Institutional Members: CEPR, NBER and Università Bocconi

WORKING PAPER SERIES

Artificial Intelligence, Algorithmic Bidding and Collusion in Online Advertising

Francesco Decarolis, Gabriele Rovigatti, Michele Rovigatti, Ksenia Shakhgildyan

Working Paper n. 708

This Version: February, 2024

IGIER – Università Bocconi, Via Guglielmo Röntgen 1, 20136 Milano –Italy
http://www.igier.unibocconi.it

Artificial Intelligence, Algorithmic Bidding and Collusion in Online Advertising*

Working Paper

Francesco Decarolis, Gabriele Rovigatti, Michele Rovigatti, Ksenia Shakhgildyan†

February 1, 2024

Abstract

Algorithms are becoming the standard tool for bidding in auctions through which digital advertising is sold. To explore how algorithmic bidding might affect functioning of these auctions, this study undertakes a series of simulated experiments where bidders employ Artificial Intelligence algorithms (Q-learning and Neural Network) to bid in online advertising auctions. We consider both the generalized second-price (GSP) auction and the Vickrey-Clarke-Groves (VCG) auction. We find that the more detailed information is available to the algorithms, the better it is for the efficiency of the allocations and the advertisers profit. Conversely, the auctioneer revenues tend to decline as more complete information is available to the advertiser bidding algorithms. We also compare the outcomes of algorithmic bidding to those of equilibrium behavior in a range of different specifications and find that algorithmic bidding has a tendency to sustain low bids both under the GSP and VCG relative to competitive benchmarks. Moreover, the auctioneer revenues under the VCG setting are either close to or lower than those under the GSP setting. In addition, we consider three extensions commonly observed in the data: introduction of a non-stategic player, bidding through a common intermediary, and asymmetry of the information across bidders. Consistent with the theory, the non-strategic player presence leads to increased efficiency, whereas bidding through a common intermediary leads to lower auctioneer revenue compared to the case of individual bidding. Moreover, in experiments with information asymmetry, more informed players earn higher rewards.

Keywords: Online Advertising, Sponsored Search Auctions, Algorithmic Bidding, Artificial Intelligence, Collusion.

---

*Francesco Decarolis and Ksenia Shakhgildyan acknowledge financial support from the PRIN 2017 (grant Prot. 2017TMF-PSH).

†Decarolis, Bocconi University and IGIER, *francesco.decarolis@unibocconi.it*, Gabriele Rovigatti, Bank of Italy, *gabriele.rovigatti@bancaditalia.it*, Michele Rovigatti, Bocconi University, *mic.rovigatti@gmail.com*, Shakhgildyan, Bocconi University and IGIER, *ksenia.shakhgildyan@unibocconi.it*.

1 Introduction

Pricing decisions are increasingly taken through computer algorithms. While these pricing algorithms have been used for a long time in specific markets like airline ticketing, only recently their usage has expanded to other sectors, such as financial markets, hotels, insurance, and essentially all online marketplaces organized as auctions, as well as other biddable markets. This process has brought about substantial benefits in terms of efficiency of how prices can rapidly adjust to changes in demand and supply conditions but has also created new challenges.

First, different artificial intelligence tools require different amount of information. Some learn from trial and error and require very little knowledge of the underlying economic environment, whereas other require its deep understanding. Nowadays, economic agents have become more and more attentive to privacy. The response by the leading platforms has anticipated and moved beyond those of regulators, deeply affecting competition in digital markets.[1] In the context of search advertising, this has meant a reduction in the amount of type of data available to advertisers, a phenomenon known as *data fogginess*. This is a well-known and hotly debated issue. In particular, many experts have argued that an unintended consequence of this evolution is that now data are exclusively inside the walled garden of the tech giants. Hence, a paradoxical consequence of privacy protection is that it has created a trade-off with market competition in data-intense industries like digital advertising. In this study, we offer a contribution to this debate by quantifying how the shift toward data fogginess is shifting the value generated by digital advertising away from advertisers and toward the platform. The key element of our analysis relies on how different artificial intelligence tools that are at the heart of any process of ad buying require different amounts of information on the economic environment, in particular, auction rules.

Second, the current debate on algorithms and collusion has identified two main channels through which algorithmic pricing might enhance collusive behavior (OECD, 2017). First, thanks to high price transparency and high-frequency trading, algorithms increase the ability to react fast and aggressively, thus making collusive strategies more stable in virtually any market structure. Second, by using automated mechanisms to implement common policies or optimize joint profits with deep learning techniques, algorithms can lead to the same outcomes of traditional "hardcore" cartels through tacit collusion.

This study contributes to the understanding of both of these challenges in the context of auctions for online advertisement. In particular, we study experimentally the behavior of two classes of algorithms (Q-learners and Neural Networks) in a workhorse search auction model of repeated bidding competition. Our first approach follows Calvano et al. (2020*a*) by setting up a series of computational experiments where Q-learner AI algorithms set bids. In our context, however, the environment is not an abstract oligopoly pricing game but a tightly defined auction market where the rules of the game match those faced by the advertisers active in buying search ads. Indeed, we consider the generalized second-price (GSP) auction,

[1] Remark: some reforms originating from the EU, like the GDPR, lead Google to a Global response, like with FLoCs. But other EU policies, lead to changes exclusively within EU countries: for instance, the choice screen auction for Android. Likely, this is not by chance: GDPR is exploited to construct walled gardens.

which is the leading mechanism to sell ad space on search engines like Google and Bing, as well as the Vickrey-Clarke-Groves (VCG) mechanism, used by Facebook for its contextual ads.

Our second approach is a novel method that is based on the Artificial Neural Network algorithm. Even though the Neural Network is one of the most popular algorithms in machine learning, we propose a new simple approach that combines the reinforcement learning idea and the simple network structure to reach convergence. There are several differences between the two approaches. First, the Q-learner optimizes behavior without actually knowing the exact rules of the game, solely based on a trial-and-error, taking the rewards as given, whereas we need to calculate the counterfactual rewards given particular states at each iteration of the Neural Network approach. Second, when the number of possible actions grows, the Q-learner suffers more from the curse of dimensionality, since when the number of actions available to the algorithms increases, the number of parameters to estimate grows exponentially.

Our baseline set of experiments explores questions of information availability and collusion by considering different configurations of slots and advertisers, involving the case of one keyword auction. Individual advertisers interact repeatedly and bid using their own Q-learner or Neural Network algorithms. In particular, we study through experimentation situations with up to five bidders, each endowed with a discrete set of possible bids. Importantly, bidders are asymmetric in terms of their valuations because, as we will show, this is key to illustrating complexities that occur in advertising auctions (asymmetric bidders, different click-through rates ratios, etc.). Indeed, not all of the experiments produce results that lead to exactly the same conclusions and exceptions are informative of how different segments of the online ad market might be impacted by the phenomena that we study. Another crucial feature in comparison to the existing literature on AI auctions is the fact that we consider algorithms that have memory. In the advertising setting, it is a fundamental feature of how the market works since advertisers base their future actions on the observables in the past.

Our main takeaways can be summarized as follows:

1. First, in terms of the informational assumptions, we find that the more information about the auction rules available to the algorithms the better it is for the efficiency of the allocations.
   (a) Neural Network Approach leads to higher efficiency. (See Table 1, Table 7)
   (b) With more informative initialization of the Q-learners, the efficiency increases. (See Table 2, Table 5 in comparison to Table 4)
2. Moreover, the more information is available to the algorithms, the better of the advertisers are, and the lower the auctioneer revenue is.
   (a) Provided enough information is available to the algorithms, the auctioneer revenues that we get using the Neural Network approach are lower than using the Q-learning algorithm. (See in Table 1 the difference between Q-learner and Neural Network when comparing experiments 1 and 2 to the others.)

(b) With more informative initialization of the Q-learners, the auctioneer revenue decreases. (See Table 2)

(c) When the scope for collusion increases through a common intermediary, for all experiments the auctioneer revenues under the Neural Network approach are lower than using the Q-learning algorithm. (See Table 7)

Next, we study whether a Nash equilibrium analysis is still adequate when AI-powered algorithms bid in the auctions. Since the algorithmic environments are not stationary, there is even no guarantee that the algorithms converge. We study the basic question of whether the bids converge toward the Nash Equilibrium (NE) predictions. Moreover, since GSP auctions have multiple NE, we compare the results to the lowest-revenue locally envy-free equilibrium (EOS) introduced by the studies of Varian (2007) and Edelman, Ostrovsky and Schwarz (2007) mentioned above. We also compare the VCG auction limit bids to the truthful bidding predicted by the literature and study the implications for search engine revenue by comparing the GSP and VCG auction formats. In our preliminary results, we find that:

3. Algorithmic bidding has a tendency to sustain low bids both under the GSP and VCG.

   (a) In GSP auction simulations most of the players tend to bid lower than predicted by the EOS bids resulting in always lower revenue than in the EOS when information is more accurate. (See Table 1 right panel and Table 2)

   (b) In VCG auction simulations players bid lower than their valuations, thus learning to coordinate relative to the truthful bidding. (See Table 4 and Table 5)

   (c) Auctioneer revenues under the VCG setting are either close to or lower than those under the GSP setting. (See Tables 4 and 5)

In addition, we consider three extensions. Those extensions account for phenomena often observed in the data. The first extension considers the auction in which in addition to the independent player, a 'noise bidder' (a player that bids non-strategically) is introduced. Hashimoto (2010) has first introduced a noise bidder in the GSP setting in order to eliminate bidders' implausible behavior. Moreover, in real auctions, there is no guarantee that every player bids strategically. For example, some players might enter an auction and explore the environment by bidding randomly. We find that in algorithmic setting:

4. The introduction of the noisy bidder leads to 100 percent efficiency (an efficient outcome for each run of each of the experiments). (See Table 6)

The second extension explores the role of intermediaries. Although in the early days of online ad auctions advertisers used to bid through their own individual accounts, nowadays most of the bidding takes place through intermediaries. A recent contribution by Decarolis and Rovigatti (2021) has shown the importance of intermediaries for digital advertising: despite Google remaining the dominant selling platform, increasing demand concentration within a handful of large advertisers has significantly eroded Google's revenue in recent years. One of the motives behind this phenomenon is that bid delegation to intermediaries creates opportunities for them to generate surplus for their clients by lowering payments by coordinating the bids of their clients. The equilibrium characterization of this form of collusion is the focus of the analysis in Decarolis, Goldmanis and Penta (2020). Their model presents sharp results in terms of the revenue and efficiency property of the GSP auction relative to the VCG auction, but it is based on a rather specific equilibrium concept. We evaluate whether AI algorithms can learn to collude and how exactly by replicating the set of experiments conducted for the case of independently competing bidders, but with the modification that a subset of the advertisers jointly bid under the same intermediary. Our findings indicate that:

5. Bidding through a common intermediary leads to lower auctioneer revenue compared to the case of individual bidding under the Neural Network Approach. (See Table 7 in comparison to Table 1)

6. Also in the case of the Q-learning, the auctioneer revenues are lower with a common agency than with independent bidders, but the two sets of revenues are closer than in the Neural Network case and this is due to the curse of dimensionality.

