Market Pricing with Reinforcement Learning

Suzie Grondin\textsuperscript{1}, Philipp Ratz\textsuperscript{2} & Arthur Charpentier\textsuperscript{2}

\textsuperscript{1} ENSAE, \textsuperscript{2} UQAM

Journées de l'Optimisation
2024 Optimization Days

HEC Montréal, May 6-8, 2024
Motivation

- $k$ agents compete in a market environment where demand functions are unknown to the agents,
- agents compete for market share and try to optimise their payoff by proposing prices $p_i$ for a product (that are higher than their production cost),
- (economic theory) well-known that competition within the market should drive prices down,
- Calvano et al. (2020) suggested a reinforcement learning framework to investigate collusion in competitive markets.

Artificial Intelligence, Algorithmic Pricing, and Collusion

By Emilio Calvano, Giacomo Calzolari, Vincenzo Demicoli, and Sergio Pastorelli

Increasingly, algorithms are supplanting human decision-makers in pricing goods and services. To analyze the possible consequences, we study experimentally the behavior of algorithms powered by Artificial Intelligence (Q-learning) in a workhorse cagilopogo model of repeated price competition. We find that the algorithms consistently learn to charge supercompetitive prices, without communicating with one another. The high prices are sustained by collusive strategies with a finite phase of punishment followed by a gradual return to cooperation. This finding is robust to asymmetries in cost or demand, changes in the number of players, and various forms of uncertainty. (JEC D21, D45, D83, L12, L13)

Software programs are increasingly being adopted by firms to price their goods and services, and this tendency is likely to continue. In this paper, we ask whether pricing algorithms may “autonomously” learn to collude. The possibility arises because of the recent evolution of the software, from rule-based to reinforcement learning programs. The new programs, powered by Artificial Intelligence (AI), are indeed much more autonomous than their precursors. They can develop their pricing strategies from scratch, engaging in active experimentation and adapting to changing environments. In this learning process, they require little or no external guidance.

In the light of these developments, concerns have been voiced by scholars and policymakers alike, that AI pricing algorithms may raise their prices above the competitive level in a coordinated fashion, even if they have not been specifically

---

*Calvano: Bologna University, System School of Economics, and CIRIEC (email: emilio.calvano@unibo.it); Calzolari: European University Institute, Bologna University, System School of Economics, and CIRIEC (email: giacomo.calzolari@eui.eu); Demicoli: Bologna University and CIRIEC (email: vincenzo.demicoli@unibo.it); Pastorelli: Bologna University (email: sergio.pastorelli@unibo.it). Jeffrey Ely was the coeditor for this article. We are grateful to the several and three anonymous reviewers for many detailed and helpful comments. We also thank, in alphabetical order, Brittany Kahl, Patrick Lampe, David Levine, Willy Mullins, Yossi Spiegel, Steve Teckouk, Emanuel Tamer, and participants at numerous conferences and seminars. Financial support from the Digital Change initiative at the Tsukuba School of Economics is gratefully acknowledged.

**See** cs.lmu.de/~lutz/teaching/teaching.html for a list of the authors’ academic activities.

---

*While revenue-management programs have been used for decades in such industries as hotels and airlines, the diffusion of pricing software has been fueled by the advent of online marketplaces. For example, in a sample of over 1,600 hotel-staying firms listed in Gartner’s Report on Hotel Management Systems in 2005 (Table 1), the result is that in 2015 more than half of the vendors had already automated their pricing. But pricing software is increasingly used also in traditional offline sectors such as gas stations; see, e.g., Mark Schubert, “Why Do Gas Station Prices Constantly Change? Blame the Algorithms,” Wall Street Journal, May 8, 2017.*
Normal-form Games

Notations

- a finite set of agents, $i \in \{1, \cdots, k\}$
- each agent $i$ can take actions $a_i \in \mathcal{A}_i$
- joint actions are $a = (a_1, \cdots, a_k) \in \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_k$
- rewards functions are $r_i : \mathcal{A} \to \mathbb{R}$

Each agent $i$ selects policy $\pi_i : \mathcal{A}_i \to [0, 1]$ (takes action $a$ with probability $\pi_i(a)$), and receives reward $r_i^{t+1}(a)$.

