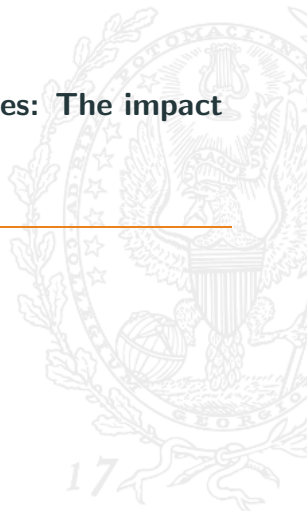


Reinforcement Learning in Trading Games: The impact of Information on Market Efficiency

Pranjal Rawat

September 26, 2023

Georgetown University



Introduction



Two important developments:

- 1 Emergence of fully computerized real-time auctions
 - Energy - Electricity, Natural Gas
 - Advertising - Sponsored Search, Display Advertising
 - Financial - NYSE, Chicago Ex, Forex, Cryptocurrencies
- 2 Critical breakthroughs in reinforcement learning algorithms:
 - Chess, Go, Starcraft, Self-driving Cars, Robotics



Will the adoption of reinforcement learning algorithms lead to algorithmic collusion, market inefficiency, and flash crashes?

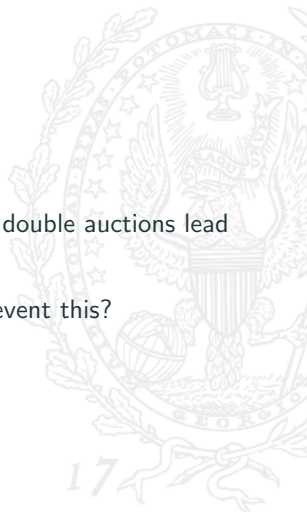
A few experiments demonstrate its possibility:

Market	Reference
One-sided Auction	Banchio-Skrzypacz (2021)
Electricity Auction	Tellidou-Bakirtzis (2006)
Bertrand Oligopoly	Calvano et al., (2020)
Cournot Oligopoly	Waltman-Kaymak (2008)
Platform	Johnson et al., (2020)

Research Questions

Can sophisticated reinforcement learning in dynamic double auctions lead to market inefficiency and algorithmic collusion?

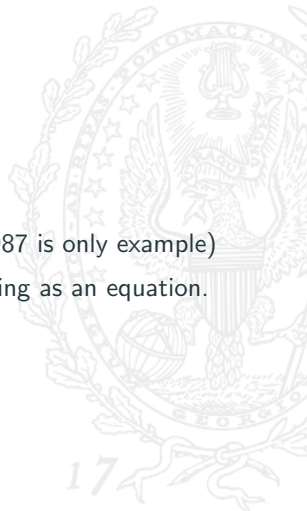
Can we develop learning-robust auction design to prevent this?



Difficult to proceed theoretically:

- Bayesian Nash equilibria not known. (Wilson 1987 is only example)
- Cannot write down sophisticated AI-based learning as an equation.

So I adopt an experimental approach.

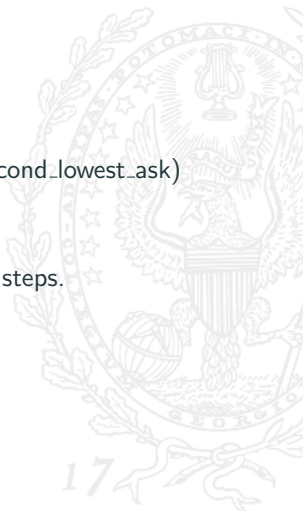


1. Pricing Mechanism \mathbb{P} :

$P = \mathbb{P}(\text{highest_bid}, \text{second_highest_bid}, \text{lowest_ask}, \text{second_lowest_ask})$

2. Public Information Set Ω :

- Environment - Number of buyers, sellers, items, steps.
- Transactions - History of winning bids and asks
- Bidding Log - History of failed bids and asks
- Identities - Identities of bidders and askers

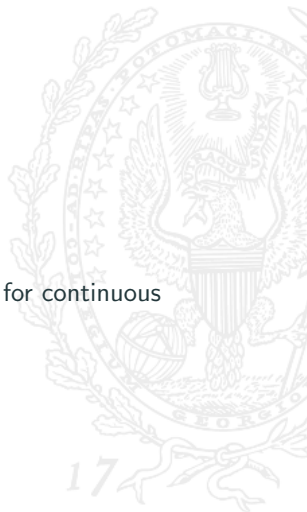


Research Contribution

Current literature does not consider:

- The role of high-dimensional information sets.
- Multi-period games
- State-of-the-art algorithms specifically designed for continuous action spaces.