Finally, the third extension focuses on the case of information asymmetry. The platform might have an incentive and certainly has an opportunity to reveal different amounts of information to different advertisers.[2] It is possible to model the asymmetry of information between the bidders in many different ways. We are currently considering the same experiments conducted for the case of competing bidders, but assuming that one of the bidders is more informed than the other in terms of the initialization. We find that:

7. More informed players earn higher rewards in comparison to the cases when the same players have less information. (See Table 8)

Our study is mostly related to two strands of research. The first one is that on digital advertising. Advertising is a financial engine for most digital platforms, and thus it is crucial to understand the dynamics of the digital economy. The sponsored search auctions on which we focus represent one of the fastest-growing and most economically relevant forms of Internet advertising, accounting for approximately half of the total revenues of this market, which in the United States alone totaled $120 billion in 2020. It is therefore not surprising that extensive research has been devoted to sponsored search (Edelman, Ostrovsky and Schwarz (2007), Varian (2007), Borgers et al. (2013), Hashimoto (2010), Che, Choi and Kim (2017), Athey and Nekipelov (2014), Gomes and Sweeney (2014), Decarolis, Goldmanis and Penta (2020) and Decarolis and Rovigatti (2021), Bergemann et al. (2021)), and digital advertising more generally (Balseiro, Besbes and Weintraub (2015), Celis et al. (2015) and Einav et al. (2018)). Within this vast and rapidly growing literature, our contribution is to analyze how AI bidding impacts players' strategies and, in turn, how this impacts payoffs and allocations.

[2] See, for example, Bergemann et al. (2021).

The second research area to which we contribute is that on pricing algorithms. Recent contributions on algorithmic pricing include the analysis of algorithmic collusion by Calvano et al. (2020*b*), Calvano et al. (2020*a*), Asker, Fershtman and Pakes (2021), Assad et al. (2020*a*), andAssad et al. (2020*b*). A different but closely related research line is that on competition in pricing algorithms pursued by Brown and MacKay (2021). This research area is likely to become more and more important as AI is expected to become a general-purpose technology Agrawal, Gans and Goldfarb (2018). In this respect, the contribution of this study is to explore the interaction of algorithmic pricing and auctions, which clearly seems the first-order question given the pervasive use of auctions across multiple types of settings. A few recent papers analyze learning in auctions, but most of the literature is concerned with the auctioneer's side (Milgrom, Tadelis et al. (2019)). There are some related papers in machine-learning literature. For example, Dütting et al. (2019) study the optimal auction design through deep learning. Nedelec, El Karoui and Perchet (2019) analyze a strategic bidder's objective through gradient descent. The most closely related paper is the one by Banchio and Skrzypacz (2022). They study the case of single item auctions with symmetric bidders, finding that the first-price auctions with no additional feedback lead to tacit-collusive outcomes, while second-price auctions do not. In contrast, we study multiple item auctions, such as those used in the online search auctions, and consider a variety of cases that illustrate complexities that occur in advertising auctions. Most crucially, in contrast to Banchio and Skrzypacz (2022), we consider algorithms with memory since, in the real world, advertisers base their future actions on past information available to them. Compared to previous studies, an important feature of our work is to address the question of information availability.

The paper proceeds as follows: Section 2 presents the theoretical framework; Section 3 describes Artificial Intelligence bidding; Section 4 describes the experiments; Section 5 describes the simulation results; Section 6 concludes.

2 Theoretical Background

In this section, we discuss the theoretical foundations of GSP and VCG auctions.

Online ad auctions are mechanisms to assign agents $i \in I = \{1, \dots, n\}$ to slots $s = 1, \dots, S$, $n \geq S$ where for simplicity we assume $n = S + 1$ (the extension to $n \geq S$ is straightforward). In our case, agents are advertisers, and slots are positions for ads on a webpage (e.g., on a social media's newsfeed for a certain set of cookies, on a search-engine result page for a given keyword, etc.). Slot $s = 1$ corresponds to the highest/best position, and so on until $s = S$, which is the slot in the lowest/worst position. For each s, we let x^s denote the 'click-through-rate' (CTR) of slot s, that is the number of clicks that an ad in position s is expected to receive, and assume that $x^1 > x^2 > \dots > x^S > 0$. We also let $x^t = 0$ for all $t > S$. Finally, we let v_i denote the per-click-valuation of advertiser i, and we label advertisers so that $v_1 > v_2 > \dots > v_n$. As in Varian (2007) and EOS, we maintain that valuations and CTRs are common knowledge.

2.1 Rules of the auctions

Both in the VCG and in the GSP auction, advertisers submit bids $b_i \in \mathbb{R}_+$, and slots are assigned according to their ranking: first slot to the highest bidder, second slot to the second-highest bidder, and so on. We denote bid profiles by $b = (b_i)_{i=1,\dots,n}$. For any profile b, we let $\rho(i)$ denote the rank of i's bid (ties are broken according to bidders' labels). Hence, with this notation, for any profile b, in either mechanism bidder i obtains position $\rho(i)$ if $\rho(i) \leq S$, and no position otherwise. The resulting payoff, ignoring payments, is thus $v_i x^{\rho(i)}$.

The GSP and VCG mechanisms only differ in their payment rule. In the GSP mechanism, the k-highest bidder gets position k and pays a price-per click equal to the $(k+1)$-th highest bid. Using our notation, given a profile of bids b, agent i obtains position $\rho(i)$ and pays a price-per-click equal to $b^{\rho(i)+1}$. Bidder i's payoff in the GSP auction, given a bids profile $b \in \mathbb{R}^n_+$, can thus be written as $u_i^{\mathcal{G}}(b) = \left(v_i - b^{\rho(i)+1}\right) x^{\rho(i)}$.

In the VCG auction, an agent pays the total allocation externality he imposes on others. In this setting, if the advertiser in position k were removed from the auction, all bidders below him would climb up one position. Hence, if other bidders are bidding truthfully (i.e., $b_j = v_j$, as will be the case in equilibrium), the total externality of the k-highest bidder is equal to $\sum_{t=k+1}^{S+1} b^t(x^{t-1} - x^t)$. We can thus write i's payoff in the VCG mechanism, given a bids profile $b \in \mathbb{R}^n_+$, as $u_i^{\mathcal{V}}(b) = v_i x^{\rho(i)} - \sum_{t=\rho(i)+1}^{S+1} b^t(x^{t-1} - x^t)$.

2.2 Equilibria

Despite the relative complexity of its payment rule, bidding behavior in the VCG is very simple, as truthful bidding (i.e., $b_i = v_i$) is a dominant strategy in this auction. In the resulting equilibrium, advertisers are efficiently assigned to positions. The VCG mechanism, therefore, is efficient and strategy-proof.

Equilibrium behavior in the GSP auction is much more complex. The GSP auction has many Nash equilibria. For this reason, EOS introduced a refinement of the equilibrium correspondence, the *lowest-revenue locally envy-free equilibrium*, which was crucial to cut through the complexity of the GSP auction.[3] As EOS showed, such equilibria induce the same allocations and payments as truthful bidding in the VCG, and they are fully characterized by the following conditions: $b_1 > b_2$, $b_i = v_i$ for all $i > S$, and for all $i = 2, \dots, S$,

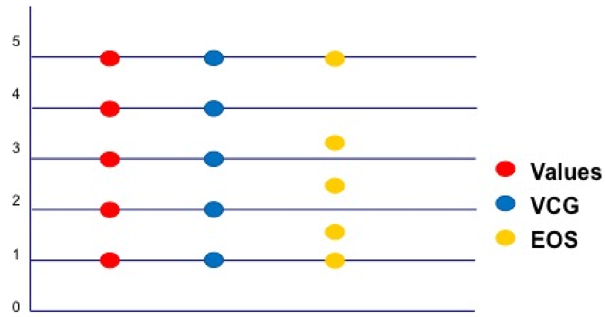
$$b_i = v_i - \frac{x^i}{x^{i-1}}(v_i - b_{i+1}). \tag{1}$$

Example: In Figure 1 we compare the equilibrium bids in GSP and VCG auction, in which bidders are known to bid truthfully. We consider an auction with four slots and five bidders. Their valuations are $v = (5, 4, 3, 2, 1)$. The CTRs for the five positions are the following: $x = (20, 10, 5, 2, 0)$. Quality scores are assumed to be one. Then the bids in EOS lowest envy-free equilibrium are: $b_5 = 1$, $b_4 = 1.6$, $b_3 = 2.3$ and

[3] A Nash equilibrium $(b_i)_{i\in I}$ is locally envy-free if $x^{\rho(i)}(v_i - b^{\rho(i)+1}) \geq x^{\rho(i)-1}(v_i - b^{\rho(i)})$ for every i. EOS refinement is the lowest-revenue Nash equilibrium which satisfies this condition. This refinement is especially important because it conforms with the search engines' tutorials on how to bid in these auctions. See, for instance, the Google AdWord tutorial in which Hal Varian teaches how to maximize profits by following this bidding strategy: `https://www.youtube.com/watch?v=tW3BRMld1c8`.

$b_2 = 3.15$ (denoted by yellow points). For details see Decarolis, Goldmanis and Penta (2020).

Figure 1: Equilibrium Bids in GSP and VCG



Notes: Valuations, equilibrium bids in the VCG auction, and equilibrium bids in the EOS equilibrium of the GSP auction in an auction with five bidders and four slots, where valuations $v = (5, 4, 3, 2, 1)$*, and* $ctr = (20, 10, 5, 2)$.

2.3 Extensions

In this paper, in addition to the analysis of the GSP and VCG auctions with independent bidders, we consider three extensions. Those extensions account for phenomena often observed in the data.

The first extension considers the auction in which in addition to the independent player, a 'noise bidder' (a player that bids non-strategically) is introduced. That extension was inspired by the paper Hashimoto (2010), which has first introduced a noise bidder in the GSP setting in order to eliminate bidders' implausible behavior. Moreover, in real actions, there is no guarantee that every player bids strategically.

The second extension considers the case of coordinated bidding. Decarolis, Goldmanis and Penta (2020) (DGP) provided a theoretical analysis of the GSP auction, when some of the advertisers' bids are placed by a common agency. In particular, DGP put forward the notion of "Recursively-stable Agency Equilibrium" (RAE), which can be used to study agency bidding in general mechanisms for online ad auctions, and general agency configurations. Crucially, the RAE's framework enables to accommodate the case of partial cartels, i.e. situations in which agencies operate side-by-side with independent bidders, which is the most relevant case in the data.

The third extension introduces the information asymmetry. Often some of the bidders are more informed than the other, especially since the platform might have an incentive and certainly has an opportunity to reveal different amount of information to different advertisers. It is also likely to be the case since different bidders enter the market at different times, and as a result are likely to have different knowledge about the environment.

3 Artificial Intelligence Algorithm

3.1 The Reinforcement Learning Approach

Reinforcement learning is the machine learning area of research that studies how agents learn to take actions in different environments by trial and error in order to maximize the reward. The main idea of the algorithm is to find the balance between exploration (trying out the new actions) and exploitation (using the obtained knowledge).