Given $\pi = (\pi_1, \cdots, \pi_k)$, expected reward (to $i$) is

$$U_i(\pi) = \sum_{a \in \mathcal{A}} P[a] \cdot r_i(a) \prod_{j=1}^{k} \pi_j(a_j)$$

(expected utility)

Classically, agents select policies that maximize their expected reward.
Normal-form Games

- **Nash equilibrium** corresponds to a situation when no unilaterally change in action help to improve reward for any agent.
- **Pareto Optimum** corresponds to a situation when we can’t improve one agent without making other agent worse off.
Normal-form Games

- Payoff matrix, here classical **Prisoner’s Dilemma**, Poundstone (1992)

<table>
<thead>
<tr>
<th></th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cooperate</strong></td>
<td>(-1, -1)</td>
<td>(-3, 0)</td>
</tr>
<tr>
<td><strong>Defect</strong></td>
<td>(0, -3)</td>
<td>(-2, -2)</td>
</tr>
</tbody>
</table>

Best action of $i$ conditional on $a_j$

Best action of $j$ conditional on $a_i$

Payoff matrix

<table>
<thead>
<tr>
<th></th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cooperate</strong></td>
<td>(-1, -1)</td>
<td>(-3, 0)</td>
</tr>
<tr>
<td><strong>Defect</strong></td>
<td>(0, -3)</td>
<td>(-2, -2)</td>
</tr>
</tbody>
</table>

Pareto optimum vs. Nash equilibrium
Repeated Games and Collusion

- at each (discrete) time $t$, agents chose actions $a_i^t \in \mathcal{A}$
- policy $\pi_i : \bar{\mathcal{H}}^t \times \mathcal{A}_i \to [0, 1]$ assigns action probabilities based on experience $\mathcal{H}^T$

\[ \mathcal{H}^t = \bigcup_{\tau=0}^{t} \mathcal{H}^\tau, \text{ where } \mathcal{H}^T = \{ \mathcal{H}^T = (a^0, \cdots, a^{T-1}) \} \]

- expected utility for agent $i$ is

\[ U_i^t(\pi) = \sum_{H^t \in \mathcal{H}^t} P[H^t | \pi] \sum_{\tau=0}^{t-1} r_i(a^\tau) \prod_{\tau=0}^{t-1} \prod_{j=1}^{k} \pi_j(H^\tau, a_j^\tau) \]

- when games are repeated (potentially) $\infty$ times,

\[ U_i(\pi) = \lim_{t \to \infty} \sum_{H^t \in \mathcal{H}^t} P[H^t | \pi] \sum_{\tau=0}^{t-1} \gamma^\tau r_i(a^\tau) \]

discount factor ($\gamma \in [0, 1)$)
Repeated Games and Collusion

\[ U_i(\pi) = \lim_{t \to \infty} \sum_{H^t \in \mathcal{H}^t} P[H^t | \pi] \sum_{\tau = 0}^{t-1} \gamma^\tau r_i(a^\tau) \]

- interpretation as random stopping-time (ends each time with probability \(1 - \gamma\)),
- interpretation as myopic (\(\gamma \to 0\)) or farsighted (\(\gamma \to 1\))
- interpretation as weighted average (exponentially weighted moving average)
Repeated Games and Collusion

- assuming the opponent plays a "trigger" strategy, also referred to as "tit-for-tat" (i.e., begins by cooperating and once the opponent defects, defects in perpetuity from the next period)
  - if I deflect, no prison at first period, then prison (-2) at each period
    \[ V_i^D = 0 + \sum_{i=t}^{\infty} \gamma^t \cdot (-2) = \frac{-2\gamma}{1 - \gamma} \]
  - if I cooperate, prison (-1) at each period
    \[ V_i^C = -1 + \sum_{t=1}^{\infty} \gamma^t \cdot (-1) = \frac{-\gamma}{1 - \gamma} \]
- I choose to cooperate if \( V_i^C > V_i^D \) \( \iff \gamma > \frac{1}{2} \)
Stochastic Games

