Algorithmic Competition and Informational Advantage in Digital Markets: Evidence from Search Auctions

Francesco Decarolis, Tommaso Pellegrinetti, Gabriele Rovigatti, Michele Rovigatti, Ksenia Shakhgildyan*

October 31, 2024

Abstract

This paper examines how proprietary algorithms used by dominant digital platforms create informational advantages in search auctions, reshaping market competition. Using experimental evidence and counterfactual simulations, we quantify the impact of algorithmic bidding on auction outcomes and competitive dynamics. Our findings reveal how platforms can leverage superior information to significantly improve their revenues, distorting competition and creating welfare losses for independent advertisers. We also show why platforms prefer selling a bidding algorithm service over directly selling data. These results highlight the need for greater scrutiny of algorithmic decision-making in platform markets, offering new insights for competition policy in digital economies.

Keywords: Online Advertising, Sponsored Search Auctions, Algorithmic Bidding, Artificial Intelligence, Collusion.
JEL Codes: C73, D82, D83, D18, D44.

*Francesco Decarolis, Bocconi University and IGIER, *francesco.decarolis@unibocconi.it*, Tommaso Pellegrinetti, Bocconi University, *tommaso.pellegrinetti@studbocconi.it*, Gabriele Rovigatti, Bank of Italy and CEPR, *gabriele.rovigatti@bancaditalia.it*, Michele Rovigatti, Bocconi University, *mic.rovigatti@gmail.com*, Ksenia Shakhgildyan, Bocconi University and IGIER, *ksenia.shakhgildyan@unibocconi.it*. This paper is part of a project, Competition in Digital Markets, that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No: 101002867).

1 Introduction

The increasing dominance of digital platforms in modern economies has brought forth fundamental challenges for competition policy and market regulation. Platforms like Google, Amazon, and Facebook operate as central intermediaries across various industries, leveraging their market position to collect vast amounts of data and use them to train sophisticated algorithms. These proprietary algorithms are integral to their business models, enabling them to optimize outcomes across markets —from search auctions to e-commerce— by exploiting informational advantages that are largely inaccessible to other market participants.

As digital platforms continue to consolidate their power, concerns about market efficiency, fairness, and long-term competition have come to the forefront. In particular, search auctions, where platforms such as Google match advertisers with user queries, have become the battleground where informational asymmetries play a crucial role. Google's dominant position in the search advertising market raises important questions about how these platforms use their proprietary algorithms to influence the advertisers' bidding behavior and shape auction outcomes. While they offer unparalleled efficiency and performance by optimizing bids based on superior information, delegation might be costly for the advertisers due to the misalignment of incentives, stifling competition, and increased dependence. Platforms no longer operate solely as auctioneers and marketplaces but also as essential providers of bidding tools to advertisers, potentially skewing auction outcomes in their favor while harming advertisers.

This paper investigates the role of algorithmic competition in search auctions, focusing on how the informational advantage of platforms affects market outcomes. While existing research has explored the mechanics of search auctions and the economics of information, we are not aware of other studies that directly examine how the deployment of advanced algorithms by platforms alters competitive dynamics. This gap is critical, given the increasing reliance of advertisers on algorithmic bidding and the opaque nature of the platforms' data advantage. In this respect, our paper offers a novel insight on the economic value of nonrival data (Jones and Tonetti [2020]).

We address this gap through a quasi-experimental strategy that uses simulations to quantify the impact of the deployment of platform bidding services on auction outcomes in search markets. We focus on search auctions, which are the most lucrative portion of digital advertising and account for more than 40% of the market. This choice is particularly suitable for our analysis because search auctions have three key characteristics: first, in this segment, players participate through algorithms that optimize their campaigns on various dimensions; second, most players delegate their bidding to specialized intermediaries, which lowers platforms' revenues; third, to decrease advertisers' and intermediaries' ability to implement effective AIAs, most platforms have obfuscated access to relevant data[1] and started offering their own services that leverage their

[1]See Decarolis et al. [2023] for a description of the effects of data obfuscation on advertisers' bidding strategies,

superior access to the data. For example, since its introduction in 2021, the share of advertisers that choose to delegate bidding to Google Performance Max algorithm grew dramatically to 80% already in 2022.[2]

Measuring the impact of AI bidding tools is virtually impossible using observational data, as there is no possibility to assess counterfactual scenarios, and platforms do not disclose their algorithms or experimental real-world data. In line with a growing literature at the intersection of economics and computer science, we use a simulation approach to compute counterfactual scenarios and assess the impact of different policies on the welfare of advertisers and platforms.[3] In doing that, we offer the methodological contribution of the paper. We provide a comprehensive analysis of the convergence results of various types of AIAs interacting and learning simultaneously in a repeated auction game.

In all our simulated exercises, we consider the case of one keyword auction played repeatedly by bidders that are symmetric in terms of their valuations of the ad, but deploy asymmetrically informed AIAs. Different classes of AIAs are feasible, depending on the amount of data revealed to the advertisers. In terms of information asymmetry, we focus on the information on the competitors' bids. This is motivated by the striking difference between the auction design across otherwise similar platforms. As an example, Yahoo! used to reveal all the bids to its advertisers, whereas Google became more and more restrictive in the amount of information disclosed, and now keeps the bids private.[4] More broadly, when AIAs are employed to bid in auctions, especially in repeated or sequential auction environments like those often found in online advertising, financial markets, or electricity markets, the information on competitor bids from past periods becomes crucial. This historical data plays a significant role in shaping bidding strategies, optimizing outcomes, and navigating the competitive landscape. By conditioning on competitors' bids, AIAs can adapt their bidding strategies over time. As a result, both from the market perspective, but also because of the straightforward link to the theory of dynamic games, the information on competitor bids is of primary interest.

We focus on the Generalized Second-Price (GSP) auctions, the mechanism through which the allocation of ad slots is organized in most of the large online platforms for search advertising, such as Google and Bing. This market design differs from that of display advertising that transited from being organized as a second-price auction in the past to the currently used first-price auction

and Bergemann and Bonatti [2015] on the role of data completeness to predict match quality for online advertising.

[2]For example, according to Google, advertisers who use Performance Max achieve on average over 18% more conversions at a similar cost per action. See New ways to multiply your results with Performance Max, Google Ads, February 2023. Indeed, since its introduction in 2021, the share of advertisers that choose to delegate bidding to Google Performance Max algorithm grew dramatically to 80% already in 2022. See Bidding for success, Think with Google, May 2022.

[3]See Calvano et al. [2020], Decarolis and Rovigatti [2021], Chen, Mislove and Wilson [2016], Competition and Authority [2018], Klein [2021], Brown and MacKay [2023], Banchio and Mantegazza [2022], Lambin [2023] and Mehta and Perlroth [2023], including papers that specifically focus on AI bidding auctions: Dütting et al. [2017], Martin et al. [2024], Bichler, Fichtl and Heidekrüger [2021] and Banchio and Skrzypacz [2023].

[4]See the discussion in [Lahaie, 2006].

system.[5]

In terms of the AIAs, we employ Q-learning algorithms to take advantage of the transparency inherit in their learning process. These algorithms exemplify a particular application of reinforcement learning techniques, and the challenge of bidding in ad auctions fundamentally embodies a reinforcement learning problem. They map into the framework of reinforcement learning due to the sequential and adaptive nature of the decision-making process, the need to learn from feedback on the rewards, as well as the challenge of balancing exploration with exploitation. This explains why reinforcement learning methods are increasingly being applied to automate and optimize bidding strategies in advertising auctions.[6] Moreover, Q-learning algorithms are often analyzed in the literature since they are characterized by parameters that have a straightforward economic interpretation, compared to the methods based on neural networks.

The amount of information disclosed by the platform and available for training determines two key design features of the algorithms: the *learning rule* and the *memory*. The former impacts the possibility of evaluating counterfactual scenarios, and the distinction to be made is between *Asynchronous* and *Synchronous* learning. *Asynchronous* learning allows the agent to learn only from actions that are actually taken, whereas *Synchronous* learning can be used when AIAs have the necessary information to calculate counterfactual scenarios ex-post. For example, when an AIA has information on the bids submitted by competitors, it is able to calculate the rewards that it would have received had it submitted different bids, and learn from these alternative choices. In terms of memory of past actions, reinforcement learning algorithms could be *StateFul* and *StateLess*. *StateFul* algorithms maintain a record of previous bids and use this information to update their future decisions. *StateLess* algorithms, on the other hand, do not retain information from previous steps. They make decisions regarding future actions based solely on the current rewards.

Hence, the type and amount of available data inform both the feasibility and the performance of the AIAs, generating four possible algorithms. These span from the less sophisticated tool with no memory and no counterfactual computation (*StateLess Asynchronous* or SLA, which is the only option for uninformed players) to the most refined one, which exploits the full access to data in order to learn from a full range of counterfactual cases (*StateFul Synchronous* or SFS). In our application, the platform-powered AIA always implements the latter, exploiting the full potential of its proprietary information; with it, the advertisers get a higher reward and the probability of winning than in the case of independent bidding - i.e., with the SLA - and have strong incentives to switch to the new system. Thus, bid delegation provides a significant advantage when utilizing the platform's services against a less sophisticated competitor.

[5]To be more specific, according to the recent DOJ case against Google, Google uses what is called randomized Generalized SP auction, see U.S. vs. Google antitrust trial.

[6]See, for example, "AI in Advertising: Real-Time Bidding & Reinforcement Learning."

However, if allowed to purchase the relevant data, the advertiser would opt for the *StateFul Asynchronous* (or SFA) algorithm, which results in even higher winning probability, cyclical patterns of convergence, and lower auctioneer's revenues than in case of bid delegation. This strongly disincentivizes the platforms from selling out the data. All this on top of the fact that, by not selling the data, the platform retains its exclusive access (now and in the future), keeps them within its "walled garden" and lowers potential competition. In this sense, our estimates reach a lower bound of the benefits of data-sharing policies from platforms. This analysis rationalizes why there is no option to buy the data from the platform in the real world, even though the players would be willing to pay for it. As a result, nearly 80% of players chose to delegate their bidding to, for example, Google's Performance Max.

In addition, we offer the methodological contribution. We first explain in detail the convergence results of various types of AIAs interacting and learning simultaneously in a repeated auction game. Moreover, we explain the origins of bid rotation (observed in symmetric games by e.g. Calvano et al. [2020]) in the case of asymmetrically informed algorithms, and the case in which they can or cannot be observed. The main channel that informs the width and the frequency of the cycles is the flow of information from the superior to the inferior AIA. When a more sophisticated player has access to competitors' bid data, in fact, its AIA manages to give feedback to the less sophisticated player and lowers auctioneer revenues.

The implications of our findings extend beyond the realm of search auctions. They raise important questions about the broader role of algorithms in digital markets, where informational advantages are increasingly concentrated in the hands of a few dominant players. As platforms continue to scale and diversify their operations, their ability to control market outcomes through proprietary algorithms poses significant challenges for regulators and policymakers tasked with ensuring fair competition. This paper thus contributes to the growing policy debate on how to regulate algorithmic competition in digital markets.[7] Our analysis suggests that current regulatory frameworks may need to be rethought to account for the unique challenges posed by platforms' informational dominance and the opaque nature of algorithmic decision-making. The results of this study underscore the need for greater transparency and oversight in algorithmic markets, where informational advantages have the potential to erode competition and consumer welfare over time.

The paper proceeds as follows: Section 2 presents the discussion of various algorithms used by the platforms; Section 3 presents the experimental design; Section 4 describes the main results, information asymmetries and platform's incentives; Section 5 digs deep into the drivers of the results; Section 6 presents multiple extensions of the main findings; Section 7 provides robustness checks; Section 8 concludes.

[7] See [Azoulay, Krieger and Nagaraj, 2024, Gans, 2024, Hagiu and Wright, 2024] on competition policy implications of the AI sector and its input market.

2 AI Tools Across Platforms

In the swiftly changing environment of digital markets such as e-commerce and digital advertising, businesses are finding AI-powered solutions for pricing optimization and ad bidding to be crucial for maintaining their competitive edge. Crucially, over the years, more and more sellers and advertisers have switched from running independent algorithms to using platform-sponsored solutions. These latter tools have leveraged the recent advancements in Artificial Intelligence and Machine Learning, offering increasingly complex and high-performing products. In this section, we provide an overview of them below.

E-commerce platforms like Amazon and eBay have integrated AI-driven pricing solutions, enabling sellers to automate pricing decisions and respond rapidly to market fluctuations.[8] The motivation is clear since these solutions help sellers maintain price competitiveness, capture demand fluctuations, and optimize revenue without the need for constant manual interventions.

AI-driven pricing is also being adopted by industries beyond the e-commerce sector. For instance, the ride-sharing platform Uber is using AI in its Surge Pricing model to adjust local ride prices, matching supply and demand on the platform.[9] Airlines are also experimenting with new applications of dynamic pricing that are based on AI models.[10] Walmart is testing AI workflows that delegate the whole negotiation process with some of its vendors.[11] Additionally, companies utilize AI algorithms to optimize price discounts calculations, helping them optimize their stock. A notable example of this is Booking.com which allows partners to dynamically select discounts to offer, ensuring the hotel remains consistently at the optimal capacity. Overall, industries that previously employed dynamic pricing are now transitioning to AI models, while those that were never exposed to such practices are increasingly experimenting with them.

Similarly, AI algorithms have transformed digital advertising by enabling advertisers to optimize two processes: bidding strategies and user targeting. The five largest online advertising platforms by market share - Google, Meta, Microsoft, Amazon, and TikTok - all offer automated bidding strategies.[12] Automation starts with asking advertisers to define a primary goal of their ad campaign, which could be, for example, increasing online sales, boosting ad views, or attracting high-value customers. Based on this objective, the platform then provides an automated bidding strategy tailored to achieve the specified goal. These algorithms learn from the auction environment and from data on a bid's performance to guide future bidding decisions. Google identifies these types of algorithms as "Automated Bidding Strategies" and offers two products: "Maximize

[8] See Amazon's Automated pricing tool and eBay's "Magical" Listing Tool.
[9] See Surge Pricing and Uber's AI Journey.
[10] See How Delta Air Lines Uses AI.
[11] See Bloomberg's Article.
[12] See Statista - Share of ad-selling companies in digital advertising revenue in the United States from 2020 to 2025.

clicks" and "Target impression".[13] Similar products are offered by Amazon ("Dynamic Bids") and Microsoft ("Performance Max", "Max Conversion", and "Max Value").[14]

Moreover, more advanced algorithms take into consideration additional information about the auction environment to determine which ads are most advantageous to bid on. Most platforms offer products that leverage user information to optimize their bidding strategies. Google refers to these as Smart Bidding strategies. These strategies consider various user-related factors, such as device type and location, whether the user has already been exposed to the advertisement campaign, the actual search query, and past behavior on the website or app. Notably, for certain ads, Google can also optimize bids based on competitors' product prices.[15] In contrast, Meta considers the predicted conversion rate for each specific user. This means that targeting decisions are made at the auction level, rather than the bid strategy level, ensuring that all bidding strategies (even those that have a fixed bid) incorporate this information.[16] Amazon offers automatic targeting on its website and can show the relevant ad to users interested in similar products, substitutes or complements. Targeting is only as effective as the amount of data an advertising platform can provide to the user. However, with the recent regulation on privacy phasing out third-party cookies from the web[17], cross-platform tracking has become increasingly challenging. As a result, companies that can offer first-party data are now at an advantage.[18]

Search plays a vital role in digital advertising, extending far beyond traditional search engines to encompass a wide range of online services: on platforms where users search for services that are often close substitutes, placement is crucial for sellers. Services like Glovo, Deliveroo, Apple Store, Uber Eats, and Expedia offer sellers the opportunity to improve the placement of their products in search results by purchasing ads. This process integrates ads closely with search results, making the ads appear very similar to the actual results. The assignment of these ad slots utilizes the same automated bidding and targeting strategies mentioned earlier to match ads with potential users better.

