Monte Carlo Tree Search for Simultaneous Ascending Auctions

Marceau Coupechoux[1]

[1]LTCI, Telecom Paris, Institut Polytechnique de Paris, France

[2]Orange Labs, France

13 juillet 2022

Joint work with Alexandre Pacaud[1,2] and Aurelien Bechler[2]

Motivations

Bidding efficiently in Simultaneous Ascending Auctions is a hot topic

- SAA has become the privilege mechanism used for spectrum auctions since its introduction in 1994 by the FCC in US
- SAA has recently been used in many countries for 5G licences (Germany [Bundesnetzagentur, 2022], Italy [European 5G Observatory, 2018], etc)
- Paul Milgrom and Robert Wilson received the Nobel Prize in Economy in 2020 mainly for their contribution to SAA (mechanism design) [Milgrom, 2000]

Gap in literature regarding how to bid efficiently in SAA

- Auction theory or exact game resolution methods are unable to compute the optimal bidding strategies due to the high complexity of the game.
- Strategical issues have always been studied separately generally in specific contexts and simplified versions of SAA [Goeree and Lien, 2014, Zheng, 2012, Brusco and Lopomo, 2002]

$\Rightarrow$ We propose a tree-search approach to the bidder strategy problem **tackling simultaneously** two strategical issues : *exposure* and *own price effect*

Outline

Simultaneous Ascending Auctions (SAA)

Brief Presentation of SAA [Milgrom, 2000, Cramton et al., 2006]

- It is an auction mechanism where m indivisible goods are sold via **separate and concurrent English auctions** between n bidders
- Bidding occurs in **multiple rounds**
- At each round :
  - Bidders submit their bids simultaneously, *activity rules* may constrain bidders to play (avoid wait-and-see strategy)
  - For every item j : The bidder having placed the highest bid becomes its temporary winner (ties randomly broken) and its bid price P_j is set to the highest bid
  - The temporary winner and bid price of each item is revealed, the minimal admissible bid for the next round is $P_j + \varepsilon$ (ε bid increment)
- Until : no new bids are submitted during a round on any object (*closing rule*)
- After closing : the objects are sold at the bid prices to the corresponding winners

Deterministic SAA with complete information

Brief Presentation of d-SAA with complete information

- Bidders take turns bidding (no more simultaneity and stochasticity)
- Temporary winner and bid price P_j are announced after each turn
- New bids are constrained to be $P_j + \varepsilon$ (discrete action space)[Goeree and Lien, 2014, Wellman et al., 2008]
- The value function of the bidders are common knowledge [Szentes and Rosenthal, 2003a, Szentes and Rosenthal, 2003b]

$\Rightarrow$ d-SAA is a sequential deterministic game with perfect and complete information

Game complexities [Van Den Herik et al., 2002]

Ex : 5G auction in Italy in 2018, $m = 12$ items, $n = 5$ bidders, $R = 171$ rounds

- State space complexity : $\sum_{i'=0}^{n-1}(1 + \sum_{i=0}^{n-1}(R - i - i')^+)^m \mathbf{1}_{\{R \geq i'\}}$ (ex : 10^{35})
- Game tree complexity : $> 2^{m(n-1)\lfloor \frac{R}{n} \rfloor}$ (ex : 10^{491})

Comparison of SAA and d-SAA extensive form

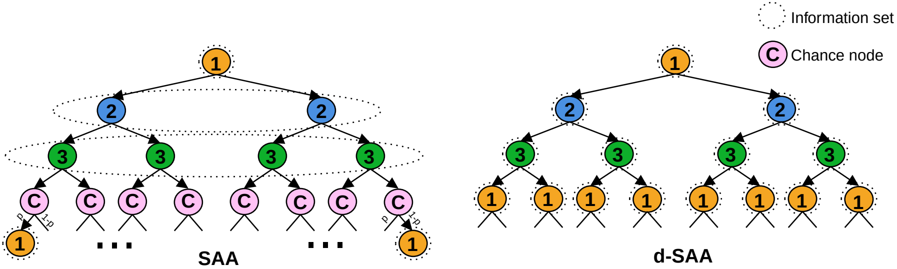


Figure – Comparison of SAA and d-SAA extensive form

⇒ Conception of simpler tree-search algorithms in d-SAA

Utility and value functions

Value functions

Each player i is defined by its value function v_i which respects the following properties :

