

Bidding efficiently in Simultaneous Ascending Auctions with budget and eligibility constraints using Simultaneous Move Monte Carlo Tree Search

Alexandre Pacaud, Aurelien Bechler and Marceau Coupechoux

Abstract—For decades, Simultaneous Ascending Auction (SAA) has been the most popular mechanism used for spectrum auctions. It has recently been employed by many countries for the allocation of 5G licences. Although SAA presents relatively simple rules, it induces a complex strategic game for which the optimal bidding strategy is unknown. Considering the fact that sometimes billions of euros are at stake in an SAA, establishing an efficient bidding strategy is crucial. In this work, we model the auction as a n -player simultaneous move game with complete information and propose the first efficient bidding algorithm that tackles simultaneously its four major strategic issues: the *exposure problem*, the *own price effect*, *budget constraints* and the *eligibility management problem*. Our solution, called SMS^α , is based on Simultaneous Move Monte Carlo Tree Search (SM-MCTS) and relies on a new method for the prediction of closing prices. By introducing a new reward function in SMS^α , we give the possibility to bidders to define their own level of risk-aversion. Through extensive numerical experiments on instances of realistic size, we show that SMS^α largely outperforms state-of-the-art algorithms, notably by achieving higher expected utility while taking less risks.

Index Terms—Simultaneous Move Monte Carlo Tree Search, Ascending Auctions, Exposure, Own price effect, Risk-aversion

I. INTRODUCTION

In order to provide high quality service and develop wireless communication networks, mobile operators need to have access to a wide range of frequencies. These frequencies are obtained in the form of licences. A licence is defined by four features: its frequency band, its geographic coverage, its period of usage and its restrictions on use. Nowadays, spectrum licences are mainly assigned through auctions. *Simultaneous Ascending Auction* (SAA), also known as *Simultaneous Multi Round Auction* (SMRA), has been the privileged mechanism used for spectrum auction since its introduction in 1994 by the US Federal Communications Commission (FCC) for the allocation of wireless spectrum rights. For instance, it has been used in Portugal [1], Germany [9], Italy [14] and the UK [23] to sell 5G licences. SAA is also expected to play a central role in future spectrum allocations, e.g. for 6G licenses. The popularity of SAA is mainly due to the relative simplicity of its rules and the generation of substantial revenue

for the regulator. Both of its creators, Paul Milgrom and Robert Wilson, received the 2020 Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel mainly for their contributions to SAA. Establishing an efficient bidding strategy for SAA is crucial for mobile operators, especially considering the large amount of money involved, e.g. Deutsche Telekom spent 2.17 billion euros in the 5G German SAA. This is the aim of this work.

SAA has a dynamic multi-round auction mechanism where bidders submit their bids simultaneously on all licences each round. It offers the freedom to adjust bids throughout the auction while taking into account the latest information about the likelihood of winning different sets of licences. Hence, a great number of bidding strategies can be applied. Unfortunately, selecting the most efficient one is a difficult task. Indeed, SAA induces a n -player simultaneous move game with incomplete information with a large state space for the solution of which no generic exact game resolution method is known [27].

In addition to the complexities tied to its general game properties, SAA presents a number of complex strategic issues. Its four major strategic issues are the *exposure problem*, the *own price effect*, *budget constraints* and the *eligibility management problem*. The exposure problem corresponds to the situation where a bidder pursues a set of complementary licences but ends up by paying more than its valuation for the ones it actually wins. The own price effect refers to the fact that bidding on a licence inevitably increases its price and, hence, decreases the utility of all bidders willing to acquire it. On the contrary, it is in the interest of all bidders to keep prices as low as possible. Budget constraints correspond to a fix budget that caps the maximum amount that a bidder can bid during an auction and, thus, can hugely impact an auction's outcome. The eligibility management problem is introduced by activity rules which penalise bidders that do not maintain a certain level of bidding activity. At the beginning of the auction, each bidder is given a certain level of eligibility. Each round a bidder fails to satisfy the activity rule, its eligibility is reduced. As bidders are forbidden to bid on sets of licences which exceed their eligibility, managing efficiently one's eligibility during the course of an auction is crucial to obtain a favourable outcome.

In terms of complexity, this bidding game shares many similarities with Real-Time Strategy Games (RTS) like *StarCraft* [5], [24]. Both are n -player simultaneous-move games with incomplete information, large state and action spaces, and present common strategic challenges, such as implicit

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received June 12, 2023

A.Pacaud and A.Bechler are with Orange Labs, France (e-mail: alexandre.pacaud@orange.com, aurelien.bechler@orange.com).

A.Pacaud and M.Coupechoux are with LINC, Telecom Paris, Institut Polytechnique de Paris, France (e-mail: marceau.coupechoux@telecom-paris.fr). The work of M. Coupechoux has been performed at LINC (lincs.fr).

player collaboration and budget management. In this work, we propose the first efficient bidding algorithm which tackles simultaneously the four strategic issues of SAA.

A. Related works

Most works on SAA, such as [12], [13], [21], have focused on its mechanism design, its efficiency and the revenue it generates for the regulator. Only a few works have addressed the bidder's point of view. These studies generally consider one of the two following formats of SAA: its original format [13] and its corresponding clock format defined hereafter. In neither of these formats, an efficient bidding strategy tackling simultaneously its four major strategic issues has yet been proposed. Generally, research has focused on trying to solve one of these strategic issues in specific simplified versions of these formats. Moreover, the solutions proposed can often only be applied to small instances.

As the original format of SAA is generally too complex to draw theoretical guarantees, a simplified clock format of SAA [15] with two types of bidders (*local* and *global*) is often considered. It presents the advantage of being a tractable model where bidders have continuous and differentiable expected utilities. Standard optimisation methods can then be applied to derive an equilibrium.

In the literature, the clock format is mainly used to analyse the exposure problem. Global bidders all have super-additive value functions. Goeree et al [15] consider the case of identical licences for which they compute the optimal dropout level of each global bidder using a Bayesian framework. They extend their work to a larger class of value functions (regional complementarities) but with only two global bidders. By modifying the initial clock format of SAA with a pause system that enables jump bidding, Zheng [36] builds a continuation equilibrium which fully eliminates the exposure problem in the case of two licences and one global bidder. Using a different pause system, Brusco and Lopomo [8] study the effect of binding public budget constraints on the structure of the unique noncollusive equilibria in the case of two licences and two global bidders. They show that such constraints can be a great source of inefficiency.

Regarding the original format of SAA, Wellman et al. [35] propose an algorithm which uses probabilistic predictions of closing prices to tackle exposure. Results seemed promising but were only obtained for a specific class of super-additive value functions.

The own price effect has also been studied in the original format of SAA. In a simple example of SAA with two licences between two bidders having the same public value function, Milgrom [21] describes a collusive equilibrium. This work was then pursued by Brusco and Lopomo [7] who build a collusive equilibrium based on signalling for SAA with two licences between two bidders having super-additive value functions. Similarly to the algorithm built to tackle exposure, Wellman et al. [35] propose an algorithm to tackle the own price effect based on the probabilistic prediction of closing prices when all licences are identical and bidders have subadditive value functions. However, obtained results were unsatisfactory as

they are significantly inferior to a simple demand reduction algorithm.

Regarding budget constraints and the eligibility management problem, little work has been done in the original format of SAA. However, it is commonly accepted that one should gradually reduce its eligibility to avoid being trapped in a vulnerable position if other bidders do not behave as expected [34].

In our previous work [26], we presented a bidding strategy computed by Monte Carlo Tree Search (MCTS) that we applied to a deterministic version of the original format of SAA with complete information. In this paper, we extend our work to simultaneous moves, budget constraints, activity rules, risk-averse rewards and larger instances. All four MCTS phases have been modified.

B. Contributions

In this paper, we consider the original format of SAA with complete information for which we propose the first bidding algorithm, named SMS^α , tackling simultaneously its four major strategic issues. We make the following contributions:

- We model the auction as a n -player simultaneous move game with complete information that we name SAA-c. No specific assumption is made on the bidders' value functions.
- We present an efficient bidding strategy (SMS^α) that tackles simultaneously the *exposure problem*, the *own price effect*, *budget constraints* and the *eligibility management problem* in SAA-c. SMS^α is based on a Simultaneous Move Monte Carlo Tree Search (SM-MCTS) [6], [32]. To the best of knowledge, it is the first algorithm that tackles the four major strategic issues of SAA.
- We introduce a hyperparameter α in SMS^α which allows a bidder to arbitrate between expected utility and risk-aversion.
- We propose a new method based on the convergence of a specific sequence for the prediction of closing prices in SAA-c. This prediction is then used to enhance the expansion and rollout phase of SMS^α .
- Through typical examples taken from the literature and extensive numerical experiments on instances of realistic size, we show that SMS^α outperforms state-of-the-art algorithms by achieving higher expected utility and better tackling the exposure problem and the own price effect in budget and eligibility constrained environments.

The remainder of this paper is organised as follows. In Section II, we define our model SAA-c and provide its game and strategic complexities. We then introduce our performance indicators. In Section III, we present our method for the prediction of closing prices. In Section IV, we present our algorithm SMS^α . In Section V, we show on typical examples taken from the literature the empirical convergence of our method for the prediction of closing prices and that SMS^α tackles efficiently the four major strategic issues. Then, by comparing SMS^α to state-of-the-art algorithms, we show through extensive numerical experiments on instances of realistic size the major increase in performance of our solution.

II. SIMULTANEOUS ASCENDING AUCTION

A. Simultaneous Ascending Auction model with complete information

Simultaneous Ascending Auction (SAA) [13], [21], [35] is one of the most commonly used mechanism design where m indivisible goods are sold via separate and concurrent English auctions between n players. Bidding occurs in multiple rounds. At each round, players submit their bids simultaneously. The player having submitted the highest bid on an item j becomes its temporary winner. If several players have submitted the same highest bid on item j , then the temporary winner is uniformly chosen at random amongst them. The *bid price* of item j , noted P_j , is then set to the highest bid placed on it. The new temporary winners and bid prices are revealed to all players at the end of each round. The auction closes if no new bids have been submitted during a round. The items are then sold at their current bid price to their corresponding temporary winners.

