{y}_r = \begin{bmatrix} \Re(\mathbf{y}_c) \\ \Im(\mathbf{y}_c) \end{bmatrix},$$

$$\mathbf{x}_r = \begin{bmatrix} \Re(\mathbf{x}_c) \\ \Im(\mathbf{x}_c) \end{bmatrix}, \quad \mathbf{n}_r = \begin{bmatrix} \Re(\mathbf{n}_c) \\ \Im(\mathbf{n}_c) \end{bmatrix}. \tag{3}$$

Dropping the subscript r in (2) for convenience, the real-valued system model is written as

$$\mathbf{y} \quad = \quad \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{4}$$

For a square QAM alphabet $\mathbb{B}$, the elements of $\mathbf{x}$ will take values from the underlying PAM alphabet $\mathbb{A}$. The ML detection rule is then given by

$$\widehat{\mathbf{x}}_{ML} \quad = \quad \arg \min_{\mathbf{x} \in \mathbb{A}^{2n_t}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad = \quad \arg \min_{\mathbf{x} \in \mathbb{A}^{2n_t}} f(\mathbf{x}), \tag{5}$$

where $f(\mathbf{x}) \stackrel{\triangle}{=} \mathbf{x}^T \mathbf{H}^T \mathbf{H}\mathbf{x} - 2\mathbf{y}^T \mathbf{H}\mathbf{x}$ is the ML cost.

III. Proposed Randomized MCMC Algorithm

The ML detection problem in (5) can be solved by using MCMC simulations, which can asymptotically provide the optimal solution [18]. We consider Gibbs sampler, which is an MCMC method used for sampling from distributions of multiple dimensions. In the context of MIMO detection, the joint probability distribution of interest is given by

$$p(x_1, \cdots, x_{2n_t} | \mathbf{y}, \mathbf{H}) \quad \propto \quad \exp\Big(- \frac{\| \mathbf{y} - \mathbf{H}\mathbf{x} \|^2}{\sigma^2} \Big). \tag{6}$$

In conventional Gibbs sampling, the algorithm starts with an initial symbol vector, denoted by $\mathbf{x}^{(t=0)}$. In each iteration of the algorithm, an updated symbol vector is obtained by sampling from distributions as follows:

$$\begin{aligned} x_1^{(t+1)} &\sim p(x_1 | x_2^{(t)}, x_3^{(t)}, \cdots, x_{2n_t}^{(t)}), \\ x_2^{(t+1)} &\sim p(x_2 | x_1^{(t+1)}, x_3^{(t)}, \cdots, x_{2n_t}^{(t)}), \\ x_3^{(t+1)} &\sim p(x_3 | x_1^{(t+1)}, x_2^{(t+1)}, x_4^{(t)}, \cdots, x_{2n_t}^{(t)}), \\ &\vdots \\ x_{2n_t}^{(t+1)} &\sim p(x_{2n_t} | x_1^{(t+1)}, x_2^{(t+1)}, \cdots, x_{2n_t-1}^{(t+1)}). \end{aligned} \tag{7}$$

The detected symbol vector in a given iteration is chosen to be that symbol vector which has the least ML cost in all the iterations up to that iteration.

MCMC Algorithm in [16]: The MCMC algorithm proposed in [16] used the joint distribution as

$$p(x_1, \cdots, x_{2n_t} | \mathbf{y}, \mathbf{H}) \quad \propto \quad \exp\Big(- \frac{\| \mathbf{y} - \mathbf{H}\mathbf{x} \|^2}{\alpha^2 \sigma^2} \Big), \tag{8}$$

where α is a temperature parameter. The algorithm used a fixed value of α in all the iterations, with the property that after the Markov chain is mixed, the probability of encountering the optimal solution is only polynomially small (not exponentially small). Though this algorithm scales well for large n_t, it's performance is highly dependent on the fixed temperature value α, the optimum value of which varies with SNR and n_t making the algorithm less attractive. In addition, the algorithm also faces the stalling problem at high SNRs (i.e., BER gets worse at high SNRs). Our proposed R-MCMC algorithm, on the other hand, does not have a temperature parameter (so optimization over α is not needed). In addition, as we will see in the performance results section, it significantly alleviates the stalling problem while scaling well for large n_t.