Full automation of the entire advertising campaign process is becoming more widespread. Platforms such as Google and Meta Ads now provide a variety of AI-powered tools that deliver fully autonomous services, combining automated bidding strategies and automated targeting. Google's Performance Max and Meta's Advantage+ utilize information about the advertiser's products, previous campaigns, and user base to automate the entire ads process, including selecting the

[13] See Google's Automated bid strategy definition.
[14] See Amazon's Bidding strategies for Sponsored Products and Microsoft Advertising Bid Management.
[15] See About Smart Bidding.
[16] About Meta's ad auctions.
[17] The two primary regulations governing the use of third-party cookies are the California Consumer Privacy Act (CCPA) in the United States and the General Data Protection Regulation (GDPR) in the European Union. Additionally, most major web browsers, except for Google, which is experimenting with blocking them, do not allow third-party cookies to be collected by default (See How Do Browsers Handle Third-Party Cookies?).
[18] See Digital Advertising: market data and analysis from Statista.

type and format of ads to display and selecting the optimal bid strategy.[19] These products operate across the various channels offered by each platform, by selecting the optimal set based on the users that the algorithm chooses to target. Those channels are Search, YouTube, Display, Discover, and Google Maps for Google, and Facebook, Instagram, and Messenger for Meta. Since the introduction of Performance Max' in 2021, the share of advertisers that choose to fully delegate bidding to Google Performance Max algorithm grew dramatically, up to 80% already in 2022.[20]

It is important to note that advertisers using most automated strategies usually do not incur additional fees. However, advertisers lose some flexibility in designing the campaign and bidding strategy. For tools that heavily utilize machine learning and AI, such as Google's Performance Max and Meta's Advantage+, advertisers are advised to use the service by ensuring their budget exceeds a minimum threshold or run their algorithm for a minimum required period. This recommendation ensures the algorithm has sufficient time to learn and optimize the campaign effectively, but it also reduces advertisers' flexibility to switch algorithms and platforms.

In addition, as Generative AI continues to advance, its use in digital advertising is becoming increasingly more common. It is estimated that as of 2023, 37% of the US Advertising Market uses generative AI.[21] Generative AI can be integrated into workflows using existing models like DALL-E, ChatGPT, Gemini, LLaMA, and others but there is also potential for Generative AI to replace most human intervention in the digital advertising processes, leaving humans to focus on the objectives of a campaign.[22] Both Meta and Google, as well as Amazon, are including the possibility to use generative AI for advertising.[23]

As the digital advertising industry continues to innovate and expand its services, recent developments are drawing increasing attention and concern from antitrust authorities. Notably, Google has been declared a monopolist in both general search services and general text advertising by the U.S. District Court for the District of Columbia, as of August 2024.[24] Additionally, the European Commission has sent Google a Statement of Objections regarding abusive practices in the online advertising technology.[25] Google is not the only platform that has recently faced scrutiny from regulatory bodies over its digital advertising practices: Amazon has been sued by the Federal Trade Commission for alleged monopolistic behavior, specifically for extracting rents by prioritizing paid advertisements over relevant search results and biasing search results in favor of its own products.[26]

[19] See Meta's Ads automation tools and Google's Performance Max.
[20] See Bidding for success, Think with Google, May 2022.
[21] Rate of generative AI adoption in the workplace in the United States 2023, by industry.
[22] McKinsey Report: How generative AI can boost consumer marketing.
[23] Meta's, Google's, and Amazon's articles about Generative AI features.
[24] United States District Court for the District of Columbia vs. Google and DOJ Mulls Google Breakup Push After Landmark Antitrust Win.
[25] Case 40670: EU Commission vs. Google.
[26] FTC vs. Amazon.

3 Experimental Design

3.1 Data Policies

Our focus is on the advertising auction environment in which one of the advertisers delegates bidding to the platform. In a counterfactual scenario instead, one of the advertisers gets the possibility to bid independently obtaining an information advantage by buying data from the platform.

For the sake of clarity, in this paper, we will focus on two extreme scenarios in terms of data knowledge, i.e. *Full Information* vs. *Limited Information*. These cases differ in terms of which data on bids is known to the advertisers:

1. *Full Information*: In every period, the bidder observes not only the current reward but also the bids of the other players.

2. *Limited Information*: The only information that the bidder observes is the reward she received after submitting a bid.

3.2 Generalized Second-Price Auction

Consider n advertisers $i \in \{1, 2, ..., n\}$ bidding in an online advertising auction. The advertisers put the values on click on the ad v_1,...,v_n. There are Z advertising slots. Slot $z = 1$ corresponds to the highest (i.e., best) position, $z = 2$ to the second-highest, and so on until $z = Z$, which is the slot in the lowest (i.e., worst) position. Following Varian [2007], the 'Click-Through-Rate' (CTR) of slot z is the number of clicks that an ad in position z is expected to receive. If an ad is placed on the first slot, it gets one x^1 clicks, on the second slot x^2 clicks, and so on. The advertiser with the highest bid wins the slot, the second-highest gets the second slot and so on. When two players submit the same bid, the winner is determined randomly. Denote the bid of advertiser i as b_i. In the GSP mechanism, the winner pays the price-per-click equal to the bid of the advertiser placed below him. As a result, the winners' reward in the GSP auction can be written as $r^i = \left(v_i - b^{\rho(i)+1}\right) x^{\rho(i)}$, where $\rho(i)$ is the rank of advertiser's i bid, whereas losing bidders get zero.

In our baseline setting, we consider two symmetric advertisers $i \in \{1, 2\}$, each valuing a click on their ad at $v_1 = v_2 = 1$. There is one available advertising slot, which guarantees a single click if an ad is placed, implying a CTR of $x^1 = 1$. We discretize the feasible set of bids $\mathcal{B}$ within the interval $[B_{\min}, B_{\max}] = [0.05, 1.4]$, with a step size of 0.05. This setup leads to a multiplicity of pure-strategy Nash Equilibria in the second-price auction. To prevent equilibria where a player remains indifferent across the entire bid range, we exclude the bid $b = 1.0$ from the action space. Consequently, the action space consists of $k = 27$ possible bids, with pure-strategy Nash Equilibria

taking the form (b_i, b_{-i}), where $b_i > 1 > b_{-i}$. Importantly, this approach differs from other studies (e.g. Banchio and Skrzypacz [2023], Martin et al. [2024]), which restrict bids to below players' valuations, allowing algorithms to learn a broader range of strategies.

3.3 Design of AIAs

The auction game described above is repeated multiple times, allowing AIAs to learn optimal bidding strategies, through trial and error. This iterative process enables the agents to maximize the expected present value of their reward stream, given by $E[\sum_{t=0}^{t=\infty} \delta^t r_t^i]$ for player i, where δ is the discount factor. Specifically, the AIAs we examine are Q-learning algorithms bidding against each other and learning simultaneously.[27] Their training entails striking a balance between exploration (testing new strategies) and exploitation (leveraging accumulated knowledge).

The knowledge The knowledge of each algorithm is represented by the Q-matrix, which is the matrix of expected rewards from each possible bid ($b \in \mathcal{B}$) in each possible state of the game ($s \in \mathcal{S}$). For each bidder i, in each period t, it is denoted by $Q_t^i(s, b)$. Here, the states of the game can contain different amounts of information about the past auction outcomes. For example, in the *Full Information* experiment setting, each state is defined by the previous bids of all players. In the first period, each cell of the Q-matrix is initialized randomly.

The experimentation To fully explore the Q-matrix, the algorithm should visit different actions in different states, even the ones that it finds not optimal, given prior knowledge. We use an ε-*greedy* exploration strategy. At each iteration, the algorithm chooses the bid that currently leads to the highest value of Q-matrix in a given state with a probability $1 - \varepsilon$, and with probability ε chooses a random bid among all possible ones.[28] We use a declining with time exploration rate $\varepsilon_t = e^{-\beta * t}$, where $\beta > 0$ is the annihilation coefficient.

Among the many features that characterize how an AIA is designed, two play a particularly key role in our analysis: the updating rule and memory of the past.

The updating rule The information obtained in period t is used for updating the Q-matrix. The algorithm starts from an initial Q-matrix alongside the reward. After choosing bid b_t^i in state s_t, the algorithm observes the reward r_t^i as well as s_{t+1}, and updates $Q_t^i(s, b)$. The updating can happen in a number of ways. We consider two main approaches that the literature describes as synchronous and asynchronous updating rules:

[27] For an overview of this type of AIA see Sutton and Andrew [2018].

[28] In what follows we assume that if several bids lead to the same value of Q-matrix in a given state, one of them is chosen randomly.

1. *Asynchronous Updating*: For each i, the cell of the Q-matrix $Q^i(s_t, b_t^i)$ is updated using the following "temporal difference" update rule:

$$Q_{t+1}^i(s_t, b_t^i) = (1-\alpha)Q_t^i(s_t, b_t^i) + \alpha * (r_t^i(b_t^i, b_t^{-i}) + \delta * \max_{b' \in \mathcal{B}} Q_t^i(s_{t+1}, b')). \tag{1}$$

   Where α is the learning rate, which determines to what extent the new information substitutes the old ("how much" the algorithm learns from new bids and received rewards). As a reminder, δ is a standard discount factor, which represents the present value of future rewards. At the same time, $Q_{t+1}^i(s, b) = Q_t^i(s, b)$ for all $s \neq s_t$, $b \neq b_t^i$. Asynchronous updating only requires knowledge of the reward received from the submitted bid.

2. *Synchronous Updating*: for each i, all cells related to the s_t state of the world are updated at once. More specifically, $Q^i(s_t, b)$ is updated for all bids $b \in \mathcal{B}$ with a reward $r_t^i(b, b_t^{-i})$ that the bidder would have received had it submitted a bid b, given the bids of other players b^{-i}:

$$Q_{t+1}^i(s_t, b) = (1-\alpha)Q_t^i(s_t, b) + \alpha * (r_t^i(b, b_t^{-i}) + \delta * \max_{b' \in \mathcal{B}} Q_t^i(s_{t+1}, b')), \tag{2}$$

   and $Q_{t+1}^i(s, b) = Q_t^i(s, b)$ for all $s \neq s_t$. Synchronous updating requires calculating the rewards for all the bids of i, $r_t^i(b, b_t^{-i})$, hence also for those bids that were not submitted $b \neq b_t^i$.

From the description above, it is clear that the feasibility of the two approaches crucially hinges on the available data. It also depends on how such data can be used to calculate the counterfactual reward associated with the actions that are not taken. In our setting, the second-price auction has very clear rules to determine the allocation of slots and payments depending on the bids received.[29] Hence, if bidder i observes the bids of competitors b^{-i}, the calculation of her reward under any possible b^i holding fixed b^{-i} is transparent and, therefore, using a synchronous updating rule is feasible. An asynchronous updating rule can be used instead even without data on b^{-i}, since its implementation requires observing only the reward associated with the bid effectively submitted by i.

The memory The data is used not only to calculate the rewards associated with the different actions but also to keep track of the state. In this regard, there are two polar cases that can be considered:

1. *StateFul algorithms*: a *StateFul* algorithm maintains a record of previous bids and uses this information to inform its decisions. In particular, in case of *Full Information*, we define for each of the players $s_t = (b_{t-1}^i, b_{t-1}^{-i})$.

[29] See Asker, Fershtman and Pakes [2023] for a discussion.

2. *StateLess algorithms*: a *StateLess* algorithm, on the other hand, does not retain information from previous steps. It makes decisions based solely on the current reward. In particular, in that case, $s_t = \emptyset$.

A key difference between *StateFul* and *StateLess* algorithms is that the former, since it has memory, can respond differently to the same received rewards, based on the previous state. This, in turn, allows for dynamic strategies that are not possible with *StateLess* algorithms. However, data requirements are greater for *StateFul* algorithms as they need to keep track of past bids, relative to the case of *StateLess* algorithms which do not require such information.

3.4 Parameterization of AIAs

Experiments with both players using *StateLess Asynchronous* AIAs are run with 1,000,000 iterations, while simulations with at least one player using *StateFul* AIA are run with 5,000,000 iterations, to ensure a satisfactory degree of exploration. In the baseline experiments, if not stated otherwise, the annihilation coefficient β is chosen such that the probability of exploring ε is $5e-06$ at the last iteration. The learning rate α is set to 0.10, and the discount factor δ is set to 0.95. Q-matrices are initialized randomly between -1 and 1. Finally, algorithms are programmed such that when the maximum Q-value for a given state is associated with multiple actions, players randomly choose one. Importantly, the results presented in the next section are robust to different hyperparameters, as well as to the use of different initializations for the Q-matrices (see Section 7).

4 Results

In this section, we examine the outcomes of two experimental settings under different information regimes. First, we consider a *Baseline* scenario in which two advertisers with *Limited Information* compete in a second-price auction using *StateLess Asynchronous* (SLA) algorithms. Second, we analyze an *Asymmetric* scenario where an advertiser with *Full Information* competes against an advertiser with *Limited Information*.[30] The former advertiser, denoted as the *informed* player, employs a *StateFul* algorithm, either *Asynchronous* (SFA) or *Synchronous* (SFS), while the latter, the *uninformed* player, uses an SLA algorithm. Finally, we discuss the strategic advantages of information in algorithmic competition, focusing on the platform's incentives to manage advertisers' bidding activities directly, rather than selling data on competitor bids.

[30] *Full Information* can be obtained either through purchasing data on competitor bids or via bid delegation services provided by the platform.

Table 1: *Results in Baseline* scenario vs. *Asymmetric* scenario

Info	Learners	(3) Runs	(4) (b_1, b_2)	(5) (r_1, r_2)	(6) (w_1, w_2)	(7) Ties Ratio	(8) Winning Bid	(9) Losing Bid	(10) (wb_1, wb_2)	(11) (lb_1, lb_2)	(12) Revenue
Panel A: Baseline	(SLA, SLA)	984	$(0.83, 0.84)$	$(0.24, 0.25)$	$(0.49, 0.51)$	0.0	1.16	0.5	$(1.16, 1.16)$	$(0.51, 0.5)$	0.5 [0.5, 0.5]
Panel B: Asymmetric	(SFA, SLA)	868	$(0.81, 0.48)$	$(0.41, 0.15)$	$(0.79, 0.21)$	0.05	0.87	0.44	$(0.97, 0.48)$	$(0.27, 0.48)$	0.44 [0.43, 0.45]
	(SFS, SLA)	899	$(0.9, 0.69)$	$(0.33, 0.16)$	$(0.68, 0.32)$	0.0	1.08	0.51	$(1.1, 1.05)$	$(0.5, 0.51)$	0.51 [0.51, 0.51]

Notes: Column 3 shows the number of simulations that successfully converged under each regime; columns 4 and 5 - the average across runs bids and rewards at convergence for each of the players in the decreasing order of valuations; column 6 - the percentage of runs in which each player wins the auction; column 7 - the percentage of runs where the auction ends up in a tie; columns 8 and 9 - the average across runs of the winning and losing bids; columns 10 and 11 - the average across runs of the winning and losing bids by player index; column 12 - the average auctioneer revenue across runs, along with its 95% confidence interval. SLA AIAs are run with 1,000,000 iterations, with $\beta = 1.22 \times 10^{-5}$. Simulations with at least one player using *StateFul* AIA are run with 5,000,000 iterations, with $\beta = 2.44 \times 10^{-6}$. For all simulations $\alpha = 0.10$, $\delta = 0.95$.

