AI Collusion in Procurement Auctions

Andrew Tai*

April 2, 2026

Abstract

Auctions are an important format for defense procurement, historically accounting for at least $1 billion per year.[1] A common format is the reverse auction, in which potential sellers submit bids for a contract, and the buyer selects the lowest bid.[2] This design is well-suited for small uniform contracts where price is the only dimension of choice; common applications include commodities and routine services.

However, this setting is also conducive to the application of AI bidding algorithms by sellers. Such algorithms generally seek to maximize profit for their owners. An obvious way to do this is to collude. AI algorithms for bidding introduces the possibility that competing algorithms collude to increase prices without explicitly communicating with each other.

I conduct simulations in a simplified environment to test this possibility. I find that Q-learning algorithms indeed converge to supra-competitive prices in reverse auctions, allowing firms to extract higher than competitive profit. The learned policies also suggest they sustain collusion by punishing competitors for deviations.

1 Introduction

Increasingly, firms use artificial intelligence algorithms to price goods and services. As early as 2015, Chen, Mislove, and Wilson (2016) find that one-third of sellers on Amazon had automated their pricing. Aggarwal et al. (2024) document extensive use of algorithms to automate bidding for advertising space in online auctions.

*DSCU. Contact: andrew.a.tai.civ@mail.mil . Disclaimer: The views expressed are those of the author and do not reflect the official policy or position of the Department of War or the U.S. Government.

[1] DoD Inspector General (2012); US GAO (2018)

[2] Alper and Boning (2003); Coughlan, Gates, and Lamping (2008)

Auctions are an important format for defense procurement, historically accounting for at least $1 billion per year.[3] A common format is the reverse auction, in which potential sellers submit bids for a contract, and the buyer selects the lowest bid.[4] This design is well-suited for small uniform contracts where price is the only dimension of choice; common applications include commodities and routine services.

However, this setting is also conducive to the application of AI bidding algorithms by sellers. Indeed, the similarities between such applications and online advertising space are clear. While details are private, bidding algorithms generally seek to maximize profit for their owners. An obvious way to do this is to collude. The application of AI algorithms for bidding introduces the possibility that competing algorithms learn to collude to increase prices, even without explicitly communicating with each other.

In general under US antitrust law, collusion is defined as agreements between competitors to fix prices or market shares, requiring some "meeting of the minds".[5] This paper takes a broader definition of "collusion" as including tacit behavior to fix prices at supra-competitive levels. Since the AI algorithms do not explicitly coordinate with each other, it may be impossible to establish that the law was violated. Yet the outcome is the same – higher profits and thus higher costs to the buyer.

A common baseline learning algorithm is Q-learning. Prior research, including Calvano, Calzolari, Denicolò, and Pastorello (2020) and Klein (2021), has shown Q-learning algorithms learn to collude to extract profit above competitive prices. In the benchmark model of competition, Bertrand competition, the general pattern is Edgeworth cycling, characterized by an initial jump to a high price then a slow decline to the competitive price. During the decline period, competing firms receive higher profits due to the above-competitive price.

The possibility of autonomous collusion in procurement auctions was an open question. The literature has not been extended to reverse auctions. Nor had it been extended to more advanced learning algorithms such deep Q-learning, which is likely closer to state-of-the-art algorithms and can better accommodate complicated environments with fine price grids. This paper applies Q-learning to simulations of reverse auctions to evaluate whether and to what extent such algorithms learn to collude. I find that Q-learning algorithms indeed converge to supra-competitive prices in reverse auctions, allowing firms to extract higher

[3] DoD Inspector General (2012); US GAO (2018)
[4] See Alper and Boning (2003); Coughlan, Gates, and Lamping (2008)
[5] See US Department of Justice (2023).

than competitive profit. Depending on the experimental parameters, the profit extracted from the seller can be as much as 3x the competitive profit. The learned policies also suggest they sustain collusion by punishing competitors for deviations.