In our model, at the beginning of the auction, the bid price of each item is set to 0. New bids are constrained to $P_j + \varepsilon$ where ε is a fixed bid increment. This reduction of the bidding space is common in the literature on SAA [15], [26], [35]. We make the classical assumption that players won't bid on items that they are currently temporarily winning [26], [35]. Hence, in our model, a winner will always pay a price for an item at most ε above the highest opponent bid.

Activity rules are introduced in SAA to penalise bidders which do not maintain a certain level of bidding activity. In our model, bidders are subject to the following simplified activity rule: the number of items temporarily won plus the number of new bids (also known as *eligibility*) by a bidder can never rise [15], [22]. For instance, suppose a bidder i is temporarily winning a set of items Y and bids on a set of items X at a given round. Its eligibility is defined as $e_i = |Y| + |X|$ and is revealed to all bidders at the end of the round. In the next round, if bidder i is temporarily winning a set of items Y' , it can only bid on a set of items X' of size $|X'| \leq e_i - |Y'|$. Its eligibility is then set to $e'_i = |X'| + |Y'| \leq e_i$. At the beginning of the auction, the eligibility of each player is set to m .

We assume that the value function v_i and budget b_i of each player i are common knowledge [26], [30], [31]. Players are not allowed to bid on a set of items that exceeds their budget. In spectrum auctions, obtaining such knowledge about competitors is considered a very difficult task. Indeed, major mobile operators invest substantial effort to refine as much as possible their estimations. Nevertheless, the complete information framework remains still particularly interesting as it provides a strategic benchmark under ideal conditions. An efficient bidding strategy within this framework can be a significant asset for mobile companies as it facilitates the analysis of possible SAA scenarios based on point-wise estimates of opponents' private information.

This simplified version of SAA induces an n -player simultaneous move game with complete information that we name SAA-c.

B. Budgets, Utility and Value functions

A player i in SAA-c is defined by its budget b_i , its value function v_i and its utility function σ_i . Without loss of generality, b_i and v_i are chosen independently. If the current bid price vector is P , a player i temporarily winning a set of items Y with current eligibility e_i can bid on a set of items X if and only if

$$\begin{cases} |X| + |Y| \leq e_i \\ \sum_{j \in X} (P_j + \varepsilon) \leq b_i - \sum_{j \in Y} P_j \end{cases} \quad (1)$$

At the end of the auction, the utility obtained by player i after winning the set of items X at bid price vector P is equal to its profit, i.e.:

$$\sigma_i(X, P) = v_i(X) - \sum_{j \in X} P_j \quad (2)$$

To respect common reinforcement learning conventions, we will sometimes denote by R^π the random variable corresponding to the utility obtained by playing policy π .

Value functions are assumed to be normalised ($v_i(\emptyset) = 0$), finite and verify the free disposal condition, i.e. for any two sets of goods X and Y such that $X \subset Y$, then $v(X) \leq v(Y)$ [19], [21]. Two disjoint sets X and Y of goods are said to be complements if $v(X + Y) > v(X) + v(Y)$ [35].

C. Extensive form

The standard representation for multi-round games is a tree representation named extensive form [20]. The game tree is a finite rooted directed tree admitting two types of nodes: *decision nodes* and *chance nodes*. At each decision node, a player has the choice between many actions each represented by a directed edge. A chance node has a fixed probabilistic distribution assigned over its outgoing edges. An information set is a set of decision nodes which are indistinguishable for the concerned player at the current position of the game [11]. This means that a player, given its current information, does not know exactly at which decision node it is playing. It only knows that it is playing at one of the decision nodes of the corresponding information set. Games where information sets are not all singletons are known as imperfect information games [29].

We represent the SAA-c game in this form with the decision nodes representing the different states of the game and the chance nodes representing the random draws of temporary winners in case of ties. At each decision node, an outgoing edge represents a set of items on which the concerned player bids if it selects this edge. Each decision node or state is defined by five features: the concerned player, the eligibility vector revealed at the end of the last round, the temporary winner of each item, the current bid price vector and the bids already submitted during the current round. The four first features are common knowledge and the last feature is hidden information for the concerned player. Therefore, all decision nodes which differ only by the last feature belong to the same information set. In Figure 1, we represent an SAA-c game between three players with their information sets and chance nodes.

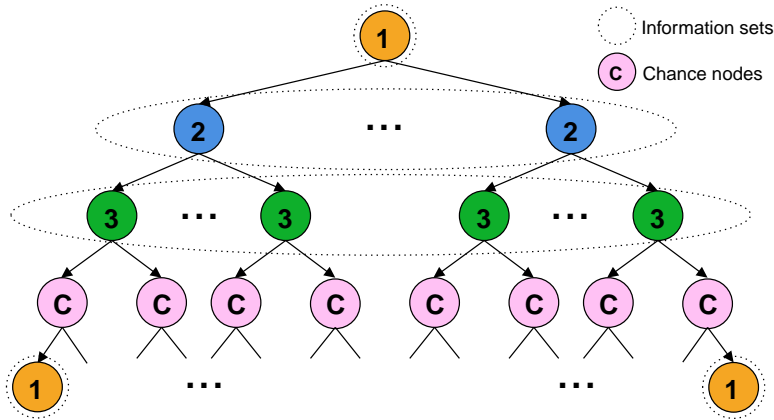


Fig. 1: Extensive form of a three player SAA-c game with information sets and chance nodes

D. Game and strategic complexities

1) *Game complexities*: To highlight the complexity of the SAA-c game, we focus on two metrics: *information set space complexity* and *game tree complexity* [33]. We define the first as the number of different information sets which can be legally reached in the game. It acts as a lower bound of the *state space complexity* [33]. The second corresponds to the number of different paths in its extensive form. It can be shown that the information set space complexity of an SAA-c with R rounds, no activity rule and unlimited budgets is equal to $n(Rn + 1)^m$ and its game tree complexity is lower bounded by $\Omega(2^{m(n-1)R})$.

Example. An SAA for 12 spectrum licences (5G) between 5 telecommunication companies was held in Italy in 2018 and ended after 171 rounds [14]. The number of possible information sets as well as a lower bound of the game tree complexity of the corresponding SAA-c game with no activity rule are respectively 10^{35} and 10^{2470} .

Adding activity rules decreases the game tree complexity as a bidder can no longer bid on a set of items which exceeds its eligibility. However, it increases the information set space complexity as a new feature (eligibility) is added to every information set.

2) *Strategic complexities*: SAA-c game also admits a number of strategic issues. The four major ones are presented below.

- **Exposure**: It is a phenomenon which happens when a player tries to acquire a set of complementary items but ends up by paying too much for the subset it actually wins at the end of the auction. Hence, the player obtains a negative utility. For instance, Table I presents a well-studied example, see e.g. [35], in a 2-item SAA-c game with a bid increment of 1 between two players with unlimited budgets (referred to as Example 1). Player 1 considers both items as perfect substitutes, i.e. it values both items equally and desires to acquire only one of the two, while player 2 considers them as perfect complements, i.e. each item is worthless without the other and desires to acquire both of them. If player 1 is temporarily winning no items and the bid price of the cheapest item is lower than 11,

it should bid on it. Otherwise, it should pass. Hence, if player 2 decides to bid on both items, it will end up exposed as it won't be able to obtain both items for a price inferior to 22. Moreover, if after a few rounds, player 2 decides to give up an item, it will still end up by paying for the other item and, hence, incur a loss.

TABLE I: Example of exposure ($\varepsilon = 1$)

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

- **Own price effect**: Competing on an item causes inevitably the rise of its bid price and, hence, the decrease in utility of all players wishing to acquire it. Thus, players have all a strong interest in maintaining the bid price of all items as low as possible. To avoid this rise, a player can concede items to its opponents hoping that they will not bid on the items it is temporarily winning in exchange. This strategy is known as *demand reduction* [3], [34]. Dividing items between players to avoid this issue is called *collusion* [7]. No communication is allowed between players. In SAA-c, players should be able to use the common knowledge of valuations and budgets to agree on a same fair split of items to tackle this issue without any communication.
- **Budget constraints**: Capping the maximum amount a bidder can spend during an auction can highly impact the auction's outcome. Indeed, it can prevent players from bidding on certain sets of items and be a source of exposure. Moreover, given this information, players can drastically change their bidding strategy. For instance, in the auction presented in Table I, if player 1 and 2 have respectively a fixed budget of 8 and 20, player 2 should bid on both items as this situation no longer presents any risk of exposure.
- **Eligibility management**: Efficient management of its own eligibility is a key factor to ensure a favourable outcome. Bidding on a high number of items to maintain high eligibility induces the own price effect. However, reducing its eligibility to form collusions can trap a bidder

in a vulnerable position if the other bidders do not behave as expected. Hence, a tradeoff must be found.