This research hopes to cover all this.

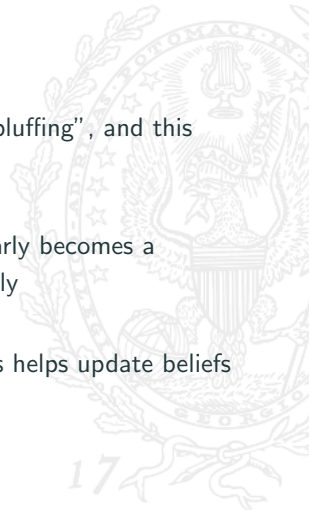


Literature



- **Game theoretic:** Chatterjee-Samuelson (1983), Myerson-Satterthwaite (1983), Sobel-Takahashi (1983), Sattettbwaite-Williams (1989), Wilson (1987), Bulow-Klemperer (1994, 1996), Pesendorfer-Swinkels (1997, 2000), Cripps-Swinkels (2006), Satterthwaite et al., (2022).
- **Experimental:** Chamberlain (1948), Smith et al., (1962, 1984), Williams et al., (1982, 1988, 1989), Gode-Sunder (1993), Rust et al. (1992,1993), Tesauro et al. (2001), Porter-Smith (2003), Chen et al., (2010), Attanasi et al. (2014), Loertscher et al., (2015).
- **Reinforcement Learning:** Watkins and Dayan (1992), Sutton and Barto (1998), Littman (1994), Mnih et al. (2015, 2016), Brockman et al., (2016), Schulman et al., (2017), Silver et al. (2018), Lillicrap et al., (2019).

- Uncertainty about others' valuations leads to “bluffing”, and this necessarily causes ex-post market inefficiency (Myerson-Satterthwaite 1983).
- As the number of traders increases, honesty nearly becomes a dominant strategy and inefficiency shrinks rapidly (Satterthwaite-Williams 1989).
- In dynamic auctions, activity in previous periods helps update beliefs and improve actions (Wilson 1987).



Key Insights - Experiments

- Efficiency in double auctions is surprisingly high (Smith et al., 1984).
- This is attained even with unintelligent traders (Gode-Sunder 1993).
- Very hard for humans to write programs that beat simple rule-of-thumb strategies (Rust et al., 1992).
- A simple background 'wait-and-watch' strategy is very effective against most human programs (Rust et al., 1992).
- Sophisticated learning algorithms will, however, with enough training outperform simple strategies (Chen et al., 2010).

Reinforcement Learning - I

- Dynamic Programming:

$$\forall s, V(s) \leftarrow \max_a [r + \gamma V(s')]$$

where $r = R(s, a)$, $s' = g(s, a)$.

- Approximate Dynamic Programming:

$$\hat{V}(s_t) \leftarrow [r_t + \gamma \hat{V}(s_{t+1})]$$

where $r_t, s_{t+1} \sim a_t, s_t$.

- Reinforcement Learning:

$$\hat{V}(s_t) \leftarrow (1 - \alpha) \hat{V}(s_t) + \alpha [r_t + \gamma \hat{V}(s_{t+1})]$$

where $r_t, s_{t+1} \sim a_t, s_t$.

Reinforcement Learning - II

Problem with DP	Solution with RL
<p>$V(s)$ table too large</p> <p>Need to know $r(s, a), g(s, a)$</p> <p>Forward search may oscillate</p> <p>Updates are heavy</p> <p>r, g may change</p>	<p>Approximate by $V(s; \theta)$</p> <p>Sampling r_t and s_{t+1}</p> <p>Soft updating: $\alpha < 1$</p> <p>Updates are incremental</p> <p>Learning is online</p>

Reinforcement Learning - III

A number of enhancements led to the breakthroughs:

- 1 Eliminating correlations in training data via **Experience Replay**: (s_i, a_i, r_i, s'_i) sampled randomly from memory.
- 2 Generalization via **Value Networks**: $\hat{V}(s; \theta)$ helps generalize to unseen states.
 - $L(\theta) = \sum_i \left[\hat{V}(s_i; \theta) - r_i - \gamma \hat{V}(s'_i; \theta) \right]^2$
 - $\theta_{t+1} = \theta_t - \alpha * \frac{dL}{d\theta}$
- 3 Stabilization via **Target Networks**: Separately trained and slow-changing \tilde{V} used to generate Bellman target.
- 4 Continuous Actions via **Policy Networks**: $\hat{a} = \hat{\pi}(s; \theta)$ to predict best action given $\hat{V}(s; \theta)$.