The results have implications for the cost and viability of auctions for procurement. If AI algorithms can collude to a high extent, the government may face increasing costs for contracts procured through auctions and similar formats. A theoretical understanding of these processes can support practical policy solutions to evolving challenges in defense procurement. Further results demonstrate the value of expanding the defense industrial base. Increasing the number of firms (even if all of them use AI algorithms for bidding) can markedly decrease the profit extracted. This paper's results also contain lessons byond of the use of AI algorithms. Humans[6] can also replicate the behavior arising in these simulations, illustrating how tacit collusion can arise without explicit communication.

I also note this paper's limitations. First, this paper is an experiment – the findings here establish theoretical possibility. In reality, firms (both using AI algorithms and not) may or may not collude. Detecting such collusion may be difficult; I refer readers to Asker (2010) for details. Second, even though Q-learning agents converge to collusive pricing, doing so can take many periods, consistent with findings in Calvano, Calzolari, Denicolò, and Pastorello (2020). In real settings, these learning periods may result in large losses. However, more advanced algorithms may converge faster. Finally, the environment is highly stylized. The model is a major simplification of any real auction environment. However, the simplicity of the model presented and simulations offers clear and easily interpretable evidence of the possibility of algorithmic collusion.

The paper proceeds as follows. Section 2 recounts basics on Q-learning. Section 3 documents the simulated experiments. Section 4 gives the empirical results, and Section 5 concludes.

2 Q-learning

This paper trains Q-learning models to auction environments. While algorithms actually used by firms are likely much more advanced, little is publicly known. However, Q-learning is a popular basic algorithm. Advantages are that it can be defined using few parameters and that its behavior is readily interpretable. To my knowledge, Waltman and Kaymak

[6]Humans can of course be seen as very advanced learning algorithms.

(2008) are the first to apply Q-learning to an economic competition model. This section recounts basics of Q-learning; familiar readers may safely skip to the next section.

Q-learning is a model-free reinforcement learning (RL) algorithm. In a repeated environment, the Q-learning algorithm records payoffs from past actions and states to learn a state-action correspondence. This correspondence is typically referred to as a **policy**. Note that this does not refer to policy in the sense of principles or laws of a government. Through the rest of this paper, except in the conclusion where it is clear, policy refers only to the technical termonology for Q-learning algorithm behavior.

The original setup, due to Watkins (1989), is a stationary Markov decision process (MDP). Notably, Q-learning is for single agent decision environments. In each period $t = 0, 1, 2, \ldots$, the agent observes a state $s_t \in S$ and chooses an action $a_t \in A(s_t)$, where the available actions perhaps depend on the state. The agent receives reward $\pi_t(s_t, a_t)$. Next period's state is drawn from a probability distribution $f(s_{t+1} \mid s_t, a_t)$, perhaps dependent on the current action and state. The agent seeks to maximize expected net present value

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t \pi_t(s_t, a_t)\right]$$

where $\delta < 1$ is the discount factor.

Maximizing this quantity is a dynamic programming problem, which is often solved analytically or computationally using the Bellman equation

$$V(s_t) := \max_{a_t \in A(s_t)} \left\{\mathbb{E}\left[\pi_t \mid s_t, a_t\right] + \delta \mathbb{E}\left[V(s_{t+1}) \mid s_t, a_t\right]\right\}$$

Intuitively, the value of a state is the combination of the reward from the action this period and the discounted value of the subsequent state, given the agent takes the optimal actions.

The Q function is similar, but lists values of pairs (s, a).

$$Q(s_t, a_t) = \mathbb{E}\left[\pi_t \mid s_t, a_t\right] + \delta \mathbb{E}\left[\max_{a_{t+1} \in A(s_{t+1})} Q(s_{t+1}, a_{t+1}) \mid s_t, a_t\right]$$

Since the MDP is stationary, the t subscripts can be generic:

$$Q(s, a) = \mathbb{E}\left[\pi \mid s, a\right] + \delta \mathbb{E}\left[\max_{a' \in A(s')} Q(s', a') \mid s, a\right]$$

where s is the current state, and s' is the next state.