Proposed R-MCMC Algorithm: The key idea behind the proposed R-MCMC algorithm is that, in each iteration, instead of updating $x_i^{(t)}$'s as per (7) with probability 1, we update them as per (7) with probability $(1 - q_i)$ and use a different update rule with probability $q_i = \frac{1}{2n_t}$. The different update rule is as follows. Generate $|\mathbb{A}|$ probability values from uniform distribution as

$$p(x_i^{(t)} = j) \sim U[0, 1], \quad \forall j \in \mathbb{A}$$

such that $\sum_{j=1}^{|\mathbb{A}|} p(x_i^{(t)} = j) = 1$, and sample $x_i^{(t)}$ from this generated pmf.

As we will see in the performance results, this simple randomization in the update rule significantly alleviates the stalling problem and achieves near-ML performance for large n_t. We note that the algorithm in [16] differs from the conventional Gibbs sampler because of the use of α, whereas our algorithm differs from conventional Gibbs sampler because of the randomized update rule described above.

Performance and Complexity of R-MCMC: The simulated BER performance of the proposed R-MCMC algorithm is shown in Figs. 1 and 2. In Fig. 1, we plot the BER performance of R-MCMC in a 16×16 V-BLAST MIMO system with 4-QAM. The performance of the MCMC algorithm in [16] for temperature parameter values $\alpha = 1, 1.5, 2, 3$ are also plotted. For both R-MCMC and MCMC in [16], the number of iterations used is $20n_t$. The sphere decoder (SD) performance is also shown for comparison. It can be seen that the performance of MCMC in [16] is very sensitive to the choice of the value of α. For e.g., for $\alpha = 1, 1.5$, we can see the BER degradation at high SNRs due to stalling problem. For $\alpha = 2$, the performance is better at high SNRs but worse in low SNRs. However, the proposed R-MCMC performs better than the MCMC in [16] (or almost the same) at all SNRs and α values shown, at the same order of complexity. In fact, the performance of R-MCMC is almost the same as the sphere decoder performance. The R-MCMC complexity is, however, significantly lower than the SD complexity. While the SD gets exponentially complex in n_t at low SNRs, the R-MCMC complexity (in number of real operations per bit) is only $O(n_t^2)$.

In Fig. 2, we plot the BER as a function of SNR for 8×8, 16×16, 32×32 and 64×64 V-BLAST MIMO systems with 4-QAM. The number of iterations is $20n_t$. The performance

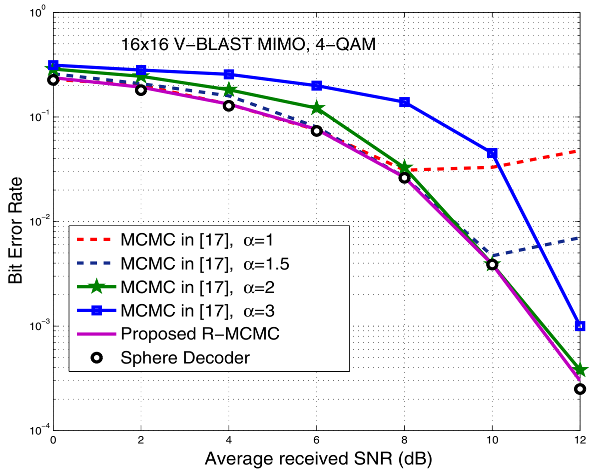

Fig. 1. BER performance of the proposed R-MCMC algorithm in comparison with those of MCMC algorithm in [17] and sphere decoder for 16×16 V-BLAST MIMO with 4-QAM.

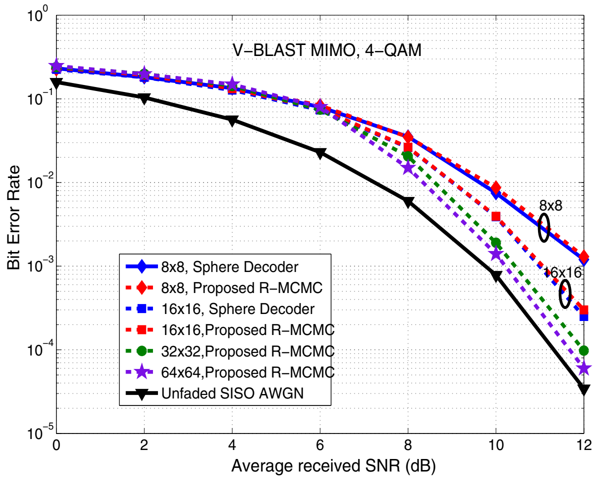

Fig. 2. BER performance of the proposed R-MCMC algorithm in 8×8, 16×16, 32×32 and 64×64 V-BLAST MIMO systems with 4-QAM.