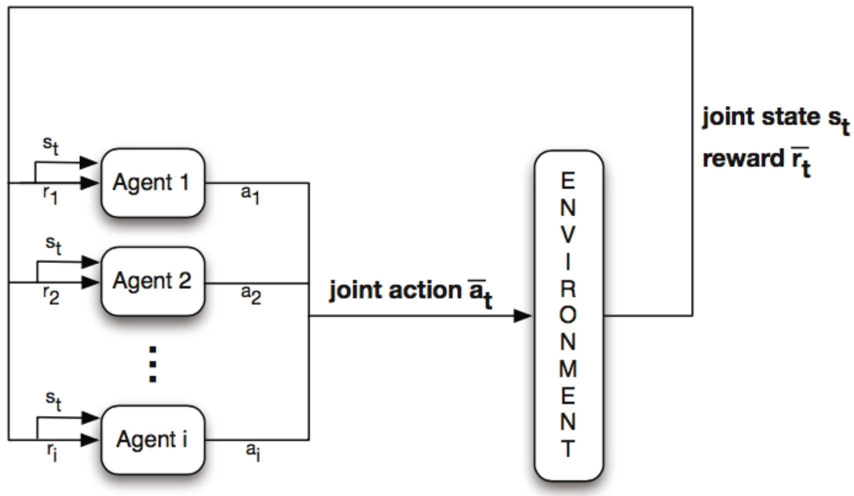
In reinforcement learning players act in an *environment* (E), which is characterized by different *states* (S) and a fixed set of rules. The players choose one out of a set of *actions* (A), whose choice - in each state - corresponds to a different *reward* (R). The aim of the player is to choose an action that maximizes the reward. The algorithm, through repetitions and experiments across different actions, "learns" the best action to be performed in each state of the world. Importantly, the algorithm learns the optimal strategy without the knowledge of the environment, but instead purely by trial-and-error. Usually, the reinforcement is modeled as a Markov decision process.

A basic reinforcement learning agent interacts with its environment in discrete time steps. Formally, at each time t, the agent receives the current state s_t and reward r_t. It then chooses an action a_t from the set of available actions, which is subsequently sent to the environment. The environment moves to a new state s_{t+1} and the reward r_{t+1} associated with the transition (s_t, a_t, s_{t+1}) is determined. The goal of a reinforcement learning agent is to learn a policy: $\pi : A \times S \rightarrow [0, 1]$, $\pi(a, s) = \Pr(a_t = a \mid s_t = s)$ which maximizes the expected cumulative reward.

Moreover, reinforcement learning can be applied to repeated games with several learning agents. To do that, the actions of other players should be treated as the state variables. Since that approach is prone to non-stationarity, an experiment should be run to study whether convergence to the equilibrium outcome would be achieved.[4]

[4]See Nowe, Vrancx and De Hauwere (2012) for more information on multi-agent Reinforcement Learning and as a reference for Figure 2.

Figure 2: Multi-agent Reinforcement Learning Model



Notes: Schematic representation of the multi-agent reinforcement learning algorithm. At every step, each of the agents takes into account current state and reward. Given state and reward, each of the agents chooses next action. Combined actions determine the next state, as well as the next rewards.

3.1.1 Q-learning

The simplest reinforcement learning algorithm is Q-learning, which is based on the recursive optimization of a Q-matrix (Q stands for quality). Q-learning finds an optimal policy in the sense of maximizing the expected value of the total reward over all successive steps, starting from the current state. Formally, the goal is to solve for Q-function:

$$Q(s,a) = \mathbb{E}(r|s,a) + \delta \mathbb{E}[\max_{a' \in A} Q(s',a')|s,a],$$

where $s' = s_{t+1}$, and $\delta < 1$ is the discount factor. The discount factor determines the importance of future rewards. A factor of 0 will make the agent "myopic" (or short-sighted) by only considering current rewards, while a factor approaching 1 will make it value mostly the long-term high reward.

In case when S and A are finite, the Q-function of each of the players can be represented as a $|S| \times |A|$ matrix, which is called **Q-matrix**. If the agent knew the Q-matrix, he could calculate the optimal action for any given state. Q-learning provides a method for estimating the Q-matrix using an update rule by an iterative procedure.

The algorithm starts from an initial Q-matrix. High initial values can encourage exploration: no matter what is the selected action, the update rule will cause it to have lower values than the other alternative, thus increasing their choice probability. After choosing some new action a_t at state s_t, the algorithm updates $Q_t(s,a)$ for $s = s_t$ and $a = a_t$ using the following "temporal difference" update rule:

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha * (r_t + \delta * \max_{a' \in A} Q_t(s',a')).$$

Here α is the learning rate. The learning rate determines to what extent the new information substitutes the old ("how much" the algorithm learns from new actions).

In order to estimate the Q-matrix, the algorithm should visit different actions in different states. Thus the algorithm should be given a direction to explore the new actions even though they might go in the direction that looks not optimal to the algorithm given prior knowledge. One of the methods is to choose the action that currently leads to the highest value of Q-matrix in a given state with a probability $1-\epsilon$, and to randomize across all possible actions with probability ϵ. We will use a declining with time exploration rate $\epsilon_t = e^{-\beta * t}$, where $\beta > 0$ is the annihilation coefficient of the exploration rate. Smaller βs will make the Q-learner explore for more periods, while larger one reduces the exploration time - at the expense of precision.

3.1.2 Q-learner application to GSP auctions

We apply Q-learners in the context of GSP auctions. In particular, we simulate an Environment with the following features. First, we assume that each of n agents has bounded memory, in particular, each state is defined by the previous actions of all players. Next, we discretize the action space. Each bidder has k equally spaced possible actions (bids) within the $[B_{min}; B_{max}]$ range, and the Q-matrix has therefore the size $k^n \times k$.

Example: Suppose that each agent can perform 10 different actions in a given state and the state is defined by each agent's action in a 5-player game. In that case, every Q-matrix has $[10^5, 10]$ shape, where 10^5 is the number of states, i.e. all possible combinations of the 10 actions of the 5 players.

$$\begin{pmatrix} Index & State & Q-matrix\,row \\ & & [Q_{a_1}, \cdots, Q_{a_{10}}] \\ 0 & '[0,0,0,0,0]' & [0.7, 0.48, 1.4, 0.65, 0.89, \cdots] \\ 1 & '[0,0,0,0,1]' & [0.5, 0.38, 0.4, 1.65, 1.29, \cdots] \\ \vdots & \vdots & \cdots \\ 10 & '[0;0,0,1,0]' & [0.18, 0.22, 0.5, 0.76, 1.56, \cdots] \\ \vdots & \vdots & \cdots \\ 10^5 & '[10,10,10,10,10]' & [0.8, 1.22, 1.3, 1.5, 1.22, \cdots] \end{pmatrix}$$

where each cell is visited and updated every time the relative (state, action) realizes, and the reward is determined. Since the action space is discrete, we introduce the random tie-breaking rule, according to which if m bidders submit the same bids, each of them wins the slot with a probability $1/m$.

The reward function is the (normalized) difference between the valuation and the price paid, multiplied by the click-through-rate:

$$r_{i,t} = \frac{(v_i - b_{i+1,t})x^i}{normalization}$$

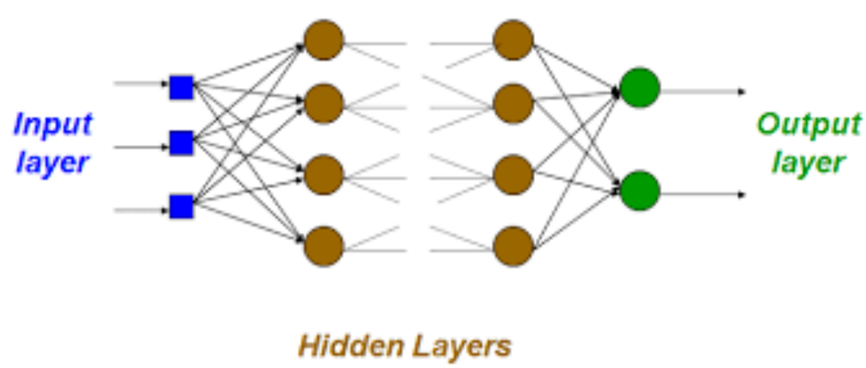
The *normalization* is the weighting factor of the reward. It ensures that the magnitude of the reward, and of Q-matrix cells, are comparable. It is constant for all the players and in all the states, so does not influence the optimal choice. In particular, we use the maximum possible reward as the normalization which is defined by the difference of maximum valuation and minimum bid, multiplied by the maximum position effect. For the initial values of the Q-matrix, we choose in each state the draw from the normal distribution centered at the valuation of the player.

3.2 The Neural Network Approach

In this section, we describe the second artificial intelligent algorithm based on Neural Networks.

An Artificial Neural Network is a function estimator. It contains layers of interconnected nodes called neurons, as they mimic in a simplified manner the functioning of the human brain. Each neuron is connected to the neurons that belong to the previous and to the next layer, but it is not connected to the neurons of its own layer. There are three types of layers, the input layer that receives external input data, the output layer used to return the results, and the hidden layer containing neurons helping to obtain an optimal result. As in figure 5, the input layer will receive a vector of real numbers which will be multiplied by the weights of its nodes. Each weight is a parameter to estimate. Then, these values are fed into the next layer and so on.

Figure 3: Neural Network Example



Notes: Basic Neural Network architecture can be represented using three types of layers of interconnected nodes. The input layer receives external input data, the output layer returns the results, and the hidden layers contain neurons that are essential to obtain an optimal result.

Let's denote by N_h the number of hidden layers and by N_{nr} the number of neurons in each of the hidden layers. Part of the model estimation is to determine the optimal number of layers and neurons. In fact, having too many neurons or layers can result in a long training time for a task that might be solved with a much smaller and faster network.

Architecture: To be more concrete, the network architecture in our application to the GSP auction for each of the players is structured as follows:

1. The external input data is the previous state. This data is received by the input layer. Similar to the case with Q-learners, each state is defined by the previous actions of all players.

2. Each layer will contain N_{nr} neurons and will be a dense layer, which means that each neuron receives input from all neurons of the previous layer.

3. The output is a k long vector. As before, k is the number of possible actions (bids) of each of the players. The output is a score assigned to each of the possible actions.

The goal is to train the network to assign, when given a state, the maximum score (which in our case is 1), to the optimal action and to assign 0 to other possible actions, therefore solving the classification problem. The chosen action for each player will be the maximum value of the output vector, the new state is defined by the combination of chosen actions.

The number of neurons N_{nr} per layer and the number of hidden layers N_h are the hyper-parameters along with the learning rate which is a parameter used for the optimization of the network.

Optimal Action and Loss: In order to train a Neural Network, we compare the output of the net with an optimal behavior and define a similarity measure, which we then optimize. This measure is called a *loss function*.