When S and A are finite, Q can be represented as a matrix, where the $Q_{s,a}$ entry is the expected reward for (s, a). If the Q matrix is known, selecting the optimal policy is straightforward. Of course, the Q matrix may be unknown and difficult to solve for analytically. The **Q-learning** algorithm estimates the Q matrix over periods of repetition.

For $t = 0$, the Q matrix is initialized at Q_0 arbitrarily (e.g. all 0s). Given Q_t and (s, a), the algorithm updates

$$Q_{t+1}(s, a) = (1 - \alpha)Q_t(s, a) + \alpha \left[\pi_t + \delta \max_{a \in A} Q_t(s', a)\right]$$

and the non-(s, a) entries remain the same. The parameter $\alpha \in (0, 1)$ is the learning rate. Intuitively, the Q matrix is updated α towards the current reward and discounted future value.

It remains to actually select a given s. This is the **exploration policy**, which is exogenously specified. Generally, Q-learning algorithms select the maximizing action with some probability, and randomly with the remaining probability. The simplest exploration policy is ε-greedy exploration: with $1 - \varepsilon$, the maximizing action (according to the current Q matrix) is chosen; with ε probability, another action is drawn uniform randomly. More sophisticated policies incorporate exploration decay, where the chance of exploration decays to 0 as $t \to \infty$. Under exploration decay policies, Q learning is guaranteed to converge to the optimal policy, shown by Watkins and Dayan (1992).

The tradeoff inherent in Q-learning is that exploration can be costly, yielding suboptimal rewards, but is necessary to learn the Q matrix to maximize future rewards.

Q-learning was originally designed for MDPs, which are single-agent and stationary environments; dynamic games are neither. In dynamic games, payoffs and state transitions may depend on both players' actions. Further, strategies may depend on the whole stream of past actions.[7] In full generality a state is thus never repeated; thus it is impossible to learn the infinite Q matrix. In this paper, the Q-learning algorithms treat the $t - 1$ period actions as the state variable, and the game structure will not depend on the past. Finally, Q-learning is not guaranteed to converge to optimal actions as in MDPs. However, this leaves open the possibility of interesting behavior (and interesting research questions).

[7] For example, consider subgame perfect Nash equilibria grim trigger strategies. The strategy punishes the opponent if he has *ever* defected, not just in the prior period.

3 Experiments

3.1 Model

The economic environment considered is a reverse auction with simultaneous bids and two players. Each stage is one such auction, which is repeated T times.

Let $i = 1, 2$ be the competing firms. In each period $t \in [1, T]$, the buyer (the "government") offers a contract via a reverse auction with simultaneous bids. Each firm submits a bid $b_{it} \in \{0, \frac{1}{k}, ...\frac{k-1}{k}, 1\}$, where k is a step-size parameter. The profits to the firms are given by

$$\pi_{it}(b_{it}, b_{jt}) = \begin{cases} b_{it} - c & \text{if } b_{it} < b_{jt} \\ \frac{1}{2}(b_{it} - c) & \text{if } b_{it} = b_{jt} \\ 0 & \text{if } b_{it} > b_{jt} \end{cases}$$

That is, the lower bidder wins the contract, gaining his bid b_{it} minus the cost of providing the service c. If the two firms tie, the contract is split. (Equivalently, can represent randomization of the winner with expected utility firms.) In this paper, I normalize $c = 0$, though stochastic costs may be an interesting future research direction.[8] Notice that $b_{1t} = b_{2t} = \frac{1}{k}$ is a Nash equilibrium profile. The firms collectively extract $\frac{1}{k}$ profit per period.

I set a ceiling on bids at $b_{it} = 1$. The reasoning for this is practical rather than theoretical – the algorithms we consider require a finite action space. However, we can motivate this feature of the model either as a "reservation bid", where the government posts that it will cancel the contract if the price is above 1, or by the presence of a third firm that will always bid on the contract at a price of 1.