of SD for 8×8 and 16×16 MIMO are also shown for comparison. We do not show the SD performance for 32×32 and 64×64 MIMO because of the prohibitive complexity of SD in such large dimensions. However, we have plotted the unfaded SISO AWGN performance which is a lower bound on the ML performance. From Fig. 2, we observe that the R-MCMC algorithm achieves almost the same performance as that of SD for 16×16 MIMO. The R-MCMC performance in 32×32 and 64×64 is also quite close to the ML performance since it is close to (within 0.5 dB at 10^{-3} BER) the unfaded SISO AWGN performance.

IV. Proposed Randomized Search Algorithm

In this section, we present the proposed RS algorithm for large-MIMO detection. A key component of the RS algorithm is a search procedure which employs a *random selection approach* to choose the candidate vectors to be tested in a local neighborhood search. The algorithm is described below.

A. Randomized Search Algorithm

Given $\mathbf{y}$ and $\mathbf{H}$, the proposed RS algorithm starts with an initial solution vector $\mathbf{x}^{(0)}$, a fixed index set $S = \{1, 2, \cdots, 2n_t\}$, and two dynamic index sets C and D which are initialized to be empty. The algorithm is iterative where each iteration results in a solution vector, which, in turn, is used as the input to the next iteration. The set C is updated only once per iteration, whereas the set D may be updated multiple times (or may not be updated) within each iteration. The set C will contain the set of those indices (i.e., symbol positions) where a symbol change in those positions relative to the solution vector of the previous iteration led to an ML cost improvement. In other words, in iteration t, the set C will add the index of the element in $\mathbf{x}^{(t)}$ which when changed, improved the ML cost. The set D will contain the set of indices where a symbol change in those positions within an iteration did not result in ML cost improvement.

Define the neighborhood set of $\mathbf{x}^{(t)}$, denoted by $\mathcal{N}(\mathbf{x}^{(t)})$, as

$$\mathcal{N}(\mathbf{x}^{(t)}) = \Big\{\mathbf{p} \in \mathbb{A}^{2n_t} : \sum_{i=1}^{2n_t} I_{(x_i^{(t)} \neq p_i)} = 1 \text{ and } j \notin \{C\} \ \forall \ x_j^{(t)} \neq p_j, \ j = 1, \cdots, 2n_t\Big\}, \qquad (9)$$

where $x_i^{(t)}$ and p_i represent the ith component of $\mathbf{x}^{(t)}$ and $\mathbf{p}$, respectively, I is an indicator function ($= 1$ if $x_i^{(t)} \neq p_i$ and 0 otherwise), and $\mathcal{N}(\mathbf{x}^{(t)})$ represents all feasible vectors of $\mathbb{A}^{2n_t}$ which are $i)$ one symbol away from $\mathbf{x}^{(t)}$, and $ii)$ the index corresponding to the symbol in which they differ is not in set C.

Step 1: Given an initial solution vector $\mathbf{x}^{(t=0)}$, find its neighborhood set, $\mathcal{N}(\mathbf{x}^{(t=0)})$.

Step 2: Randomly select an element m from the index set $\{S - C - D\}$. Choose a subset of vectors from $\mathcal{N}(\mathbf{x}^{(t)})$, denoted by $\{\mathbf{d}(j), \ j = 1, 2, \cdots, |\mathbb{A}| - 1\}$, such that $\mathbf{d}(j)$'s differ from $\mathbf{x}^{(t)}$ in the mth position, $m \in \{S - C - D\}$. It is noted that $j \in \{1, \cdots, |\mathbb{A}| - 1\}$, since, for each symbol in a given position, there are $|\mathbb{A}| - 1$ possible other symbols. Let $g(\mathbf{x}^{(t)} \rightarrow \mathbf{d}(j))$ denote the difference in the ML cost between $\mathbf{x}^{(t)}$ and $\mathbf{d}(j)$, i.e.,