Table 1 presents the experimental results. Panel A summarizes outcomes for the *Baseline* scenario, while Panel B shows results for the *Asymmetric* scenario, where Player 1 is the *informed* player. The first row in Panel B reports the case where the *informed* player uses the SFA algorithm, and the second row reports the case with the SFS algorithm. Each experiment consists of 1,000 independent simulations. Column 3 shows the number of simulations that successfully converged under each regime. For each converged simulation, we first averaged the metrics of interest over the last 500 iterations (referred to as *limit* iterations), then computed the averages across simulations. Columns 4 and 5 report the average bids and rewards for each player, listed by player index. Column 6 shows the percentage of runs in which each player wins the auction, excluding limit iterations that resulted in a tie.[31] Column 7 shows the percentage of *limit* iterations that ended in a tie. Column 8 reports the average winning bid, and Column 9 reports the average losing bid. Columns 10 and 11 provide a breakdown of the average winning and losing bids by player index. Finally, Column 12 presents the average auctioneer revenue across all simulations, along with its 95% confidence interval (in square brackets).[32] Simulations where convergence was not achieved are discarded. In each experiment, the convergence rate is higher than 85%. See Appendix A.3 for the definition and discussion of convergence.

4.1 Baseline Experiments

In simulations where both players have *Limited Information* and use SLA algorithms, the players converge to similar bids, rewards, and winning rates. The average winning bid is 1.16, while the

[31] Ties are excluded from the calculation of rewards, winning rates, winning and losing bids (Columns 5, 6, 8, 9, 10, and 11). This simplifies the interpretation of results, without compromising the accuracy of these metrics, provided that ties are infrequent and resolved randomly.

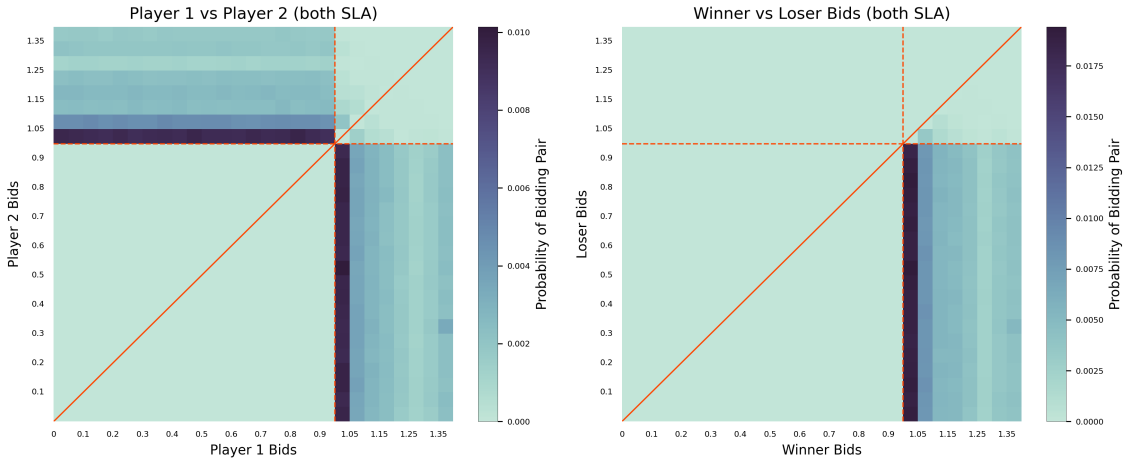
[32] Confidence intervals are calculated using the standard deviation of the average revenue (over *limit* iterations) across simulations.

average losing bid is 0.5. Since winning the first slot results in one click, the auctioneer's average revenue in this setting is also 0.5. No significant differences are observed in the average winning and losing bids when analyzed by player index. Notably, ties are absent, ensuring a distinct winner in each iteration.

Figure 1 provides detailed insights into the distribution of players' bids at convergence. This visualization reveals that winning bids always exceed players' valuations, while losing bids are generally distributed uniformly below the valuation. Being in the form ($b^i > 1 > b^{-i}$), the vast majority of *limit* iterations achieves one of the static Nash Equilibria of the game.

Notably, given the convergence criteria and parameterization of these experiments, *StateLess* algorithms converge to either a single action that yields the highest reward or to a set of actions with equal Q-values, chosen randomly with equal probability. Any alternative equilibrium would not meet the constant argmax condition required for convergence (see Appendix A.2 and A.3). The winning player's strategy falls into the former category, bidding a constant amount that varies across simulations due to the stochastic nature of the learning process. In contrast, the losing player's strategy involves random selection among actions that do not secure the slot, leading to indifference between several options. Since the winning player consistently bids above one, the Q-values for the losing player's actions converge towards non-positive values. Q-values for actions within the range $[0.05, 0.95]$ converge to zero, as they yield null rewards. Bids higher than the winning bid result in negative rewards, causing their Q-values to converge to negative values. Similarly, bids above the player's valuation but below the winning bid also converge to negative values in most simulations, as these actions receive negative rewards when the winner explores lower bids, yet remains above 1.0. Once these Q-values become negative and exploration diminishes, these actions are no longer updated and are excluded at convergence.

Figure 1: Bid Pairs at Convergence - SLA vs. SLA Algorithms



Notes: This plot shows the heatmap of bid pairs at convergence in an experiment with two SLA AIAs agents. The left plot shows the probability distribution of bid pairs for Player 1 and Player 2, while the right plot displays the probability of winning versus losing bids.Color intensity reflects the likelihood of each bid pair. Dashed lines represent the Nash equilibrium at 0.95 and the symmetric equilibrium.

The distribution of winning bids at convergence suggests that bids slightly above 1.0 tend to be more effective strategies for the winning player. In the exploitation phase, as the losing player's Q-values converge to non-positive values, their optimal action frequently shifts, preventing any single action from remaining optimal for extended periods. The loser often changes strategies, and bids closer to 1.0 tend to yield higher rewards for the winner. These bids make actions within the range of $[1.05, 1.4]$ less appealing to the opponent, accelerating their convergence to bids below 1.0.

For example, when the winning player repeatedly bids 1.05, the loser experiences negative rewards for all bids between 1.10 and 1.4, which stabilizes the loser's bids within the $[0.05, 0.95]$ range. Conversely, if the winner bids 1.30, the loser only obtains negative rewards for a narrow range of bids, specifically $[1.35, 1.4]$. As a result, bids in the range of $[1.05, 1.25]$ become more frequent in the latter case, since they are associated with non-negative rewards. This increases payments for the winning player, potentially prompting a strategic shift towards lower bids over time. This dynamic explains the observed concentration of winning bids around lower values slightly above 1.0, particularly at 1.05.

The results from this experiment align with findings from similar studies, such as Banchio and Skrzypacz [2023] and Decarolis et al. [2023]. A key distinction from Banchio and Skrzypacz [2023] is the presence of a unique pure-strategy Nash Equilibrium at (0.95, 0.95) in their second-price auction experiments. This result stems from their constraints of limiting the action space below player's valuation.[33] However, in second-price auctions, where players do not pay their bids, such

[33] Our simulations replicate the outcomes of Banchio and Skrzypacz [2023] when using the same bid grid structure, as shown in Table 15 in Appendix A.4.

restrictions narrow the set of possible equilibria. By removing these constraints and allowing bids above players' valuations, more favorable outcomes are achievable. With a larger action space, players' average rewards increase nearly tenfold (≈ 0.25 vs. ≈ 0.025). This indicates that players can benefit from a more flexible bid grid, enabling them to converge to equilibria where one player wins all iterations of the game while paying generally less than in the restricted grid equilibrium. [34]

4.2 Asymmetric Experiments

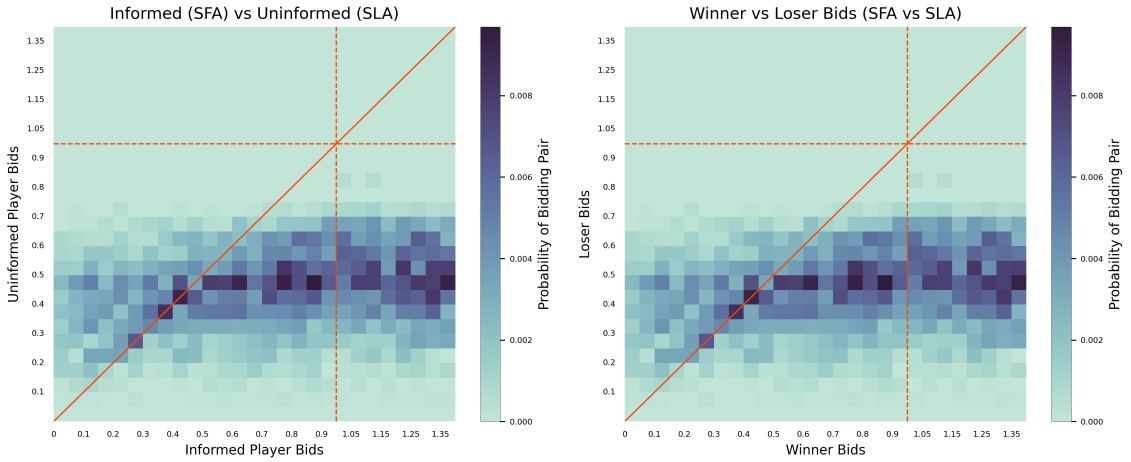
4.2.1 SFA vs. SLA Experiments

The deployment of an SFA algorithm allows the *informed* player to significantly outperform the opponent, achieving higher rewards (0.41 vs. 0.25) and winning rates (79% vs. 50%) compared to the *Baseline* scenario. Additionally, both winning and losing bids are notably lower (0.87 vs. 1.16 and 0.44 vs. 0.50, respectively), leading to a substantial decrease in auctioneer revenues (0.44 vs. 0.50). Interestingly, the *uninformed* player's winning bid is considerably lower than that of the *informed* player (0.48 vs. 0.97) and falls below the players' valuation. Furthermore, the *informed* player's average losing bid is much lower than the opponent's (0.27 vs. 0.48).

The heatmap in Figure 2 illustrates the distribution of players' bids at convergence. Two aspects stand out: the left panel reveals that the *uninformed* player frequently wins, while the right panel shows that instances of winning bids falling below the players' valuations are more common compared to the *Baseline* scenario.

[34] This conclusion holds when the argmax selection is either *random* or *conservative*. Otherwise, the loser would consistently bid 0.95, resulting in a winner's expected reward similar those under the restricted grid (see Appendix A.4). However, players can adjust the argmax parameter to optimize their rewards.

Figure 2: Bid Pairs at Convergence - SFA vs. SLA Algorithms



Notes: This plot shows the heatmap of bid pairs at convergence in an experiment with one SFA and one SLA AIA agent. The left plot shows the probability distribution of bid pairs for Player 1 and Player 2, while the right plot displays the probability of winning versus losing bids. Color intensity reflects the likelihood of each bid pair. Dashed lines represent the Nash equilibrium at 0.95 and the symmetric equilibrium.

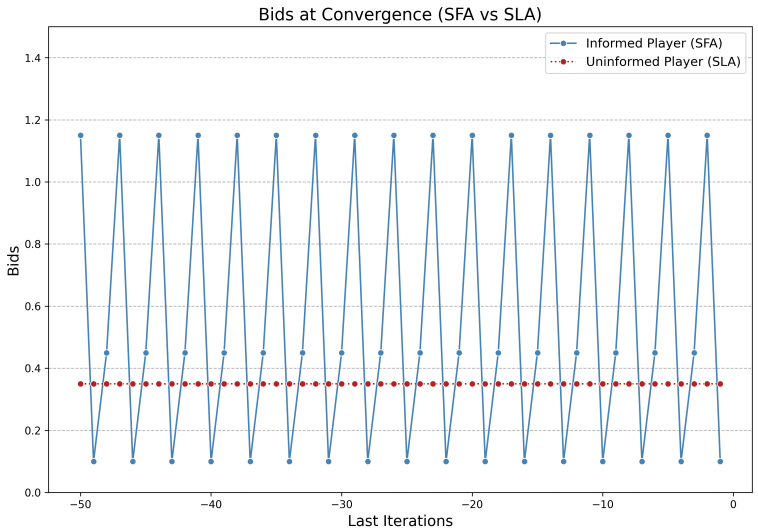
Upon examining the convergence of individual simulations, we find that the cyclical equilibria achieved by the SFA algorithm explain these results. For instance, Figure 3 shows bids from the last 50 iterations of a single simulation, where the SFA algorithm cycles between three distinct bids (0.10, 0.45, 1.15). This cycle includes a phase where the SFA algorithm is outbid by the opponent, who instead converges to a constant bid (0.35). Such bid rotation equilibria are observed in all converged simulations, and explain the lack of static Nash Equilibria in *limit* iterations. The SFA algorithm is consistently outbid by the SLA algorithm at one phase of the cycle, which accounts for the winning rates reported in Table 1. The cycle length varies between 2 and 12 iterations, with an average of 4.83.[35]

4.2.2 SFS vs. SLA Experiments

When deploying an SFS algorithm, the *informed* player maintains an informational advantage over the *uninformed* player, leading to significantly higher rewards (0.33 vs. 0.25) and winning rates (68% vs. 50%) relative to the *Baseline* scenario. However, the SFS algorithm does not outperform the SFA algorithm in terms of rewards (0.33 vs. 0.41). This outcome is due to both higher winning rates (0.32 vs. 0.21) and higher average losing bids (0.51 vs. 0.48) achieved by the SLA when competing against the SFS. Unlike the SFA vs. SLA case, bid rotation equilibria are absent in this experiment. At convergence, the *informed* player either consistently wins or consistently loses. Consequently, the average losing bid is higher than in the SFA vs. SLA scenario (0.51 vs.

[35] A detailed discussion of the emergence of these cycling patterns, for this and other cases, is provided in Section 5.

Figure 3: Bid Dynamics at Convergence in a Single Simulation - SFA vs. SLA Algorithms

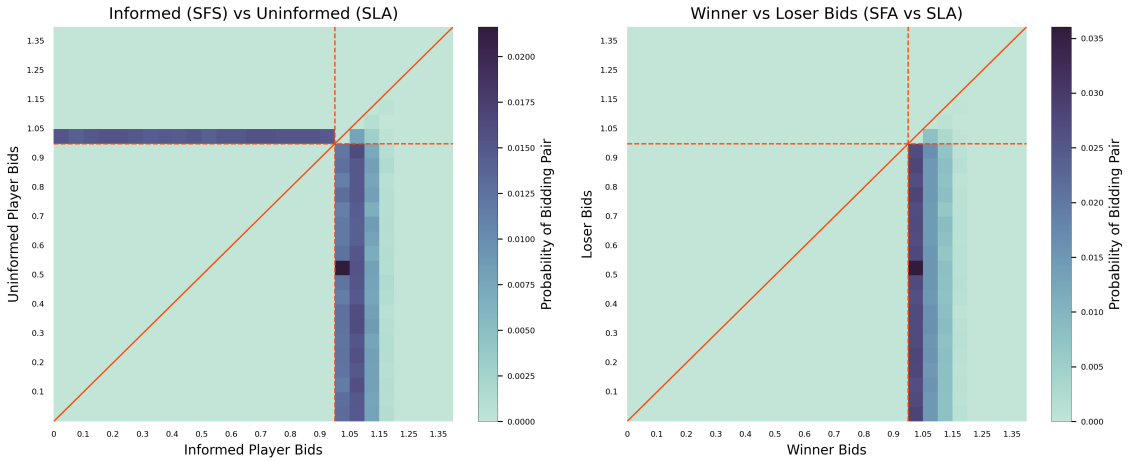


Notes: This plot shows bid patterns from an SFA vs. SLA experiment, displaying the final 50 iterations of a single simulation.