Single Agent Learning



Game \rightarrow *Rounds* \rightarrow *Period* \rightarrow *Step* \rightarrow Bid/ask and Buy/Sell

- Game: $n_{\text{buyers}}, n_{\text{sellers}}, n_{\text{rounds}}, n_{\text{periods}}, n_{\text{steps}}, n_{\text{tokens}}, K$
- Round: $\text{tokenvalues} \sim \text{Uniform}(0, K)$
- Period: Reset token allocation
- Step:
 - Bid/Ask: $(\text{bid1}, \text{bid2}, \dots), (\text{ask1}, \text{ask2}\dots), \text{cbid}, \text{cask}$
 - Buy/Sell: $\text{buy}, \text{sell}, \text{price}, \text{bprofit}, \text{sprofit}$

Example I

Bidder 1 is a DRL Agent, all others offer randomly.

State $s_t \in \mathbb{R}^{40}$ is a two-period activity log.

Buyer values:

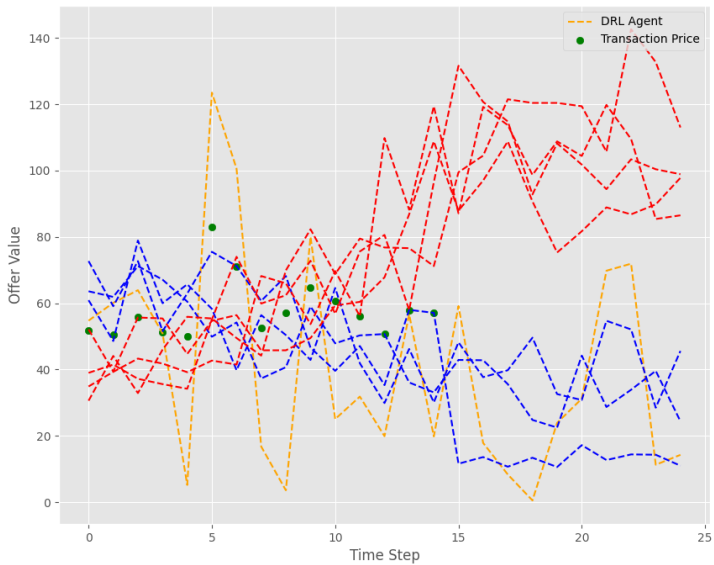
102.3	99.5	72.1	53.3	51.6	46.3
81.5	80.4	71.2	55.7	44.7	40.1
91.7	83.3	65.1	55.9	30.4	26.0
93.8	87.8	71.9	65.5	60.2	19.1

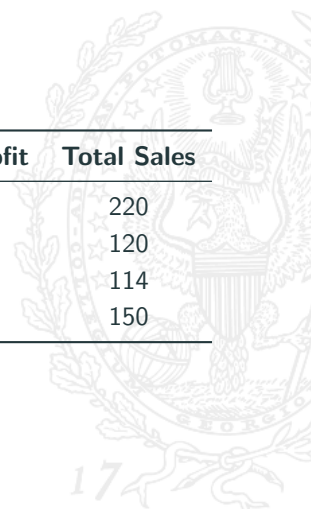
Seller costs:

32.5	33.0	53.9	57.0	90.6	96.0
26.6	30.4	33.1	50.7	81.0	82.1
23.9	32.8	41.5	56.8	95.9	99.1
38.3	40.4	45.4	72.7	86.2	92.6