We define the *optimal action* for a player as the action which grants the highest possible reward given the actions of the other players in the state s_t. Since in the GSP and VCG auctions more than one action can grant the same reward, there are multiple optimal actions for each possible state. Once we have the indices of the optimal actions, we define the loss function in the following way. We create a k-long vector y which has all zero entries besides the indices of the optimal actions which are assigned one. Next using a loss function, we compare y to the output of the Neural Network $\hat{y}_i$, which is also a k-long vector of probabilities. There are several ways to define the loss function. We use a Binary Cross-Entropy, which is also widely used in binary classification problems:

$$Loss = -\sum_{i=1}^{k}[y_i \cdot \log(\sigma(\hat{y}_i)) + (1 - y_i) \cdot \log(1 - \sigma(\hat{y}_i))],$$

where $\sigma(\cdot)$ is a sigmoid function.[5] A sigmoid is applied for numerical stability.

Optimization and Updating: Once we have defined the loss function, the optimization is done through back-propagation. In particular, we use the Adam optimizer (Kingma and Ba (2014)).

[5]The exact function we use in Python is torch.nn.BCEWithLogitsLoss. Standard Binary Cross-Entropy is given by $Loss = -\sum_{i=1}^{k}[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$ instead. When $y_i = 1$, the lower the value of $\hat{y}_i$ the bigger is the loss term. In contrast, when $y_i = 0$, the bigger the value of $\hat{y}_i$ the lower is the loss term.

3.3 Comparison of the Algorithms

There are several crucial differences between the proposed algorithms. First, technically, the Q-learning allows solving the dynamic problem with any exogenously given discount rate. As a limit case, the static problem with $\delta = 0$ can be considered. On the other hand, the proposed Neural Network approach aims at solving the one-shot game.

Second, there are prominent differences in terms of informational assumptions. Each of the Neural-learners requires information on the counterfactual rewards given bids of the other players, which Q-learners learn solely based on the trial-and-error without the knowledge of the reward function and the environment. On the other hand, the Neural Network approach does not depend on the initialization, whereas for the Q-learners, as we are going to see later, the initialization plays a fundamental role.

Moreover, the main drawback of the Q-learning approach is the curse of dimensionality. When the number of possible actions grows, the Q-learner suffers more from the curse of dimensionality, since when the number of possible actions increases the number of parameters to estimate grows exponentially. Instead, for the Neural Network approach, the increase in the number of possible actions does not lead to a substantial increase in the Neural Network nodes and their initial weights.

4 Experiments Description

We start from three experiments of similar nature, but with an increasing number of players from 2 to 5.

Experiment 1. *(2-bidder case:) Consider an auction with one slot and two bidders. Their valuations are* $v = (2, 1)$*. The CTRs are the following:* $x = (2, 0)$*.*

We define the action space for each of the player on the interval $[B_{min}, B_{max}] = [0.2, 3]$, with $k = 15$ possible actions so that the step between the actions is 0.2.

Experiment 2. *(3-bidder case) Consider an auction with two slots and three bidders. Their valuations are* $v = (3, 2, 1)$*. The CTRs are the following:* $x = (5, 2, 0)$*.*

We define the action space for each of the player on the interval $[B_{min}, B_{max}] = [0.2, 3]$, with $k = 29$ possible actions so that the step between the actions is 0.1.

Experiment 3. *(5-bidder case) Consider an auction with four slots and five bidders. Their valuations are* $v = (5, 4, 3, 2, 1)$*. The CTRs for the five positions are the following:* $x = (20, 10, 5, 2, 0)$*.*

We define the action space for each of the player on the interval $[B_{min}, B_{max}] = [0.2, 5]$, with $k = 25$ possible actions so that the step between the actions is 0.2.

Moreover, we also consider three extra examples from papers on GSP actions.

Experiment 4. *(Hashimoto (2010)) Consider an auction with three slots and four bidders. Their valuations are* $v = (100, 90, 60, 30)$*. The CTRs for the three positions are the following:* $x = (4, 2, 1)$*.*

We define the action space for each of the player on the interval $[B_{min}, B_{max}] = [20, 100]$, with $k = 17$ possible actions.

Experiment 5. *(Milgrom and Mollner (2014)) Consider an auction with three slots and three bidders. Their valuations are* $v = (15, 10, 5)$*. The CTRs for the three positions:* $x = (100, 3, 1)$*.*

We define the action space for each of the player on the interval $[B_{min}, B_{max}] = [2, 15]$, with $k = 27$ possible actions so that the step between the actions is 0.5.

Experiment 6. *(Borgers et al. (2013)) Consider an auction with three slots and three bidders. Their valuations are* $v = (16, 15, 14)$*. The CTRs for the three positions are the following:* $x = (3, 2, 1)$*.*

We define the action space for each of the player on the interval $[B_{min}, B_{max}] = [5, 16]$, with $k = 23$ possible actions so that the step between the actions is 0.5.

5 Simulation Results

In this section, we discuss our findings using both types of AI algorithms: Q-learning and Neural Networks. We start by presenting results for the GSP auction with independent players. Next, we compare results to the ones of the VCG auction. We also compare the outcomes of algorithmic bidding to those of equilibrium behavior in a range of different specifications and find that algorithmic bidding has a tendency to sustain low bids both under the GSP and VCG. Finally, we consider two extensions of the GSP setting. The first one introduces the 'noise bidder'. The second extension explores the role of a common intermediary placing bids on behalf of multiple bidders.

5.1 GSP with independent bidders

We start with the preview of the results for all of the experiments summarized in Table 1.

Table 1: Bids, Efficiency and Revenues in the Six Experiments - GSP

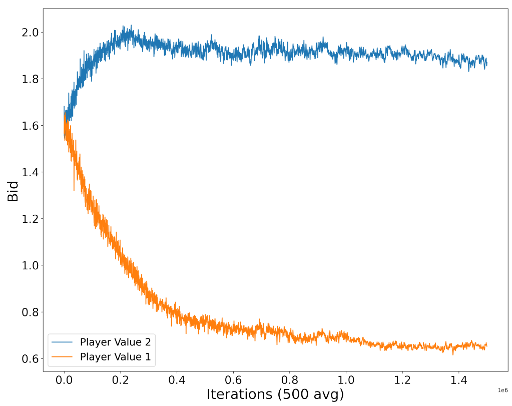
	Independent Players										
	EOS		Q-learners				Neural-learners				
	Bids	Revenue	Bids	Eff.	NE	Revenue	Bids	Eff.	NE	Revenue
Exp. 1	$(b_1, 1)$	2	$(1.87, 0.65)$	100%	100%	1.3	$(2.04, 1.04)$	100%	100%	2.07
						[1.23, 1.37]				[1.78, 2.36]
Exp. 2	$(b_1, 1.6, 1)$	10	$(2.22, 1.66, 0.90)$	98%	100%	9.15	$(2.22, 1.67, 1.08)$	100%	100%	10.46
						[9.0, 9.3]				[9.95, 10.97]
Exp. 3	$(b_1, 3.15, 2.3, 1.6, 1)$	96	$(3.90, 3.16, 2.45, 1.64, 0.87)$	93%	87.5%	98.06	$(3.9, 2.92, 2.27, 1.58, 0.88)$	98%	100%	90.82
						[97.4, 98.72]				[88.48, 93.16]
Exp. 4	$(b_1, 67.5, 45, 30)$	390	$(74.1, 66.7, 48.3, 24.8)$	80%	55%	403.6	$(78.7, 59.5, 42.6, 23.9)$	96%	100%	344.30
						[393.5, 413.7]				[332.12, 356.48]
Exp. 5	$(b_1, 9\frac{4}{5}, 3\frac{1}{3})$	990	$(11.8, 8.7, 4.4)$	100%	100%	884.7	$(13.35, 8.17, 3.41)$	100%	100%	827.23
						[867.7, 901.7]				[756.17, 898.29]
Exp. 6	$(b_1, 9\frac{2}{3}, 7)$	43	$(11.7, 10.6, 10.3)$	40%	0%	52.36	$(11.04, 8.54, 6.62)$	86%	94%	37.59
						[51.67, 53.05]				[36.71, 38.47]

Notes: Experiment 1: $v = (2, 1)$, $ctr = 2$; *Experiment 2*: $v = (3, 2, 1)$, $ctr = (5, 2)$; *Experiment 3*: $v = (5, 4, 3, 2, 1)$, $ctr = (20, 10, 5, 2)$; *Experiment 4*: $v = (100, 90, 60, 30)$, $ctr = (4, 2, 1)$; *Experiment 5*: $v = (15, 10, 5)$, $ctr = (100, 3, 1)$; *Experiment 6*: $v = (16, 15, 14)$, $ctr = (3, 2, 1)$. *In columns 2 and 3 the EOS equilibrium bids and auctioneer revenues are presented as a benchmark. Columns 4-7 report the simulation results for the experiments using the Q-learning AI algorithm, whereas columns 8-11 correspond to the results using the Neural Network approach. For both approaches we report the average across runs limit strategies (bids) for each of the players in the decreasing order of valuations; the percent of all runs that converge to an efficient outcome (the bids are alligned with the valuations); the percent of runs that converge to Nash equilibrium; the average auctioneer revenue across runs as well as the 95% confidence interval for the revenue in squared brackets.*

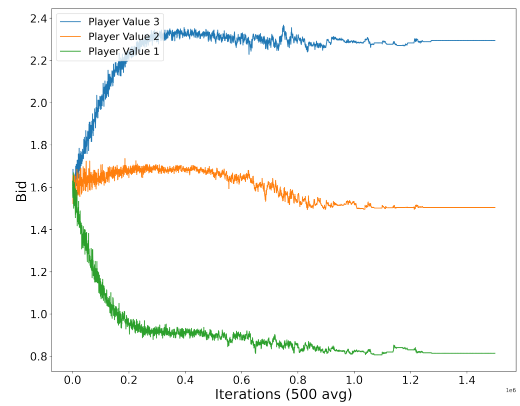
There are several notable features. As we have already previewed, the major difference between the Q-learning algorithm and Neural Network is the informational assumption, each of the Neural-learners requires the information on the counterfactual rewards given bids of the other players.