3.2 Q-learning setup

To specify the Q-learning algorithm, we need the state variables and exploration policy.

To keep the model simple, I set the state to be the previous period's bids:

$$s_t = (b_{1,t-1}, b_{2,t-1})$$

This allows the Q-learning algorithms to condition the present period's action a_t on the prior period's bids. Then $|A| = k+1$ and $|S| = (k+1)^2$. Contrasted to a more complicated state

[8] See Ballestero (2025) for an example.

variable, this restricts the complexity of the algorithms' policies. The primary advantage is faster learning, as there are fewer states to explore.

The exploration policy is ε-greedy with decay defined by

$$\varepsilon_t = \beta^t$$

where $\beta > 0$ is a decay parameter. In $t = 0$, the algorithm chooses randomly. As t increases, exploration becomes less likely.

3.3 Experimental setup

I first summarize the parameters and note the baseline values for the simulations. To summarize, the parameters are:

- $T = 100,000$, the number of repeated stages
- $k = 6$, the the discrete price step parameter
- $\alpha = 0.1$, the learning rate
- $\beta = 0.99995$, the exploration decay
- $\delta = 0.95$, the discount factor

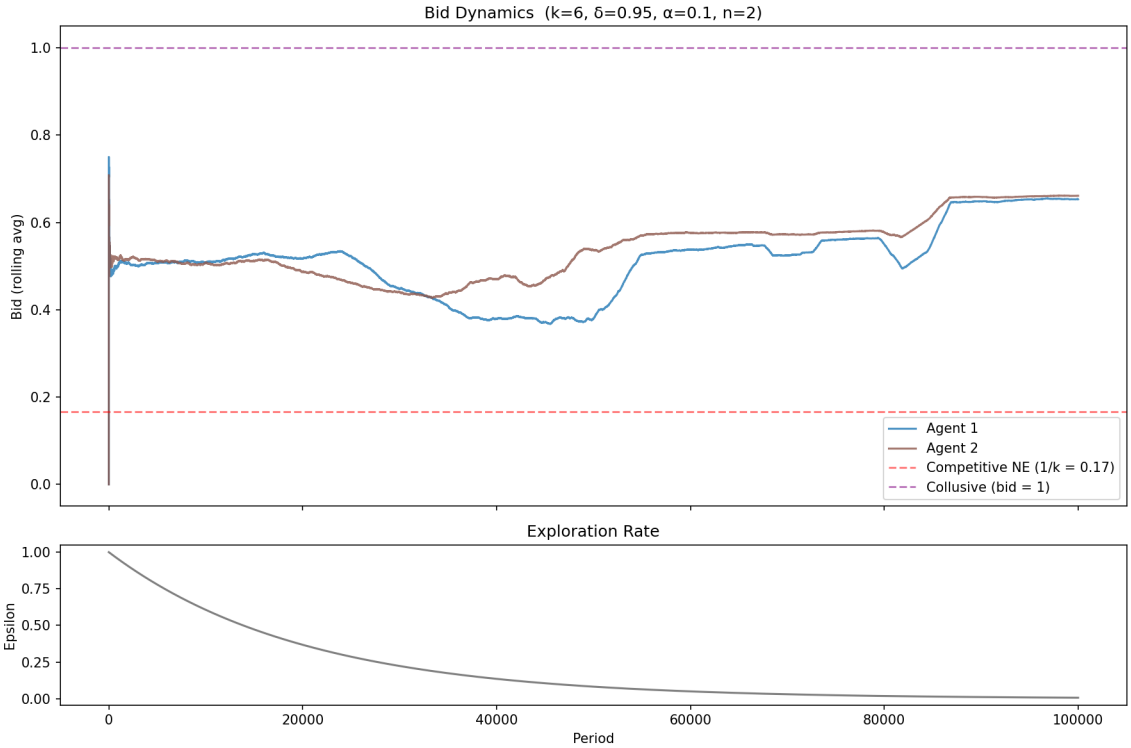
T is chosen to be computationally feasible. The step size k is chosen to match the setting of Maskin and Tirole (1988). The learning parameter α matches common benchmarks in computer science literature (c.f. Calvano, Calzolari, Denicolò, and Pastorello (2020)). The exploration decay β is chosen to to yield $\sim 1\%$ chance of exploration after at $t = 100,000$; that is,