Period 0

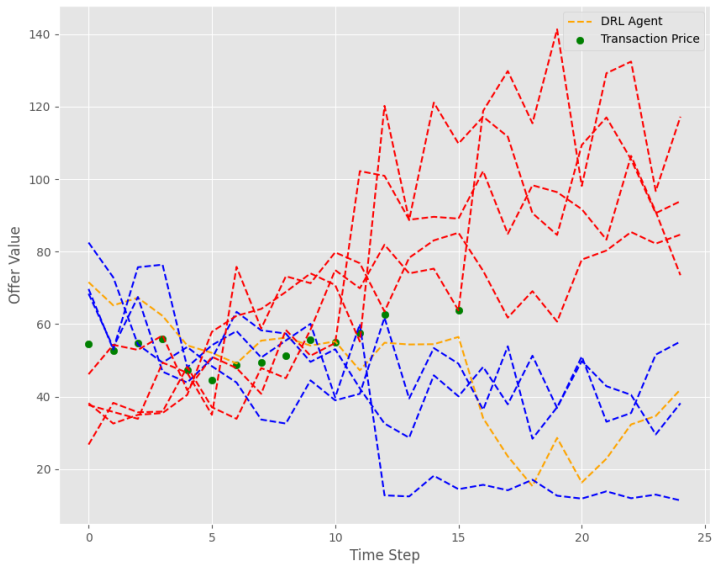




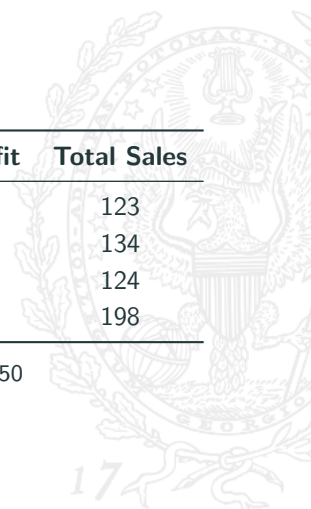
Bidder Index	Avg. Price	Total Buyer Profit	Total Sales
1	75.064227	-459.03	220
2	58.377083	2287.75	120
3	58.046053	2586.95	114
4	57.832000	3387.80	150

Table 1: Overview Periods 0-40

Period 3515



Period 3510-3550



Bidder Index	Price	Total Buyer Profit	Total Sales
0	54.686870	4389.415	123
1	56.454851	2554.350	134
2	57.703629	2663.150	124
3	57.277020	3706.750	198

Table 2: Overview Periods 3510-3550

Period 3515

step	bids	asks	current_bid	current_bid_idx	current_ask	current_ask_idx	buy	sell	price	bprofit	sprofit
0	[71.57, 68.4, 82.5, 69.7]	[37.7, 38.2, 26.8, 46.2]	82.50	2	26.8	2	True	True	54.65	37.05	30.75
1	[65.19, 53.6, 72.9, 53.0]	[35.8, 32.6, 38.3, 54.3]	72.90	2	32.6	1	True	True	52.75	30.55	26.15
2	[67.27, 67.5, 54.4, 75.7]	[33.9, 35.0, 35.7, 52.9]	75.70	3	33.9	0	True	True	54.80	39.00	22.30
3	[62.29, 46.9, 49.4, 76.4]	[49.4, 35.5, 35.9, 56.9]	76.40	3	35.5	1	True	True	55.95	31.85	25.55
4	[54.14, 43.9, 53.7, 48.1]	[46.6, 40.5, 48.2, 41.8]	54.14	0	40.5	1	True	True	47.32	54.98	14.22
5	[52.09, 50.5, 48.4, 54.2]	[35.0, 57.9, 37.1, 51.0]	54.20	3	35.0	0	True	True	44.60	27.30	11.60
6	[49.16, 63.4, 43.9, 58.1]	[75.8, 62.3, 33.9, 47.9]	63.40	1	33.9	2	True	True	48.65	32.85	15.85
7	[55.46, 58.3, 33.7, 50.8]	[58.8, 64.2, 47.9, 40.8]	58.30	1	40.8	3	True	True	49.55	30.85	11.25
8	[56.26, 57.4, 32.6, 55.5]	[73.2, 68.9, 45.1, 58.4]	57.40	1	45.1	2	True	True	51.25	19.95	9.75