$$\begin{aligned} g(\mathbf{x}^{(t)} \rightarrow \mathbf{d}(j)) &= f(\mathbf{x}^{(t)}) - f(\mathbf{d}(j)) \\ &= \|\mathbf{y} - \mathbf{H}\mathbf{x}^{(t)}\|^2 - \|\mathbf{y} - \mathbf{H}\mathbf{d}(j)\|^2 \\ &= \mathbf{x}^{(t)T}\mathbf{H}^T\mathbf{H}\mathbf{x}^{(t)} - \mathbf{d}(j)^T\mathbf{H}^T\mathbf{H}\mathbf{d}(j) - 2\mathbf{y}^T\mathbf{H}(\mathbf{x}^{(t)} - \mathbf{d}(j)) \end{aligned} \qquad (10)$$

Let $\mathbf{G} \triangleq \mathbf{H}^T\mathbf{H}$, $\mathbf{z} \triangleq \mathbf{H}^T\mathbf{y}$, and $\beta(j) \triangleq g(\mathbf{x}^{(t)} \rightarrow \mathbf{d}(j))$. By definition, $\mathbf{d}(j)$ can be rewritten as

$$\mathbf{d}(j) = \mathbf{x}^{(t)} + \lambda_m \mathbf{e}_m, \qquad (11)$$

where $\mathbf{e}_m$ denotes the vector with its mth entry only as one and all other entries as zeros, and λ_m belongs to a set of integers such that $\mathbf{d} \in \mathbb{A}^{2n_t}$. For e.g., if $\mathbb{A} = \{-3, -1, +1, +3\}$, then the possible integer values that λ_m can take are $\{-6, -4, -2, 0, 2, 4, 6\}$. Now, (10) can be simplified as

$$\beta(j) = 2\lambda_m z_m - 2\lambda_m \mathbf{e}_m^T \mathbf{G}\mathbf{x}^{(t)} - \lambda_m^2 \mathbf{G}_{m,m}, \qquad (12)$$

where z_m denotes the mth element of $\mathbf{z}$, and $\mathbf{G}_{i,j}$ denotes the element in the ith row and jth column of $\mathbf{G}$.

Step 3: Compute

$$\beta_{max} = \max_j \{\beta(j)\}_{j=1}^{|\mathbb{A}|-1}, \qquad (13)$$

$$max\_idx = \arg\max_j \{\beta(j)\}_{j=1}^{|\mathbb{A}|-1}. \qquad (14)$$

Two cases, namely, $\beta_{max} \geq 0$ and $\beta_{max} < 0$, are possible.

- If $\beta_{max} \geq 0$, then
  make $t = t + 1$, $\mathbf{x}^{(t)} = \mathbf{d}(max\_idx)$, add m to C, find the new neighborhood set $\mathcal{N}(\mathbf{x}^{(t)})$, and go to *Step 2* if $C \neq S$; else output $\mathbf{x}^{(t)}$ as the final solution and stop.
- If $\beta_{max} < 0$, then
  include m in D, and go to *Step 2* if $D \neq \{S - C\}$; else output $\mathbf{x}^{(t)}$ as the final solution and stop.

It is noted that the maximum number of iterations possible is $2n_t$, and the size of the neighborhood set $\mathcal{N}(\mathbf{x}^{(t)})$ decreases by $|\mathbb{A}| - 1$ in each iteration.

B. Multiple-Restart Randomized Search

Running the above RS algorithm with a certain random initial vector allows to explore only 'some' parts of the solution space. Exploring other parts of the solution space can yield better solution vectors. This can be achieved by running the RS algorithm several times (parameterized by $L-1$, referred to as the *number of restarts*) such that each time a 'different' part of the solution space is likely to be explored without increasing the order of complexity. This can be realized through starting the RS algorithm with different random initial vectors each time, which works as follows: 1) Choose a random initial vector; 2) Run the RS algorithm; 3) Repeat the steps 1 and 2, $L-1$ times; 4) Output the solution vector having the least ML cost among the L solution vectors as the final solution vector and stop.