Now, agents interact in a shared stochastic environment,

- a finite set of agents, \( i \in \{1, \ldots, k\} \)
- \( S \) (finite) set of states
- each agent \( i \) can take actions \( a_i \in \mathcal{A}_i \)
- joint actions are \( a = (a_1, \ldots, a_k) \in \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_k \)
- \( \mathbb{P} \) is some state transition model, \( \mathbb{P}[s_{t+1} | s_t, a], S \times (S \times \mathcal{A}) \)
- rewards functions are \( r_i : S \times \mathcal{A} \to \mathbb{R} \)

**MDP (Markov Decision Process)**

The game starts with some \( s^0 \in S \), and at time \( t \), agent \( i \)

- observes state \( s^t \) and past joint action \( a^{t-1} \)
- choose action \( a^t_i \) with probability \( \pi_i(s^t, a^t), \pi_i(H^t, a^t) \)
- receives reward \( r_i(s^t, a^t) \)

then, game transitions into next state \( s^{t+1} \) with probabilities \( \mathbb{P}[\cdot | s_t, a^t] \), and repeat
Stochastic Games

Given policies \( \pi = (\pi_1, \cdots, \pi_k) \), expected return to agent \( i \) in state \( s \) is

\[
U_i(s, \pi) = \sum_a \left( \prod_{j=1}^{k} \pi_j(s, a_j) \right) \cdot \left[ r_i(s, a) + \gamma \sum_{s' \in S} P[s'|s, a] \cdot U_i(s', \pi) \right]
\]

Analogous to Bellman equation.

Use discount rate \( \gamma \in [0, 1) \) to have finite return

With heterogeneous agent, several solution concepts have been introduced

- Minimax solution
- Nash-correlated equilibrium
- Pareto-optimality
- Social welfare and fairness
- No-regret
Stochastic Games

- **Minimax solution**, Von Neumann and Morgenstern (1944)

  In a two-player zero-sum game, profile $\pi_i, \pi_j$ is maximin/minimax if

  $U_i(\pi_i, \pi_j) = \max_{p_i} \min_{p_j} U_i(p_i, p_j) = \min_{p_j} \max_{p_i} U_i(p_i, p_j) = -U_j(\pi_i, \pi_j)$

  Reward that can be guaranteed against worst-case opponent

- **Nash (correlated) equilibrium**, Fink (1964), Aumann (1974)

  Policy $\pi = (\pi_1, \cdots, \pi_k)$ is a Nash equilibrium if

  $\forall i \in \{1, \cdots, k\}, \forall p_i, U_i(p_i, \pi_{-i}) \leq U_i(\pi_i, \pi_{-i})$

  No agent can improve reward by unilaterally deviating from profile (every agent plays best-response to other agents)

  In two-player zero-sum game, minimax is same as a Nash equilibrium
Stochastic Games

- **Pareto-optimality**

Policy \( \pi = (\pi_1, \ldots, \pi_k) \) is a Pareto-optimal if there is no other profile \( p \) such that

\[
\forall i \in \{1, \ldots, k\}, \ U_i(p) \geq U_i(\pi) \text{ and } \exists j \in \{1, \ldots, k\}, \ U_j(p) > U_j(\pi)
\]

(we can’t improve one agent without making other agent worse off)

- **Social Welfare and Fairness**

\[
\text{Welfare}(\pi) = \sum_{i=1}^{k} U_i(\pi) \text{ and } \text{Fairness}(\pi) = \prod_{i=1}^{k} U_i(\pi)
\]

Policy \( \pi = (\pi_1, \ldots, \pi_k) \) is welfare/fairness-optimal if it maximises

\[
p \mapsto \text{Welfare}(p) / \text{Fairness}(p)
\]

Any welfare/fairness-optimal \( \pi \) is also Pareto-optimal
**Q learning**

- define \( Q^\pi(s, a) = \mathbb{E}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| S_t = s, A_t = a \right] \)

- if \( S \) and \( A \) are finite, store \( Q \) values in a matrix / table, (a row for each state and a column for each action)