E. Performance indicators

The natural metric used to measure the performance of a strategy is the *expected utility*. However, given the fact that a specific instance of a spectrum auction (i.e. same frequency bands, same operators, etc ...) is generally only held once and an operator just participates to a few different instances, comparing strategies only on the basis of their *expected utility* is not sufficient. Indeed, given the huge amount of money involved, potential losses due to exposure should also be taken into account. To measure this risk, we decompose the expected utility as follows:

$$\mathbb{E}(R^\pi) = \mathbb{P}(R^\pi \geq 0) \mathbb{E}(R^\pi | R^\pi \geq 0) + \underbrace{\mathbb{P}(R^\pi < 0) \mathbb{E}(R^\pi | R^\pi < 0)}_{\text{Exposure}} \quad (3)$$

where π is a policy and R^π is a random reward obtained by playing π in an SAA-c game. We introduce the term $-\mathbb{P}(R^\pi < 0) \mathbb{E}(R^\pi | R^\pi < 0)$ as a metric of potential exposure which should be minimised. We name it *expected exposure* and estimate it by taking the opposite of all losses incurred by a strategy divided by the number of plays. Moreover, we define the *exposure frequency* as $\mathbb{P}(R^\pi < 0)$. This is estimated by the number of times a strategy incurs a loss divided by the number of plays. To increase its expected utility, one can either try to acquire a set of items with higher value or reduce the price paid for the items won. Hence, to highlight the rise in expected utility due to better tackling the own price effect, we use the *average price paid per item won*. To ensure that a strategy divides efficiently items between bidders and that no item is returned to the auctioneer unnecessarily, we consider the *ratio of items won*.

III. PREDICTING CLOSING PRICES

SMS^α is based on a SM-MCTS whose expansion and rollout phases rely on the following bidding strategy and prediction of closing prices, i.e., an estimation of the price of each item at the end of the auction.

A. Constrained point-price prediction bidding

We start by extending the definition of point-price prediction bidding (*PP*) [35] to budget and eligibility constrained environments.

Definition 3.1. In an SAA-c game with m objects and a current bid price vector P , a point-price prediction bidder with budget b , a current eligibility e , an initial prediction of closing prices P^{init} and a set of temporarily won items Y computes the subset of goods

$$X^* = \underset{\substack{X \subset \{1, \dots, m\} \setminus Y \\ \sum_{j \in X \cup Y} \rho_j(P^{init}, P, Y) \leq b \\ |X| + |Y| \leq e}}{\arg \max} \sigma(X \cup Y, \rho(P^{init}, P, Y)) \quad (4)$$

breaking ties in favour of smaller subsets and lower-numbered goods. It then bids $P_j + \varepsilon$ on all items j belonging to X^* . The function $\rho : (P^{init}, P, Y) \rightarrow \mathbb{R}_+^m$ maps an initial prediction of closing prices, a current bid price vector and a set of items temporarily won to an estimation of closing prices. For any item j , it follows the below update rule:

$$\rho_j(P^{init}, P, Y) = \begin{cases} \max(P_j^{init}, P_j) & \text{if } j \in Y \\ \max(P_j^{init}, P_j + \varepsilon) & \text{otherwise} \end{cases} \quad (5)$$

A point-price prediction bidder only considers sets of items *within budget* b given its prediction of closing prices $\rho(P^{init}, P, Y)$, i.e., only sets of items X such that $\sum_{j \in X \cup Y} \rho_j(P^{init}, P, Y) \leq b$. Moreover, it can only bid on sets of items which does not exceed its eligibility e .

If closing prices are correctly estimated and independent of the bidding strategy, then playing *PP* is optimal for a player. However, in practice, closing prices are usually tightly related to a player's bidding strategy. Playing *PP* with a null prediction of closing prices ($P^{init} = 0$) is known as straightforward bidding (*SB*) [21]. The efficiency of the bidding strategy *PP* highly depends on the accuracy of the initial prediction of closing prices P^{init} . For instance, if P^{init} largely underestimates the actual closing price of each item, then when the current bid price $P \geq P^{init}$ component-wise, playing *PP* with initial prediction P^{init} gives the same strategy as *SB*. However, if P^{init} overestimates too much the actual closing price of each item, then the bidder might stop playing prematurely in order to avoid exposure.

B. Computing an initial prediction of closing prices

Several methods exist in the literature for computing an initial prediction of closing prices P^{init} in budget constrained environments. However, they all seem to present some limitations in SAA-c. For instance, the well known Walrasian price equilibrium [2] does not always exist when preferences exhibit complementarities as it is the case in Example 1. Standard tâtonnement processes, such as the one used to compute *expected price equilibrium* [35], return the same price vector regardless of the auction's specificities (e.g., bid increment ε). The final prediction is then completely independent of the auction mechanism of SAA-c which is problematic. Computing an initial prediction by using only the outcomes of a single strategy profile is relevant only if bidders actually play according to this strategy profile. For instance, simulating SAA-c games where all bidders play *SB* and using the average closing prices as initial prediction is relevant if the actual bidders play *SB*. We propose hereafter a prediction method based on the convergence of a specific sequence which aims at tackling all of these issues.

Conjecture 3.1. Let Γ be an instance of an SAA-c game. Let $f_\Gamma(P)$ be a random variable returning the closing prices of Γ when all bidders play *PP* with initial prediction P . The sequence $p_{t+1} = \frac{1}{t+1} \mathbb{E}[f_\Gamma(p_t)] + (1 - \frac{1}{t+1})p_t$ with p_0 the null vector of prices converges to a unique element p^* .

The fact that f_Γ is a random variable comes from the tie-breaking rule which introduces stochasticity in Γ . By taking

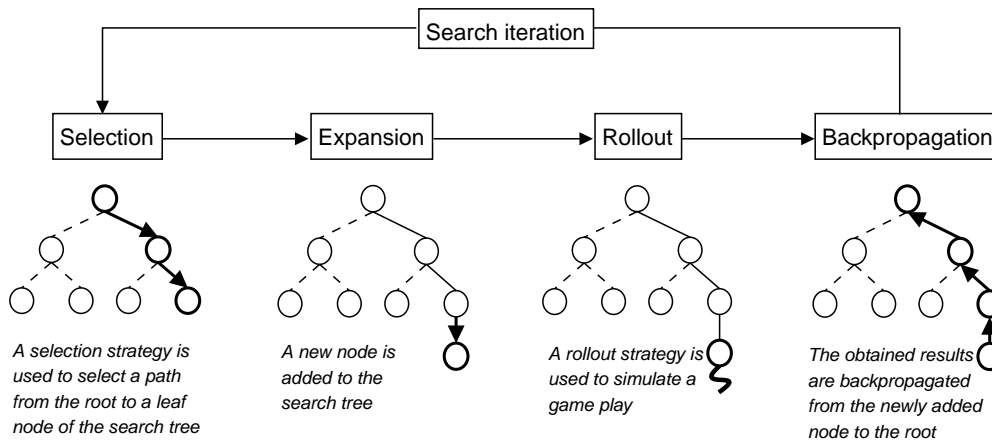


Fig. 2: MCTS scheme

its expectation $\mathbb{E}[f_{\Gamma}(p_t)]$ at each iteration t , we ensure our deterministic sequence p_t to always converge to the same fixed point p^* . Hence, all players using our method share the same prediction of closing prices p^* . In practice, we perform a Monte-Carlo estimation of $\mathbb{E}[f_{\Gamma}(p_t)]$ by simulating many SAA-c games. In small instances, it is possible to obtain a closed-form expression of $\mathbb{E}[f_{\Gamma}(p_t)]$ and, from that, prove the convergence of sequence p_t .

Example. Suppose that both players play *PP* with $P^{init} = p_0$ in Example 1. During the first round, player 1 bids on item 1 and player 2 bids on both items. There is 50% chance that player 1 temporarily wins item 1 and 50% chance that player 2 temporarily wins item 1. If player 1 wins item 1 during the first round, player 2 bids on item 1 during the second round while player 1 passes. In the third round, player 1 bids on item 2 while player 2 passes. In the fourth round, player 2 bids on item 2 while player 1 passes. Hence, the bid price of item 1 (respectively item 2) is odd (respectively even) if temporarily won by player 1. When the bid price $P = (12, 11)$ and both items are temporarily won by player 2, player 1 drops out of the auction as, by definition of *PP*, it prefers smaller subsets of items for a same predicted utility. If player 2 wins item 1 during the first round, the bid price of item 1 (respectively item 2) is even (respectively even) if temporarily won by player 1. The closing price are then $P = (11, 11)$. Therefore, $f_{\Gamma}(p_0)$ has 50% chance of returning $(12, 11)$ and 50% chance of returning $(11, 11)$. Hence, $\mathbb{E}[f_{\Gamma}(p_0)] = (11.5, 11)$. By performing a similar analysis, we can show that $\forall p \in \mathbb{R}^2, \mathbb{E}[f_{\Gamma}(p)] \in [0, 11.5]^2$ and obtain the following closed-form expression for any $p \in [0, 11.5]^2$:

$$\mathbb{E}[f_{\Gamma}(p)] = \begin{cases} (1, 0) & \text{if } p_1 + p_2 \geq 20 \text{ and } p_1 \leq p_2 \\ (0, 1) & \text{if } p_1 + p_2 \geq 20 \text{ and } p_1 > p_2 \\ (11.5, 11) & \text{if } p_1 + p_2 < 20 \text{ and } p_1 \leq p_2 \\ (11, 11.5) & \text{if } p_1 + p_2 < 20 \text{ and } p_1 > p_2 \end{cases} \quad (6)$$

From there, it is easy to show that sequence p_t converges to $p^* = (10, 10)$ in Example 1.

The general proof of the conjecture is left for future work. Computing an initial prediction of closing prices as above

has mainly three advantages compared to other methods in the literature. (1) We observe that this sequence converges in all undertaken SAA-c game instances. (2) This method takes into account the auction's mechanism through f_{Γ} . (3) This prediction of closing price is not based only on the outcomes of a single specific strategy profile. Indeed, depending on the value of p_t , different strategy profiles are used across iterations. At a fixed iteration t , a single strategy profile is used to compute $\mathbb{E}[f_{\Gamma}(p_t)]$ as the strategy returned by *PP* only depends on its initial prediction $P^{init} = p_t$.