C. Performance and Complexity of RS

The BER performance of the proposed RS algorithm are shown in Figs. 3 and 4. In Fig. 3, we plot the BER as a function of number of restarts, $L-1$, in a 16×16 V-BLAST MIMO system with 4-QAM at SNR = 10 and 12 dB. For these two SNRs, the sphere decoder (SD) performance is also shown for comparison. It can be seen that with no restarts (i.e., $L-1=0$), the performance of the RS algorithm is far from the SD performance. However, increasing the number of restarts improves the BER significantly in such a way that almost SD performance is achieved in about 60 restarts.

In Fig. 4, we plot the BER as a function of SNR for 8×8, 16×16, 32×32 and 64×64 V-BLAST MIMO systems with 4-QAM. The number of restarts for each point was chosen through simulation such that any further increase in number of restarts does not yield significant BER improvement. For e.g., at 8 dB SNR, the chosen values for L are: $L = 16$ for 8×8, $L = 24$ for 16×16, $L = 32$ for 32×32, and $L = 32$ for 64×64. From Fig. 4, we observe that, like R-MCMC algorithm, the RS algorithm also achieves almost the same performance as that of SD for 16×16 MIMO. The RS performance in 32×32 and 64×64 is also very close to ML lower bound (i.e., SISO AWGN performance).

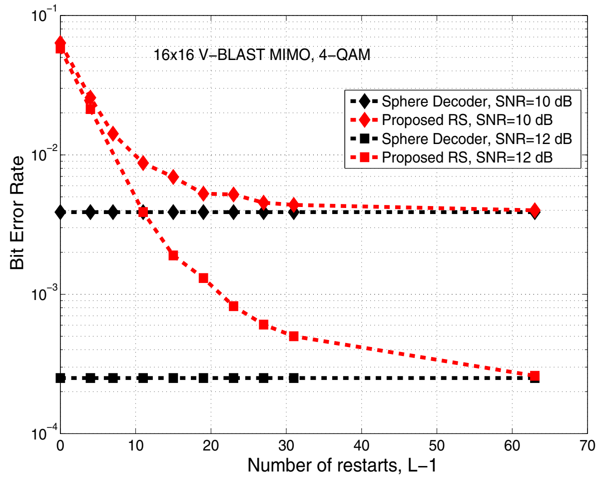

Fig. 3. BER performance of the proposed RS algorithm as a function of number of restarts, $L-1$, in 16×16 V-BLAST MIMO system with 4-QAM and SNR = 10 dB, 12 dB.

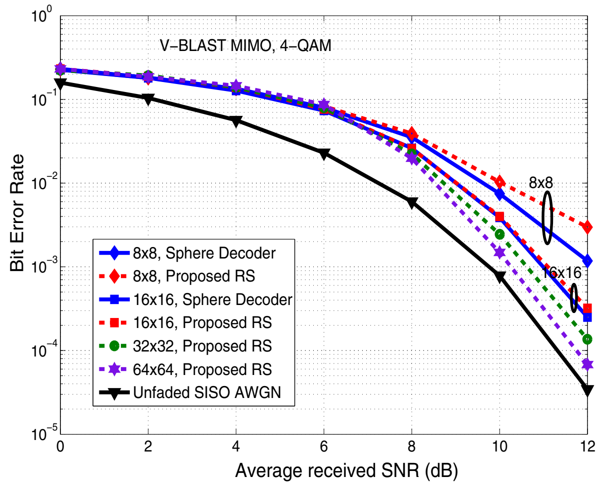

Fig. 4. BER performance of the proposed RS algorithm in 8×8, 16×16, 32×32, and 64×64 V-BLAST MIMO systems with 4-QAM. Sphere decoder performance for 8×8 and 16×16 MIMO are also shown.

Through simulations, we computed the complexity of the RS algorithm in terms of average number of real operations required per bit to achieve a target BER. We found that the complexity of the proposed RS algorithm is $O(n_t^{1.4})$, which is quite attractive for large-MIMO systems.