Figure 4: Bid Convergence in the Six Experiments - GSP

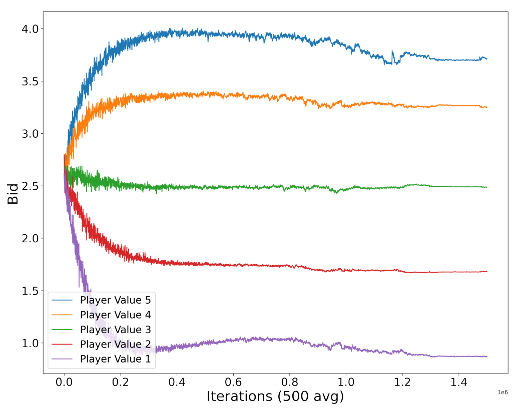
(a) Experiment 1



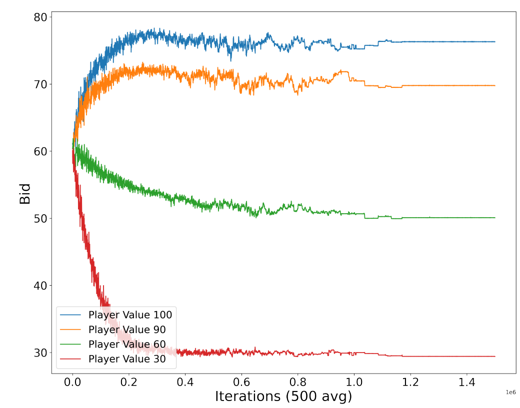
(b) Experiment 2



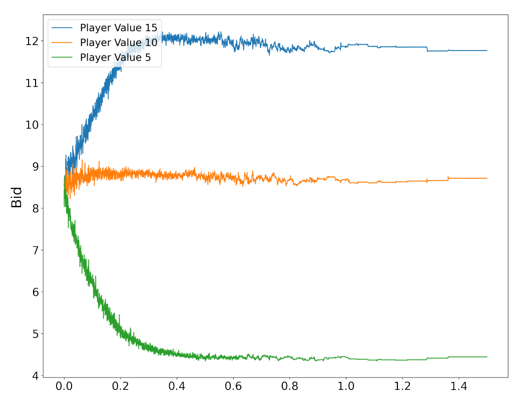
(c) Experiment 3



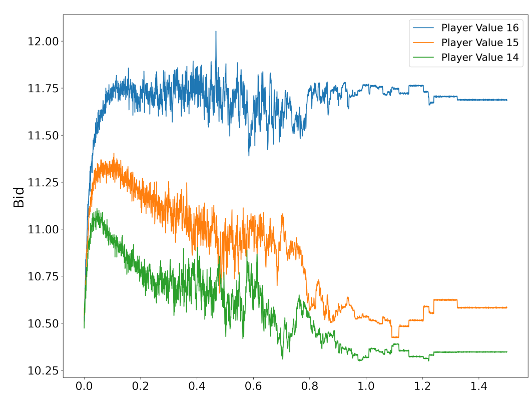
(d) Experiment 4



(e) Experiment 5



(f) Experiment 6

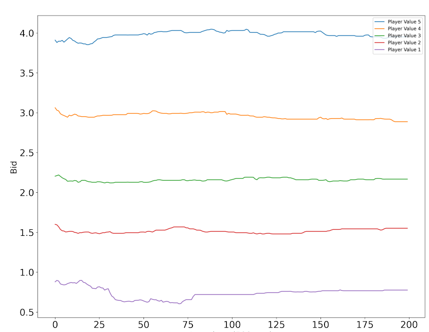


Notes: Each of the panels represents the bidding behavior of each of the single players that use Q-learning algorithm as a function of the iterations in each of the six experiments described in Section 4. We plot moving averages over 500 iterations.

The Q-learners, instead, learn solely based on the trial-and-error without the knowledge of the reward function and the environment. First, apart from experiments 1 and 2, the auctioneer revenues that we get using the Neural Network approach are lower than using the Q-learning algorithm and lower than the EOS equilibrium revenues. In Experiments 1 and 2 under the Neural Network approach, the lowest bidder that doesn't get a slot bids higher than his value. That is likely to be due to the fact that with only 1 or 2 slots available, Neural-learners lack the ability to explore all possible actions. Instead, once there are enough slots and bidders for exploration, the Neural-learners allow the bidders to converge to an equilibrium with lower than in EOS revenues. Second, the probability with which an algorithm convergences to an efficient equilibrium and to Nash Equilibrium is higher for the Neural-learners than for the Q-learners with baseline initialization. The difference is very prominent for Experiment 6. In that experiment, the valuations are very close to each other, and all allocations of positions to bidders are possible equilibrium allocations. Thus, more information about the auction rules is better for the players and for the efficiency.

A graphical illustration can also be informative of the types of bidding behavior exhibited by the two types of algorithms. In Figure 4, the six plots report, separately for each one of the six experiments, the actions (bids) of single players (moving averages across 500 iterations). We explore the convergence of the Q-learning algorithm using $T = 1,500,000$ iterations. The hyperparameters are: $\delta = 0.95$, $\beta = 1e-05$, and $\alpha = 0.5$. We plot the average bids over 50 runs of the algorithm. For comparison, in Figure 5, we also provide for the Neural Network the plot of the actions (bids) of single players (moving averages across 500 iterations) for our leading example corresponding to Experiment 3 (the case of the 5-bidder game) above. Instead of Q-learners, all players use Neural Networks to learn the optimal bidding strategies. The hyperparameters are: $N_{nr} = 20$, $N_h = 1$, $lr = 0.01$. Instead of $T = 1,500,000$ iterations as in case with Q-learners, we consider $T = 100,000$ iterations.

Figure 5: 5-bidder case: Neural Networks



Notes: Bidding behavior of each of the single players that use Neural Network algorithm as a function of the iterations in Experiment 3 with five bidders described in Section 4. We plot moving averages over 500 iterations.

As we can see, the limit values are quite similar to the ones of the Q-learning algorithm, apart from the players with the highest and lowest valuations. Moreover, the convergence is much faster compared to the Q-learning algorithm.

Table 2: Information Precision: Q-learners with Different Initializations

	Independent Players							
	Q-learners (Baseline Gaussian Initialization)				Q-learners (Informed Initialization)			
	Bids	Eff.	NE	Revenue	Bids	Eff.	NE	Revenue
Exp. 1	$(1.87, 0.65)$	100%	100%	1.3	$(1.92, 0.67)$	100%	100%	1.34
				[1.23, 1.37]				[1.30,1.38]
Exp. 2	$(2.22, 1.66, 0.90)$	98%	100%	9.15	$(2.37, 1.49, 0.78)$	100%	100%	8.99
				[9.0,9.3]				[8.75,9.23]
Exp. 3	$(3.90, 3.16, 2.45, 1.64, 0.87)$	93%	87.5%	98.06	$(3.84, 2.99, 2.21, 1.31, 0.59)$	100%	100%	89.67
				[97.4,98,72]				[88.95,90.39]
Exp. 4	$(74.1, 66.7, 48.3, 24.8)$	80%	55%	403.6	$(74.09, 64.19, 44.40, 24.75)$	100%	100%	370.31
				[393.5,413.7]				[368.94,371.68]
Exp. 5	$(11.8, 8.7, 4.4)$	100%	100%	884.7	$(14.00, 8.21, 3.74)$	100%	100%	857.28
				[867.7,901.7]				[849.93,864.63]
Exp. 6	$(11.7, 10.6, 10.3)$	40%	0%	52.36	$(8.70, 8.16, 7.86)$	100%	0%	39.63
				[51.67,53.05]				[39.16,40.10]

Notes: Experiment 1: $v = (2, 1)$*, ctr* $= 2$*; Experiment 2:* $v = (3, 2, 1)$*, ctr* $= (5, 2)$*; Experiment 3:* $v = (5, 4, 3, 2, 1)$*, ctr* $= (20, 10, 5, 2)$*; Experiment 4:* $v = (100, 90, 60, 30)$*, ctr* $= (4, 2, 1)$*; Experiment 5:* $v = (15, 10, 5)$*, ctr* $= (100, 3, 1)$*; Experiment 6:* $v = (16, 15, 14)$*, ctr* $= (3, 2, 1)$*. Columns 2-5 report the simulation results for the experiments using Q-learning AI algorithm with baseline initialization that only requires knowledge of players valuations, whereas columns 6-9 correspond to the results using Q-learning AI algorithm with informed initialization. For both approaches we report the average across runs limit strategies (bids) for each of the players in the decreasing order of valuations; the percent of all runs that converge to an efficient outcome (the bids are alligned with the valuations); the percent of runs that converge to Nash equilibrium; the average auctioneer revenue across runs as well as the 95% confidence interval for the revenue in squared brackets.*

Importantly, the results for the Q-learning algorithm depend on the initialization of the Q-matrix, whereas the Neural Network approach leads to convergence to an equilibrium in most of the cases even given random initialization of the Neural Network weights. Once the Q-matrix is initialized using the more informed initialization (see columns 5-8 of Table 2), the limit values of the bids as well as the auctioneer revenues are closer to the results of the Neural Network approach. This informed initialization in each state sets the value of the cell of Q-matrix at $t = 0$ at the discounted payoff that would accrue to player i if opponents randomized uniformly (similar to the baseline initialization in Calvano et al. (2020*a*)):

$$Q_{i,0}(s, a_i) = \frac{\Sigma_{a_{-i} \in A_{n-1}} \Pi_i(a_i, a_{-i})}{(1-\gamma)\|A\|^{n-1}}.$$

In comparison with the baseline initialization of the Q-learners, informed initialization always leads to an efficient outcome and in all experiments but Experiment 6 to Nash Equilibrium in every single run. Moreover, the auctioneer revenues are lower. Thus, once again, more information, in this case in the form of initialization, is better for the players and for the efficiency.

In addition to the results above, we have also ran Q-learning simulations for a much longer time, and reached $T = 15,000,000$ iterations of the algorithm for each of the players. Even though we believe that exploration of $T = 1,500,000$ that we have used in the baseline simulations is a reasonable assumption and is also consistent with Calvano et al. (2020a), by allowing algorithms to explore more, we substitute for the lack of initial information. As a result, compared to Table 1, the efficiency percentage increases to 100% under all initializations, including a random initialization. Moreover, auctioneer revenues decrease and become comparable with the revenues that were obtained using the Neural Network Approach. As before, the auctioneer revenues are lower than predicted by the EOS equilibrium. Thus, more information, in this case in terms of the increased exploration, is better for the anvertisers and for the efficiency. Importantly, in reality players are likely not to have enough time to explore all possible combinations of states many times, and as a result, the information provided by the platform is crucial for learning.