IV. SM-MCTS BIDDING STRATEGY

A. Brief presentation of MCTS

Given the large state space and game tree complexities, it is practically impossible to explore the SAA-c game tree exhaustively as soon as we depart from very small instances. Thus, only a small portion of the game tree, called the search tree, can be explored. MCTS is a search technique that iteratively builds a search tree using simulations through a process named search iteration (see Figure 2). Each search iteration is divided into four steps. (1) The *selection phase* selects a path from the root to a leaf node of the search tree. (2) The *expansion phase* chooses one or more children to be added to the search tree from the selected leaf node according to the available actions. (3) The *simulation phase* simulates the outcome of the game from the newly added node. (4) The *backpropagation phase* propagates backwards the outcome of the game from the newly added node to the root in order to update the diverse statistics stored in each selected node of the search tree. This process is repeated until some predefined computational budget (time, memory, iteration constraint) is reached. Before running SMS^{α} , we compute our initial prediction of closing prices p^* as presented in Section III-B.

B. Risk-averse rewards

Ideally, one would want to maximise its expected profit while minimising its risk of exposure. However, achieving both objectives simultaneously may present potential conflicts.

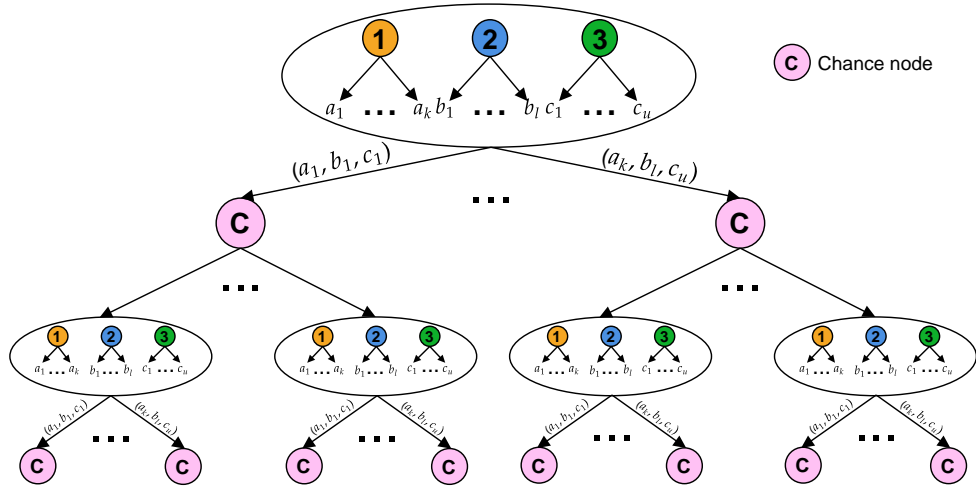


Fig. 3: SM-MCTS tree structure with explicit chance nodes for an SAA-c game with 3 players

Indeed, taking risks can either be highly beneficial or lead to exposure depending on how the other players react. To do so, we introduce a new risk-averse reward incorporating both targets. For any strategy π , we define:

$$R_{\alpha}^{\pi} = (1 + \alpha \mathbb{1}_{R^{\pi} < 0}) R^{\pi} \quad (7)$$

where α is a hyperparameter which controls the risk aversion of $SM S^{\alpha}$. Note that

$$\mathbb{E}(R_{\alpha}^{\pi}) = \mathbb{E}(R^{\pi}) + \alpha \mathbb{P}(R^{\pi} < 0) \mathbb{E}(R^{\pi} | R^{\pi} < 0) \quad (8)$$

where $\mathbb{P}(R^{\pi} < 0) \mathbb{E}(R^{\pi} | R^{\pi} < 0)$ is the term corresponding to the losses induced by exposure in Equation (3). Moreover, we define for any vector of price P and any set of items X , $\sigma^{\alpha}(X, P) = (1 + \alpha \mathbb{1}_{\sigma(X, P) < 0}) \sigma(X, P)$ which is a modified utility taking into account both of our objectives. We name it risk-averse utility. When the term utility is not explicitly specified as risk-averse, we refer to the utility described in Equation (2).

The use of a linear scalarization function is a classical approach in multi-objective optimisation, multi-objective reinforcement learning [4], constrained MDP [17] or POMDP [18].

C. Search tree structure

For the selection phase of $SM S^{\alpha}$, we use a Simultaneous Move MCTS (SM-MCTS) [6], [32] (Figure 3). At each selection step, we select an n -tuple where each index i corresponds to the action maximising the selection index of player i given only its information set. By doing so, bids are selected simultaneously and independently. Each selection step corresponds to a complete bidding round of SAA-c. Hence, the depth of our search tree corresponds to how many rounds ahead $SM S^{\alpha}$ can foresee. The search tree nodes are defined by the eligibility of each bidder, the temporary winner and current bid price of each item. The vertices correspond to players' joint actions. Chance nodes are explicitly included in the search tree to break ties. There are three main advantages of using an SM-MCTS instead of an MCTS applied to a serialised game

tree, i.e. turning SAA-c into a purely sequential game with perfect information. The first advantage is that it maintains the simultaneous move nature of SAA-c. The second advantage is that it does not increase the number of information sets making our learning process more efficient. The third advantage is that the number of selection steps to complete a bidding round of SAA-c is reduced from n to 1. Thus, the number of players n is no longer a burden for planning a bidding strategy over many rounds.

D. Selection

At each selection step, players are asked to bid on the set of items which maximises their selection index. The selection phases ends when a terminal state of the SAA-c game or a non-expanded node, i.e. configuration of temporary winners, bid prices and eligibilities not yet added to the search tree, is reached. Our selection index is a direct application of the *Upper Confidence bound applied to Trees* (UCT) [16] to risk-averse rewards. Unlike usual applications of UCT, the size of the risk-averse reward support is unknown so we proceed to an online estimation of it. Each player i chooses to bid on the set of items x_i with highest score q_{x_i} at information set I_i :

$$q_{x_i} = \frac{r_{x_i}^{\alpha}}{n_{x_i}} + \max(c_{x_i}^{\alpha} - a_{x_i}^{\alpha}, \varepsilon) \sqrt{\frac{2 \log(\sum_{x'_i} n_{x'_i})}{n_{x_i}}} \quad (9)$$

where $r_{x_i}^{\alpha}$ is the sum of risk-averse rewards obtained after bidding on x_i at I_i , n_{x_i} is the number of times player i has bidden on x_i at I_i , ε is the bid increment, $a_{x_i}^{\alpha}$ is the estimated lower bound and $c_{x_i}^{\alpha}$ is the estimated higher bound of the risk-averse reward support when bidding on x_i at I_i . Thus, $\max(c_{x_i}^{\alpha} - a_{x_i}^{\alpha}, \varepsilon)$ acts like the size of the risk-averse reward support when bidding on x_i at I_i .

E. Expansion

The high branching factor due to the exponential growth of the game tree's width with the number of items m prevents in-depth inspection of promising branches. Thus, it is necessary

to reduce the action space at each information set of the search tree [29]. To do so, each time a non-expanded node is added to the search tree, we select a maximum number N_{act} of promising actions per information set. In order to obtain a relevant estimation of the risk of exposure when playing a specific move, it is important that we select individual actions and not joint actions. If a too low value is chosen for N_{act} , then a bidder can only play a very limited number of moves which affects its coordination with other bidders as well as its possibilities to avoid exposure. Thus, given thinking time constraints, a tradeoff between in-depth inspection and a wide range of possible moves must be found.

Passing its turn without bidding on any item is always included in the N_{act} selected actions. This enables SMS^α to obtain shallow terminal nodes in its search tree which correspond to potential collusions between bidders and, thus, reduces the own price effect. The remaining $N_{act} - 1$ actions correspond to the moves leading to the $N_{act} - 1$ highest predicted utilities in strategy PP with initial prediction p^* . Hence, the actions are chosen independently for each player, ensuring that the choices made for one player have no influence on the selections made by another. More formally, for each player i at information set I_i temporarily winning set of items Y_i with eligibility e_i , the action of bidding on set of items X_i is selected if $\sigma_i^\alpha(Y_i \cup X_i, \rho(p^*, P, Y_i))$ is one of the $N_{act} - 1$ highest values with P the current bid price. Only sets of items X_i verifying $\sum_{j \in X_i \cup Y_i} \rho_j(p^*, P, Y_i) \leq b_i$ and $|X_i| + |Y_i| \leq e_i$ are considered. Statistics for each action are then initialised as follows:

$$\begin{aligned} * r_{x_i}^\alpha &\leftarrow 0 \\ * n_{x_i} &\leftarrow 0 \\ * a_{x_i}^\alpha &\leftarrow +\infty \\ * c_{x_i}^\alpha &\leftarrow -\infty \end{aligned}$$

F. Rollout

From the newly added node, an SAA-c game is simulated until the game ends. Players are asked to bid at each round of the rollout. The default strategy is usually to bid on a random set of items. However, it leads to absurd outcomes in this case with very high prices as player rarely all pass. Therefore, we propose an alternative approach. At the beginning of each rollout phase, we set $p_i^* = p^* + \eta_i$ with $\eta_i \sim U([- \varepsilon, \varepsilon]^m)$. Each player i then plays PP with initial prediction of closing prices p_i^* during the entire rollout. Noise is added to our initial prediction p^* to diversify players' bidding strategy and, hence, improve the quality of our sampling. To ensure that the prediction p_i^* is always coherent with p^* in all auctions, η_i is chosen in $[- \varepsilon, \varepsilon]^m$. Hence, no absurd prediction can be drawn which could negatively impact our sampling. At the end of the rollout, an n -tuple is returned corresponding to the risk-averse utility obtained by each player.