$$\beta^{100,000} = 0.01$$

The tradeoff in selecting β is the necessity of visiting each Q matrix cell enough times to learn an optimal policy, versus losses resulting from suboptimal actions. Heuristically, with decay β and large T, there are in expectation $\approx \frac{1}{1-\beta}$ exploration actions. At our standard parameters, there are $(k+1)^2 = 49$ cells in the Q matrix and 20,000 exploration actions; in expectation each cell is visited 408 times via random exploration.

Each iteration (of $T = 100,000$ periods) constitutes one experimental observation. I repeat this setup 100 times and report results.

Figure 1: Bids over time



4 Results

I first show bids and profits from the first experimental observation.

Players' bids increase to around $b_{it} = 2/3$, significantly above the competitive bids of $b_{it} = 1/6$. Correspondingly, the firms collect $\pi_{it} = 1/3$ each, a combined profit of $2/3$ out of the maximum profit 1. Figure 1 shows bids over time; Figure 2 shows profits over time. The two agents largely mirror each other, starting around the middle of the bid grid. Interestingly, one agent appears to dip towards Nash equilibrium before increasing toward $2/3$. I interpret this behavior as tacit collusion, as both agents sustain bids roughly 4x competitive bids.

Figure 3 displays the final learned policies. These are also suggestive of collusive behavior. For example, notice Agent 1's row for $b_{1,t-1} = 2/3$. If agent 2 undercut this bid at $b_{2,t-1} = 1/2$, agent 1 retaliates by bidding $b_{1t} = 1/6$. Though this evidence is only suggestive, it resembles limited punishment strategies in repeated games.