Table 3: Comparison between Q-learners with 15M iterations with different initializations

Independent Players - GSP												
	Q-learners: Baseline Gaussian Initalization				Q-learners: Informed Initalization				Q-learners: Random Initalization			
	Bids	Eff.	NE	Revenue	Bids	Eff.	NE	Revenue	Bids	Eff.	NE	Revenue
Exp. 1	$(1.97, 0.81)$	100.0%	100.0%	1.57	$(1.99, 0.82)$	100.0%	100.0%	1.58	$(1.99, 0.8)$	100.0%	100.0%	1.56
				[1.35, 1.79]				[1.36, 1.8]				[1.34, 1.78]
Exp. 2	$(2.27, 1.62, 0.87)$	100.0%	100.0%	9.74	$(2.24, 1.6, 0.94)$	100.0%	100.0%	9.81	$(2.3, 1.64, 0.95)$	100.0%	100.0%	10.07
				[9.16, 10.33]				[9.14, 10.48]				[9.47, 10.66]
Exp. 3	$(3.7, 3.02, 2.3, 1.66, 0.9)$	100.0%	100.0%	92.1	$(3.82, 2.95, 2.2, 1.57, 0.84)$	100.0%	100.0%	90.78	$(3.59, 2.9, 2.17, 1.63, 1.36)$	100.0%	84.0%	92.15
				[87.72, 96.48]				[87.27, 94.28]				[87.14, 97.16]
Exp. 4	$(71.28, 64.94, 45.59, 23.59)$	100.0%	100.0%	352.7	$(73.29, 62.83, 43.76, 25.68)$	100.0%	100.0%	355.0	$(68.77, 63.61, 44.26, 30.35)$	100.0%	90.0%	353.26
				[334.93, 370.48]				[336.6, 373.41]				[332.54, 373.99]
Exp. 5	$(12.7, 7.57, 4.55)$	100.0%	100.0%	802.87	$(12.69, 7.62, 4.89)$	100.0%	100.0%	818.47	$(12.91, 7.56, 6.21)$	100.0%	100.0%	874.15
				[722.32, 883.43]				[737.28, 899.66]				[797.08, 951.22]
Exp. 6	$(9.78, 9.36, 8.9)$	100.0%	0.0%	41.33	$(9.54, 8.74, 7.75)$	100.0%	0.0%	39.56	$(9.9, 9.52, 9.23)$	92.0%	0.0%	42.2
				[39.29, 43.37]				[38.43, 40.69]				[39.97, 44.43]

Notes: Experiment 1: $v = (2, 1)$*,* $ctr = 2$*; Experiment 2:* $v = (3, 2, 1)$*,* $ctr = (5, 2)$*; Experiment 3:* $v = (5, 4, 3, 2, 1)$*,* $ctr = (20, 10, 5, 2)$*; Experiment 4:* $v = (100, 90, 60, 30)$*,* $ctr = (4, 2, 1)$*; Experiment 5:* $v = (15, 10, 5)$*,* $ctr = (100, 3, 1)$*; Experiment 6:* $v = (16, 15, 14)$*,* $ctr = (3, 2, 1)$*. Columns 2-5 report the simulation results for the experiments using Q-learning AI algorithm with baseline initialization that only requires knowledge of players valuations, columns 6-9 correspond to the results using Q-learning AI algorithm with informed initialization, whereas 6-9 correspond to the results using Q-learning AI algorithm with random initialization. For all initializations we report the average across runs limit strategies (bids) for each of the players in the decreasing order of valuations; the percent of all runs that converge to an efficient outcome (the bids are alligned with the valuations); the percent of runs that converge to Nash equilibrium; the average auctioneer revenue across runs as well as the 95% confidence interval for the revenue in squared brackets.*

5.2 VCG with independent bidders

Next, we compare the algorithmic results for the VCG auction with the ones for the GSP. Importantly, we find that under VCG revenues are either close or lower than those under GSP auctions.

Table 4: Bids, Efficiency and Revenues in the Six Experiments - VCG Case

	Independent Players										
	EOS		Q-learners: GSP				Q-learners: VCG				
	Bids	Revenue	Bids	Eff.	NE	Revenue	Bids	Eff.	NE	Revenue
Exp. 1	$(b_1, 1)$	2	(1.87, 0.65)	100%	100%	1.3	(1.87, 0.63)	100%	100%	1.27
						[1.23, 1.37]				[1.22, 1.31]
Exp. 2	$(b_1, 1.6, 1)$	10	(2.22, 1.66, 0.90)	98%	100%	9.15	(2.37, 1.78, 0.86)	98%	100%	8.76
						[9.0, 9.3]				[8.55, 8.98]
Exp. 3	$(b_1, 3.15, 2.3, 1.6, 1)$	96	(3.90, 3.16, 2.45, 1.64, 0.87)	93%	87.5%	98.06	(4.29, 3.73, 2.85, 1.99, 0.91)	96%	58%	90.9
						[97.4, 98.72]				[89.95, 91.84]
Exp. 4	$(b_1, 67.5, 45, 30)$	390	(74.1, 66.7, 48.3, 24.8)	80%	55%	403.6	(83.04, 78.18, 55.15, 27.92)	80%	38%	348.91
						[393.5, 413.7]				[343.9, 353.92]
Exp. 5	$(b_1, 9\frac{4}{5}, 3\frac{1}{3})$	990	(11.8, 8.7, 4.4)	100%	100%	884.7	(11.84, 8.79, 4.92)	100%	100%	871.24
						[867.7, 901.7]				[856.104, 886.367]
Exp. 6	$(b_1, 9\frac{2}{3}, 7)$	43	(11.7, 10.6, 10.3)	40%	0%	52.36	(14.01, 13.36, 12.2)	58%	0%	37.53
						[51.67, 53.05]				[36.84, 38.23]

Notes: Experiment 1: $v = (2, 1)$, $ctr = 2$; *Experiment 2*: $v = (3, 2, 1)$, $ctr = (5, 2)$; *Experiment 3*: $v = (5, 4, 3, 2, 1)$, $ctr = (20, 10, 5, 2)$; *Experiment 4*: $v = (100, 90, 60, 30)$, $ctr = (4, 2, 1)$; *Experiment 5*: $v = (15, 10, 5)$, $ctr = (100, 3, 1)$; *Experiment 6*: $v = (16, 15, 14)$, $ctr = (3, 2, 1)$. *In columns 2 and 3 the EOS equilibrium bids and auctioneer revenues are presented as a benchmark. Columns 4-7 report the simulation results for the experiments using the Q-learning AI algorithm, whereas columns 8-11 correspond to the results using the Neural Network approach. For both approaches we report the average across runs limit strategies (bids) for each of the players in the decreasing order of valuations; the percent of all runs that converge to an efficient outcome (the bids are alligned with the valuations); the percent of runs that converge to Nash equilibrium; the average auctioneer revenue across runs as well as the 95% confidence interval for the revenue in squared brackets.*

As in the previous section, we additionally investigate results under "more informative" initialization of the Q-learners.

Table 5: Q-learning with Informed Initialization - VCG Case

	Independent Players							
	Q-learners: GSP (Informed Initialization)				Q-learners: VCG (Informed Initialization)			
	Bids	Eff.	NE	Revenue	Bids	Eff.	NE	Revenue
Exp. 1	(1.92, 0.67)	100%	100%	1.34	(1.86, 0.63)	100%	100%	1.26
				[1.30, 1.38]				[1.18,1.34]
Exp. 2	(2.37, 1.49, 0.78)	100%	100%	8.99	(2.71, 1.52, 0.73)	100%	100%	7.50
				[8.75,9.23]				[7.27,7,7.72]
Exp. 3	(3.84, 2.99, 2.21, 1.31, 0.59)	100%	100%	89.67	(4.87, 3.86, 2.84, 1.91, 0.88)	100%	100%	91.25
				[88.95,90.39]				[90.63,91.86]
Exp. 4	(74.09, 64.19, 44.40, 24.75)	100%	100%	370.31	(88.76, 80.72, 56.90, 29.87)	100%	90%	364.84
				[368.94,371.68]				[364.00,365.68]
Exp. 5	(14.00, 8.21, 3.74)	100%	100%	857.28	(14.47, 8.915.01)	100%	100%	851.39
				[849.93,864.63]				[844.72,858.07]
Exp. 6	(8.70, 8.16, 7.86)	100%	0%	39.63	(14.19, 13.79, 13.31)	100%	0%	40.41
				[39.16,40.10]				[31.16,49.66]

5.3 Extension 1: Introduction of the noise bidder

In this subsection, we consider the same settings as in the experiments with independent players above, but add an additional bidder who at every iteration of the algorithm chooses the bid among all possible bids randomly, a noise bidder in the spirit of Hashimoto (2010).

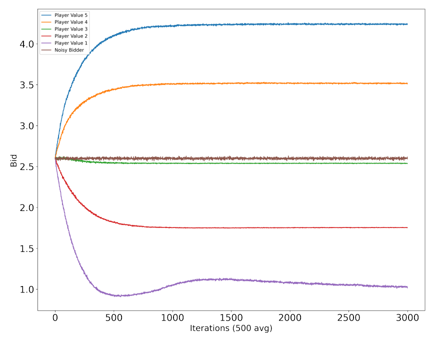
Table 6: The Six Experiments with an Extra Noise Bidder

	With an Extra Noise Bidder					
	Q-learners			Neural-learners		
	Convergence	Eff.	Revenue	Convergence	Eff.	Revenue
Exp. 1	(1.93, 0.96)	100%	3.04	(2.20, 1.04)	100%	3.31
			[2.76,3.32]			[3.02,3.60]
Exp. 2	(2.70, 1.93, 1.10)	100%	13.74	(2.63, 2.10, 1.16)	100%	14.45
			[13.05,14.43]			[14.15,14.75]
Exp. 3	(4.48, 3.74, 2.81, 1.87, 1.15)	100%	124.69	(4.19, 3.13, 2.56, 2.07, 0.81)	100%	112.03
			[119.77,129.61]			[109.84,114.22]
Exp. 4	(89.73, 81.94, 58.03, 30.43)	100%	504.48	(88.09, 75.0, 60.0, 28.84)	100%	488.63
			[485.33,523.63]			[481.77,495.50]
Exp. 5	(13.52, 9.47, 5.72)	100%	1088.72	(14.80, 8.20, 5.00)	100%	1025.41
			[1032.88,1144.56]			[987.07,1063.74]
Exp. 6	(14.85, 14.06, 13.19)	100%	79.14	(14.07, 12.61, 12.65)	100%	74.58
			[77.42,80.87]			[73.85,75.30]

Notes: Experiment 1: $v = (2, 1)$, $ctr = 2$; *Experiment 2*: $v = (3, 2, 1)$, $ctr = (5, 2)$; *Experiment 3*: $v = (5, 4, 3, 2, 1)$, $ctr = (20, 10, 5, 2)$; *Experiment 4*: $v = (100, 90, 60, 30)$, $ctr = (4, 2, 1)$; *Experiment 5*: $v = (15, 10, 5)$, $ctr = (100, 3, 1)$; *Experiment 6*: $v = (16, 15, 14)$, $ctr = (3, 2, 1)$. *Columns 2-4 report the simulation results for the experiments using the Q-learning AI algorithm, whereas columns 5-7 correspond to the results using the Neural Network approach. For both approaches we report the average across runs limit strategies (bids) for each of the players in the decreasing order of valuations; the percent of all runs that converge to an efficient outcome (the bids are alligned with the valuations); the average auctioneer revenue across runs as well as the 95% confidence interval for the revenue in squared brackets.*

We find that as predicted by the theory both in the case of Q-learning and the Neural Network approach, the introduction of the noisy bidder dramatically increases efficiency and leads to an efficient outcome for each run of each of the experiments. For Experiment 3, we explore the convergence of the Q-learning algorithm using $T = 1,500,000$ iterations. The hyperparameters are: $\delta = 0.95$, $\beta = 1e - 05$, and $\alpha = 0.5$. We plot the average results over 50 runs in Figure 6. In comparison to the behavior without the non-strategic bidder, the convergence of the algorithm is faster.

Figure 6: Experiment 3, with an Extra Noise Bidder



Notes: Bidding behavior of each of the single players as well as additional 'noise bidder' that use Q-learning algorithm as a function of the iterations in Experiment 3 described in Section 4. We plot moving averages over 500 iterations.

5.4 Extension 2: delegated bidding to an agency

In this section in contrast to the previous experiments in which every player was bidding independently, players 1 and 3 (player with the highest valuation and the third-highest) bid through an intermediary, with a single Q-matrix and a total reward equal to the sum of both players' rewards. The results for all of the experiments 1-6 with this additional condition are summarized in Table 7.