- Normalisation : $v_i(\emptyset) = 0$; Finite : $\forall X, v_i(X) < +\infty$
- Free disposal : $\forall X, Y, X \subset Y$ implies $v_i(X) \leq v_i(Y)$ [Milgrom, 2000]

Complements and substitutes

- A set of goods X exhibits **complementarities** with a disjoint set of goods Y if $v(X+Y) > v(X) + v(Y)$
- A set of goods X exhibits **substitutabilities** with a disjoint set of goods Y if $v(X+Y) < v(X) + v(Y)$

Utility function

At the end of the auction, if player i wins the set of goods X and the bid price vector is P, then its utility is :

$$\sigma_i(X, P) = v_i(X) - \sum_{j \in X} P_j$$

Exposure problem

Definition

The exposure problem refers to the possibility that, by bidding on a set of complementary goods, a bidder ends up paying more than its valuation for the subset it actually wins as the goods have become too expensive

Example 1

	$v(\{1\})$	$v(\{2\})$	$v(\{1,2\})$
Player 1	12	12	12
Player 2	0	0	20

- A rational strategy for player 1 is :
  - To pass its turn if currently winning an item or the bid price of both items is greater than $12 - \varepsilon$ (e.g. $\varepsilon = 1$)
  - To bid on the cheapest item otherwise
- Given the fact that player 1 plays rationally, if player 2 bids on an item, player 2 will end up exposed as it will not be able to acquire both items for a price inferior to 22

$\Rightarrow$ No efficient bidding strategy is known to avoid this problem in the general case

Own price effect

Definitions

- **Own price effect :** Each bid on an item increases its price and decreases the utility of bidders willing to acquire it. Each bidder has its *own effect* on the prices [Weber, 1997]
- **Demand reduction strategy** [Weber, 1997, Ausubel et al., 2014] : Reduce demand to keep prices low and coordinate on a split of the items (this is a *collusion* [Brusco and Lopomo, 2002])

Example 2 ($\varepsilon = 1$)

	$v(\{1\})$	$v(\{2\})$	$v(\{1,2\})$
Player 1	10	10	20
Player 2	10	10	20

- If players don't form a collusion, the final bid price of each item will be 10. Both players end up with a utility of 0.
- If players form a collusion, then they both acquire an item for a price of $\varepsilon = 1$ and end up with a utility of 9.

Outline

Point price prediction bidding

Point-price prediction bidding (PP) [Wellman et al., 2008]

- A **point-price prediction bidder** (PP) computes the subset of goods

$$X^* = \underset{X}{\operatorname{argmax}}\, \sigma(X, \rho(\mathcal{B}))$$

  breaking ties in favour of smaller subsets and lower-numbered goods.
- The bidder bids $P_j + \varepsilon$ on all items j that it is not currently winning in X^*.
- The function $\rho : \mathcal{B} \to \mathbf{R}_+{}^m$ maps the bidder's information state $\mathcal{B}$ to an estimation of the final price vector $\rho(\mathcal{B})$.
- ρ may use only the initial estimation of the final price vector $\rho(\mathcal{B}_0)$:

$$\rho_j(\mathcal{B}) = \begin{cases} \max(\rho_j(\mathcal{B}_0), P_j) & \text{if winning good } j \\ \max(\rho_j(\mathcal{B}_0), P_j + \varepsilon) & \text{otherwise} \end{cases}$$

Straightforward Bidding (SB) [Milgrom, 2000]

The Straightforward Bidding strategy (SB) corresponds to a PP with null initial estimation of the final price vector ($\rho(\mathcal{B}_0) = 0$)