9	[54.1, 49.6, 44.5, 60.1]	[71.3, 74.0, 58.5, 51.3]	60.10	3	51.3	3	True	True	55.70	9.80	15.30
10	[55.26, 53.1, 39.0, 39.8]	[79.8, 70.8, 74.9, 54.8]	55.26	0	54.8	3	True	True	55.03	44.47	9.63
11	[47.26, 42.3, 40.8, 60.0]	[76.8, 55.2, 69.9, 102.2]	60.00	3	55.2	1	True	True	57.60	2.60	6.90
12	[54.88, 32.6, 61.8, 12.8]	[63.5, 120.2, 81.9, 100.9]	61.80	2	63.5	0	True	True	62.65	2.45	8.75
13	[54.4, 28.7, 39.5, 12.5]	[78.3, 88.7, 74.0, 88.8]	54.40	0	74.0	2	False	False	NaN	0.00	0.00
14	[54.48, 45.9, 53.4, 18.2]	[83.1, 121.1, 75.3, 89.6]	54.48	0	75.3	2	False	False	NaN	0.00	0.00
15	[56.48, 40.1, 49.1, 14.5]	[85.2, 109.8, 63.8, 89.1]	56.48	0	63.8	2	True	False	63.80	8.30	7.00
16	[34.09, 48.2, 36.4, 15.7]	[74.7, 117.2, 118.8, 102.2]	48.20	1	74.7	0	False	False	NaN	0.00	0.00
17	[23.55, 37.9, 53.9, 14.2]	[61.8, 111.7, 129.8, 84.9]	53.90	2	61.8	0	False	False	NaN	0.00	0.00
18	[15.36, 51.3, 28.4, 17.1]	[69.1, 90.6, 115.4, 98.3]	51.30	1	69.1	0	False	False	NaN	0.00	0.00
19	[28.64, 36.9, 37.1, 12.7]	[60.7, 84.6, 141.3, 96.4]	37.10	2	60.7	0	False	False	NaN	0.00	0.00
20	[16.34, 51.1, 49.8, 11.9]	[77.8, 109.4, 98.1, 91.7]	51.10	1	77.8	0	False	False	NaN	0.00	0.00
21	[22.91, 33.1, 42.9, 13.9]	[80.3, 117.0, 129.2, 83.3]	42.90	2	80.3	0	False	False	NaN	0.00	0.00
22	[32.35, 35.5, 40.5, 12.0]	[85.4, 105.4, 132.4, 106.4]	40.50	2	85.4	0	False	False	NaN	0.00	0.00
23	[34.68, 51.6, 29.6, 13.0]	[82.2, 90.6, 96.7, 91.0]	51.60	1	82.2	0	False	False	NaN	0.00	0.00
24	[41.99, 55.2, 38.2, 11.4]	[84.7, 93.9, 117.2, 73.5]	55.20	1	73.5	3	False	False	NaN	0.00	0.00