G. Backpropagation

The results obtained during the rollout phase are propagated backwards to update the statistics of the selected nodes. Let V_i^α be the risk-averse utility obtained by player i at the end of the rollout. Let x_i be the set of items on which player i

bidded at information state I_i for one of the selected nodes. The statistics stored for I_i are updated as follows:

$$\begin{aligned} * r_{x_i}^\alpha &\leftarrow r_{x_i}^\alpha + V_i^\alpha \\ * n_{x_i} &\leftarrow n_{x_i} + 1 \\ * a_{x_i}^\alpha &\leftarrow \min(a_{x_i}^\alpha, V_i^\alpha) \\ * c_{x_i}^\alpha &\leftarrow \max(c_{x_i}^\alpha, V_i^\alpha) \end{aligned}$$

H. Transposition table

Transposition tables are a common search enhancement used to considerably reduce the size of the search tree and improve performance of MCTS within the same computational budget [10]. By using such tables, we prevent the expansion of redundant nodes in our search tree and share the same statistics between transposed information states. This results in a significant improvement in performance of SMS^α for the same amount of thinking time.

To identify each information set in the search tree, our hash function is based on two functions h_1 and h_2 . The first returns a different integer for each combination of bid prices and allocations. The second returns a different integer for each eligibility vector. Hence, our hash function assigns a unique value to each information set in the search tree. More precisely, due to computational constraints, we can only assign a unique value for every node in the search tree with a depth lower than R_{max} . R_{max} is a hyperparameter corresponding to an upper bound of the maximal depth (or rounds) in the final search tree. An example of function h_1 assigning a different integer for each combination of bid prices and allocations in a search tree of maximal depth R_{max} is given in Algorithm 1. It uses as inputs the bid price vector P^0 at the root of the search tree, the bid price vector P and the temporary winner A_j of each item j at a given node. If $A_j = 0$, then item j is temporarily allocated to the auctioneer.

In practice, given the thinking time constraints in our experimental results, choosing $R_{max} = 10$ is more than sufficient to guarantee a final search tree with maximal depth lower than R_{max} . Hence, our hash function acts as a perfect hash function as no type-1 error or type-2 error occurs [37].

Algorithm 1 Example of function h_1

Inputs Game: n, m, ε

Inputs Root Node: Bid price vector P^0

Inputs Node: Bid price vector P , Allocation vector A

Hyperparameter: R_{max}

$h = 0$

$step = R_{max} \times n$

for $j = 1, 2, \dots, m$ **do**

if $A_j > 0$ **then** $h += (R_{max} \times (A_j - 1) + \frac{P_j - P_j^0}{\varepsilon})step^{j-1}$

end if

end for

return h

I. Final move selection

The final move which is returned by SMS^α is the action which maximises the player's expected risk-averse reward at the root node. More formally, SMS^α returns $\arg \max_{x_i} \frac{r_{x_i}^\alpha}{n_{x_i}}$ for player i .

V. EXPERIMENTS

In this section, we start by analysing the convergence rates of sequence p_t , notably through Example 1. Then, we show that our algorithm SMS^α largely outperforms state-of-the-art existing bidding algorithms in SAA-c, mainly by tackling own price effect and exposure more efficiently. This is first shown through typical examples taken from the literature and, then, through extensive experiments on instances of realistic size. We compare SMS^α to the following four strategies:

- MS^λ : An MCTS algorithm described in [26] which relies on two risk-aversion hyperparameters λ^r and λ^o .
- EPE: A PP strategy using expected price equilibrium [35] as initial prediction.
- SCPD: A distribution price prediction strategy using self-confirming price distribution [35] as initial distribution prediction.
- SB: Straightforward bidding [21].

The four strategies MS^λ , EPE, SCPD and SB initially rely on the definition of PP for unconstrained environments [35]. We extend them to budget and eligibility constrained environments in the same way as it is done in Definition 3.1. In all experiments, none of the bidders are aware of their opponents' strategy.

Each algorithm is given respectively 150 seconds of thinking time. Initial prediction of closing prices are done offline before the auction starts and, therefore, are excluded from the thinking time. This step usually takes a few minutes. All experiments are run on a server consisting of Intel@Xeon@E5-2699 v4 2.2GHz processors. In all upcoming experiments, the hyperparameter α of SMS^α takes the value 7 and the risk-aversion hyperparameters λ^r and λ^o of MS^λ both take the value 0.025. The maximum number of expanded actions per information set N_{act} of SMS^α is set to 20. The choice of the hyperparameters is motivated in Section V-D.

A. Convergence of sequence p_t

One of the main advantages of using our method to compute an initial prediction is the convergence of sequence p_t . Even though this convergence has only been observed and not proven, it is possible to derive rates of convergence in small instances. For instance, in Example 1, it can be shown that $\forall t \geq 1$, p_t belongs to the diamond defined by the points $(10 - \frac{10}{t}, 10 - \frac{9}{t})$, $(10 - \frac{9}{t}, 10 - \frac{10}{t})$, $(10 + \frac{7}{4t}, 10 + \frac{3}{4t})$ and $(10 + \frac{3}{4t}, 10 + \frac{7}{4t})$ which converges to $p^* = (10, 10)$. We represent in Figure 4 the sequence p_t^1 with its corresponding lower bound and upper bound.

In larger instances, we observe similar rates of convergence. However, computing such bounds seems unrealistic as obtaining a closed-form expression of $\mathbb{E}[f_\Gamma(p_t)]$ seems untractable.

B. Test experiments

One of the greatest advantages that MCTS methods have over other bidding algorithms is the capacity to judge pertinently in which situations adopting a demand reduction strategy is more beneficial. Indeed, through the use of its search tree, an MCTS method is capable of determining if

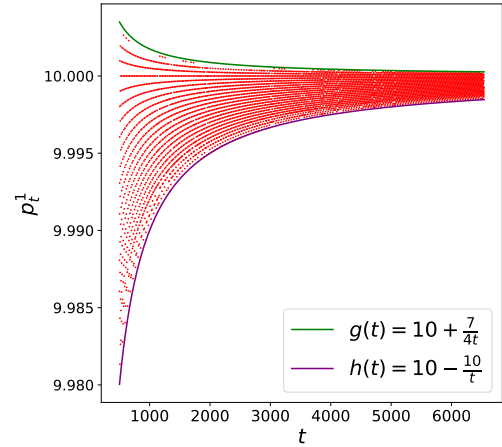


Fig. 4: Convergence of sequence p_t^1 with its respective upper bound $g(t)$ and lower bound $h(t)$ in Example 1.

it is more profitable to concede items to its opponents to keep prices low or to bid greedily. To highlight this feature, we propose the following experiment in a 2-item auction between two players with additive value functions. Each player values each item at $l = 10$. Player 1 has a budget $b_1 \geq 20$. Given that, the optimal strategy for player 2 is to bid on the cheapest item if it is not temporarily winning any item. Otherwise, it should pass. The optimal strategy for player 1 fully depends on its opponent's budget b_2 . For an infinitesimal bid increment ε ,

- If $b_2 \leq \frac{l}{2}$, player 1's optimal strategy is to play straightforwardly and it obtains an expected utility of $l - 2b_2$.
- If $b_2 \geq \frac{l}{2}$, player 1's should adopt a demand reduction strategy and it obtains an expected utility of l .

We plot in Figure 5 the expected utility $\mathbb{E}(\sigma_1)$ of player 1 for each strategy given player 2's budget b_2 . The three algorithms SB, EPE, SCPD always suggest to player 1 to bid greedily and never propose a demand reduction strategy even when it is highly profitable ($b_2 > \frac{l}{2}$). However, both MCTS methods perfectly adopt the appropriate strategy. This experiment highlights the fact that SMS^α selects the most profitable strategy and tackles own price effect, at least in simple budget and eligibility constrained environments.

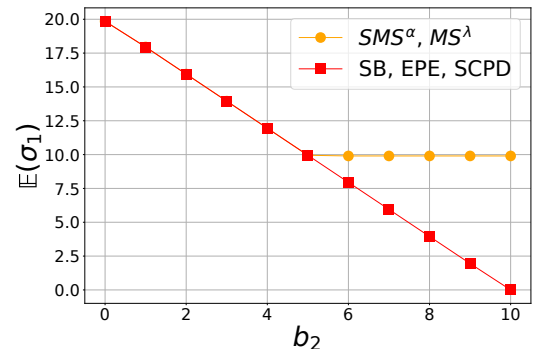


Fig. 5: Evolution of player 1's expected utility $\mathbb{E}(\sigma_1)$ depending on strategy versus player 2's budget b_2 given that player 2 plays optimally ($\varepsilon = 0.1$).

Furthermore, SMS^α is capable of avoiding obvious exposure. To highlight this feature, we use the SAA-c game presented in Example 1 where player 2's budget $b_2 = 16$. The optimal strategy for player 1 is to play straightforwardly. Similarly to the preceding experiment, the optimal strategy for player 2 fully depends on its opponent's budget b_1 .

- If $b_1 < 8$, player 2's optimal strategy is to play straightforwardly.
- If $b_1 \geq 8$, player 2's optimal strategy is to drop out of the auction to avoid exposure.

We plot in Figure 6 the expected utility $\mathbb{E}(\sigma_2)$ of player 2 for each strategy given player 1's budget b_1 . The two algorithms SCPD and SB always suggest to player 2 to bid straightforwardly leading player 2 to exposure when $b_1 \geq 8$. MS^λ never leads player 2 to exposure. However, it suggests to drop out prematurely of the auction in some situations with no risk of exposure and, hence, incurs a loss of easy profit ($b_1 = 7$). SMS^α and EPE perfectly adopt the optimal strategy. This experiment highlights the fact that SMS^α perfectly adopts the most profitable strategy and tackles efficiently exposure, at least in simple budget and eligibility constrained environments.