Predicting the final price vector

Walrasian equilibrium

- $D_i(p) = \text{argmax}_X \, \sigma(X, p)$ is the demand set of bidder i at price p
- A **Walrasian equilibrium** is a price vector p and an allocation $(X_1, ..., X_n)$ such that $X_i \in D_i(p)$ for every bidder i and all items are allocated (market clearance)
- Walrasian equilibrium doesn't always exist (ex : Example 1)

Expected Price equilibrium (EPE) [Wellman et al., 2008]

- **EPE :** Tâtonnement process used to find a Walrasian equilibrium if it exists.

$$p(t+1) = p(t) + \alpha(t)(x(p(t)) - 1) \quad (x \text{ the demand function})$$

- **Problem :** Does not take in account the auction's mechanism

Self-Confirming Point Price Prediction [Wellman et al., 2008]

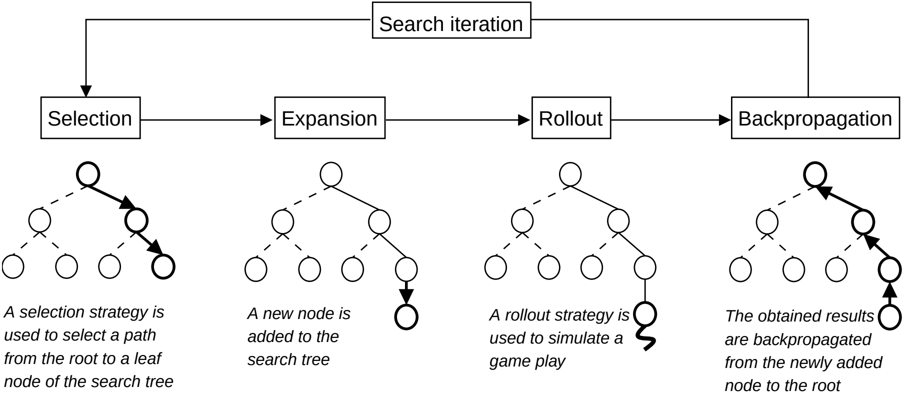
A **Self-Confirming Point Price Prediction** is a price vector p such that, if all bidders play PP with initial estimation $\rho(\mathcal{B}_0) = p$, the final price vector is equal to p. It does not always exist (ex : Example 1).

Outline

Monte Carlo Tree Search (MCTS)

Search Tree

As it is impossible to explore the whole game tree, only a small portion of it is explored. In MCTS, it is constructed iteratively.



MCTS

Success and theoretical guarantees in two-player zero-sum deterministic games

- **Theoretical guarantees :** It has been shown that the probability of playing a suboptimal action with the MCTS variant *Upper Confidence bounds applied to Trees* (UCT) converges to zero at polynomial rate as the number of search iterations grows to infinity [Kocsis and Szepesvári, 2006].
- Great success in various games such as *Go* [Coulom, 2006, Lee et al., 2009] or *Othello* [Robles et al., 2011].
- In March 2016, the algorithm **AlphaGo** beats the world champion of Go 4-1 [Silver et al., 2016]

$\Rightarrow$ d-SAA is a n-player non-zero sum game. No theoretical guarantees regarding MCTS exist for such games. We use the MCTS-max^n which is the most popular variant for such games [Nijssen, 2013].

Selection phase

Selection index

We use a penalised variant of UCT. At parent node y, our selection strategy chooses the child x with the highest score q_x :

$$q_x = \frac{r_x}{n_x} + \max(b_x - a_x, \varepsilon)\sqrt{\frac{2\log(n_y)}{n_x}} - no\_object(x) - risky\_move(x) \qquad (1)$$

where

- r_x is the sum of rewards found in the subtree with root x
- n_x is the number of visits of child node x
- n_y is the number of visits of parent node y
- ε is the bid increment
- a_x is the estimated lower bound of the reward support found in the subtree with root x
- b_x is the estimated higher bound of the reward support found in the subtree with root x