Deep reinforcement learning can perform well against simple opponents in complex dynamic strategic games.



Multi-Agent Learning



Example II: Duopsony

Both Bidders are DRL, sellers are honest.

State $s_t \in \mathbb{R}^{13}$ is a one-period activity log.

Theoretical eqbm price: 38.9

Buyer values:

93.7	74.2	72.1	64.1	56.8	53.8	50.6	29.6
97.4	79.8	75.6	60.8	52.8	47.6	43.6	31.1

Seller values:

23.0	27.6	32.9	33.1	76.7	85.1	87.6	97.4
11.1	28.5	30.4	41.0	64.7	65.6	78.2	82.0
28.6	37.7	53.2	62.3	62.6	77.3	82.3	101.6
29.7	29.8	36.3	39.2	44.0	81.9	93.7	100.1
16.3	28.2	44.4	60.4	67.9	78.5	84.9	86.2
42.4	48.2	52.4	58.0	67.3	80.6	82.7	90.8

Period 59

Collusion!

	rnd	period	step	bids	asks	current_bid	current_bid_idx	current_ask	current_ask_idx	buy	sell	price	sale	bprofit	sprofit
2950	0	59	0	[0.94, 0.97]	[23.0, 11.1, 28.6, 29.7, 16.3, 42.4]	0.97	1	11.1	1	True	False	11.1	1	86.3	0.0
2951	0	59	1	[0.94, 0.8]	[23.0, 28.5, 28.6, 29.7, 16.3, 42.4]	0.94	0	16.3	4	True	False	16.3	1	77.4	0.0
2952	0	59	2	[0.74, 0.8]	[23.0, 28.5, 28.6, 29.7, 28.2, 42.4]	0.80	1	23.0	0	True	False	23.0	1	56.8	0.0
2953	0	59	3	[0.74, 0.76]	[27.6, 28.5, 28.6, 29.7, 28.2, 42.4]	0.76	1	27.6	0	True	False	27.6	1	48.0	0.0
2954	0	59	4	[0.74, 0.61]	[32.9, 28.5, 28.6, 29.7, 28.2, 42.4]	0.74	0	28.2	4	True	False	28.2	1	46.0	0.0
...
3116	0	62	16	[0.3, 0.31]	[76.7, 41.0, 53.2, 39.2, 44.4, 42.4]	0.31	1	39.2	3	False	False	NaN	0	0.0	0.0
3117	0	62	17	[0.3, 0.31]	[76.7, 41.0, 53.2, 39.2, 44.4, 42.4]	0.31	1	39.2	3	False	False	NaN	0	0.0	0.0
3118	0	62	18	[0.3, 0.31]	[76.7, 41.0, 53.2, 39.2, 44.4, 42.4]	0.31	1	39.2	3	False	False	NaN	0	0.0	0.0
3119	0	62	19	[0.3, 0.31]	[76.7, 41.0, 53.2, 39.2, 44.4, 42.4]	0.31	1	39.2	3	False	False	NaN	0	0.0	0.0
3120	0	62	20	[0.3, 0.31]	[76.7, 41.0, 53.2, 39.2, 44.4, 42.4]	0.31	1	39.2	3	False	False	NaN	0	0.0	0.0

Example III: Limited Information

Both Bidders and sellers are DRL.

State $s_t \in \mathbb{R}^1$ contains information only about timestep.

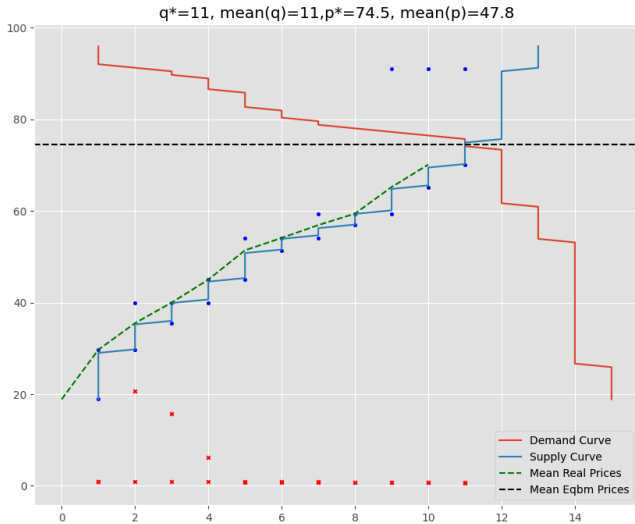
Theoretical eqbm price: 74.5

Buyers: $\begin{bmatrix} 91.5 & 91.2 & 86.5 & 82.2 & 78.6 & 76.6 & 53.5 & 16.3 \\ 95.9 & 89.2 & 79.7 & 78.0 & 75.8 & 74.0 & 61.0 & 26.3 \end{bmatrix}$

Sellers: $\begin{bmatrix} 18.9 & 40.0 & 54.1 & 56.9 & 91.0 & 100.1 & 102.6 & 103.9 \\ 29.7 & 35.5 & 45.0 & 51.4 & 59.4 & 65.2 & 70.1 & 75.3 \end{bmatrix}$

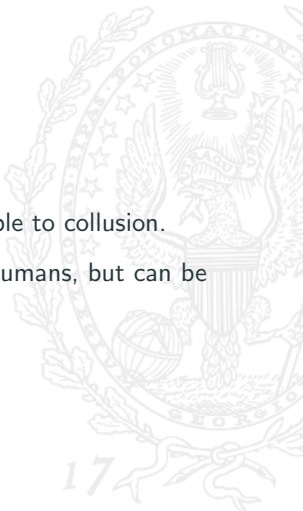
Period 49

Collusion!



Algorithms can not only collude, but also be vulnerable to collusion.

Double auctions are efficient with random play and humans, but can be inefficient with “super-human” algorithms.



Example IV: More Information

Both Bidders and sellers are DRL.