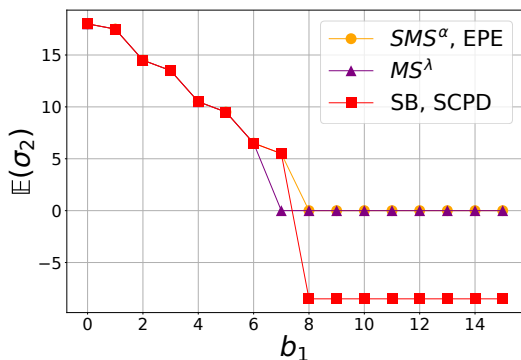


Fig. 6: Evolution of player 2's expected utility $\mathbb{E}(\sigma_2)$ depending on strategy versus player 1's budget b_1 given that player 1 plays optimally in Example 1 ($\varepsilon = 1$).

C. Extensive experiments

In this section, we study instances of realistic size with $n = 4$ and $m = 11$. Each experimental result has been run on 1000 different SAA-c instances. With the exception of [26], all experimental results in the literature are obtained for specific settings of SAA, i.e., using value functions with some specific property such as superadditivity [15], [27], [35]. Hence, it is difficult to conclude on the effectiveness of a method in more generic settings. Therefore, we propose a more general approach to generate value functions by making no additional assumption on its form. Budgets are drawn randomly.

Setting. Let Γ be an instance of SAA-c with n bidders, m items and bid increment ε . Each player i has a budget $b_i \sim U([b_{min}, b_{max}])$ with U the uniform distribution. Its value function v_i is built as follows: $v_i(\emptyset) = 0$ and, for any set of goods X ,

$$v_i(X) \sim U([\max_{j \in X} v_i(X \setminus \{j\}), V + \max_{j \in X} v_i(X \setminus \{j\}) + v_i(\{j\})]) \quad (10)$$

Drawing value functions through a uniform distribution is widely used for creating auction instances [27], [35]. In our setting, the lower-bound ensures that v_i respects the *free disposal* [21] condition. The upper-bound caps the maximum surplus of complementarity possibly gained by adding an item j to the set of goods $X \setminus \{j\}$ by V . As valuations are always finite, any value function can be represented by our setting for a sufficiently large V . For $V = 0$, only subadditive functions are considered. For $V > 0$, goods can either be complements or substitutes. In our experimental results, value functions and budgets are generated for each instance as above with $\varepsilon = 1$, $b_{min} = 10$, $b_{max} = 40$ and $V = 5$.

In the upcoming analysis, the average price paid per item won, the ratio of items won, the expected exposure and the exposure frequency are obtained by confronting a strategy A to a strategy B . To facilitate our study, each measure of A against B is obtained by averaging the results obtained for the three following strategy profiles: (A, B, B, B) , (A, A, B, B) and (A, A, A, B) . For instance, if $A = SMS^\alpha$ and $B = SB$, the average price paid per item won by SMS^α in these three strategy profiles is respectively: 5.96, 5.46 and 4.62. Hence, the average price payer per item won by SMS^α against SB is 5.35.

1) *Expected Utility:* To facilitate our analysis, we study the normal form game in expected utility where each player has the choice between playing SMS^α or another strategy A . The same empirical game analysis approach was employed by Wellman et al. in [35]. More precisely, we map each strategy profile to the estimated expected utility obtained by each player in the 1000 SAA-c instances. The four resulting empirical games for each possible strategy A are given in Figure 7.

For example, in Figure 7b, if all bidders play EPE, each bidder obtains an expected utility of 10.8. In the case of three EPE bidders and one SMS^α bidder, the SMS^α bidder obtains an expected utility of 21.5. Hence, if all bidders play EPE, a bidder can double its expected utility by switching to SMS^α . Therefore, deviating to SMS^α is profitable if all bidders play EPE. This is also the case for the three other possible deviations in Figure 7b. Hence, in the empirical game where bidders have the choice between playing SMS^α or EPE, each bidder has interest in playing SMS^α . We can clearly see that all deviations to SMS^α are also strictly profitable in the three other empirical games. Hence, in each empirical game, a bidder should play SMS^α to maximise its expected utility. Therefore, the strategy profile $(SMS^\alpha, SMS^\alpha, SMS^\alpha, SMS^\alpha)$ is a Nash equilibrium of the normal-form SAA-c game in expected utility with strategy set $\{SMS^\alpha, MS^\lambda, EPE, SCPD, SB\}$.

Moreover, the strategy profile where all bidders play SMS^α has a substantially higher expected utility than any other strategy profile where all bidders play the same strategy. This is mainly due to the fact that SMS^α tackles efficiently the own price effect. For instance, in Figure 7, the expected utility of the strategy profile where all bidders play SMS^α is respectively 1.13, 1.68 and 3.94 times higher than the ones where all bidders play EPE, MS^λ and SCPD.

The fact that the expected utility obtained by the strategy

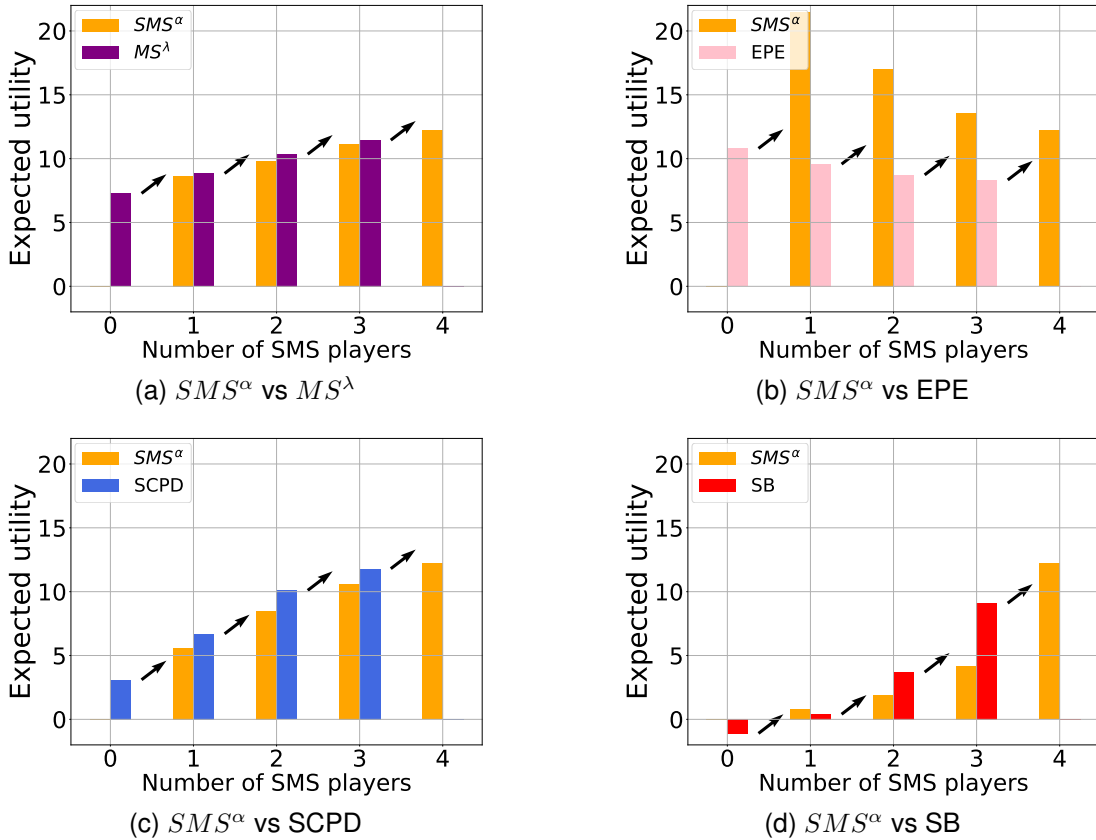


Fig. 7: Normal-form expected utility for an SAA-c game with five strategies

profile where all bidders play EPE is relatively close to the one where all bidders play SMS^α can be explained as follows. To compute their expected price equilibrium as initial prediction of closing prices, all EPE bidders in our experiments share the same initial price vector and adjustment parameter in their tâtonnement process. This tâtonnement process is independent of the auction's mechanism and only relies on the estimated valuations of the players. Hence, as SAA-c is a game with complete information, all EPE bidders share the same initial prediction of closing prices and can therefore split up the items between them more or less efficiently.

Not all algorithms have the ability of achieving good coordination between bidders. For instance, the strategy profile where all bidders play SB leads to a negative expected utility. Hence, in this specific case, bidders would have preferred not to participate in the auction. This highlights the fact that playing SB is a very risky strategy and mainly leads to exposure.

We believe that SMS^α outperforms the four other strategies for the three following reasons:

- it better judges when to perform demand reduction or to bid greedily.
- it better tackles the own price effect without putting itself in a vulnerable position because of eligibility constraints.
- it better tackles the exposure problem.

2) *Own Price Effect*: To analyse the own price effect, we plot in Figure 8a the average price paid per item won and in Figure 8b the ratio of items won by each strategy A against

every strategy B displayed on the x-axis. For instance, if $A = SMS^\alpha$ and $B = EPE$, the average price paid per item won and the ratio of items won by SMS^α against EPE are respectively 1.53 and 0.31. They both correspond respectively to the orange bar above index EPE on the x-axis. If $A = EPE$ and $B = SMS^\alpha$, then the average price paid per item won and the ratio of items won by EPE against SMS^α are respectively 2.53 and 0.18. They both correspond respectively to the pink bar above index SMS^α on the x-axis.

In Figure 8a, we can clearly see that SMS^α acquires items at a lower price in average than the other strategies against SMS^α , SCPD and SB. For instance, SMS^α spends 13.3%, 17.1%, 44.9% and 49.8% less per item won against SCPD than MS^λ , EPE, SCPD and SB respectively. Moreover, against MS^λ and EPE, only EPE spends slightly less than SMS^α per item won.