Selection : First penalty term

Penalty term *no_object*

- **Objective :** Discourage bidders to pass their turn if they have got nothing to lose by bidding on an additional item.
- A player i' will no longer bid on an item j if $\forall X \in S_{-j}, P_j \geq v_{i'}(X + \{j\}) - v_{i'}(X)$
- $\Pi_j^i = \max_{i' \in \{1,\dots,n\} \setminus \{i\}} \max_{X \in S_{-j}} v_{i'}(X + \{j\}) - v_{i'}(X)$ is the minimal price from which item j is considered as undesired by all opponents of i.
- Let P^x be the price vector at child node x, i the player bidding at parent node y and X_x^i the set of goods temporarily won by player i at x

$$no\_object(x) = \begin{cases} \max_{j \in \{1,\dots,m\} \setminus X_x^i} (v_i(X_x^i + \{j\}) - v_i(X_x^i) - P_j - \varepsilon)^+ \\ \text{if } \{j' \in X_x^i, P_x^{j'} < \Pi_{j'}^i\} = \emptyset \\ 0 \quad \text{otherwise} \end{cases} \quad (2)$$

Selection : Second penalty term

Penalty term *risky_move*

- **Objective :** Deter players from bidding on sets of goods which might lead to exposure.
- A set of goods X is said to lead to exposure at price vector p if $\exists Y \subseteq X, \sigma_i(Y, p) < 0$
- Let P^x be the price vector at child node x, r the root player, i the player bidding at parent node y and X_x^i the set of goods temporarily won by player i at node x

$$risky\_move(x) = \begin{cases} \lambda^r v_i(\{1, ..., m\}) & \text{if } X_x^i \text{ can lead to exposure for } i = r \text{ at price } P^x \\ \lambda^o v_i(\{1, ..., m\}) & \text{if } X_x^i \text{ can lead to exposure for } i \neq r \text{ at price } P^x \\ 0 & \text{otherwise} \end{cases}$$

- λ^r and λ^o have opposite effects on the algorithm's risk aversion
  - λ^r controls the risk aversion of the root player
  - λ^o controls the risk aversion of the root player's opponents

Rollout phase

- The default rollout strategy is to play randomly. However, in d-SAA, it leads to absurd outcomes with potentially very high prices.
- Our rollout strategy is PP with a new method to estimate the final price vector $\rho(\mathcal{B}_0)$

$$p(t+1) = \frac{1}{t+1} f(p(t)) + (1 - \frac{1}{t+1}) p(t)$$

  where $f(p)$ is the final price vector obtained when all players play PP with initial prediction p
    - $p(t)$ always converges when $m = n = 2$ and items exhibit complementarities
    - We conjecture the convergence in the general case
- Rollout algorithm :
    - Compute the limit P^* of $p(t)$.
    - Set $\rho(\mathcal{B}_0) = P^* + \eta$ where η is a random variable which follows a bounded uniform distribution (introduce diversity and improve sampling)
    - Simulate PP with this estimation.

Outline

Simulation settings

- **MCTS settings :** $\lambda_r = \lambda_o = 0.07$ (grid-search)
- We compare to SB [Milgrom, 2000], EPE [Wellman et al., 2008], SCPD [Wellman et al., 2008], UCB (with no selection penalties) and MCTS^{np} with no selection penalties
- Each algorithm is given a maximum of 30 seconds CPU thinking time.

Test experiment : Exposure

	$v(\{1\})$	$v(\{2\})$	$v(\{1,2\})$
Player 1	12	12	12
Player 2	0	0	20

Table – Example 1

- MCTS, MCTS^{np}, UCB and EPE suggest player 2 not to bid and, hence, avoids exposure.
- SB and SCPD expose player 2 by inciting player 2 to bid on both items
- MCTS is able to avoid obvious exposure.