Figure 2: Profits over time

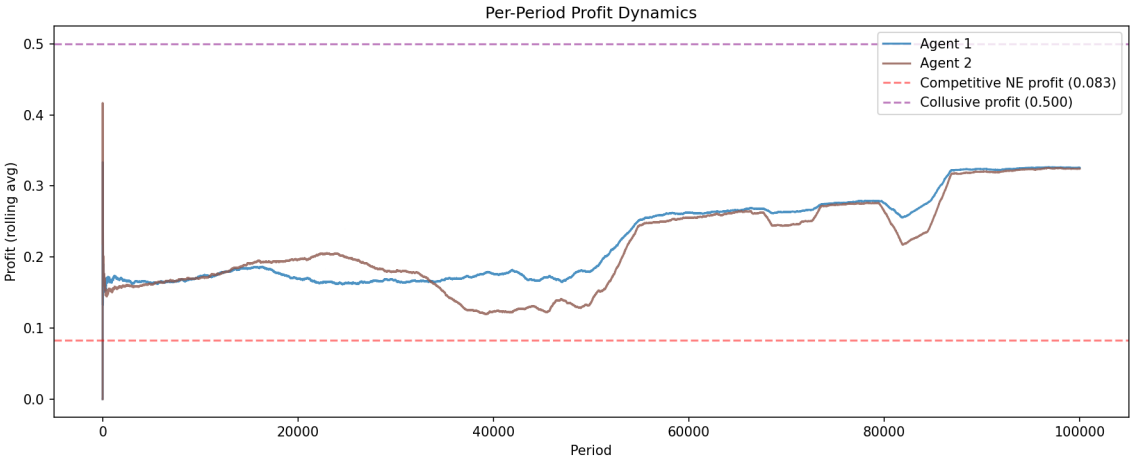


Figure 3: Learned policies

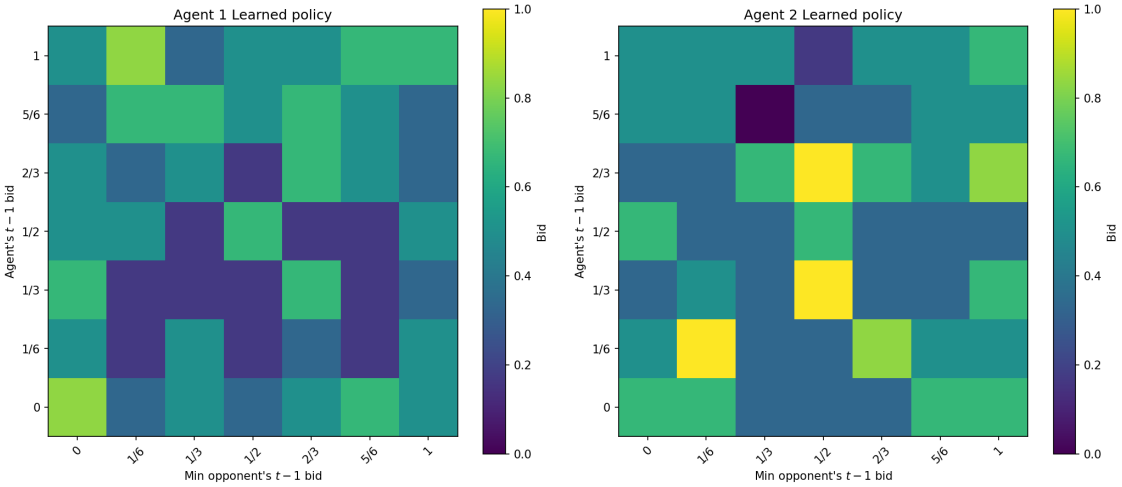
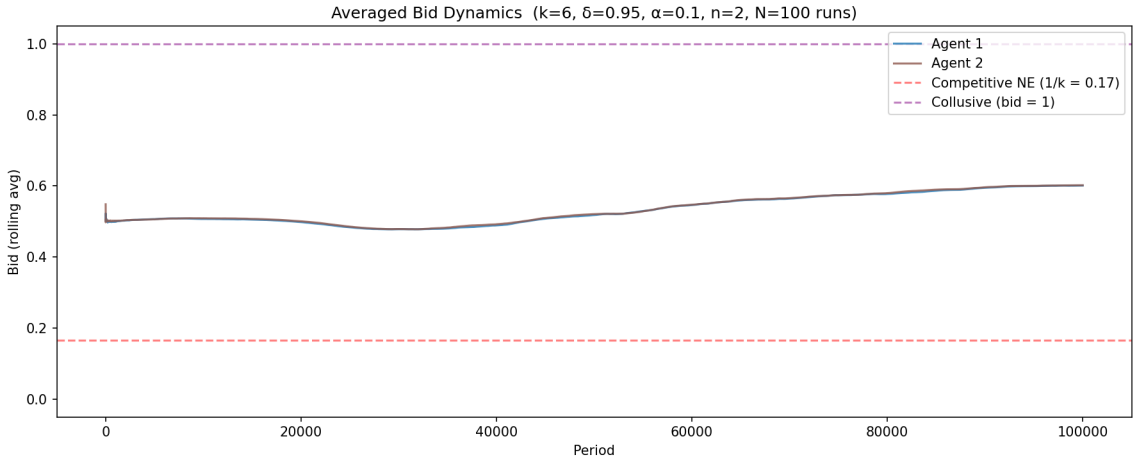


Figure 4: Bids over time, average of 100 simulations



4.1 Average results

To demonstrate that the pattern presented is not coincidental, I repeat the experiment 100 times. Figure 4 shows the average per-period bids across 100 repetitions. Unsurprisingly, the two agents' averages are almost identical. By $t = 100,000$, the Q-learning algorithms on average bid around $b_{it} = 0.6$, well above the competitive bids of $b_{it} = 1/6$. Figure 5 shows the 90% empirical confidence interval for a single agent's bids; that is, among the 100 repeated experiments, the 5th to 95th percentile per-period bids. These results demonstrate that supra-competitive behavior is remarkably consistent. Even at the 5th percentile, firm 1 is bidding $b_{1t} = 0.5$ by the final period.