Table 7: Simulations results for the GSP experiments with a coalition $C = \{1, 3\}$.

	Agency Bidding									
	UC-RAE		Q-learners				Neural-learners			
	Convergence	Revenue	Convergence	Eff.	NE	Revenue	Convergence	Eff.	NE	Revenue
Exp. 2	$(b_1, 1.2, 0)$	6	$(2.04, 1.76, 0.70)$	84%	82%	10.00	$(2.11, 1.38, 0.53)$	98 %	94%	7.88
						[9.61,10.40]				[1.21,14.55]
Exp. 3	$(b_1, 2.9, 1.8, 1.6, 1)$	86	$(3.65, 3.31, 1.88, 1.82, 0.88)$	42%	26%	94.89	$(3.72, 2.83, 1.98, 1.54, 0.80)$	90%	82%	83.76
						[93.57,96.20]				[77.01,90.43]
Exp. 4	$(b_1, 60, 30, 30)$	330	$(72.43, 67.89, 35.93, 26.46)$	70%	56%	362.03	$(75.75, 57.60, 39.67, 29.70)$	90%	84%	329.43
						[354.20,369.86]				[322.76,322.76]
Exp. 5	$(b_1, 9.7, 0)$	970	$(10.96, 8.73, 4.77)$	98%	88%	887.05	$(12.68, 7.91, 4.25)$	100%	100%	803.38
						[864.31,909.80]				[791.58,815.17]
Exp. 6	$(b_1, 5, 0)$	15	$(11.94, 10.57, 10.29)$	38%	0%	52.26	$(10.83, 7.82, 5.32)$	95%	55%	34.01
						[51.56,52.97]				[23.46,44.58]

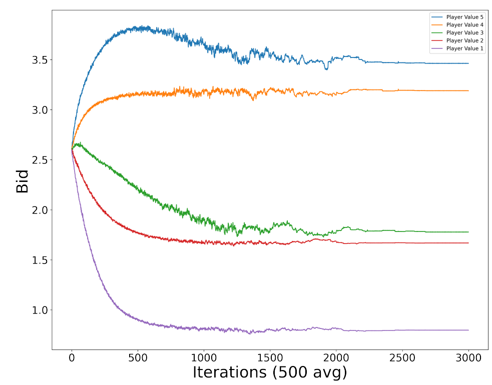
Notes: Experiment 1: $v = (2, 1)$, $ctr = 2$; Experiment 2: $v = (3, 2, 1)$, $ctr = (5, 2)$; Experiment 3: $v = (5, 4, 3, 2, 1)$, $ctr = (20, 10, 5, 2)$; Experiment 4: $v = (100, 90, 60, 30)$, $ctr = (4, 2, 1)$; Experiment 5: $v = (15, 10, 5)$, $ctr = (100, 3, 1)$; Experiment 6: $v = (16, 15, 14)$, $ctr = (3, 2, 1)$. Coalition in all the experiments is $C = \{1, 3\}$.
In columns 2 and 3 the UC-RAE equilibrium bids and auctioneer revenues are presented as a benchmark. Columns 4-7 report the simulation results for the experiments using the Q-learning AI algorithm, whereas columns 8-11 correspond to the results using the Neural Network approach. For both approaches we report the average across runs limit strategies (bids) for each of the players in the decreasing order of valuations; the percent of all runs that converge to an efficient outcome (the bids are aligned with the valuations); the percent of runs that converge to Nash equilibrium; the average auctioneer revenue across runs as well as the 95% confidence interval for the revenue in squared brackets.

Similar to the case of independent players, the auctioneer revenues under the Neural Network approach are always lower than using the Q-learning algorithm, even though the limit strategy of the player with the highest valuation is higher for the Neural-learner.

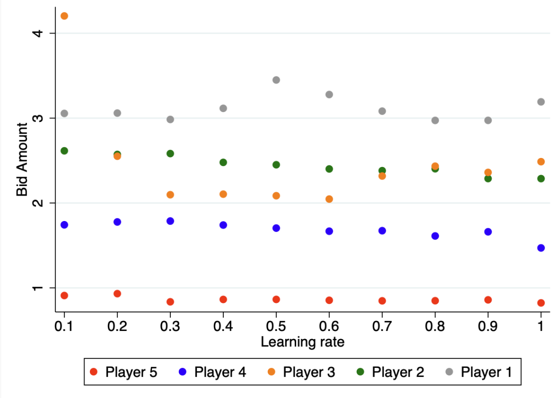
Q-learner: Again, next we consider in detail our leading Example (Experiment 3). We define the action space for each of the player on the interval $[B_{min}, B_{max}] = [0.2, 5]$, with $k = 13$ possible actions so that the step between the actions is 0.4. We explore the convergence of the algorithm using $T = 1,500,000$ iterations. The hyperparameters are: $\delta = 0.95$, $\beta = 1e - 05$, and $\alpha = 0.2$. We plot the average results over 50 runs in panel (a) of Figure 7. In comparison, recall that the bids in UC-RAE equilibrium are: $b_5 = 1$, $b_4 = 1.6$, $b_3 = 1.8$ and $b_2 = 2.9$.

Figure 7: Agency Bidding

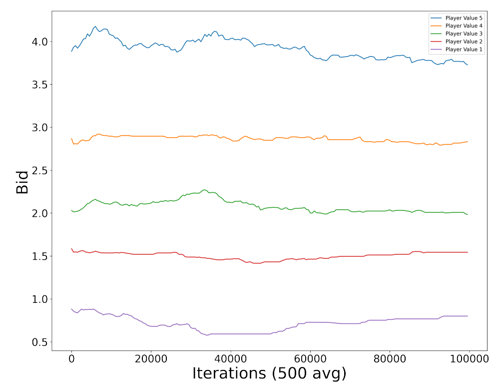
(a) Experiment 3 with agency - Q-learners



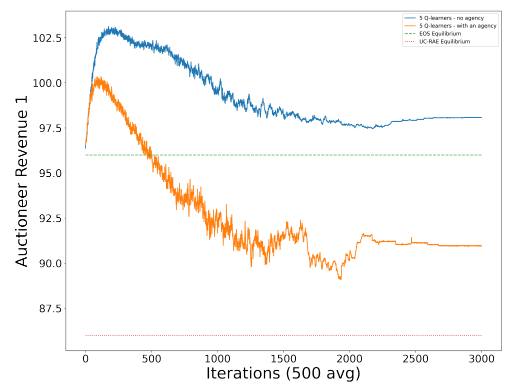
(b) Experiment 3 with agency - Learning rate



(c) Experiment 3 with agency - Neural Networks



(d) Experiment 3 - Auctioneer revenue with and without agency



Notes: Panels a and c represent respectfully: bidding behavior of each of the single players that use Q-learning algorithm, and bidding behavior of each of the single players that use Neural Network algorithm, as a function of the iterations in Experiment 3 in which, additionally, player with value 5 and player with value 3 bid under common intermediary. We plot moving averages over 500 iterations. Panel b report the bids at convergency as a function of the learning rate α for Q-learning algorithm. Panel d report the auctioneer revenues as a function of the iterations in case when players use Q-learning algorithm with and without an agency.

Moreover, we explore the behavior of the AI algorithm under different learning rates α. Recall that this parameter determines to what extent the new information substitutes the old. Panel (b) of Figure 7 depicts the optimal bids of the players to which the Q-learning algorithm converged given different values of α. In the next iteration of the paper, we plan to explore the dependence on the hyperparameters in greater detail.

Neural Network: We reach similar conclusions for the case with an agency bidding in Experiment 3 using the Neural Network approach. The hyperparameters are: $N_{nr} = 20$, $N_h = 1$, $lr = 0.01$. Instead of $T = 1,500,000$ iterations as in case with Q-learners, we consider $T = 100,000$ iterations. As we can see, in panel (c) of Figure 7 the limit values are not that different from the ones of the Q-learning algorithm, apart from the player with the second-highest valuation. Moreover, the convergence is much faster compared to the Q-learning algorithm.

Finally, we compare the results of Experiments 3 with and without an agency in terms of auctioneer revenue. Our findings indicate that bidding through the intermediary leads to the lower auctioneer revenue compared to the case of individual bidding which can be seen in the following panel (d) of Figure 7. This chart represents the auctioneer revenue as a function of the number of AI algorithm iterations. The blue path corresponds to Experiment 3 with independent bidding, whereas the orange line – to the case in which two agents play through the common intermediary. In comparison, the equilibrium auctioneer revenue in EOS is 96 and in UC-RAE 86.

Moreover, as can be seen from Table 1 compared to Table 7, the auctioneer revenues in the case of agency bidding are lower than in the case of independent players both using the Q-learning and Neural Network approaches for almost all of the scenarios.

5.5 Extension 3: information asymmetry

The platform might have an incentive and certainly has an opportunity to reveal different amount of information to different advertisers. It is possible to model the asymmetry of information between the bidders in many different ways. In this section in contrast to the previous experiments in which every player was bidding using the same algorithm, we consider the case in which one bidder is more informed than the other. In particular, in the case of Q-learner, one bidder uses informed initialization defined in Section 5.1, whereas all other bidders use baseline initialization. The results for all variations of experiment 3 in which only one bidder is more informed are summarized in Table 8.

Table 8: Bids, Efficiency, Individual Rewards, and Revenues in Experiment 3 - GSP

	Independent Players - Q-learners with Asymmetric Initializations				
	Bids	Individual Rewards	Eff.	NE	Revenue
Exp. 3 bidder 1	$(3.9, 3.06, 2.34, 1.69, 1.0)$	$(35.65, 11.66, 5.86, 2.26, 0.26)$	90.0%	82.0%	97.81
					[93.46, 102.15]
Exp. 3 bidder 2	$(3.8, 3.06, 2.28, 1.71, 1.02)$	$(33.45, 17.39, 5.46, 2.18, 0.27)$	94.0%	80.0%	93.82
					[90.68, 96.95]
Exp. 3 bidder 3	$(3.61, 3.02, 2.3, 1.59, 0.93)$	$(31.38, 17.09, 8.6, 2.33, 0.39)$	88.0%	86.0%	88.9
					[83.93, 93.86]
Exp. 3 bidder 4	$(3.59, 3.02, 2.36, 1.55, 0.79)$	$(28.89, 16.28, 8.68, 3.72, 0.43)$	78.0%	62.0%	89.38
					[83.54, 95.21]
Exp. 3 bidder 5	$(3.81, 3.25, 2.41, 1.79, 0.84)$	$(28.37, 15.02, 6.48, 2.87, 0.35)$	90.0%	78.0%	97.04
					[92.01, 102.07]

Notes: Experiment 3 bidder i: $v = (5, 4, 3, 2, 1)$, $ctr = (20, 10, 5, 2)$; *bidder i uses a Q-learner with informed initialization. We report the average across runs limit strategies (bids) for each of the players in the decreasing order of valuations; average across runs rewards for each of the players in the decreasing order of valuations; the percent of all runs that converge to an efficient outcome (the bids are alligned with the valuations); the percent of runs that converge to Nash equilibrium; the average auctioneer revenue across runs as well as the 95% confidence interval for the revenue in squared brackets.*

In addition to the results reported in previous tables, we also report the average individual rewards for each of the players. The first row corresponds to the case in which only bidder 1 is more informed, row two - to the with more informed bidder 2, and so on. Once we look at the individual rewards of bidder i across the rows, we observe that for all the bidders, the case with the highest reward is the one in which this bidder is more informed than the others.[6] For example, bidder 1 gets a reward 35.65 in case when he is the more informed player, and that reward is the highest for him across all other scenarios. As a result, we conclude that more informed players get higher rewards.