Test experiment : Own price effect

	$v(\{1\})$	$v(\{2\})$	$v(\{1,2\})$
Player 1	h	h	$2h$
Player 2	ℓ	ℓ	2ℓ

- $0 \leq \ell \leq h$ and ε infinitesimal
- If $h \leq 2(h-\ell)$, Player 1 should bid on both items until Player 2 drops out
- If $h \geq 2(h-\ell)$, Player 1 should form a collusion with Player 2 by conceding an item
- Player 2 optimal strategy is to bid on the cheapest item if its bid price is lower than $\ell - \varepsilon$ and is currently winning no items and pass otherwise

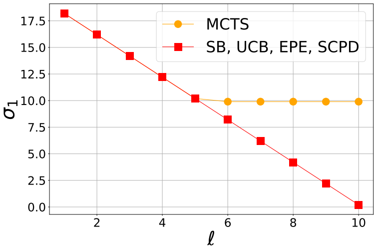
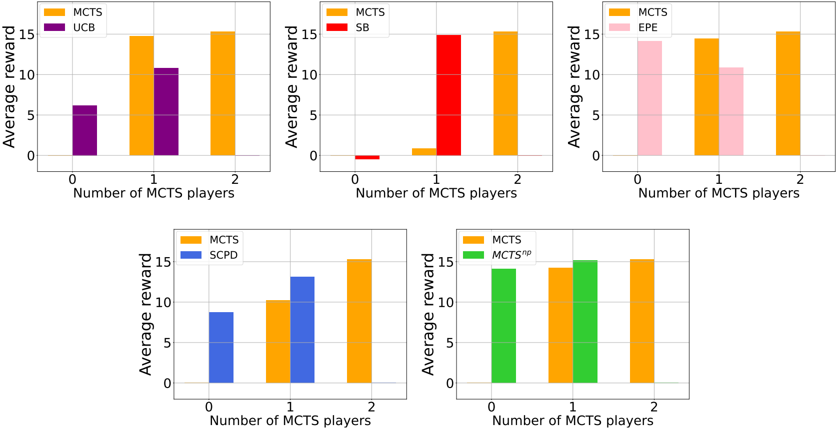


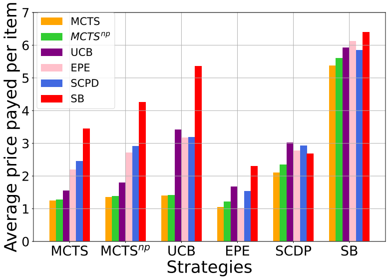
Figure – Evolution of player 1's utility σ_1 in Test experiment [Brusco and Lopomo, 2002] ($h = 10$, $\varepsilon = 0.1$) given that player 2 plays optimally

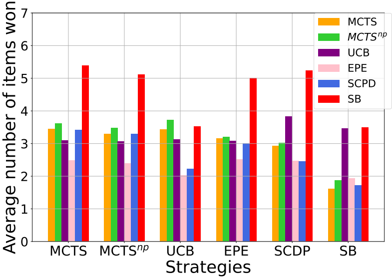
Extensive Experiments : Utility $(n = 2, m = 7, \varepsilon = 1)$



$\Rightarrow$ (MCTS,MCTS) is the only pure Nash equilibrium of the Normal form game in expected payoff with six strategies

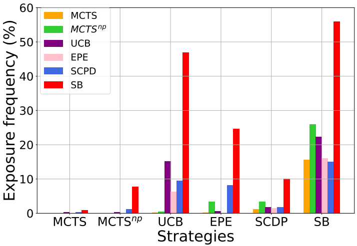
Extensive Experiments : Own Price Effect $(n = 2, m = 7, \varepsilon = 1)$

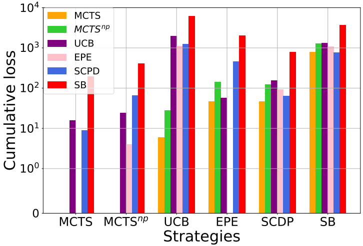




- MCTS obtains the lowest average price per item won against every strategy
- MCTS is fairly competitive in terms of number of acquired items

Extensive Experiments : Exposure $(n = 2, m = 7, \varepsilon = 1)$





- (MCTS,MCTS) never suffers from exposure
- MCTS has very low exposure frequency against every strategy, except SB
- MCTS is less exposed and has lower losses than MCTS^{np} against other strategies

Outline

Conclusion and future works

Conclusion

- First algorithm to tackle **simultaneously** the *exposure problem* and *own price effect* in a simplified version of SAA
- MCTS : a promising approach to derive auction strategies in SAA

Complementary works

- Our algorithm is easily extended to budget constraints
- Algorithm remains efficient and robust to significant errors in the valuation estimates

Future work

- Increase in the number of players
- Adding simultaneity and incomplete information to our SAA model

Thank you for your attention !