State $s_t \in \mathbb{R}^{13}$ is a one period activity log

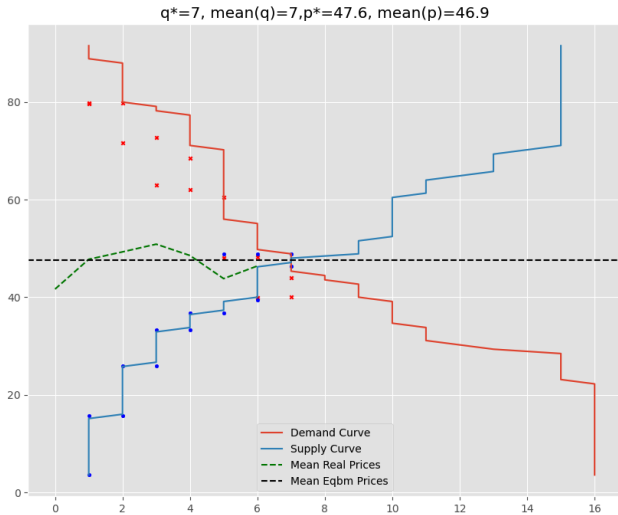
Theoretical eqbm price: 47.6

Buyers: $\begin{bmatrix} 91.5 & 91.2 & 86.5 & 82.2 & 78.6 & 76.6 & 53.5 & 16.3 \\ 95.9 & 89.2 & 79.7 & 78.0 & 75.8 & 74.0 & 61.0 & 26.3 \end{bmatrix}$

Sellers: $\begin{bmatrix} 18.9 & 40.0 & 54.1 & 56.9 & 91.0 & 100.1 & 102.6 & 103.9 \\ 29.7 & 35.5 & 45.0 & 51.4 & 59.4 & 65.2 & 70.1 & 75.3 \end{bmatrix}$

Period 49

Truthtelling!

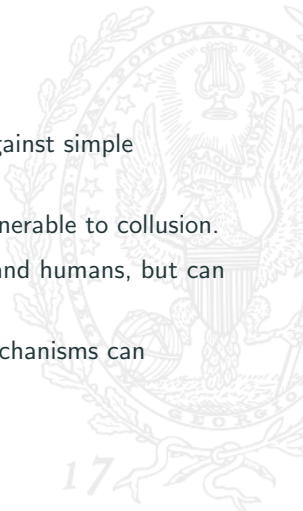


Truth-telling!

rnd	period	step	bids	asks	current_bid	current_bid_idx	current_ask	current_ask_idx	buy	sell	price	sale	bprofit	sprofit
0	46	0	[91.5, 88.6]	[15.8, 3.6]	91.50	0	3.6	1	True	True	47.55	1	43.95	43.95
0	46	1	[79.9, 88.6]	[15.8, 25.9]	88.60	1	15.8	0	True	True	52.20	1	36.40	36.40
0	46	2	[79.9, 70.5]	[33.3, 25.9]	79.90	0	25.9	1	True	True	52.90	1	27.00	27.00
0	46	3	[77.3, 70.5]	[33.3, 36.7]	77.30	0	33.3	0	True	True	55.30	1	22.00	22.00
0	46	4	[55.5, 70.5]	[48.8, 36.7]	70.50	1	36.7	1	True	True	53.60	1	16.90	16.90
0	46	5	[55.5, 44.7]	[48.8, 39.4]	55.50	0	39.4	1	True	True	47.45	1	8.05	8.05
0	46	6	[49.5, 44.7]	[48.8, 46.4]	49.50	0	46.4	1	True	True	47.95	1	1.55	1.55
0	46	7	[39.5, 44.7]	[48.8, 61.1]	44.70	1	48.8	0	False	False	NaN	0	0.00	0.00
0	46	8	[39.5, 44.7]	[48.8, 61.1]	44.70	1	48.8	0	False	False	NaN	0	0.00	0.00
0	46	9	[39.5, 44.7]	[48.8, 61.1]	44.70	1	48.8	0	False	False	NaN	0	0.00	0.00
0	46	10	[39.5, 44.7]	[48.8, 61.1]	44.70	1	48.8	0	False	False	NaN	0	0.00	0.00

Conclusions

- 1 Deep reinforcement learning can perform well against simple opponents in complex dynamic strategic games.
- 2 Algorithms can not only collude, but also be vulnerable to collusion.
- 3 Double auctions are efficient with random play and humans, but can be inefficient with “super-human” algorithms.
- 4 Adapting information disclosures and pricing mechanisms can improve market efficiency.



Next Steps..

- Fully randomized experiment to estimate the precise impact of information disclosures and pricing mechanism on market efficiency.
- Find which environmental and algorithmic factors cause market inefficiency.
- Explore how effect of auction design interacts with other factors - environmental and algorithmic.
- Zoom into the learning process and answer when and how does inefficiency arise.