4.2 Increasing the number of firms

I also introduce a third firm. The environment remains the same and completely symmetric. That is, the lowest bidder wins, and firms split profits uniformly among winners if there is a tie. As figure 6 shows, supra-competitive pricing is preserved. However, the magnitude is reduced. By the last period on average, firms bid $b_{it} = \frac{2}{6}$, still above the competitive equilibrium of $b_{it} = \frac{1}{6}$, though greatly reduced from 0.6.

This result highlights the importance of the number of firms. While pricing remains supra-competitive, it appears that increasing the number of firms decreases the profit extracted by the sellers.

Figure 5: Bids over time, average of 100 simulations, 90% empirical confidence interval

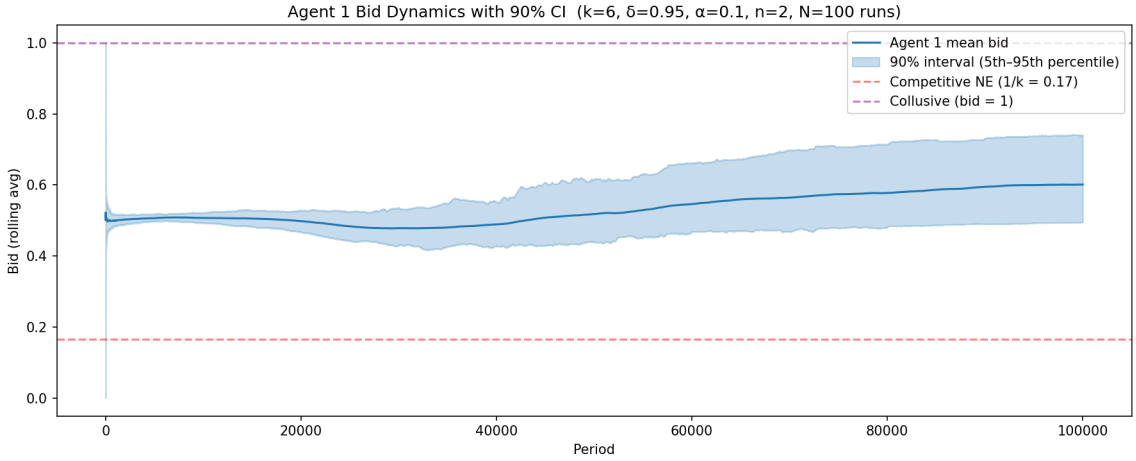
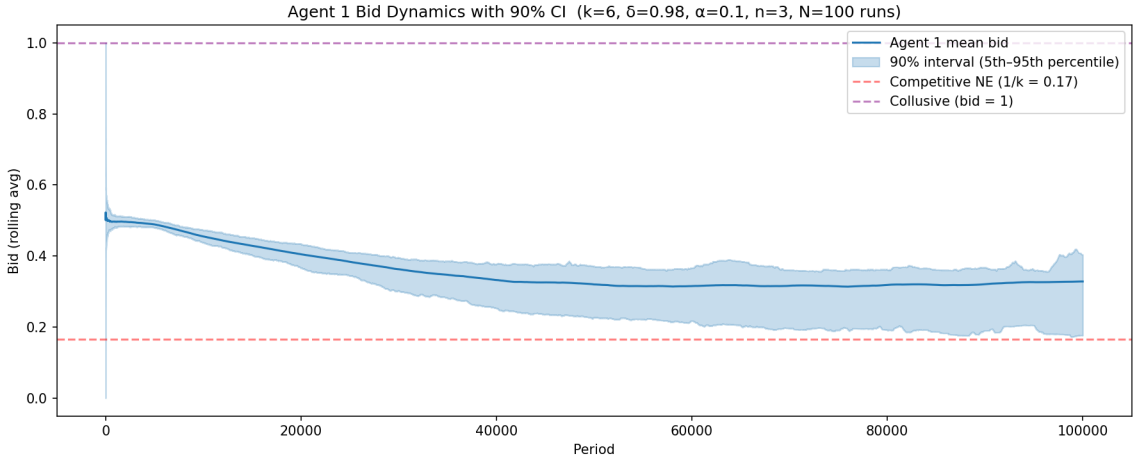