0.44), and auctioneer revenues return to *Baseline* levels (0.51).

Figure 4 presents a heatmap showing the distribution of players' bids at convergence. The right panel reveals that the *informed* player's average winning bid is slightly higher than the *uninformed* player's. When the *uninformed* player wins, they always bid 1.05 at convergence, while the *informed* player's bid varies across simulations, with an average of 0.90. This occasionally allows the *uninformed* player to safely bid above their valuation at convergence. The left panel highlights the similarities between the SFS vs. SLA and the *Baseline* scenario: the winner consistently bids above valuation, while the loser's bids are uniformly distributed below valuation. No bid rotation is observed, and the winning player's strategy at convergence is a constant bid. As in the *Baseline* case, a Nash Equilibrium is reported in the vast majority of *limit* iterations.

Figure 4: Bid Pairs at Convergence - SFS vs. SLA Algorithms



Notes: This plot shows the heatmap of bid pairs at convergence in an experiment with one SFS and one SLA AIA agent. The left plot shows the probability distribution of bid pairs for Player 1 and Player 2, while the right plot displays the probability of winning versus losing bids. Color intensity reflects the likelihood of each bid pair. Dashed lines represent the Nash equilibrium at 0.95 and the symmetric equilibrium.

4.3 Information Asymmetries and Platform's Incentives

The experimental results highlight a significant information advantage for advertisers with full access to competitor bids. Deploying a *StateFul* algorithm against an *uninformed* opponent substantially increases rewards, regardless of the updating rule implemented. Notably, the SFA algorithm delivers a larger reward increase (64%) compared to the SFS algorithm (32%), as the former can learn bid rotation strategies. In contrast, the SFS algorithm does not exhibit these strategies, leading to higher bids from the opponent and, consequently, higher payments. As a result, platform revenues differ based on the amount of information available to advertisers and their algorithmic design choices, with revenues being 12% lower when the *informed* player uses an SFA algorithm compared to an SFS algorithm.

These findings illustrate the tension between the platform's incentives to maximize revenue and the advertisers' incentives to maximize rewards. The platform can profit by either selling data on competitor bids or by offering bid delegation services that leverage such data.

In the first scenario, the platform loses control over the design of the algorithm used by the *informed* player, who would implement an SFA algorithm. This would lead to the highest reward increase for the advertiser but reduce auction revenues.[36]

In the second scenario, the platform retains control over algorithm design and sells access to a proprietary SFS algorithm. This approach would still boost the *informed* advertiser's rewards

[36]Selling data remains profitable if the platform can capture enough of the *informed* player's additional rewards through data pricing.

compared to the *Baseline*, without negatively impacting auction revenues. These results suggest the platform's preference for controlling the information provided to advertisers, which rationalizes the rise of bid delegation services in online advertising markets.

5 Drivers of Information Advantage

5.1 How Bid Rotation is Sustained

As described in Section 4, the SFA vs. SLA experiment converges to equilibria characterized by cyclical patterns (see, for example, Figure 3). In these cycles, the SLA algorithm consistently places a low bid, while the SFA algorithm rotates between a set of bids, one of which is lower than the opponent's. These bid rotation schemes are central to the equilibria observed in asymmetric games. On one hand, the periodic positive profit prevents the SLA algorithm from shifting to a different strategy. On the other hand, the SFA algorithm finds it advantageous to accept a loss in one iteration of the cycle in exchange for higher rewards in subsequent periods.

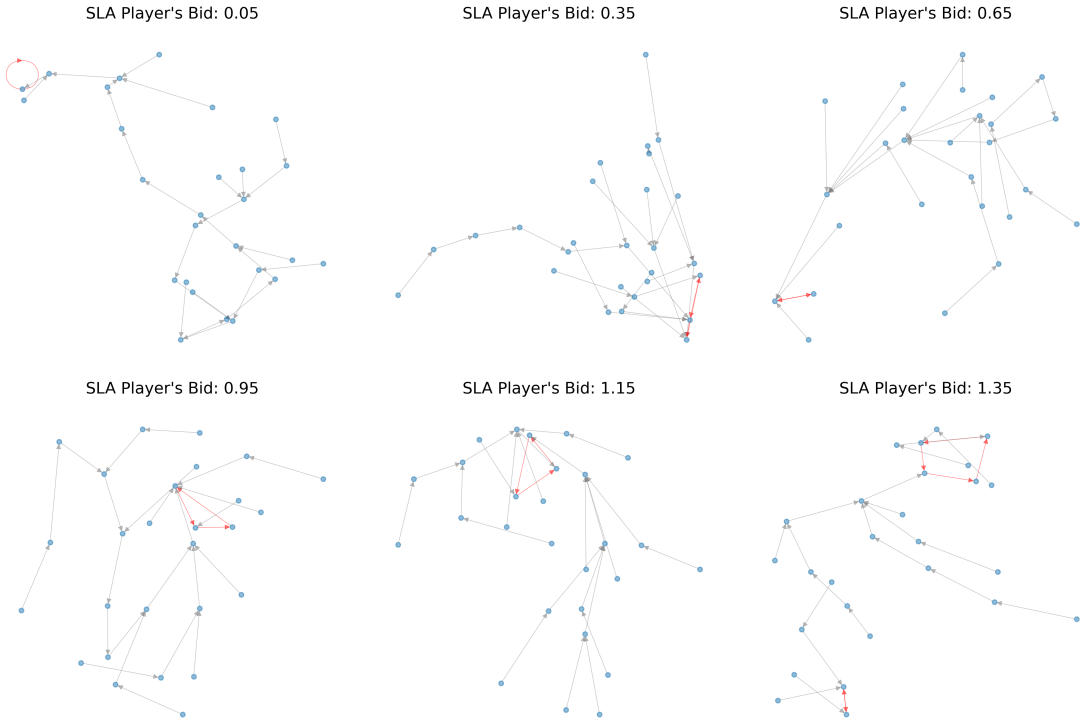
In this section, we explore the algorithmic features that drive the emergence of these equilibria. For an extensive discussion of how bid rotation equilibria emerge in SFA vs. SLA bidding, see Appendix A.1.

The memory Endowed with memory, the SFA algorithm can learn more complex strategies compared to the SLA. Specifically, the ability to learn state-specific best responses allows the SFA to develop dynamic strategies consisting of bid cycles repeated over time. This is illustrated in Figure 5, which shows the strategy learned by the SFA algorithm in a single SFA vs. SLA simulation. The states are represented as nodes, while the transitions resulting from the best responses are depicted as directed edges. Since these transitions are derived by holding the opponent's bid constant, the panels show *partial* best response networks-i.e., how the system would evolve from any accessible state, based on the strategy learned by a *StateFul* algorithm and the fixed strategy of the opponent. The cyclical equilibria are identified by red edges and are often present in this type of networks.

Without memory, the *partial* best response networks would have a completely different structure. Indeed, the strategies of *StateLess* algorithms consist of either playing a single action or randomizing uniformly over a set of actions with the same Q-values.[37]

[37]This also depends on the type of argmax used. See further discussion in Appendix A.2.

Figure 5: Partial Best Response Networks - SFA Algorithm



Notes: This plot shows the system dynamics induced by the best responses of the SFA algorithm, given the opponent's fixed action at convergence in a single simulation. Nodes represent states while directed edges represent transitions. Red edges identify cycles.

The patience Considering the stream of future rewards associated with a given bid is crucial for the emergence of dynamic strategies, such as bid rotation. In Equation 1, both the immediate reward ($r_t^i(b_t^i, b_t^{-i})$) and the discounted future reward ($\delta \max_{b' \in \mathcal{B}} Q_t^i(s_{t+1}, b')$) are included in the temporal difference update rule. This enables bid rotation cycles to be sustainable, as the null rewards from losing the auction with a low bid in one phase can be offset by higher rewards in subsequent phases. This is illustrated in Figure 3, where the informed player loses the auction in one phase (0.10 vs. 0.35) but benefits from low payments in the remaining phases (0.45 vs. 0.35 and 1.15 vs. 0.35).

Without future rewards considered in the temporal difference update rule, a series of null immediate rewards would eventually lead to a strategic shift in the state where the player loses the auction, as the opponent could be overbid profitably.

The first row in Table 2 reports results from a variation of the SFA vs. SLA experiment, where both algorithms disregard future rewards in optimizing their Q-matrix. Technically, we set the parameter $\delta = 0$ to implement this. In this scenario, the SFA algorithm never wins at convergence and earns a null reward. The information advantage disappears entirely. When learning dynamic

strategies is not possible, the larger Q-matrix makes the learning process more challenging for the SFA algorithm, who is systematically outperformed by the SLA algorithm. Cyclical equilibria do not emerge, and the convergence patterns resemble those in the *Baseline* scenario, with the winning player placing a single bid higher than the players' valuation.

Table 2: Mechanisms Behind Main Results

	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SFA vs SLA ($\delta = 0$)	97	$(0.54, 1.11)$	$(0.0, 0.46)$	$(0.0, 1.0)$	0.0	1.11	0.54	$(-, 1.11)$	$(0.54, -)$	0.54
										[0.54, 0.54]
SLS vs SLA	100	$(0.55, 1.13)$	$(0.0, 0.45)$	$(0.0, 1.0)$	0.0	1.13	0.55	$(-, 1.13)$	$(0.55, -)$	0.55
										[0.54, 0.56]
SFS vs SLA ($\delta = 0$)	100	$(0.51, 1.06)$	$(0.0, 0.49)$	$(0.0, 1.0)$	0.0	1.06	0.51	$(-, 1.06)$	$(0.51, -)$	0.51
										[0.51, 0.51]

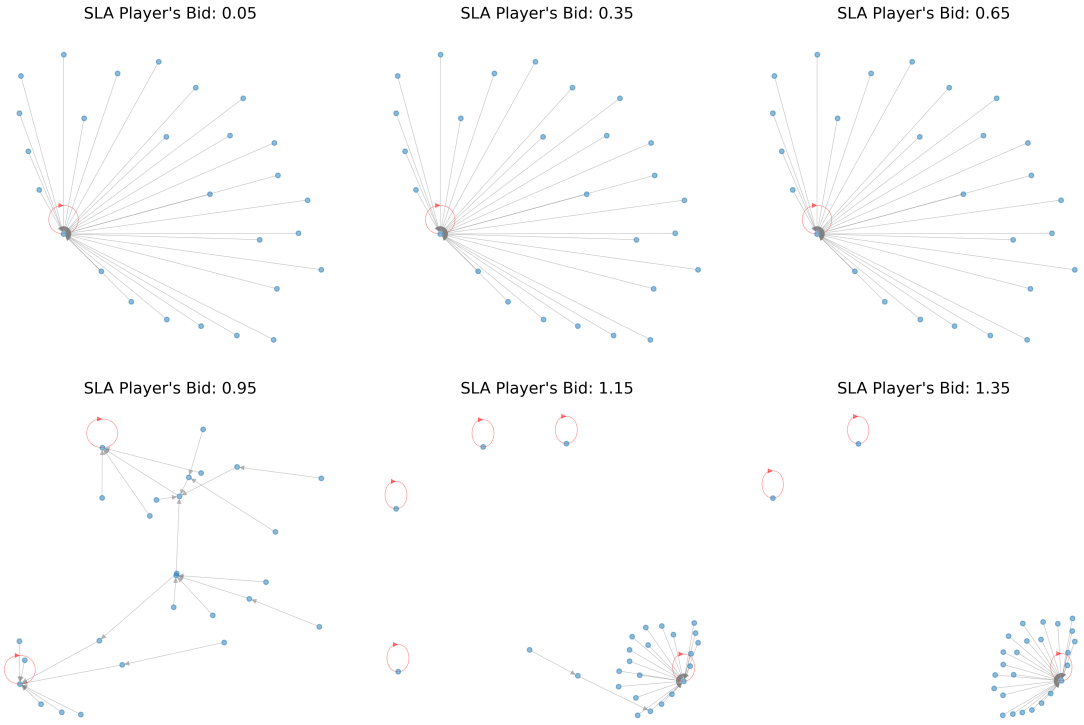
Notes: Results of experiments where $\delta = 0$ if there is at least one *SateFul* AIAs (row 1 and 3) and where a SLA plays against a SLS AIA (row 2). Column 3 shows the number of simulations that successfully converged under each regime; columns 4 and 5 - the average across runs bids and rewards at convergence for each of the players in the decreasing order of valuations; column 6 - the percentage of runs in which each player wins the auction; column 7 - the percentage of runs where the auction ends up in a tie; columns 8 and 9 - the average across runs of the winning and losing bids; columns 10 and 11 - the average across runs of the winning and losing bids by player index; column 12 - the average auctioneer revenue across runs, along with its 95% confidence interval. SLA AIAs are run with 1,000,000 iterations, with $\beta = 1.22 \times 10^{-5}$. Simulations with at least one player using *StateFul* AIA are run with 5,000,000 iterations, with $\beta = 2.44 \times 10^{-6}$. For all simulations $\alpha = 0.10$.

Asynchronous Learning A cyclical equilibrium over a set of states is not sustainable in the SFS vs. SLA experiment. In this case, the suboptimal action within the cycle cannot persist as the optimal action. For instance, in Figure 3, there is a state, $s = (1.10, 0.35)$, where the SFA algorithm's preferred action is 0.10, even though it leads to a null reward given the opponent's best response of 0.35. If all Q-values in that state were updated in every iteration, bids higher than 0.35 would be repeatedly updated with positive immediate rewards. Eventually, overbidding the opponent would become the best response in every state involved in the cycle, eliminating the possibility of bid rotation. A formal discussion of this dynamic is provided in Appendix A.2.

Figure 6 illustrates the *partial* best response networks from a single simulation where the SFS algorithm competes against an SLA opponent. The structure of the network differs significantly from that in Figure 5. Cycles are absent, and, with a fixed opponent bid, the system converges to a fixed point.[38]

[38]This phenomenon can be described as the propagation of the best responses. Throughout the simulation, Q-values in states defined by the convergence action increase more rapidly than in other states. Since the Q-value of the convergence action is updated across all states, regardless of whether it is the optimal action, a strong preference for this action develops because of the large discounted future reward associated with it. For example, if the maximum Q-value in state $(1.2, 0.6)$ is significantly higher than in other states of the form $(b_1, 0.6)$, the Q-value of 1.2 in those states will grow due to its high discounted future reward. Importantly, this occurs even if 1.2 is not the best response in a particular state, because of the synchronous update rule.

Figure 6: Partial Best Response Networks - SFS Algorithm



Notes: This plot shows the system dynamics induced by the best responses of *StateFul Synchronous* algorithm, given the opponent's fixed action at convergence in a single simulation. Nodes represent states while directed edges represent transitions. Red edges identify fixed points of the system.