<table>
<thead>
<tr>
<th>state</th>
<th>action</th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C,C)</td>
<td></td>
<td>( Q((C,C),C) )</td>
<td>( Q((C,C),D) )</td>
</tr>
<tr>
<td>(C,D)</td>
<td></td>
<td>( Q((C,D),C) )</td>
<td>( Q((C,D),D) )</td>
</tr>
<tr>
<td>(D,C)</td>
<td></td>
<td>( Q((D,C),C) )</td>
<td>( Q((D,C),D) )</td>
</tr>
<tr>
<td>(D,D)</td>
<td></td>
<td>( Q((D,D),C) )</td>
<td>( Q((D,D),D) )</td>
</tr>
</tbody>
</table>

Conditional on the state, pick an appropriate action
### Q learning

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C,C)</td>
<td></td>
<td>$Q((C,C),C)$</td>
<td>$Q((C,C),D)$</td>
</tr>
<tr>
<td>(C,D)</td>
<td></td>
<td>$Q((C,D),C)$</td>
<td>$Q((C,D),D)$</td>
</tr>
<tr>
<td>(D,C)</td>
<td></td>
<td>$Q((D,C),C)$</td>
<td><strong>$Q((D,C),D)$</strong></td>
</tr>
<tr>
<td>(D,D)</td>
<td></td>
<td>$Q((D,D),C)$</td>
<td>$Q((D,D),D)$</td>
</tr>
</tbody>
</table>

Once an action is picked, update:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[ R_{t+1} + \gamma \max_a \{ Q(S_{t+1}, a) \} - Q(S_t, A_t) \right]$$

- **learning rate**: $\alpha$
- **reward**: $R_{t+1}$
- **discounted optimal value at next step**: $\gamma \max_a \{ Q(S_{t+1}, a) \}$

Pick random action $a$, $\varepsilon_t$

$$\max_a \{ Q(S_{t+1}, a) \}, 1 - \varepsilon_t$$

Where $\varepsilon_t = \exp(-\beta t)$
Q learning

- Agents learn a policy $\pi(\cdot|s)$ selection an action, based on current state $s$
- $\varepsilon$-greedy Q-learning policy (Watkins (1989))

\[
Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[ R_{t+1} + \gamma \max_a \{Q(S_{t+1}, a)\} - Q(S_t, A_t) \right]
\]

- $\alpha \in (0, 1)$ learning rate (or step size)
  - $\alpha \to 1$ agent completely ignores prior knowledge, only values the most recent information
- $\gamma \in (0, 1)$ discount factor
  - $\gamma \to 0$, agent completely ignores the future rewards
Q learning

- $\varepsilon \in (0, 1)$ (actually $\varepsilon_t = \exp[-\beta t]$, $\beta > 0$),
  - introduces randomness into the algorithm, forcing us to try different actions
  - $\varepsilon \to 0$ never explore but always exploit the knowledge we already have
  - $\varepsilon \to 1$ force the algorithm to always take random actions and never use past knowledge
  - $\varepsilon_t \to 0$ as $t \to \infty$ lets an agent explore more at earlier stages (when experience is scarce)
- the aim is to have a balance between exploration and exploitation
  - if we make infinite trials, each action is taken an infinite number of times.
    - (the epsilon-greedy action selection policy discovers the optimal actions for sure)
Q learning

- **Prisoner’s Dilemma** game, Poundstone (1992), with **2 agents**
  both have only **2 choices** C (Collusion price) and D (Nash),
- suppose $\alpha = 5\%$, $\delta = 99\%$
- In this simple problem exploration is key, given an adequate $\beta$ results are collusive for fewer than 100,000 repetitions of the game
For a small enough $\beta$, results are always collusive in the end, $t \to \infty$. 
For a small $\beta$, results are always collusive in the end, $n \to \infty$

**Learning Curves for Different $\beta_1$ Values and $\beta_2 = 1$**

- Without communication, if the two prisoners use Q-learning algorithms to make their choices, they will be able to achieve the Pareto optimum without communicating on their strategy.
Market Pricing Iterative Game

- instead of gangsters, consider companies, willing to sell a product (Calvano et al. (2020)’s experiment)
- action \( a_i \) is choice of price \( p_i \)
- reward is associated to the demand for the product (Boltzmann experimentation model),

\[
D_i(p_i, p_j) = \frac{\exp(4(2 - p_i))}{\exp(4(2 - p_i)) + \exp(4(2 - p_j)) + 1}
\]