Figure 6: 3 firms, bids over time, average of 100 simulations, 90% empirical confidence interval



5 Conclusion

In this paper, I simulate a simplified sealed bid reverse auction environment, which governments often use to procure goods and services. I find that Q-learning algorithms consistently learn to collude, extracting multiple times the competitive equilibrium profit. On average, the firms extract three times the competitive profit. While these results are theoretical rather than empirical, they have major implications for the use of auctions and similar formats to procure goods and services. In this paper's simplified environment, the government pays on average three times the competitive price by the last period.

Since the algorithms do not explicitly communicate with each other, such behavior would not generally be considered antitrust violations. Indeed, Q-learning algorithms simply learn to maximize profit; they do not receive any explicit instructions to employ anticompetitive behavior. I stop short of recommending particular government policy responses, though this paper's results demonstrate the importance of developing policies to address AI collusion. Such policies can include both explicit changes to the law to restrict use of algorithms and changes to the auction environment to reduce their collusive behavior. In this paper, I have simulated the auction environment with particular parameters – it may be helpful future work to examine behavior across parameters, especially the number of firms; such results may help determine what features of the environment promote algorithmic collusion.

References

Aggarwal, G., Badanidiyuru, A., Balseiro, S. R., Bhawalkar, K., Deng, Y., Feng, Z., ... & Zuo, S. (2024). Auto-bidding and auctions in online advertising: A survey. ACM SIGecom Exchanges, 22(1), 159-183.

Alper, O. & Boning, W. B. (2003). Using Procurement Auctions in the Department of Defense. CNA Corporation. CRM D0007515.A3.

Asker, J. (2010). A study of the internal organization of a bidding cartel. American Economic Review, 100(3), 724-762.

Ballestero, G. (2025). Algorithmic collusion under sequential pricing and stochastic costs. Available at SSRN 5203084.

Calvano, E., Calzolari, G., Denicolo, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. American Economic Review, 110(10), 3267-3297.

Chen, L., Mislove, A., & Wilson, C. (2016, April). An empirical analysis of algorithmic pricing on amazon marketplace. In Proceedings of the 25th international conference on World Wide Web (pp. 1339-1349).

Coughlan, P., Gates, W., & Lamping, J. (2008). Innovations in defense acquisition auctions: Lessons learned and alternative mechanism designs. Acquisition Research Program.

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

Maskin, E., & Tirole, J. (1988). A theory of dynamic oligopoly, II: Price competition, kinked demand curves, and Edgeworth cycles. Econometrica: Journal of the Econometric Society, 571-599.

US Department of Justice. (2023.) Federal Antitrust Crime: A Primer for Law Enforcement Personnel. Accessed from https://www.justice.gov/atr/page/file/1091651/dl?inline.

U.S. Government Accountability Office. (2018). Report GAO-18-446.

U.S. DoD Inspector General. (2012). DODIG-2012-098.

Waltman, L., & Kaymak, U. (2008). Q-learning agents in a Cournot oligopoly model. Journal of Economic Dynamics and Control, 32(10), 3275-3293.

Watkins, C. J. C. H. 1989. Learning from Delayed Rewards. PhD dissertation. King's College.

Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8(3), 279-292.