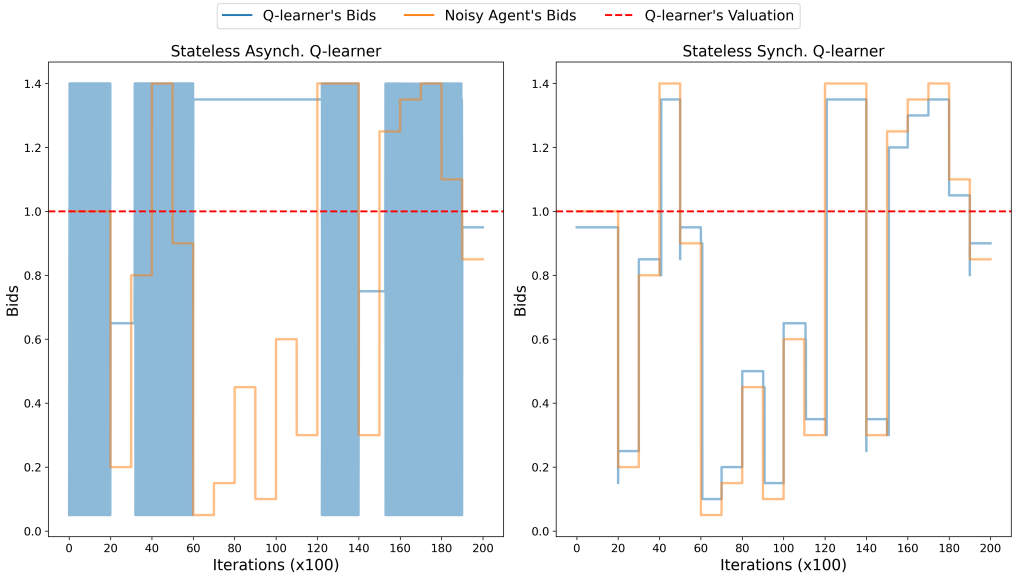
5.2 The Role of Synchronous Update

The *Synchronous* update mechanism involves updating the Q-values for all possible actions at each iteration. This creates a stronger relationship between the competitor's bid at time t and the player's optimal bid at $t+1$ compared to the *Asynchronous* update. Figure 7 shows the evolution of bids in a simulation where a *StateLess* algorithm competes against a noisy agent that randomly changes its bid periodically. The left panel illustrates the bids of an *Asynchronous* algorithm, while the right panel shows those of a *Synchronous* one. The SLS learner adapts instantly to the opponent's bid to maximize rewards. Specifically, when the noisy agent bids above (below) the opponent's valuation (the red horizontal line), the SLS algorithm underbids (overbids). This contrasts with the delayed bid adjustments of the SLA algorithm.[39] However, this increased responsiveness does not necessarily provide a strategic advantage. In fact, when the

[39]The greater responsiveness of the SLS algorithm is due to the synchronous updating of all Q-values in each state. Let $\bar{b}$ represent the noisy agent's bid and v the algorithm's valuation. With *Synchronous* updates, if $\bar{b_t} > v$, then $Q_{t+1}(b)$ decreases for all $b \in [\bar{b_t}, b_{max}]$, reflecting reduced attractiveness of higher bids due to negative rewards. Conversely, if $\bar{b_t} < v$, the attractiveness of higher bids increases in the next iteration. This dynamic is absent in the *Asynchronous* case, where only $Q(b_t)$ is updated.

SLS algorithm competes against an SLA, the former consistently underperforms. The second row of Table 2 shows results from 100 independent simulations, indicating that the SLS algorithm tends to settle in low-bid equilibria where it bids lower than the opponent. The excessive responsiveness of the *Synchronous* update penalizes the *informed* player in this experiment. Additionally, the analysis shows that winning bids are generally lower than in scenarios where both algorithms are SLA (Table 1, first row). This is due to the stronger connection between the bids of an algorithm and its payments under the *Synchronous* update. As a result, bids closer to players' valuations, which are associated with lower payments, emerge more frequently as the equilibrium strategy in the repeated game.

Figure 7: Bidding Dynamics: Asynchronous vs. Synchronous Q-Learning



Notes: This plot shows the evolution of bids in a simulation where a Q-learner competes against a noisy agent that randomly changes its bid every 1,000 periods. Exploration is set to zero (i.e. the Q-learner always plays its greedy action). The left panel shows the bids of an SLA Q-learner. In the left panel, an area is employed to visualize Q-learner's dynamic bid behavior, characterized by rapid shifts between various bids as it explores the entire bid grid. The right panel shows the bids of an SLS Q-learner.

5.3 The Role of Memory and Patience

In the SFA vs. SLA experiment, the information advantage arises from the ability of the SFA algorithm to learn bid rotation strategies. In contrast, in the SFS vs. SLA experiment, the advantage is attributable to the combined implementation of memory and *Synchronous* update.

The introduction of memory produces two effects. First, the presence of multiple states weakens the interdependence discussed in Section 5.2: while the opponent's bid at time t influences the best response in state s_t, the system may transition to a different state at time $t+1$. As a result, the

impact of the current opponent's bid on the *Synchronous* algorithm's strategy in the next period tend to be reduced if the algorithm has memory. Second, when players are patient ($\delta > 0$), Q-values of different actions are updated based on distinct discounted future rewards, as each action leads the system into a different state in the next iteration.

Row 3 of Table 2 shows that memory alone does not provide an information advantage. In 100 independent simulations, an SFS algorithm with $\delta = 0$ invariably fails to win the auction against an SLA algorithm. The over-sensitivity to the opponent's bids entrenches the SFS algorithm in low-bid equilibria at convergence. As seen in Table 1 (Row 3), these equilibria are avoided when future payoffs are incorporated into the Q-value updates. Therefore, the information advantage of the SFS algorithm stems from the combination of memory and patience.

6 Extensions

6.1 Symmetric Full Information

As a first extension, we explore a scenario where both players have *Full Information* and can implement *StateFul* algorithms. Panel A of Table 3 reintroduces the SLA vs. SLA experiment as a benchmark. Panel B presents the outcomes of experiments where both players are *informed* and use either SFA (first row) or SFS (second row) algorithms. In all cases, both players have symmetric valuations $v_1 = v_2 = 1$.

Table 3: Symmetric Information Experiments Results

Info	Learners	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
Panel A: Baseline	(SLA, SLA)	984	$(0.83, 0.84)$	$(0.24, 0.25)$	$(0.49, 0.51)$	0.0	1.16	0.5	$(1.16, 1.16)$	$(0.51, 0.5)$	0.5 [0.5, 0.5]
Panel B: Symmetric Info	(SFA, SFA)	100	$(0.71, 0.7)$	$(0.31, 0.31)$	$(0.51, 0.49)$	0.01	1.04	0.38	$(1.04, 1.03)$	$(0.37, 0.38)$	0.38 [0.36, 0.4]
	(SFS, SFS)	90	$(0.77, 0.81)$	$(0.23, 0.26)$	$(0.46, 0.54)$	0.0	1.07	0.51	$(1.07, 1.07)$	$(0.52, 0.51)$	0.51 [0.5, 0.52]

Notes: Results of experiments with symmetric players. Column 3 shows the number of simulations that successfully converged under each regime; columns 4 and 5 - the average across runs bids and rewards at convergence for each of the players in the decreasing order of valuations; column 6 - the percentage of runs in which each player wins the auction; column 7 - the percentage of runs where the auction ends up in a tie; columns 8 and 9 - the average across runs of the winning and losing bids; columns 10 and 11 - the average across runs of the winning and losing bids by player index; column 12 - the average auctioneer revenue across runs, along with its 95% confidence interval. SLA AIAs are run with 1,000,000 iterations, with $\beta = 1.22 \times 10^{-5}$. Simulations with at least one player using *StateFul* AIA are run with 5,000,000 iterations, with $\beta = 2.44 \times 10^{-6}$. For all simulations $\alpha = 0.10$, $\delta = 0.95$. Experiment with *StateLess* AIAs have a total of 1000 runs, all others 100.

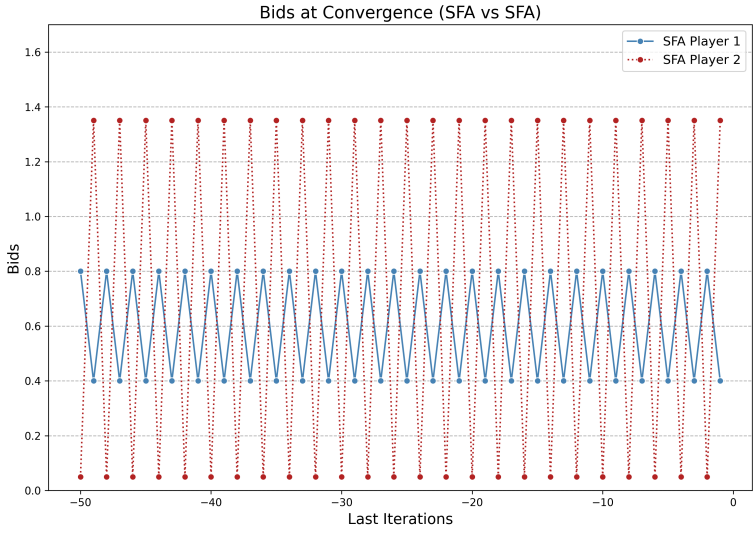
Compared to the benchmark, in the experiment with SFA algorithms, both players earn higher rewards, while auctioneer revenues are lower. This result stems from both players exhibiting cyclical behavior at convergence. Figure 8 visualizes this process in a single run, where the cycle length is 2 for both players. While the cycle length and the bids within each cycle vary across simulations,

both players consistently earn positive rewards within the cycles, and the average revenues across runs are equal between players.

In the experiment with SFS algorithms, the rewards for both players and auctioneer revenues are similar to those in the benchmark. The winner bids an average of 1.07 at convergence, while the loser randomly bids between 0.05 and 1.05. Both players have an equal probability of being the highest bidder, and no cycles emerge in this setup.

These results offer additional insights into the platform's incentives to share competitors' data. First, the comparison between the SFA vs. SFA experiment and the benchmark supports the findings of Decarolis et al. [2023], indicating that auctioneer revenues are lower in a symmetric *Full Information* setting compared to a symmetric *Limited Information* setting. Second, the comparison between the SFA vs. SFA and SFA vs. SLA experiments reveals that platforms may have an incentive to grant asymmetric access to competitor data.

Figure 8: Bids at Convergence in a Single Simulation - Two SFA Algorithms



Notes: This plot shows the emergence of cyclical behavior at convergence of two symmetric players that use SFA algorithms.

6.2 First-Price Auctions

As a second extension, we examine the impact of different information regimes in a first-price auction. Panel A in Table 4 reports the results of an experiment where both players have *Limited Information* and use SLA algorithms. Panel B shows results for experiments with asymmetric information. Specifically, the first row shows result for the SFA vs. SLA experiment, and the second row reports the outcome of the SFS vs. SLA experiment.

Table 4: First-Price Auction Results

Info	Learners	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
Uninformed Baseline	(SLA, SLA)	50	$(0.32, 0.32)$	$(0.34, 0.34)$	$(0.5, 0.5)$	1.0	0.32	0.32	$(0.32, 0.32)$	$(0.32, 0.32)$	0.32
											[0.29, 0.35]
Asymmetric Info	(SFA, SLA)	50	$(0.52, 0.47)$	$(0.36, 0.13)$	$(0.69, 0.31)$	0.41	0.52	0.47	$(0.52, 0.48)$	$(0.47, 0.48)$	0.52
											[0.49, 0.55]
	(SFS, SLA)	50	$(0.94, 0.94)$	$(0.03, 0.03)$	$(0.5, 0.5)$	1.0	0.94	0.94	$(0.94, 0.94)$	$(0.94, 0.94)$	0.94
											[0.91, 0.97]

Notes: Results of experiments with a first-price auction environment. Column 3 shows the number of simulations that successfully converged under each regime; columns 4 and 5 - the average across runs bids and rewards at convergence for each of the players in the decreasing order of valuations; column 6 - the percentage of runs in which each player wins the auction; column 7 - the percentage of runs where the auction ends up in a tie; columns 8 and 9 - the average across runs of the winning and losing bids; columns 10 and 11 - the average across runs of the winning and losing bids by player index; column 12 - the average auctioneer revenue across runs, along with its 95% confidence interval. SLA AIAs are run with 1,000,000 iterations, with $\beta = 1.22 \times 10^{-5}$. Simulations with at least one player using *StateFul* AIA are run with 5,000,000 iterations, with $\beta = 2.44 \times 10^{-6}$. For all simulations $\alpha = 0.10$, $\delta = 0.95$. 50 runs, full convergence in SLA vs SLA and SFA vs SLA, but no convergence in SFS vs SLA (1/50). Therefore, kept all simulations for the table. In general, SLA vs SLA and SFS vs SLA report an extremely high percentage of ties at convergence.

In the SLA vs. SLA experiment, players exhibit the same average bids across runs, consistent with the second-price results (Table 1). In the first-price auction setting, bids are significantly lower than the two Nash Equilibria of the game (0.95, 0.95 and 0.90, 0.90) and are also lower than in the second-price setting (Table 1, first row). As explained in Banchio and Skrzypacz [2023], SLA algorithms in first-price auctions tend to coordinate on lower bids.

In the case of asymmetric information, outcomes differ depending on the type of updating rule featured in the *StateFul* algorithm. In the SFA vs. SLA experiment, the SFA algorithm slightly overbids the uninformed opponent by setting short cycles, averaging 2.46 iterations. In at least one iteration, the SFA algorithm either matches the opponent's bid, allowing the auction winner to be chosen randomly, or slightly underbids. This behavior allows the SLA algorithm to receive some rewards, discouraging her from deviating. Interestingly, in the SFS vs. SLA experiment, nearly all runs converge to symmetric bids of 0.95 from both players. In this setting, each player earns the lowest revenues compared to the other first-price experiments. Thus, the platform can still profit from selling asymmetric access to competition data, although in this case its incentives are aligned with the incentives of the *informed* player. SFA vs. SLA is the most favorable scenario for both the platform and the informed advertiser.

6.3 Asymmetric Information with Three Players

As a final extension, we study the effect of information regimes in a GSP auction with three players. Panel A of Table 5 presents the results of the results of a benchmark experiment where all players have *Limited Information* and use SLA algorithms. Panel B summarizes experiments where an *informed* player competes against two *uninformed* opponents. The *informed* player uses an SFA

algorithm in the first row, and an SFS algorithm in the second row. All players have symmetric valuations ($v_1 = v_2 = v_3 = 1$).[40] The auction involves two slots, with CTRs of $x^1 = 5$ for the first slot and $x^2 = 2$ for the second.

Table 5: Three Players Auctions Results

Info	Learners	Runs	(b_1, b_2, b_3)	(r_1, r_2, r_3)	Revenue
Panel A: Baseline	(SLA, SLA, SLA)	97	$(0.98, 0.96, 0.97)$	$(0.24, 0.22, 0.24)$	6.3
					[6.3, 6.3]
Panel A: Asymmetric Info	(SFA, SLA, SLA)	100	$(0.51, 0.68, 0.66)$	$(0.82, 1.38, 1.3)$	3.51
					[3.33, 3.63]
	(SFS, SLA, SLA)	87	$(1.08, 0.92, 0.9)$	$(0.46, 0.22, 0.15)$	6.17
					[5.98, 6.32]

Notes: Results of experiments with three players. Column 3 shows the number of simulations that successfully converged under each regime; columns 4 and 5 - the average across runs bids and rewards at convergence for each of the players in the decreasing order of valuations; column 6 - the average auctioneer revenue across runs, along with its 95% confidence interval. SLA AIAs are run with 1,000,000 iterations, with $\beta = 1.22 \times 10^{-5}$. Simulations with at least one player using *StateFul* AIA are run with 15,000,000 iterations, with $\beta = 2.44 \times 10^{-6}$. For all simulations $\alpha = 0.10$, $\delta = 0.95$. Bids are restricted to increment of 0.1 due to computational constraints.