V. Message Passing-Aided R-MCMC and RS for Higher-Order QAM

In the previous two sections, we saw that the proposed R-MCMC and RS algorithms achieved near-ML performance for large n_t and 4-QAM. However, we found that their higher-order QAM performances are far from ML performance. This observation is illustrated in Fig. 5, where the RS performance is found to be 5 dB away from SD performance at 10^{-2} symbol error rate (SER) and R-MCMC performance is 2 dB away from SD performance at 10^{-2} SER. This observation motivated us to look for ways to improve the performance of R-MCMC and RS for higher-order QAM. We are able to significantly improve the higher-order QAM performance of R-MCMC by using improved priors; these priors are obtained from a low-complexity message passing (MP) algorithm. We

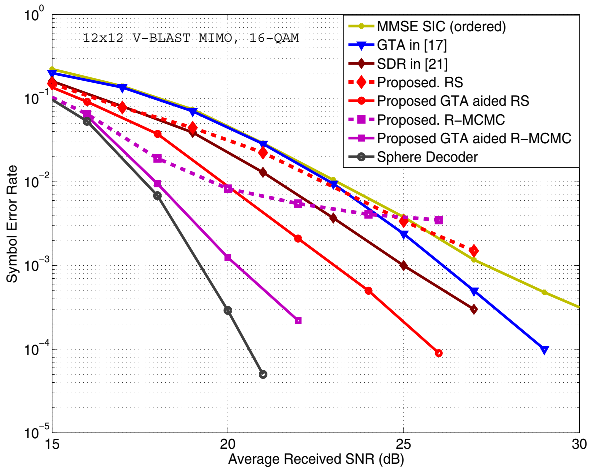

Fig. 5. SER comparison of the proposed MP-aided RS and R-MCMC algorithms with the proposed RS and R-MCMC algorithms and other detection algorithms: MMSE-SIC, SDR in [21], GTA in [17], and sphere decoder in 12×12 V-BLAST MIMO with 16-QAM.

refer to this algorithm as *message passing aided R-MCMC*. Likewise, by running the RS using the output vector from the message passing algorithm as the initial vector (in addition to the L runs with random initial vectors), the higher-order QAM performance of RS is significantly improved. We call this algorithm as *message passing aided RS*.

MP Aided R-MCMC: Recently, in [17], a Markov random field (MRF) based message passing algorithm has been reported for MIMO detection. The algorithm used a Gaussian tree approximation (GTA) to convert the fully-connected graph representing the MIMO system into a tree, and carry out message passing on the resultant approximated tree. We note that though the per-bit complexity of this GTA algorithm is also $O(n_t^2)$ like R-MCMC, its higher-order QAM performance is far from SD performance. This is illustrated in Fig. 5, where GTA performance is away from SD performance by about 5 dB at 10^{-2} SER in 12×12 MIMO. In the R-MCMC algorithm in Sec. III, the priors are taken to be uniform. In the MP aided R-MCMC, however, we use the output beliefs from the GTA algorithm as the priors to the R-MCMC algorithm.

MP Aided RS: The proposed MP aided RS works as follows. Run the RS algorithm L times with random initial vectors as described in Sec. IV. In addition, run RS with the output vector from GTA as the initial vector. Choose the best vector among these $L+1$ solution vectors and declare it as the final solution vector.

Performance of MP Aided R-MCMC/RS: In Fig. 5, we show the SER performance of the proposed MP aided R-MCMC/RS algorithms for 16-QAM in 12×12 MIMO. Performance of other algorithms including MMSE-SIC, semi-definite relaxation (SDR) in [21], GTA algorithm in [17], and sphere decoding are also shown for comparison. The MP aided R-MCMC/RS algorithms are seen to outperform the SDR, GTA and MMSE-SIC detectors. Performance of MP aided RS gets closer to SD performance (close to within about 2 dB at 10^{-2} SER) compared to RS without MP aid. The MP aided R-MCMC performance gets even closer to SD performance (close to within 0.25 dB and 1 dB at 10^{-2} and 10^{-3} SER).

VI. Conclusions

We presented two algorithms for low-complexity near-ML detection in MIMO systems with tens of antennas. The proposed R-MCMC algorithm was shown to achieve near-ML performance and alleviate the stalling problem by adopting a randomized sampling based on a probabilistic rule. The proposed RS algorithm was also shown to achieve near-ML performance by the use of randomized selection of candidate vectors and multiple restarts. The per-bit complexities of R-MCMC and RS are $O(n_t^2)$ and $O(n_t^{1.4})$, respectively. A message passing aided scheme that enabled R-MCMC and RS to perform well in higher-order QAM was also presented.