We are also working on simulations that are based on different assumptions on information asymmetry. One corresponds to a different state variable: we assume that the player observes only his own bid and the bid below, corresponding to the price he pays, and not all the bids. Moreover, uncertainty about the click-through-rates for some players is another way to model asymmetry.[7]

6 Conclusions

In this study, we analyzed the role of AI in search advertising auctions. Through computational experiments, we evaluated the performance of bidding algorithms powered by AI across multiple, stylized auction games. We have three main results. First, in terms of the informational assumptions, we find that the more information available to the algorithms the better it is for the efficiency of the allocations. Second, the more information is available to the algorithms, the better of the advertisers are, and the lower the auctioneer

[6] Except for bidder 5 which expects to get zero reward and gets a positive one only due to inefficient outcomes.
[7] See, for example, Bergemann et al. (2021).

revenue is. Third, algorithmic bidding has a tendency to sustain low bids both under the GSP and VCG.

These results highlight several important features regarding how AI might shape the working of the ad auctions in terms of both efficiency of the ad slot allocation and the transfer of revenues between advertisers and the selling platform. Moreover, our findings might help to understand recent decisions by the dominant selling platforms. In the context of search advertising, advertisers on Google have experienced a reduction in the amount of type of data available to optimize their bidding strategies. This deliberate increase in the extent of *data fogginess*, while responding to the growing privacy concerns among search users, creates a trade-off with market competition as the AI tools offered by the platform itself are at an advantage relative to those available to advertisers (and their intermediaries). In future iterations of this study, we plan to even more directly address this risk by implementing versions of the experiments analyzed in this paper where competing algorithms are fed with a heterogeneous amount of information, thus simulating the informational advantage of the platform's AI tools. Moreover, while under the stylized version of the GSP auction described above bidders are ranked by their bids, on search engines like Google, advertisers are ranked across keywords by the product of their bid and a quality score that embeds Google choices. Quality scores might be used to alter the outcomes of bidders' competition, for instance, to try to prevent collusion by adjusting quality scores in ways that would make bid coordination less effective. Thus, we plan to investigate that as well.

References

Agrawal, Ajay, Joshua Gans, and Avi Goldfarb. 2018. *Prediction machines: the simple economics of artificial intelligence.* Harvard Business Press.

Asker, John, Chaim Fershtman, and Ariel Pakes. 2021. "Artificial Intelligence and Pricing: The Impact of Algorithm Design."

Assad, Stephanie, Robert Clark, Daniel Ershov, and Lei Xu. 2020*a*. "Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market."

Assad, Stephanie, Robert Clark, Daniel Ershov, and Lei Xu. 2020*b*. "Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market."

Athey, Susan, and Denis Nekipelov. 2014. "A Structural Model of Sponsored Search Advertising Auctions." *mimeo.*

Balseiro, Santiago R., Omar Besbes, and Gabriel Weintraub. 2015. "Repeated Auctions with Budgets in Ad Exchanges: Approximations and Design." *Management Science*, 61(4): 864–884.

Banchio, Martino, and Andrzej Skrzypacz. 2022. "Artificial Intelligence and Auction Design." *Available at SSRN 4033000.*

Bergemann, Dirk, Paul Duetting, Renato Paes Leme, and Song Zuo. 2021. "Calibrated Click-Through Auctions: An Information Design Approach." *arXiv preprint arXiv:2105.09375.*

Borgers, Tilman, Ingemar Cox, Martin Pesendorfer, and Vaclav Petricek. 2013. "Equilibrium Bids in Sponsored Search Auctions: Theory and Evidence." *American Economic Journal: Microeconomics*, 5(4): 163–187.

Brown, Zach Y., and Alexander MacKay. 2021. "Competition in Pricing Algorithms." *NBER Working Paper.*

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. 2020*a*. "Artificial Intelligence, Algorithmic Pricing, and Collusion." *American Economic Review*, 110(10): 3267–97.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò, Joseph E. Harrington, and Sergio Pastorello. 2020*b*. "Protecting consumers from collusive prices due to AI." *Science*, 370(6520): 1040–1042.

Celis, L. Elisa, Gregory Lewis, Markus Mobius, and Hamid Nazerzadeh. 2015. "Buy-It-Now or Take-a-Chance: Price Discrimination Through Randomized Auctions." *Management Science*, 60(12): 2927–2948.

Che, Yeon-Koo, Syngjoo Choi, and Jinwoo Kim. 2017. "An experimental study of sponsored-search auctions." *Games and Economic Behavior*, 102: 20 – 43.

Decarolis, Francesco, and Gabriele Rovigatti. 2021. "From Mad Men to Maths Men: Concentration and Buyer Power in Online Advertising." *American Economic Review*, forthcoming.

Decarolis, Francesco, Maris Goldmanis, and Antonio Penta. 2020. "Marketing Agencies and Collusive Bidding in Online Ad Auctions." *Management Science*, Vol.66, No 10, 4433-4454.

Dütting, Paul, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. 2019. "Optimal auctions through deep learning." 1706–1715, PMLR.

Edelman, Benjamin, Michael Ostrovsky, and Michael Schwarz. 2007. "Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords." *American Economic Review*, 97(1): 242–259.

Einav, Liran, Chiara Farronato, Jonathan Levin, and Neel Sundaresan. 2018. "Auctions versus Posted Prices in Online Markets." *Journal of Political Economy*, 126(1): 178–215.

Gomes, Renato, and Kane Sweeney. 2014. "Bayes-Nash Equilibria of the Generalized Second Price Auction." *Games and Economic Behavior*, 86: 421–437.

Hashimoto, Tadashi. 2010. "Equilibrium selection, inefficiency, and instability in internet advertising auctions." *Inefficiency, and Instability in Internet Advertising Auctions (June 8, 2010)*.

Kingma, Diederik P, and Jimmy Ba. 2014. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980.*

Milgrom, Paul, and Joshua Mollner. 2014. "Equilibrium selection in multi-player games with auction applications." Citeseer.

Milgrom, Paul R, Steven Tadelis, et al. 2019. "How artificial intelligence and machine learning can impact market design." *The economics of artificial intelligence: An agenda*, 567–585.

Nedelec, Thomas, Noureddine El Karoui, and Vianney Perchet. 2019. "Learning to bid in revenue-maximizing auctions." 4781–4789, PMLR.

Nowe, Ann, Peter Vrancx, and Yann-Michael De Hauwere. 2012. *Game Theory and Multi-agent Reinforcement Learning.*

OECD. 2017. *Algorithms and Collusion.* OECD Background paper.

Varian, Hal. 2007. "Position auctions." *International Journal of Industrial Organization*, 25(6): 1163–1178.

Appendices

A Results for experiments with zero discount rate

The auctioneer revenues in case with $\delta = 0$ are slightly lower than in case with $\delta = 0.95$, but they still comparable with the revenues given by the Neural Network approach.

Table 9: Comparison between Q-learners with 15M iterations with different initializations and $\delta = 0$

	Independent Players - GSP											
	Q-learners: Baseline Gaussian Initialization				Q-learners: Informed Initialization				Q-learners: Random Initialization			
	Bids	Eff.	NE	Revenue	Bids	Eff.	NE	Revenue	Bids	Eff.	NE	Revenue
Exp. 1	$(1.91, 0.56)$	100.0%	100.0%	1.08	$(1.93, 0.55)$	100.0%	100.0%	1.05	$(1.9, 0.57)$	100.0%	100.0%	1.09
				[0.85, 1.31]				[0.82, 1.29]				[0.86, 1.33]
Exp. 2	$(2.25, 1.55, 0.84)$	100.0%	100.0%	9.31	$(2.25, 1.54, 0.84)$	100.0%	100.0%	9.27	$(2.24, 1.55, 0.85)$	100.0%	100.0%	9.31
				[8.78, 9.85]				[8.74, 9.8]				[8.79, 9.84]
Exp. 3	$(3.82, 2.9, 2.17, 1.56, 0.85)$	100.0%	100.0%	89.1	$(3.83, 2.9, 2.17, 1.57, 0.86)$	100.0%	100.0%	89.23	$(3.81, 2.89, 2.17, 1.54, 0.85)$	100.0%	100.0%	88.84
				[85.56, 92.64]				[85.69, 92.78]				[85.21, 92.47]
Exp. 4	$(74.92, 62.01, 44.78, 23.25)$	100.0%	100.0%	349.35	$(74.87, 61.72, 44.51, 23.2)$	100.0%	100.0%	347.75	$(74.37, 61.78, 44.38, 23.65)$	100.0%	100.0%	347.18
				[334.15, 364.56]				[332.5, 362.99]				[331.45, 362.91]
Exp. 5	$(12.33, 7.01, 2.38)$	100.0%	100.0%	709.12	$(12.4, 6.94, 2.39)$	100.0%	100.0%	702.67	$(12.33, 7.06, 2.51)$	100.0%	100.0%	713.48
				[639.5, 778.73]				[633.56, 771.79]				[645.11, 781.84]
Exp. 6	$(9.67, 8.37, 7.49)$	100.0%	2.0%	37.45	$(10.11, 8.57, 6.01)$	100.0%	100.0%	35.59	$(9.93, 8.47, 6.62)$	100.0%	100.0%	36.18
				[36.2, 38.7]				[34.14, 37.05]				[34.75, 37.6]

Notes: Experiment 1: $v = (2, 1)$*,* $ctr = 2$*; Experiment 2:* $v = (3, 2, 1)$*,* $ctr = (5, 2)$*; Experiment 3:* $v = (5, 4, 3, 2, 1)$*,* $ctr = (20, 10, 5, 2)$*; Experiment 4:* $v = (100, 90, 60, 30)$*,* $ctr = (4, 2, 1)$*; Experiment 5:* $v = (15, 10, 5)$*,* $ctr = (100, 3, 1)$*; Experiment 6:* $v = (16, 15, 14)$*,* $ctr = (3, 2, 1)$*. Columns 2-5 report the simulation results for the experiments using Q-learning AI algorithm with baseline initialization that only requires knowledge of players valuations, columns 6-9 correspond to the results using Q-learning AI algorithm with informed initialization, whereas 6-9 correspond to the results using Q-learning AI algorithm with random initialization. For all initializations we report the average across runs limit strategies (bids) for each of the players in the decreasing order of valuations; the percent of all runs that converge to an efficient outcome (the bids are alligned with the valuations); the percent of runs that converge to Nash equilibrium; the average auctioneer revenue across runs as well as the 95% confidence interval for the revenue in squared brackets.*