As expected, players report equal rewards in the benchmark experiment where all players are *uninformed*. Within each run, the winner consistently overbids her valuation, often setting a bid of 1.1 or higher. Similarly, the losing players typically set identical bids of 0.9. Since this experiment is entirely symmetric, the ranking of players across simulations is purely due to chance. Thus, the average bids and revenues for each player across runs are nearly identical.

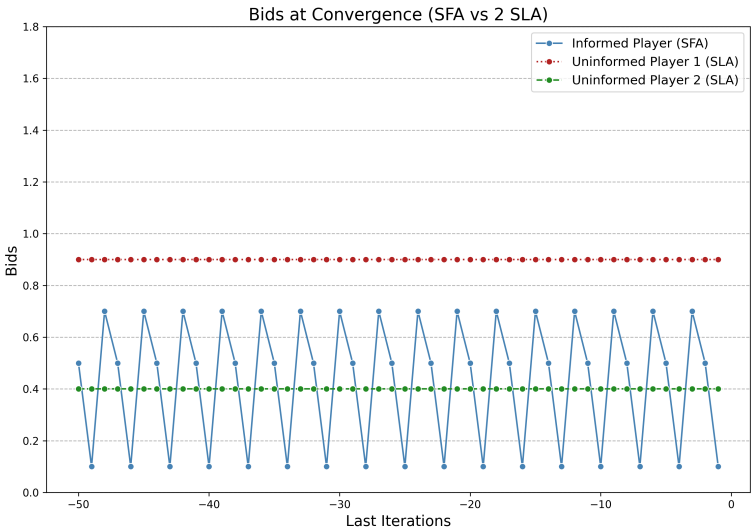
In the SFA vs. 2 SLA experiment, bid rotation equilibria emerge in all simulations. In most cases, the SFA algorithm bids between the two SLA algorithms. Occasionally, the SFA algorithm underbids the second opponent to share some revenue, providing incentives for the opponent not to deviate from their current action. Figure 9 visualizes this process in a single simulation. As in the two-players experiments, the structure of the cycles varies across simulations. The cycle lengths range from 2 to 7 bids per player, with an average length of 3.32.

In the SFS vs. 2 SLA experiment, most runs converge to an equilibrium where the informed player bids 1.1, while the two uninformed players bid 0.9. The SFS algorithm sets a high enough bid to discourage the two SLA algorithms from bidding higher, as doing so would result in negative rewards. By placing identical bids, the SLA players secure a positive expected reward. Their final ranking is purely due to chance, and their rewards at convergence are the same when the number of the simulation grows large.

Thus, the best thing the platform can do in this environment is not to share any data.

[40] Note that the size of the action space is restricted to 13 actions (from 0.1 to 1.4, excluding 1.0, with a step size of 0.1) for computational reasons.

Figure 9: Bids at Convergence in a Single Simulation - One SFA algorithm vs. Two SLA algorithms



Notes: This plot shows the emergence of cyclical behavior as the informed player (SFA) converges in a three-player simulation, where the other two symmetric players employ SLA algorithms.

7 Robustness Checks

We examine several variations of parameters of the main experiments in order to assess the robustness of our results. We provide all tables and a more detailed discussion in Section A.4 of the Appendix.

First, we test the robustness of the results of our main experiments to alternative parameterizations: (1) we alter the CTR of the first slot from 1 to 2, while keeping the second slot's CTR at 0; (2) we modify the number of iterations for the *StateFul* AIA's experiments, initially set at 5,000,000, to 2,000,000 and then 10,000,000; (3) we change the Q-matrix initialization method; (4) we adjust the initial value of δ from 0.95 to 0.5 and 0.1; (5) we add to the bids grid the players' valuation; (6) we vary the strategy that is used in the case of ties in the Q-values from random as did in the baseline experiment to a *greedy* and a *conservative* type; finally, (7) we restricted the bid grid below players' valuation as in Banchio and Skrzypacz [2023]. We summarize the findings below:

Alternative Click-Through-Rate Varying the CTR in the experiments yields results that are largely consistent with the main experiments, except for a predictable change in auctioneer revenues when CTR become more valuable.

Alternative Number of Iterations Altering the number of iterations from to the original 5,000,000 to 2,000,000 and 10,000,000 provides different results based on whether the *informed* player uses an *Asynchronous* or *Synchronous* update. For the SLA algorithm, the reduced number of iterations has no significant effect. The additional number of iterations allows the *uninformed* player to explore more extensively, also on higher bids, and the bids rotation equilibria settles at a higher value. In contrast, the opposite holds true for the SFS algorithm, where the reduced number of runs is insufficient for convergence. Nevertheless, results similar to the main experiments are achieved by both variations in the number of iterations.

Alternative Q-matrix Initialization Method Modifying the Q-matrix initialization method to reflect the discounted average reward for each player when opponents randomize uniformly (as Calvano et al. [2020]) does not significantly impact the results. The outcomes are nearly identical to those of the main experiments, which employed a random initialization method.

Alternative Discount Factor The results of different values of the discount factor produces results consistent with the discussion in Section 5.1 regarding the SFA algorithm. In contrast, a different but intuitive outcome is observed for the experiments with an SFS algorithm. The reduced discount factor benefits the *informed* player, allowing it to achieve earlier convergence and preventing the *uninformed* player from winning in any of the runs. When only the *informed* player's discount factor is set to zero, results are consistent with those reported in Table 2.

Bids Grid With Valuations Included The inclusion of the players' valuation (i.e. $b = 1.0$) in the grid of bids leads to marginally higher average bids, but this does not significantly impact the overall outcome, which remains comparable to the main experiments.

Alternative Choice Type under Q-values Ties As expected, changing the choice type under ties in the actions available to players does not significantly impact the SFA vs. SLA experiments, since the equilibrium runs converge to a bid cycle rather than a fixed bid. In contrast, the other experiments exhibit different results, as the *uninformed* player no longer plays randomly within the bid range of $[0.05, 0.95]$. It is instead forced to choose either the highest available bid among those with tied Q-values (a *greedy* strategy) or the lowest one (a *conservative* strategy). This, in turn, affects the *informed* player and auctioneer revenues. Specifically, when the bids are higher (*greedy* strategy), the revenues are lower for the *informed* player and higher for the platform, whereas in the case of the *conservative* strategy, the opposite holds true.

Big Grids Restricted Below Players' Valuation We replicate the findings of Banchio and Skrzypacz [2023] by conducting experiments in which bids are constrained to the interval [0.05,

0.95], ensuring that no player bids above their valuation of the spot. Additionally, we extend the original study by introducing not only symmetric SLA players but also by analyzing an SFA vs. SLA auction and an SFS vs. SLA auction. In the symmetric experiment, our results are consistent with the original paper, reporting convergence to the Nash Equilibrium of the game. In the SFA vs. SLA experiment, we observe outcomes akin to those highlighted in Section 4.2.1, reaffirming prior findings. As expected, the results for the SFS vs. SLA experiment differ, as the SFS player cannot overbid their valuation. This leads to convergence toward outcomes similar to those observed in the symmetric SLA setup (Table 3, second row).

8 Conclusion

This paper investigates how algorithmic competition and informational advantages, leveraged by major digital platforms, reshape the dynamics of online search auctions. By focusing on the interplay between proprietary AI tools and the strategic behavior of advertisers, we reveal critical insights into the functioning of digital markets. Our findings demonstrate that platforms, by offering bid delegation services or restricting access to critical data, can significantly alter auction outcomes, often to their advantage. The core mechanism lies in the platforms' ability to exploit asymmetric information, placing independent advertisers at a disadvantage compared to those who delegate their bidding to platform-sponsored algorithms.

Through a series of simulations, we provide evidence of the substantial performance gap between advertisers using platform-sponsored algorithms and those using independent bidding tools. Our results show that platforms' proprietary algorithms, equipped with superior information, allow advertisers who delegate their bidding to gain significant market advantages. However, we also find that the misalignment of incentives between platforms and advertisers creates a tension: while advertisers benefit from bid delegation, they are also subjected to lock-in effects, reinforcing their dependency on platform services.

From a policy perspective, our study raises important questions about the competitive implications of algorithmic bidding and data access in digital markets. Our results suggest that platforms' control over key data and the ability to design AI tools with superior learning capabilities exacerbate information asymmetries, reducing competition and increasing market concentration. These dynamics call for careful regulatory scrutiny, particularly in markets where a handful of platforms dominate the landscape and have both the incentive and ability to control access to critical market data.

In conclusion, this paper contributes to the growing literature on algorithmic competition by highlighting how platforms' informational advantages, through proprietary AI tools, affect auction outcomes and competitive dynamics in digital advertising. We suggest that policymakers consider

new approaches to mitigate the risks associated with these asymmetries, potentially through data-sharing regulations or greater transparency requirements for algorithmic bidding practices. As digital markets continue to evolve, understanding the strategic implications of algorithmic tools will be critical for fostering competition and ensuring fair market outcomes.

References

Asker, John, Chaim Fershtman, and Ariel Pakes. 2023. "The Impact of AI Design on Pricing." *Journal of Economics and Management Strategies*, forthcoming.

Azoulay, Pierre, Joshua L Krieger, and Abhishek Nagaraj. 2024. "Old moats for new models: Openness, control, and competition in generative ai." National Bureau of Economic Research.

Banchio, M., and A. Skrzypacz. 2023. "Artificial Intelligence and Auction Design." *Working Paper*.

Banchio, M., and G. Mantegazza. 2022. "Adaptive Algorithms and Collusion via Coupling." *Working Paper*.

Bergemann, Dirk, and Alessandro Bonatti. 2015. "Selling cookies." *American Economic Journal: Microeconomics*, 7(3): 259–294.

Bichler, Martin, M. Fichtl, and Stefan Heidekrüger. 2021. "Learning equilibria in symmetric auction games using artificial neural networks." *Nature Machine Intelligence*, 3: 687–695.

Brown, Zach Y, and Alexander MacKay. 2023. "Competition in pricing algorithms." *American Economic Journal: Microeconomics*, 15(2): 109–156.

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

Chen, Le, Alan Mislove, and Christo Wilson. 2016. "An empirical analysis of algorithmic pricing on amazon marketplace." *Proceedings of the 25th international conference on World Wide Web*, 1339–1349.

Competition, and Markets Authority. 2018. "Pricing algorithms: Economic working paper on the use of algorithms to facilitate collusion and personalised pricing." *CMA Working Paper CMA94*.

Decarolis, Francesco, and Gabriele Rovigatti. 2021. "From Mad Men to Maths Men: Concentration and Buyer Power in Online Advertising." *American Economic Review*, 111(10): 3299–3327.

Decarolis, Francesco, Gabriele Rovigatti, Michele Rovigatti, and Ksenia Shakhgildyan. 2023. "Artificial Intelligence & Data Obfuscation: Algorithmic Competition in Digital Ad Auctions." *CEPR Discussion Paper No. 18009*.

Dütting, Paul, Zhe Feng, Harikrishna Narasimhan, and David C. Parkes. 2017. "Optimal Auctions through Deep Learning." *CoRR*, abs/1706.03459.

Gans, Joshua S. 2024. "Market Power in Artificial intelligence." *Working Paper*.

Hagiu, Andrei, and Julian Wright. 2024. "Artificial intelligence and competition policy." *Working Paper*.

Jones, Charles I, and Christopher Tonetti. 2020. "Nonrivalry and the Economics of Data." *American Economic Review*, 110(9): 2819–2858.

Klein, Timo. 2021. "Autonomous algorithmic collusion: Q-learning under sequential pricing." *The RAND Journal of Economics*, 52(3): 538–558.

Lahaie, Sébastien. 2006. "An analysis of alternative slot auction designs for sponsored search." *Proceedings of the 7th ACM Conference on Electronic Commerce*, 218–227.

Lambin, Xavier. 2023. "Less than meets the eye: simultaneous experiments as a source of algorithmic seeming collusion." *Available at SSRN 4498926*.

Martin, Bichler, Gupta Alok, Mathews Laura, and Oberlechner Matthias. 2024. "Low Revenue in Display Ad Auctions: Algorithmic Collusion vs. Non-Quasilinear Preferences." *Working Paper*.

Mehta, Aranyak, and Andres Perlroth. 2023. "Auctions without commitment in the auto-bidding world." *arXiv preprint arXiv:2301.07312*.

Sutton, Richard S., and G. Barto Andrew. 2018. *Reinforcement Learning: An Introduction.* MIT Press.

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

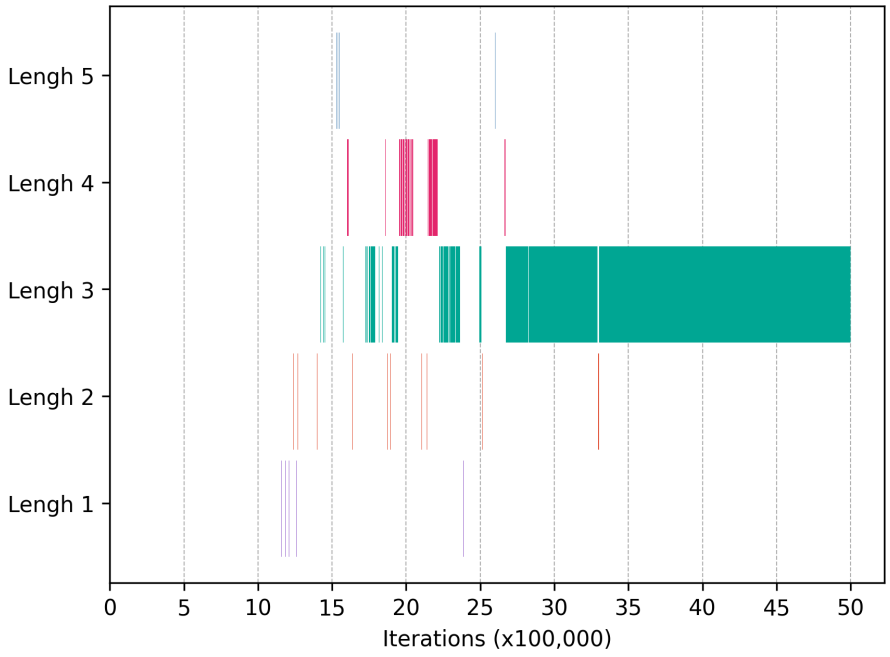
A Appendix

A.1 The Emergence of Cycling Behavior

In Section 5.1, we identified the elements that sustain an equilibrium with cycling bids. In this Appendix, we provide further details on how this type of equilibrium systematically emerges in simulations where an SFA algorithm competes against an SLA one. Although the presence of multiple states and the stochastic nature of the environment make a formal and complete understanding of the learning process in complex experiments, some stable patterns can be identified.

Cycling behavior is repeatedly and extensively observed throughout the simulations that we ran. Figure 10 tracks the emergence of cycles of different lengths in a single simulation. In around 20% of detected cycles, the SFA algorithm always wins the auction, while in roughly 10% of cycles the SLA algorithm always wins.[41] These types of cycles cannot be sustained over time, as a series of null rewards would eventually prompt a strategic shift by the losing player, breaking the cycle.