Regarding strategy profiles where all bidders play the same strategy, the one corresponding to SMS^α has an average price paid per item won 1.70, 2.35 and 2.98 times lower than MS^λ , SCPD and SB respectively. Moreover, by looking at Figure 8b, we can see that all items are allocated when all bidders play SMS^α . Being capable of splitting up all items at a relatively low price explains why the expected utility of the strategy profile where all bidders play SMS^α is substantially higher than the ones where all bidders play a same other strategy. Only obtaining items at a low price is not sufficient. For instance, when all bidders play EPE, the average price paid

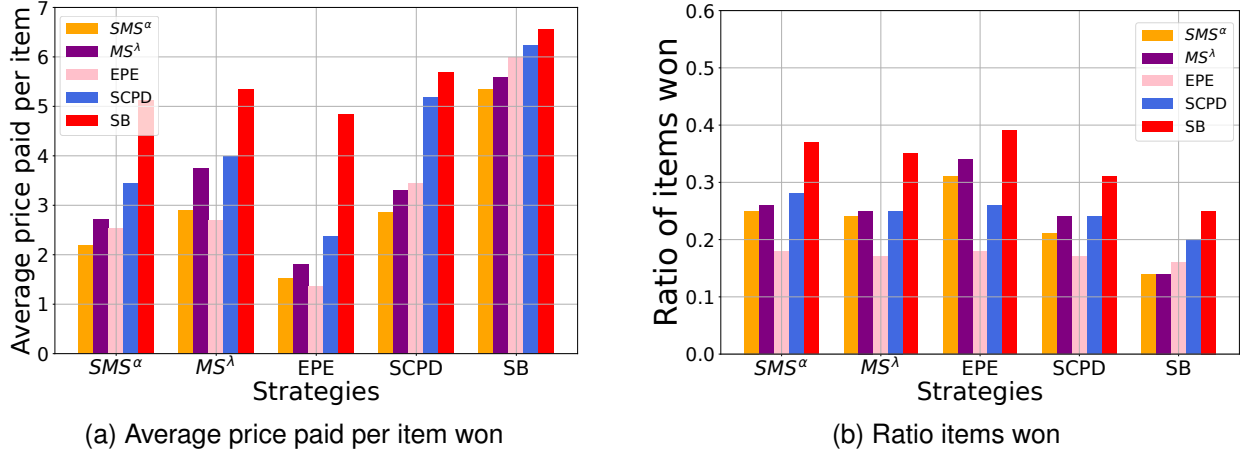


Fig. 8: Own price effect analysis for an SAA-c game with five strategies

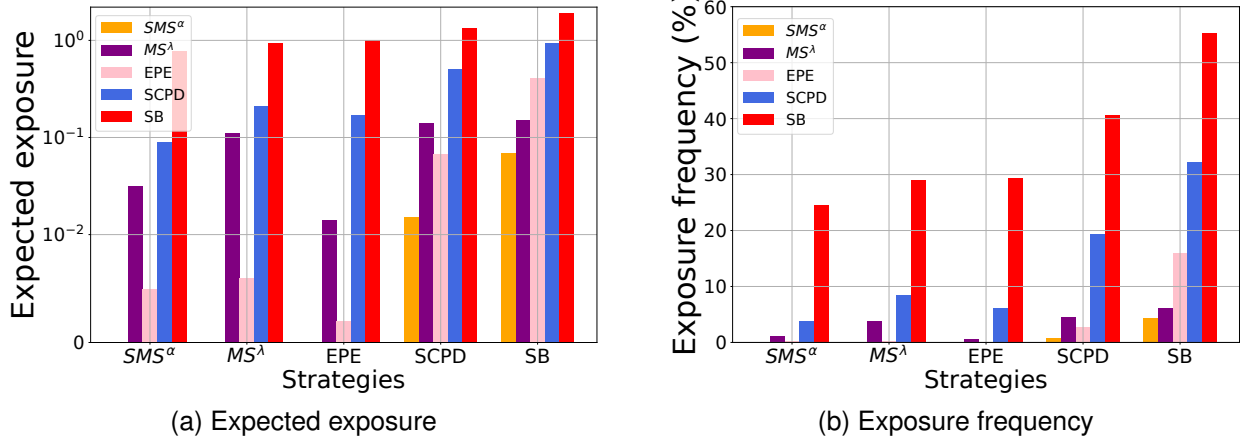


Fig. 9: Exposure analysis for an SAA-c game with five strategies

per item won is 1.6 times lower than when all bidders play SMS^α . However, only 72% of all items are allocated. Hence, this strategy profile achieves a lower expected utility than if all bidders had played SMS^α .

Moreover, the fact that the average price per item won when all bidders play EPE is relatively close to ε raises an important strategic issue. Indeed, to obtain such low price, EPE bidders drastically reduce their eligibility during the first round without considering the fact that they might end up in a vulnerable position. Hence, an EPE bidder can easily be deceived. This explains why a bidder doubles its expected utility if it decides to play SMS^α instead of EPE when all its opponents are playing EPE in Figure 7b. After the first round, SMS^α easily takes advantage of the weak position of its opponents. By gradually decreasing its eligibility, a SMS^α bidder tackles efficiently the own price effect and avoids putting itself in vulnerable positions.

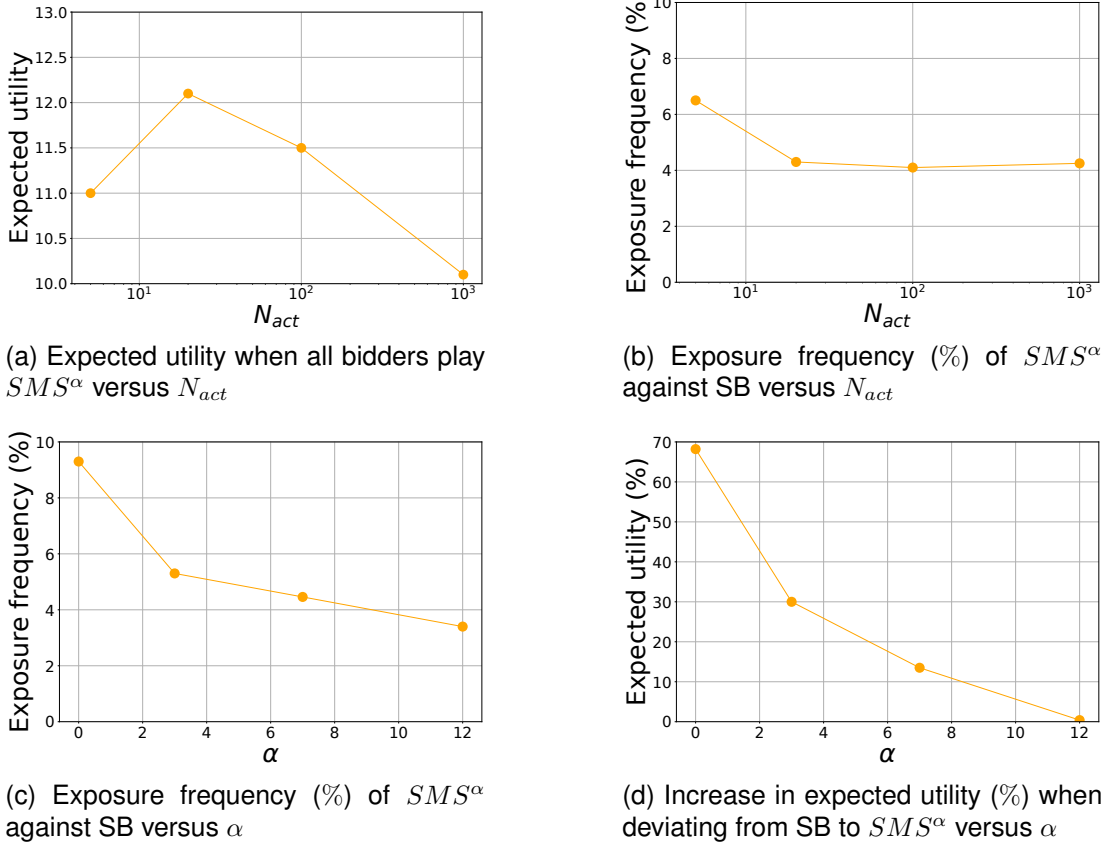
3) *Exposure*: To analyse exposure, we plot in Figure 9a the expected exposure of each strategy A against every strategy B displayed on the x-axis. Similarly, we plot in Figure 9b the exposure frequency of each strategy A against every strategy B displayed on the x-axis. For instance, if $A = SMS^\alpha$ and

$B = SB$, the expected exposure and exposure frequency of SMS^α against SB are respectively 0.07 and 4.4%. They both correspond respectively to the orange bar above index SB on the x-axis in Figure 9a and Figure 9b.

Firstly, in the situation where all bidders decide to play the same strategy, SMS^α has the remarkable property of never leading to exposure. This is not the case for the four other strategies. Secondly, SMS^α is the only strategy which never suffers from exposure against MS^λ and EPE. Thirdly, even against SCPD and SB, SMS^α is rarely exposed. It has the lowest expected exposure and exposure frequency. For instance, SMS^α induces 9.3, 4.5, 34 and 90 times less expected exposure against SCPD than MS^λ , EPE, SCPD and SB respectively. Moreover, regarding exposure frequency, by playing SMS^α a bidder has 6.6, 4, 27.6 and 58.1 times less chance of ending up exposed against SCPD than MS^λ , EPE, SCPD and SB respectively.

Hence, not only does SMS^α achieve higher expected utility than state-of-the-art algorithms but it also takes less risks.

We exclusively focus on empirical games of size ($n = 4$, $m = 11$) with only two distinct strategies. However, SMS^α also remains highly efficient for varied instance sizes and when

Fig. 10: Impact of N_{act} and α on SMS^α

bidders have access to a wider range of bidding strategies. For instance, it still achieves higher expected utility than state-of-the-art algorithms, notably by better tackling the four major strategic issues of SAA. We omit the results here due to space constraints; additional experiments are provided in [25].