(model of price competition with logit demand and constant marginal costs)
- and \( R_i(p_i, p_j) = (p_i - c_i) \cdot D_i(p_i, p_j) \) where \( c_i = 1 \)
- **15 choices** of prices between, \( p^N - \xi(p^C - p^N) \) and \( p^M + \xi(p^C - p^N) \)

| 1.425 | 1.464 | 1.502 | 1.541 | 1.579 | 1.618 | 1.656 | 1.695 | 1.734 | 1.772 | 1.811 | 1.849 | 1.888 | 1.926 | 1.965 |
Market Pricing Iterative Game

<table>
<thead>
<tr>
<th>$p_i$</th>
<th>$$1.926$</th>
<th>$$1.464$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$1.926$</td>
<td>(0.34,0.34)</td>
<td>(0.11,0.36)</td>
</tr>
<tr>
<td>$$1.464$</td>
<td>(0.36,0.11)</td>
<td>(0.23,0.23)</td>
</tr>
</tbody>
</table>

Pareto optimum ($p^C = 1.926$) vs. Nash equilibrium ($p^N = 1.464$)

- agents have **15 choices** of prices between, $p^N - \xi(p^M - p^N)$ and $p^M + \xi(p^M - p^N)$ (equitably distributed).
- each agent have 15 different choices, which implies : $15^2 = 225$ states, hence, Q-table’s size is $225 \times 15$ i.e. **3375 cells**
## Market Pricing Iterative Game