References I

Ausubel, L. M., Cramton, P., Pycia, M., Rostek, M., and Weretka, M. (2014).
Demand reduction and inefficiency in multi-unit auctions.
The Review of Economic Studies, 81(4) :1366–1400.

Brusco, S. and Lopomo, G. (2002).
Collusion via signalling in simultaneous ascending bid auctions with heterogeneous objects, with and without complementarities.
The Review of Economic Studies, 69(2) :407–436.

Bundesnetzagentur (2022).
Spectrum for wireless access for the provision of telecommunications services.
https://www.bundesnetzagentur.de/EN/Areas/Telecommunications/Companies/FrequencyManagement/ElectronicCommunicationsServices/ElectronicCommunicationServices_node.html.

Coulom, R. (2006).
Efficient selectivity and backup operators in monte-carlo tree search.
In *International conference on computers and games*, pages 72–83. Springer.

References II

Cramton, P. et al. (2006).
Simultaneous ascending auctions.
Combinatorial auctions, pages 99–114.

European 5G Observatory (2018).
Italian 5g spectrum auction.
https://5gobservatory.eu/italian-5g-spectrum-auction-2/.

Goeree, J. K. and Lien, Y. (2014).
An equilibrium analysis of the simultaneous ascending auction.
Journal of Economic Theory, 153 :506–533.

Kocsis, L. and Szepesvári, C. (2006).
Bandit based monte-carlo planning.
In *European conference on machine learning*, pages 282–293. Springer.

References III

Lee, C.-S., Wang, M.-H., Chaslot, G., Hoock, J.-B., Rimmel, A., Teytaud, O., Tsai, S.-R., Hsu, S.-C., and Hong, T.-P. (2009).
The computational intelligence of mogo revealed in taiwan's computer go tournaments.
IEEE Transactions on Computational Intelligence and AI in games, 1(1) :73–89.

Milgrom, P. (2000).
Putting auction theory to work : The simultaneous ascending auction.
Journal of political economy, 108(2) :245–272.

Nijssen, J. A. M. (2013).
Monte-Carlo tree search for multi-player games.
PhD thesis, Maastricht University.

References IV

Robles, D., Rohlfshagen, P., and Lucas, S. M. (2011).
Learning non-random moves for playing othello : Improving monte carlo tree search.

In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pages 305–312. IEEE.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016).
Mastering the game of go with deep neural networks and tree search.
nature, 529(7587) :484–489.

Szentes, B. and Rosenthal, R. W. (2003a).
Beyond chopsticks : Symmetric equilibria in majority auction games.
Games and Economic Behavior, 45(2) :278–295.

Szentes, B. and Rosenthal, R. W. (2003b).
Three-object two-bidder simultaneous auctions : chopsticks and tetrahedra.
Games and Economic Behavior, 44(1) :114–133.

References V

Van Den Herik, H. J., Uiterwijk, J. W., and Van Rijswijck, J. (2002).
Games solved : Now and in the future.
Artificial Intelligence, 134(1-2) :277–311.

Weber, R. J. (1997).
Making more from less : Strategic demand reduction in the fcc spectrum auctions.
Journal of Economics & Management Strategy, 6(3) :529–548.

Wellman, M. P., Osepayshvilli, A., MacKie-Mason, J. K., and Reeves, D. (2008).
Bidding strategies for simultaneous ascending auctions.
B.E. J. Theoret. Econom.

Zheng, C. Z. (2012).
Jump bidding and overconcentration in decentralized simultaneous ascending auctions.
Games and Economic Behavior, 76(2) :648–664.