D. Influence and selection of N_{act} and α

The values of hyperparameters N_{act} and α are obtained through grid search. We propose hereafter an analysis of the impact of both hyperparameters on the performance of SMS^α and explain our choice of setting N_{act} to 20 and α to 7.

First, we compare SMS^α when $\alpha = 7$ for the following values of N_{act} : 5, 20, 100 and 1000. We plot the expected utility when all bidders play SMS^α versus N_{act} in Figure 10a. We can see that the expected utility drastically increases between 5 and 20 and, then, slowly decreases between 20 and 1000. The fact that the expected utility seems to be a unimodal function presenting a maximum for $N_{act} = 20$ illustrates perfectly the tradeoff to be made between in-depth inspection and a wide range of possible moves per player. Moreover, in Figure 10b, we represent the exposure frequency of SMS^α against SB versus N_{act} . We can see that between 5 and 20 the exposure frequency drops by 50% whereas it remains relatively constant afterwards. This highlights the fact that having an overly restricted range of moves reduces the possibilities to avoid exposure. Hence, in order to maximise

coordination between bidders and minimise exposure, we set N_{act} to 20.

Secondly, we compare SMS^α with $N_{act} = 20$ for the following values of α : 0, 3, 7 and 12. We represent the exposure frequency of SMS^α against SB versus α in Figure 10c. We clearly see that the exposure frequency decreases when α grows. Thus, by increasing α , SMS^α tackles better the exposure problem. In Figure 10d, we plot the increase in expected utility when deviating from SB to SMS^α when two opponents play SB and the last opponent plays SMS^α . We observe that the profitability of the deviation considerably decreases when α grows and nearly reaches 0 when $\alpha = 12$. Hence, the reduction of exposure when α increases is accompanied by a decrease in profitability of deviating to SMS^α . The hyperparameter α thus allows the bidder to arbitrate between expected utility and risk-aversion. In order to maintain all deviations towards SMS^α profitable in each empirical game illustrated in Figure 7 and minimise the risk of exposure, we set α to 7.

It is also important to note that increasing α reduces the own price effect and enables a better coordination between bidders. This is a natural effect of risk-aversion where a bidder tends to avoid a rise in price. For instance, the average price per item won all bidders play SMS^α is respectively 3.1 and 2 for α equal to 0 and 12. This explains why the expected utility when all bidders play SMS^{12} is 1.36 times greater than the expected utility when all bidders play SMS^0 .

VI. CONCLUSIONS AND FUTURE WORK

This paper introduces the first efficient bidding strategy that tackles simultaneously the *exposure problem*, the *own price effect*, *budget constraints* and the *eligibility management problem* in a simplified version of SAA (SAA-c). It is a SM-MCTS whose expansion and rollout phase relies on a new method for the prediction of closing prices. By introducing hyperparameter α , we give the freedom to bidders to arbitrate between expected profit and risk-aversion. Our solution SMS^α largely outperforms state-of-the-art algorithms on instances of realistic size in generic settings.

For future works, it would be interesting to see if SMS^α can be improved by modifying some of its search iteration phases. For instance, another selection index could be used such as *EXP3* or *Regret Matching* [6]. Moreover, following the successful model of AlphaZero [28], exploring approaches which combine MCTS with neural networks might also be a promising direction.

An important future work is the design of an efficient bidding strategy for SAA with incomplete information. To address this challenge, one could investigate determinization approaches. These solutions are based on complete information schemes like SMS^α , which thus appears as an important building block for future advancements.

Finally, although SMS^α was initially designed for bidders, it could also be used by mechanism designers to verify the impact of new rules on the efficiency and revenue of SAA.

REFERENCES

- [1] ANACOM. Anacom commences procedure to amend the 5g auction regulation in order to prohibit use of smaller bidding increments. <https://www.anacom.pt/render.jsp?contentId=1695741>, 2021.
- [2] Kenneth Joseph Arrow, Frank Hahn, et al. General competitive analysis. 1971.
- [3] Lawrence M Ausubel, Peter Cramton, Marek Pycia, Marzena Rostek, and Marek Weretka. Demand reduction and inefficiency in multi-unit auctions. *The Review of Economic Studies*, 81(4):1366–1400, 2014.
- [4] Leon Barrett and Srin Narayanan. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*, pages 41–47, 2008.
- [5] Nicolas A Barriga, Marius Stanescu, and Michael Buro. Game tree search based on nondeterministic action scripts in real-time strategy games. *IEEE Transactions on Games*, 10(1):69–77, 2017.
- [6] Branislav Bošanský, Viliam Lisý, Marc Lanctot, Jiří Čermák, and Mark HM Winands. Algorithms for computing strategies in two-player simultaneous move games. *Artificial Intelligence*, 237:1–40, 2016.
- [7] Sandro Brusco and Giuseppe Lopomo. Collusion via signalling in simultaneous ascending bid auctions with heterogeneous objects, with and without complementarities. *The Review of Economic Studies*, 69(2):407–436, 2002.
- [8] Sandro Brusco and Giuseppe Lopomo. Simultaneous ascending auctions with complementarities and known budget constraints. *Economic Theory*, 38(1):105–124, 2009.
- [9] Bundesnetzagentur. Spectrum for wireless access for the provision of telecommunications services. https://www.bundesnetzagentur.de/EN/Areas/Telecommunications/Companies/FrequencyManagement/ElectronicCommunicationsServices/ElectronicCommunicationServices_node.html, 2022.
- [10] Benjamin E Childs, James H Brodeur, and Levente Kocsis. Transpositions and move groups in monte carlo tree search. In *2008 IEEE Symposium On Computational Intelligence and Games*, pages 389–395. IEEE, 2008.
- [11] Peter I Cowling, Edward J Powley, and Daniel Whitehouse. Information set monte carlo tree search. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):120–143, 2012.
- [12] Peter Cramton et al. Spectrum auctions. *Handbook of telecommunications economics*, 1:605–639, 2002.
- [13] Peter Cramton et al. Simultaneous ascending auctions. *Combinatorial auctions*, pages 99–114, 2006.
- [14] European 5G Observatory. Italian 5g spectrum auction. <https://5gobservatory.eu/italian-5g-spectrum-auction-2/>, 2018.
- [15] Jacob K Goeree and Yuanchuan Lien. An equilibrium analysis of the simultaneous ascending auction. *Journal of Economic Theory*, 153:506–533, 2014.
- [16] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.
- [17] Jongmin Lee, Geon hyeong Kim, Pascal Poupart, and Kee-Eung Kim. Monte-carlo tree search for constrained mdps. In *IJCAI 2018*, 2018.
- [18] Jongmin Lee, Geon-Hyeong Kim, Pascal Poupart, and Kee-Eung Kim. Monte-carlo tree search for constrained pomdps. *Advances in Neural Information Processing Systems*, 31, 2018.
- [19] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.
- [20] Michael Maschler, Shmuel Zamir, and Eilon Solan. *Game theory*. Cambridge University Press, 2020.
- [21] Paul Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of political economy*, 108(2):245–272, 2000.
- [22] Paul Milgrom and Paul Robert Milgrom. *Putting auction theory to work*. Cambridge University Press, 2004.
- [23] Ofcom. Award of the 700 mhz and 3.6-3.8 ghz spectrum bands. Technical report, Ofcom, 2020. https://www.ofcom.org.uk/_data/assets/pdf_file/0020/192413/statement-award-700mhz-3.6-3.8ghz-spectrum.pdf.
- [24] Santiago Ontanón, Gabriel Synnaeve, Alberto Uriarte, Florian Richoux, David Churchill, and Mike Preuss. A survey of real-time strategy game ai research and competition in starcraft. *IEEE Transactions on Computational Intelligence and AI in games*, 5(4):293–311, 2013.
- [25] Alexandre Pacaud, Aurelien Bechler, and Marceau Coupechoux. Bidding efficiently in simultaneous ascending auctions with budget and eligibility constraints using simultaneous monte carlo tree search. *arXiv preprint arXiv:2307.11428v2*, 2024.
- [26] Alexandre Pacaud, Marceau Coupechoux, and Aurelien Bechler. Monte carlo tree search bidding strategy for simultaneous ascending auctions. In *2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, pages 322–329. IEEE, 2022.
- [27] Daniel M Reeves, Michael P Wellman, Jeffrey K MacKie-Mason, and Anna Osepayshvili. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39(1):67–85, 2005.
- [28] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhruv Kumar, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [29] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Małdziuk. Monte carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review*, pages 1–66, 2022.
- [30] Balázs Szentes and Robert W Rosenthal. Beyond chopsticks: Symmetric equilibria in majority auction games. *Games and Economic Behavior*, 45(2):278–295, 2003.
- [31] Balázs Szentes and Robert W Rosenthal. Three-object two-bidder simultaneous auctions: chopsticks and tetrahedra. *Games and Economic Behavior*, 44(1):114–133, 2003.
- [32] Mandy JW Tak, Marc Lanctot, and Mark HM Winands. Monte carlo tree search variants for simultaneous move games. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8. IEEE, 2014.
- [33] H Jaap Van Den Herik, Jos WHM Uiterwijk, and Jack Van Rijswijk. Games solved: Now and in the future. *Artificial Intelligence*, 134(1-2):277–311, 2002.
- [34] Robert J Weber. Making more from less: Strategic demand reduction in the fcc spectrum auctions. *Journal of Economics & Management Strategy*, 6(3):529–548, 1997.
- [35] Michael P Wellman, Anna Osepayshvili, Jeffrey K MacKie-Mason, and Daniel Reeves. *Bidding strategies for simultaneous ascending auctions*. B.E. J. Theoret. Econom, 2008.
- [36] Charles Z Zheng. Jump bidding and overconcentration in decentralized simultaneous ascending auctions. *Games and Economic Behavior*, 76(2):648–664, 2012.
- [37] Albert L Zobrist. A new hashing method with application for game playing. *ICGA Journal*, 13(2):69–73, 1990.