References

[1] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telecommun.,* vol. 10, no. 6, pp. 585-595, November 1999.

[2] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, 2003.

[3] K. Vishnu Vardhan, Saif K. Mohammed, A. Chockalingam, B. Sundar Rajan, "A low-complexity detector for large MIMO systems and multicarrier CDMA systems," *IEEE JSAC Spl. Iss. on Multiuser Detection for Adv. Commun. Sys. Net.*, vol. 26, no. 3, pp. 473-485, April 2008.

[4] B. Cerato and E. Viterbo, "Hardware implementation of low-complexity detector for large MIMO," *IEEE ISCAS'2009*, pp. 593-596, Taipei, May 2009.

[5] Saif K. Mohammed, Ahmed Zaki, A. Chockalingam, and B. Sundar Rajan, "High-rate space-time coded large-MIMO systems: Low-complexity detection and channel estimation," *IEEE Jl. Sel. Topics in Sig. Proc.*, vol. 3, no. 6, pp. 958-974, December 2009.

[6] P. Li and R. D. Murch, "Multiple output selection-LAS algorithm in large MIMO systems," *IEEE Commun. Lett.,* pp. 399-401, May 2010.

[7] N. Srinidhi, Saif K. Mohammed, A. Chockalingam, B. Sundar Rajan, "Low-complexity near-ML decoding of large non-orthogonal STBCs using reactive tabu search," *Proc. IEEE ISIT'2009*, Seoul, July 2009.

[8] N. Srinidhi, Saif K. Mohammed, A. Chockalingam, B. Sundar Rajan, "Near-ML signal detection in large-dimension linear vector channels using reactive tabu search," arXiv:0911.4640v1 [cs.IT] 24 Nov 2009.

[9] N. Srinidhi, Tanumay Datta, A. Chockalingam, and B. Sundar Rajan, "Layered tabu search algorithm for large-MIMO detection and a lower bound on ML performance," *Proc. of IEEE GLOBECOM'2010*.

[10] Saif K. Mohammed, A. Chockalingam, and B. Sundar Rajan, "Low-complexity near-MAP decoding of large non-orthogonal STBCs using PDA," *Proc. IEEE ISIT'2009*, Seoul, June-July 2009.

[11] Suneel Madhekar, Pritam Som, A. Chockalingam, and B. Sundar Rajan, "Belief propagation based decoding of large non-orthogonal STBCs," *Proc. IEEE ISIT'2009*, Seoul, June-July 2009.

[12] Pritam Som, Tanumay Datta, A. Chockalingam, and B. Sundar Rajan, "Improved large-MIMO detection based on damped belief propagation," *Proc. IEEE ITW'2010*, Cairo, January 2010.

[13] R. Chen, J. S. Liu, and X. Wang, "Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications," *IEEE Trans. Sig. Proc.*, vol. 50, no. 2, pp. 255-270, Feb 2002.

[14] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," *IEEE Trans. on Sig. Proc.*, vol. 54, no. 5, pp. 1896-1909, May 2006.

[15] H. Zhu, B. Farhang-Boroujeny, and R. R. Chen, "On performance of sphere decoding and Markov chain Monte Carlo methods," *IEEE Sig. Proc. Lett.,* vol. 12, no, 10, pp. 669-672, Oct 2005.

[16] M. Hansen, B. Hassibi, A. G. Dimakis, and W. Xu, "Near-optimal detection in MIMO systems using Gibbs sampling," *Proc. IEEE ICC'2009*, Honolulu, December 2009.

[17] J. Goldberger and A. Leshem, "MIMO detection for high-order QAM based on a Gaussian tree approximation," *arXiv:1001.5364v1[cs.IT] 29 Jan 2010*.

[18] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge Univ. Press, 2003.

[19] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2nd Edition, 2004.

[20] O. Haggsstrom, *Finite Markov Chains and Algorithmic Applications*, Cambridge Univ. Press, 2002.

[21] N.D. Sidiropoulos and Z.Q. Luo, "A semidefinite relaxation approach to MIMO detection for high-order QAM constellations," *IEEE Sig. Proc. Letters*, vol. 13, no. 9, pp. 525-528, September 2006.