Figure 10: Bid Cycles in a Single Simulation



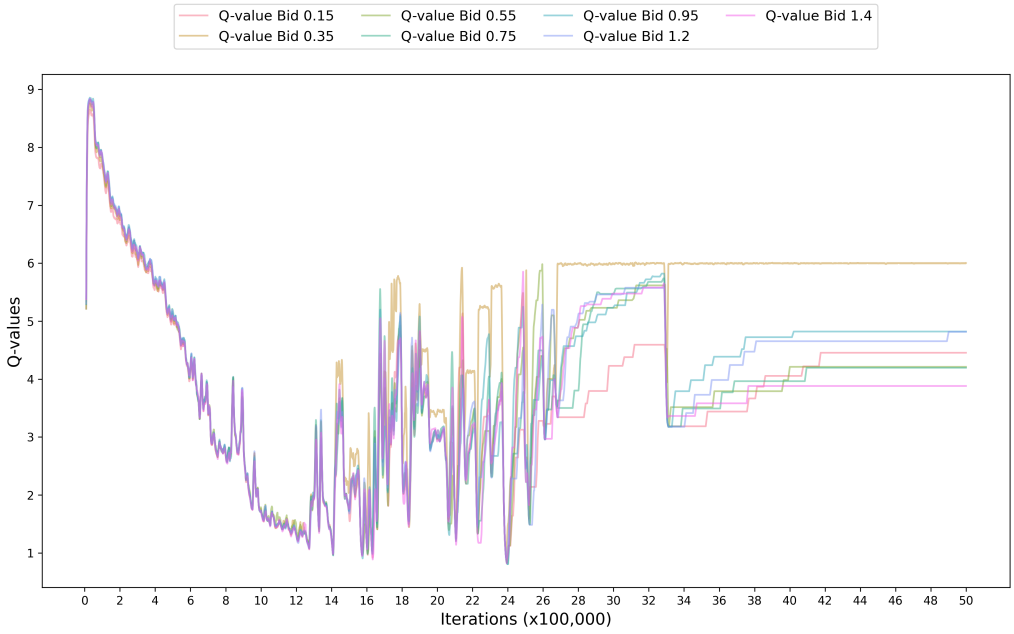
Notes: Presence of bids cycles in a single simulation. The plot shows that cycles of different length are extensively observed throughout the simulation. Cycles repeated less than 10 times were not plotted.

In contrast, cycles where bid rotation is present could be sustained for longer periods, as the latter stabilizes the optimal actions of the losing player. In the remaining 70% of cycles, the SLA algorithm generally wins all but one iteration. Eventually, the system converges to a cycle of length three (Figure 3) that includes the states: $(0.1, 0.35)$, $(0.45, 0.35)$, $(1.15, 0.35)$. The positive

[41] Here a victory is defined as a *strict* victory, i.e. $b^{SFA} > b^{SLA}$.

reward earned by the SLA algorithm within the cycle stabilizes his optimal action at 0.35, with an increase in the Q-value of the action associated with that bid. Figure 11 plots the evolution of (a subset of) Q-values of the SLA algorithm, emphasizing the presence of periods where a strong and stable preference emerges for a given action. These periods result from bid rotation equilibria, wherein the *uninformed* player earns positive rewards within the cycle.

Figure 11: Q-values of StateLess Asynchronous Player



Notes: Evolution of Q-values of the uninformed player (sampled every 1,000 iterations) in a single simulation where the informed player employs a SFA Q-learner. A rolling average of 10 iterations is applied to smooth the lines. The Q-values are calculated for the bids [0.15, 0.35, 0.55, 0.75, 0.95, 1.2, 1.4]. In the second part of the simulation, periods of stable Q-values are observed as the result of bid rotation equilibria.

Cycles in bids generally emerge by chance, and are sustained by the factors highlighted in Section 5.1.

The fact that bid rotation equilibria emerge by chance may also explain why they are often suboptimal. Table 1 shows that the winning bid of the SLA algorithm averages to 0.478. However, bid rotation would be more advantageous from the perspective of the *informed* player if the *uninformed* player were to bid a lower amount.

A.2 Impossibility of Cycles in SFS vs. SLA Experiments

A.2.1 Setting

Consider a baseline setting with two players and one slot for sale. The first advertiser, Player 1, is the *informed* player and employs an SFS algorithm, while Player 2, the *uninformed* player, uses an SLA algorithm. $\mathcal{B} = (b_1, b_2, \ldots, b_K)$ is the set of possible bids, where K represents the number of

distinct bids available to each player. A state s in period t in case of the SFS algorithm is given by the bids of the players in the past period, $s_t = (b^1_{t-1}, b^2_{t-1})$, whereas a state is empty, $s_t = \emptyset$, for the case of SLA algorithm. The state space is defined as $S = \mathcal{B} \times \mathcal{B}$. Generally, convergence is evaluated after T periods. Since in this section we consider only the players' behavior at convergence, here we denote by $t = 1$ the first period once convergence is reached. The convergence horizon is denoted as $\mathcal{T}$.

Let $A^i_t(s)$ be the set of greedy actions in state s at time t for player i, i.e. $A^i_t(s) = \{b \in \mathcal{B} : Q^i_t(s, b) = \max_{b \in \mathcal{B}} Q^i_t(s, b)\}$. Convergence means that players always bid their greedy action in the current state, and the set of greedy actions in a given state does not change. When $|A^i_t(s)| > 1$, the bid is chosen randomly from the set. Furthermore, when k highest Q-values in a given state can be strictly ordered, we define $a^i_{1t}(s), a^i_{2t}(s), \ldots, a^i_{kt}(s)$ as the bids satisfying $Q^i_t(s, a^i_{1t}(s)) > Q^i_t(s, a^i_{2t}(s)) > \cdots > Q^i_t(s, a^i_{kt}(s))$.

A.2.2 Impossibility of Cycles in SFS vs. SLA Experiments

In this setting, cyclical bidding behavior at convergence is impossible for the SLA Q-learner. Specifically, a cyclical pattern of distinct bids, such as $(b_1, b_2, \ldots, b_L, b_1, b_2, \ldots, b_L, \ldots)$ with $L \geq 2$, cannot emerge as a stable strategy over the convergence horizon $\mathcal{T}$. The reasoning is straightforward: since the Q-learner is *StateLess*, its strategy is determined by the argmax of its Q-values, $Q(b_1), Q(b_2), ..., Q(b_k)$, which do not depend on the state. If a unique bid corresponds to the maximum Q-value, the action set A^i_t is a singleton, meaning that player i will always choose the same bid at convergence. If the Q-values are tied, the greedy action is chosen randomly from A^i_t, preventing a stable cyclical behavior.

Proposition 1. *In a repeated Second-Price auction where an SFS Q-learner competes against an SLA Q-learner, cyclical state transitions of the form $\mathcal{S} = (s_1, s_2, \ldots, s_L, s_1, s_2, \ldots, s_L, \ldots)$, $L \geq 2$ cannot occur at convergence, where $s_t = (b^1_{t-1}, b^2_{t-1})$. There always exists a $t \in \mathcal{T}$ such that the SFS Q-learner deviates from $\mathcal{S}$.*

Proof. For cyclical behavior to be observed over $\mathcal{T}$, it is necessary that the greedy action in each state remains unique:

$$A^i_t(s_l) = \{a^i_1(s_l)\} \quad \forall l \in \{1, 2, \ldots, L\},\ \forall t \in \mathcal{T},\ \forall i \in \{1, 2\} \tag{3}$$

For the SLA Q-learner, this condition holds if and only if its action is a fixed one at convergence, $A^2_t = \{b^2_*\}\ \forall t \in \mathcal{T}$, since this algorithm does not condition future actions on the state. Thus, $\mathcal{S} = ((a^1_1(s_1), b^2_*), (a^1_1(s_2), b^2_*), \ldots, (a^1_1(s_L), b^2_*), \ldots)$.

Next, consider the SFS Q-learner. Let $\mathcal{S}$ be a sequence of states at convergence. For simplicity, let $L = 2$ and define the states that constitute $\mathcal{S}$ as $s_1 = s_A = (b_A, b^2_*)$ and $s_2 = s_B = (b_B, b^2_*)$.

Since the state is defined by the greedy actions in the past period $\arg\max_{b' \in \mathcal{B}} Q_t^1(s_A, b') = b_B$ and $\arg\max_{b' \in \mathcal{B}} Q_t^1(s_B, b') = b_A$. Action b_B is taken in state s_A, whereas action b_A is taken in state s_B. Because of the Synchronous update, rule Q-values corresponding to the actions not taken in a given state are also being updated.

Thus, the Q-value update rule of the SFS Q-learner is given by:[42]

$$Q_{t+1}(s,b) = Q_t(s,b) + \alpha \left(r(b) + \delta \max_{b' \in \mathcal{B}} Q_t((b, b_*^2), b') - Q_t(s,b) \right) \quad \forall b \in \mathcal{B}. \tag{4}$$

When state transitions follow $\mathcal{S}$, the following system of equations holds:

$$\begin{cases} Q_{t+1}(s_A, b_A) = Q_t(s_A, b_A) + \alpha \left(r(b_A) + \delta Q_t(s_A, b_B) - Q_t(s_A, b_A) \right) \\ Q_{t+1}(s_A, b_B) = Q_t(s_A, b_B) + \alpha \left(r(b_B) + \delta Q_t(s_B, b_A) - Q_t(s_A, b_B) \right) \\ Q_{t+1}(s_B, b_B) = Q_t(s_B, b_B) + \alpha \left(r(b_B) + \delta Q_t(s_B, b_A) - Q_t(s_B, b_B) \right) \\ Q_{t+1}(s_B, b_A) = Q_t(s_B, b_A) + \alpha \left(r(b_A) + \delta Q_t(s_A, b_B) - Q_t(s_B, b_A) \right) \end{cases} \tag{5}$$

This system converges to the following steady state (we plug in $Q_{t+1} = Q_t = Q^*$ and solve for the Q-values):

$$\begin{cases} Q^*(s_A, b_A) &= \frac{\delta r(b_B) + r(b_A)}{1 - \delta^2} \\ Q^*(s_A, b_B) &= \frac{\delta r(b_A) + r(b_B)}{1 - \delta^2} \\ Q^*(s_B, b_B) &= \frac{\delta r(b_A) + r(b_B)}{1 - \delta^2} \\ Q^*(s_B, b_A) &= \frac{\delta r(b_B) + r(b_A)}{1 - \delta^2} \end{cases} \tag{6}$$

As a reminder, here:

$$r(b) = \begin{cases} 0 & \text{if } b \leq b_*^2 \\ (v^1 - b_*^2) & \text{if } b > b_*^2 \end{cases} \tag{7}$$

If b_A and b_B yield equal reward $r(b_A) = r(b_B)$, the four Q-values defined in (6) converge to $r/(1-\delta)$. This means that at convergence both $a_1(s_A)$ and $a_1(s_B)$ are not defined since both actions b_A and b_B lead to the same Q-value, and as a result, (3) cannot be satisfied.

If instead b_A and b_B are such that $r(b_A) \neq r(b_B)$, given the reward function in (7), either $b_A < b_*^2 < b_B$ or $b_B < b_*^2 < b_A$. Suppose $b_A < b_*^2 < b_B$ and $r(b_B) > r(b_A) = 0$. This means that in the state s_B the initial greedy bid b_A is suboptimal at convergence since $Q^*(s_B, b_B) > Q^*(s_B, b_A)$ (as $\delta < 1$). Thus, there exists a $t \in \mathcal{T}$ such that $Q_t(s_B, b_B) > Q_t(s_B, b_A)$. This means that at some point the greedy action in state B will change from b_A to b_B, breaking the cycle described by $\mathcal{S}$, and violating (3).

In general, because of the *synchronous* update, there always exists $t^* \in \mathcal{T}$ such that

[42] We omit index 1 for simplicity.

$Q_{t^*}((b_l, b_*^2), b_l) \geq Q_{t^*}((b_l, b_*^2), a_1^1(s_l))$ for at least one l in $1, 2, \ldots, L$.[43] This means that, in at least one state that makes up the cycle described by $\mathcal{S}$, the Q-value of the bid that defines the state itself at some point will be greater or equal to the initial optimal bid in that state, violating (3).

□

A.3 Discussion of Convergence

In multiagent reinforcement learning, the reward structure and update rules depend on other players' actions, making each player's optimization problem inherently non-stationary. As bidders experiment and learn, the same action can yield different rewards, which explains the lack of general convergence results for Q-learning in games.

Importantly, convergence can be verified *ex-post*. Following the definition in Calvano et al. [2020], in our main experiments with *StateFul* and *StateLess* Q-learners, we consider convergence achieved if: (i) the *informed* player's optimal strategy remains unchanged for 100,000 consecutive periods, i.e. $b_t^i(s) = \arg\max_{b \in \mathcal{B}}[Q_{i,t}(s, b)]$ remains constant over 100,000 iterations for each state s; (ii) the *uninformed* player's optimal strategy remains unchanged for at least 1,000 consecutive periods. Whenever these two conditions are met, we assume the learning process has completed and stable behavior has been attained. The session is terminated upon convergence or after 5 million iterations, whichever comes first.

Only a few runs failed to converge, and consistent with Calvano et al. [2020] and Banchio and Skrzypacz [2023], we considered only converged runs for all charts and tables. Table 6 presents statistics on the number of Q-value updates in our main experiments. At convergence, Q-matrix cells are typically visited hundreds or thousands of times, depending on the algorithm used.

Convergence is only reached when experimentation is nearly complete, as even a small exploration rate keeps the environment too non-stationary. In the *StateFul Synchronous* experiments, 5 million iterations are required for the probability of exploration to decrease to 5×10^{-6}, following the formula $\exp(-5,000,000 \times 2.44 \times 10^{-6}) = 5 \times 10^{-6}$. Convergence is faster in the *StateLess Asynchronous* experiments due to the Q-matrix reducing to a Q-vector. This is why in experiments where both players employ SLA algorithms 1,000,000 periods and $\beta = 1.22e-05$ were sufficient to achieve convergence with adequate exploration.

[43] The extension to $L > 2$ is straightforward. One needs to define a system of equations as in (5). The system includes L equations for each of the L states that make up $\mathcal{S}$ and find the steady state by setting $Q_{t+1} = Q_t = Q^*$.

Table 6: Summary statistics of Q-matrix cells counts for the different players.

Player (Experiment)	Count	Mean	σ	Min	25%	50%	75%	Max
SLA Player 1 (SLA vs SLA)	27	37,037	29,681	20,446	22,419	25,408	35,947	160,579
SLA Player 2 (SLA vs SLA)	27	37,037	29,917	20,751	22,818	25,250	34,141	155,400
SFA Player (SFA vs SLA)	19,683	237	312	9	48	97	297	3,148
SLA Player (SFA vs SLA)	27	172,526	141,015	23,087	71,257	124,519	248,676	589,912
SFS Player (SFS vs SLA)	19,683	6,444	14,419	382	629	927	3,158	97,175
SLA Player (SFS vs SLA)	27	173,990	175,894	33,756	104,486	136,524	184,133	941,647

A.4 Robustness Checks

This section presents the robustness checks, building upon the initial discussion in Section 7. We provide a detailed report and include the results tables for the experiments.

Alternative Click-Through-Rate We execute an experiment identical to the experiment presented in Section 4 with the exception that the initial click-trough-rate (CTR) for the first ad slot is increased from 1 to 2. The second slot still keeps a CTR value of 0.

By doubling the CTR from 1 to 2, we observe similar results to the main experiment, as shown in Table 7. The only notable difference is the anticipated increase in auctioneer revenue, which is a direct consequence of the CTR being 2. As auctioneer revenue is calculated by multiplying the number of clicks by the second-highest bidder's bid, the additional clicks on the slots naturally lead to a higher revenue.

Table 7: Alternative Click Trough Rate (CTR) Experiments Results

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SLA vs SLA	981	$(0.83, 0.83)$	$(0.5, 0.49)$	$(0.51, 0.49)$	0.0	1.16	0.51	$(1.15, 1.16)$	$(0.51, 0.5)$	1.01 [1.01, 1.01]
SFA vs SLA	90	$(0.85, 0.55)$	$(0.69, 0.3)$	$(0.78, 0.22)$	0.03	0.91	0.5	$(1.01, 0.55)$	$(0.32, 0.55)$	1.01 [0.96, 1.06]
SFS vs SLA	90	$(0.91, 0.69)$	$(0.66, 0.32)$	$(0.68, 0.32)$	0.0	1.08	0.51	$(1.1, 1.05)$	$(0.5, 0.51)$	1.02 [1.01, 1.03]

Notes: The CTRs are $(x_1, x_2) = (2, 0)$

Alternative Number Of Iterations In this experiment, we investigate the impact of varying the number of iterations on the results. In the main experiment, we set the maximum number of iterations to 5,000,000 for all *StateFul* AIA experiments. To ensure robustness, we also conducted

additional experiments with reduced (2,000,000) and increased (10,000,000) iteration counts, as reported in Table 8.

When comparing the results of the SFA vs. SLA experiments across different iteration counts, we observe that the differences between 2,000,000 and 5,000,000 iterations are relatively small and may be attributed to the variation in the number of runs performed for the robustness checks versus the main experiments. However, when the iteration count is increased to 10,000,000, the results exhibit a notable change: the SLA algorithm tends to adopt a higher constant bid. This strategy is likely learned due to the higher probability of small deviations in these extended runs. As for the SFA algorithm, this strategy makes bid rotation less profitable, as a higher bid is now necessary to win the slot.

The variations of the SFS vs. SLA experiments exhibit mostly align with the main results presented in Table 1. Notably, the number of converged runs at 10,000,000 iterations is similar to that of the main experiment, but the SFS algorithm is now able to win more runs. In contrast, the 2,000,000 iteration count appears to be insufficient for some experiments to converge, as evidenced by the lower number of convergent runs compared to the main experiment. Nevertheless, winning shares and bids for the convergent runs are consistent with Table 1.

Table 8: Alternative Interations Experiments Results

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SFA vs SLA (2'000'000 Iter.)	97	$(0.8, 0.51)$	$(0.38, 0.17)$	$(0.76, 0.24)$	0.02	0.86	0.46	$(0.97, 0.51)$	$(0.31, 0.51)$	0.46 [0.43, 0.49]
SFA vs SLA (10'000'000 Iter.)	68	$(0.87, 0.67)$	$(0.28, 0.17)$	$(0.7, 0.3)$	0.04	1.0	0.55	$(1.08, 0.83)$	$(0.43, 0.6)$	0.55 [0.51, 0.59]
SFS vs SLA (2'000'000 Iter.)	70	$(0.93, 0.66)$	$(0.36, 0.14)$	$(0.73, 0.27)$	0.0	1.08	0.51	$(1.1, 1.05)$	$(0.5, 0.51)$	0.51 [0.51, 0.51]
SFS vs SLA (10'000'000 Iter.)	100	$(1.04, 0.58)$	$(0.43, 0.06)$	$(0.88, 0.12)$	0.0	1.1	0.51	$(1.11, 1.05)$	$(0.5, 0.51)$	0.51 [0.5, 0.52]

Alternative Q-matrix initialization Method To investigate the potential impact of initialization methods on our experimental results, we modify the Q-matrix initialization approach from random to informed. This informed initialization method, as used in Calvano et al. [2020], involves initializing each player's Q-matrix such that each row represents the discounted average reward obtained when opponents randomize uniformly.

Table 9 shows that an informed initialization yields almost identical results to the main experiments, except for some slight deviations in certain runs of the SFA vs. SLA experiment. These can be attributed to the fewer number of simulations run in the robustness check experiment.

Table 9: Informed Initialisation Experiments Results

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SLA vs SLA	100	$(0.85, 0.82)$	$(0.26, 0.23)$	$(0.53, 0.47)$	0.0	1.17	0.51	$(1.15, 1.18)$	$(0.5, 0.51)$	0.51
										[0.51, 0.51]
SFA vs SLA	89	$(0.83, 0.43)$	$(0.44, 0.18)$	$(0.78, 0.22)$	0.01	0.88	0.39	$(1.02, 0.42)$	$(0.22, 0.43)$	0.39
										[0.37, 0.41]
SFS vs SLA	91	$(0.88, 0.7)$	$(0.32, 0.17)$	$(0.65, 0.35)$	0.0	1.08	0.5	$(1.09, 1.05)$	$(0.5, 0.51)$	0.5
										[0.5, 0.5]

Notes: Informed initialization creates a Q-matrix in the first iteration of the experiment such that each row represents the discounted average reward obtained when opponents randomize uniformly.

Alternative Discount Factor We test different variations of the discount factor δ by adjusting its value for the original 0.95 to 0.5 and 0.1. Note that a decrease of δ is interpreted as a reduction in the present value of future rewards. We first show results where changes to the discount factor affect both players symmetrically. Then, we discuss an experiment where only the informed player experiences a variation in the discount factor with $\delta = 0$.

We find that when the learning rate is low, the SFA player struggles to execute a successful bid rotation strategy. This limitation arises from the larger Q-matrix compared to the SLA player and the impossibility to successfully perform bid rotations. Consequentially, in both scenarios where δ is set to 0.5 and 0.1, the SLA algorithm consistently wins at convergence. These results align with our discussion in Subsection 5.1

In contrast, the SFS algorithm benefits from the reduced discount factor, as it wins all the auctions, resulting in larger profits compared to the main experiments. Unlike the main experiments, where there is a possibility for the SLA algorithm to win, the reduced discount factor leads to a consistent outcome. Furthermore, the bids in this setting are aligned with those observed in the SFS algorithm's winning runs of the main experiments, suggesting that the decreased discount factor primarily affects the runs where it should have won. In these cases, the SLA algorithm's ability to successfully set a higher bid than the informed player is diminished, as its opponent secures the optimal 1.05 bid earlier on into the experiment (1'500'000 iterations before the convergence check begins).

Table 10: Alternative Discount Factor Experiments Results

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SFA vs SLA ($\delta = 0.1$)	95	$(0.5, 1.05)$	$(-0.0, 0.5)$	$(0.0, 1.0)$	0.0	1.05	0.5	$(1.4, 1.05)$	$(0.5, 1.05)$	0.5
										[0.5, 0.5]
SFA vs SLA ($\delta = 0.5$)	99	$(0.5, 1.05)$	$(0.0, 0.5)$	$(0.0, 1.0)$	0.0	1.05	0.5	$(nan, 1.05)$	$(0.5, nan)$	0.5
										[0.5, 0.5]
SFS vs SLA ($\delta = 0.1$)	100	$(1.05, 0.5)$	$(0.5, 0.0)$	$(1.0, 0.0)$	0.0	1.05	0.5	$(1.05, nan)$	$(nan, 0.5)$	0.5
										[0.5, 0.5]
SFS vs SLA ($\delta = 0.5$)	100	$(1.07, 0.51)$	$(0.5, 0.0)$	$(0.99, 0.01)$	0.0	1.08	0.5	$(1.08, 1.05)$	$(0.51, 0.5)$	0.5
										[0.5, 0.5]

When the discount factor is set to zero only for the *informed* player, we find no significant

changes to the results presented in Section 5.1. Furthermore, we observe that when $\delta = 0$, the *informed* player is unable to win any auction.

Table 11: Results with Asymmetric Discount Factors

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SFA vs SLA ($\delta_1 = 0, \delta_2 = 0.95$)	100	$(0.56, 1.15)$	$(0.0, 0.44)$	$(0.0, 1.0)$	0.0	1.15	0.56	$(nan, 1.15)$	$(0.56, nan)$	0.56 [0.55, 0.57]
SFS vs SLA ($\delta_1 = 0, \delta_2 = 0.95$)	100	$(0.51, 1.07)$	$(0.0, 0.49)$	$(0.0, 1.0)$	0.0	1.07	0.51	$(nan, 1.07)$	$(0.51, nan)$	0.51 [0.5, 0.52]

Bids Grid With Players' Valuation Included In the main experiments, algorithms were endowed with bid grids that excluded the players' valuation of slot being auctioned. For a more in-depth explanation, please refer to Section 3.2. We introduced a bid value of 1.0 to observe its impact on the experiment.

Although the average bids are marginally higher due to the inclusion of the bid 1.0, our results show no substantial difference compared to the main experiment.

Table 12: Results when Bids Grid Include Players' Valuation

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SLA vs SLA	93	$(0.88, 0.82)$	$(0.24, 0.2)$	$(0.55, 0.45)$	0.01	1.14	0.56	$(1.13, 1.14)$	$(0.57, 0.55)$	0.56 [0.55, 0.57]
SFA vs SLA	96	$(0.78, 0.46)$	$(0.43, 0.15)$	$(0.79, 0.21)$	0.04	0.83	0.42	$(0.93, 0.45)$	$(0.28, 0.46)$	0.42 [0.39, 0.45]
SFS vs SLA	95	$(0.97, 0.68)$	$(0.33, 0.11)$	$(0.77, 0.23)$	0.01	1.09	0.56	$(1.11, 1.03)$	$(0.52, 0.57)$	0.56 [0.55, 0.57]

Notes: Players can choose between 28 different actions as the bid value of 1 belongs to the bids set of both players. That is $1 \in \mathcal{B}^i$.

Alternative Choice Type under Q-values Ties We examine the effect of alternative tie-breaking strategies for the Q-values of a player on the experimental outcomes, building on the random approach employed in the baseline experiments. In particular, we investigate two additional strategies: the *greedy* strategy, which selects the highest bid among those with tied Q-values, and the *conservative* strategy, which chooses the tied action with the lowest bid.

As shown in Table 13, the results indicate that, except for the SFA against SLA players' experiment, the bids from other players are higher. This leads to an increase in auctioneer revenues. The increase in bids can be attributed to the losing SLA player consistently bidding 0.95 instead of bidding at random in the set of bids that ranges between 0.05 and 0.95. When facing another SLA player, the winning bid is set at 1.05 or higher to discourage deviations. If the *informed* player employs an SFA algorithm, the experiment will be identical to the ones presented in Section4.2.1, as the bid cycle equilibrium is not affected by alternative choice types. However, in the SFS vs. SLA experiment, the winning SFS player's bid does not consistently converge to 1.05, instead often converging to higher values in most runs.

Table 13: *Greedy* Choice Under Ties Experiment Results

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SLA vs SLA	100	$(1.02, 1.04)$	$(0.02, 0.03)$	$(0.45, 0.55)$	0.0	1.11	0.95	$(1.11, 1.12)$	$(0.95, 0.95)$	0.95
										[0.95, 0.95]
SFA vs SLA	95	$(0.8, 0.51)$	$(0.39, 0.16)$	$(0.76, 0.24)$	0.04	0.88	0.45	$(0.97, 0.59)$	$(0.32, 0.49)$	0.45
										[0.42, 0.48]
SFS vs SLA	96	$(1.21, 0.99)$	$(0.03, 0.02)$	$(0.57, 0.43)$	0.0	1.25	0.95	$(1.4, 1.05)$	$(0.95, 0.95)$	0.95
										[0.95, 0.95]

Notes: The greedy choice type selects the tied action with the highest associated bid .

Table 14 shows that adopting a *conservative* strategy to resolving tied Q-values leads to lower overall bids in all games. This is because, when a player is indifferent between multiple actions, it selects the one with the lowest associated bid. This effect is particularly pronounced in the SLA vs. SLA and SFS vs. SLA experiments, where the losing player, being indifferent to bids within the $[0.05, 0.95]$ range, consistently chooses the 0.05 bid at convergence. Consequently, the winning player's revenues increase significantly, while the auctioneer's revenues decrease compared to the main experiments.

Table 14: *Conservative* Choice Under Ties Experiment Results

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SLA vs SLA	100	$(0.56, 0.61)$	$(0.45, 0.5)$	$(0.47, 0.53)$	0.0	1.13	0.05	$(1.14, 1.11)$	$(0.05, 0.05)$	0.05
										[0.05, 0.05]
SFA vs SLA	92	$(0.76, 0.49)$	$(0.4, 0.18)$	$(0.75, 0.25)$	0.04	0.84	0.43	$(0.94, 0.55)$	$(0.3, 0.47)$	0.43
										[0.4, 0.46]
SFS vs SLA	100	$(1.03, 0.11)$	$(0.89, 0.06)$	$(0.94, 0.06)$	0.0	1.09	0.05	$(1.1, 1.05)$	$(0.05, 0.05)$	0.05
										[0.05, 0.05]

Notes: The *conservative* choice type selects the tied action with the lowest associated bid

Big Grids Restricted Below Players' Valuation We build an experiment with the bids grids identical to the ones provided in Banchio and Skrzypacz [2023] in order to check the robustness of our results. That is, bid grids for this experiment are limited to the range $[0.05, 0.95]$ and do not include values above a player's valuation of the ad spot (that is $v_i = 1$). This contrasts with our main experiments, where the bids grids include values above 1, so that players are able to learn a wider range of strategies. A more detail discussion can be found in Section 3.2. We begin by replicating the original paper employing two symmetric SLA players. Next, we expand by assessing the key contributions of this work in the environment of Banchio and Skrzypacz [2023]'s experiments. We do so by introducing two asymmetric players –one of the SFA type and the other of the SFS type.

When we employ players that are identical to the one of the original paper, our results are consistent. Specifically, we observe convergence to the Nash Equilibrium with bids $(0.95, 0.95)$.

When exploring the asymmetric algorithms, we obtain results consistent with those presented in Section 4.2. More specifically, when considering auctions using the SFA algorithm, the observed patterns closely resemble those of our main experiment. The SFA algorithm remains the

most profitable choice for players, as it effectively leverages bid cycles to exploit its informational advantage.

As anticipated, the results differ significantly form the main experiments when contrasting auctions involving SFS and SLA players. In this case, the SFS player is unable to fully exploit its informational advantage by overbidding its own valuation, as the bids are now restricted to below 0.95. The player eventually settles into the Nash Equilibrium, where it wins half of the lotteries. Consequently, platform revenues in this scenario are comparable to those observed in the uninformed experiment.

Table 15: Results with Bid Grids Restricted Below Players' Valuation (Banchio and Skrzypacz [2023])

Experiment	Runs	(b_1, b_2)	(r_1, r_2)	(w_1, w_2)	Ties Ratio	Winning Bid	Losing Bid	(wb_1, wb_2)	(lb_1, lb_2)	Revenue
SLA vs SLA (1'000'000 Iter.)	100	$(0.95, 0.95)$	$(0.02, 0.03)$	$(0.5, 0.5)$	1.0	0.95	0.95	$(0.95, 0.95)$	$(0.95, 0.95)$	0.95 [0.95, 0.95]
SFA vs SLA (5'000'000 Iter.)	100	$(0.68, 0.54)$	$(0.36, 0.2)$	$(0.73, 0.27)$	0.1	0.73	0.44	$(0.82, 0.49)$	$(0.26, 0.51)$	0.48 [0.44, 0.52]
SFS vs SLA (5'000'000 Iter.)	100	$(0.92, 0.92)$	$(0.04, 0.04)$	$(0.5, 0.5)$	1.0	0.92	0.92	$(0.92, 0.92)$	$(0.92, 0.92)$	0.92 [0.89, 0.95]