<table>
<thead>
<tr>
<th></th>
<th>1.425</th>
<th>1.464</th>
<th>1.502</th>
<th>1.541</th>
<th>1.579</th>
<th>1.618</th>
<th>1.656</th>
<th>1.695</th>
<th>1.734</th>
<th>1.772</th>
<th>1.811</th>
<th>1.849</th>
<th>1.888</th>
<th>1.926</th>
<th>1.965</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.425</td>
<td>0.202</td>
<td>0.218</td>
<td>0.232</td>
<td>0.250</td>
<td>0.268</td>
<td>0.285</td>
<td>0.299</td>
<td>0.314</td>
<td>0.328</td>
<td>0.340</td>
<td>0.351</td>
<td>0.362</td>
<td>0.371</td>
<td>0.380</td>
<td>0.388</td>
</tr>
<tr>
<td>1.464</td>
<td>0.217</td>
<td>0.219</td>
<td>0.235</td>
<td>0.250</td>
<td>0.265</td>
<td>0.280</td>
<td>0.293</td>
<td>0.306</td>
<td>0.318</td>
<td>0.329</td>
<td>0.339</td>
<td>0.348</td>
<td>0.356</td>
<td>0.364</td>
<td>0.370</td>
</tr>
<tr>
<td>1.502</td>
<td>0.201</td>
<td>0.218</td>
<td>0.235</td>
<td>0.252</td>
<td>0.268</td>
<td>0.284</td>
<td>0.300</td>
<td>0.314</td>
<td>0.328</td>
<td>0.340</td>
<td>0.351</td>
<td>0.362</td>
<td>0.371</td>
<td>0.380</td>
<td>0.388</td>
</tr>
<tr>
<td>1.541</td>
<td>0.197</td>
<td>0.215</td>
<td>0.232</td>
<td>0.250</td>
<td>0.268</td>
<td>0.286</td>
<td>0.302</td>
<td>0.318</td>
<td>0.333</td>
<td>0.348</td>
<td>0.361</td>
<td>0.373</td>
<td>0.384</td>
<td>0.394</td>
<td>0.403</td>
</tr>
<tr>
<td>1.579</td>
<td>0.191</td>
<td>0.209</td>
<td>0.227</td>
<td>0.246</td>
<td>0.265</td>
<td>0.284</td>
<td>0.302</td>
<td>0.319</td>
<td>0.336</td>
<td>0.351</td>
<td>0.366</td>
<td>0.380</td>
<td>0.392</td>
<td>0.404</td>
<td>0.414</td>
</tr>
<tr>
<td>1.618</td>
<td>0.183</td>
<td>0.201</td>
<td>0.220</td>
<td>0.240</td>
<td>0.259</td>
<td>0.279</td>
<td>0.298</td>
<td>0.317</td>
<td>0.335</td>
<td>0.352</td>
<td>0.368</td>
<td>0.383</td>
<td>0.397</td>
<td>0.409</td>
<td>0.421</td>
</tr>
<tr>
<td>1.656</td>
<td>0.174</td>
<td>0.192</td>
<td>0.211</td>
<td>0.231</td>
<td>0.251</td>
<td>0.271</td>
<td>0.291</td>
<td>0.311</td>
<td>0.330</td>
<td>0.348</td>
<td>0.366</td>
<td>0.382</td>
<td>0.398</td>
<td>0.412</td>
<td>0.425</td>
</tr>
<tr>
<td>1.695</td>
<td>0.164</td>
<td>0.182</td>
<td>0.201</td>
<td>0.221</td>
<td>0.241</td>
<td>0.262</td>
<td>0.282</td>
<td>0.303</td>
<td>0.323</td>
<td>0.342</td>
<td>0.361</td>
<td>0.379</td>
<td>0.395</td>
<td>0.410</td>
<td>0.425</td>
</tr>
<tr>
<td>1.734</td>
<td>0.153</td>
<td>0.171</td>
<td>0.190</td>
<td>0.209</td>
<td>0.229</td>
<td>0.250</td>
<td>0.271</td>
<td>0.292</td>
<td>0.312</td>
<td>0.333</td>
<td>0.353</td>
<td>0.372</td>
<td>0.389</td>
<td>0.406</td>
<td>0.421</td>
</tr>
<tr>
<td>1.772</td>
<td>0.142</td>
<td>0.160</td>
<td>0.178</td>
<td>0.197</td>
<td>0.217</td>
<td>0.237</td>
<td>0.258</td>
<td>0.279</td>
<td>0.300</td>
<td>0.321</td>
<td>0.341</td>
<td>0.361</td>
<td>0.380</td>
<td>0.398</td>
<td>0.414</td>
</tr>
<tr>
<td>1.811</td>
<td>0.132</td>
<td>0.148</td>
<td>0.165</td>
<td>0.184</td>
<td>0.203</td>
<td>0.223</td>
<td>0.244</td>
<td>0.265</td>
<td>0.286</td>
<td>0.307</td>
<td>0.328</td>
<td>0.349</td>
<td>0.368</td>
<td>0.386</td>
<td>0.404</td>
</tr>
<tr>
<td>1.849</td>
<td>0.121</td>
<td>0.136</td>
<td>0.153</td>
<td>0.170</td>
<td>0.189</td>
<td>0.209</td>
<td>0.229</td>
<td>0.249</td>
<td>0.271</td>
<td>0.292</td>
<td>0.313</td>
<td>0.333</td>
<td>0.353</td>
<td>0.372</td>
<td>0.390</td>
</tr>
<tr>
<td>1.888</td>
<td>0.111</td>
<td>0.125</td>
<td>0.141</td>
<td>0.157</td>
<td>0.175</td>
<td>0.194</td>
<td>0.213</td>
<td>0.234</td>
<td>0.254</td>
<td>0.275</td>
<td>0.296</td>
<td>0.316</td>
<td>0.336</td>
<td>0.356</td>
<td>0.374</td>
</tr>
<tr>
<td>1.926</td>
<td>0.101</td>
<td>0.114</td>
<td>0.129</td>
<td>0.144</td>
<td>0.161</td>
<td>0.179</td>
<td>0.198</td>
<td>0.217</td>
<td>0.237</td>
<td>0.257</td>
<td>0.278</td>
<td>0.298</td>
<td>0.318</td>
<td>0.338</td>
<td>0.356</td>
</tr>
<tr>
<td>1.965</td>
<td>0.092</td>
<td>0.104</td>
<td>0.117</td>
<td>0.132</td>
<td>0.147</td>
<td>0.164</td>
<td>0.182</td>
<td>0.200</td>
<td>0.220</td>
<td>0.239</td>
<td>0.260</td>
<td>0.279</td>
<td>0.298</td>
<td>0.318</td>
<td>0.336</td>
</tr>
</tbody>
</table>

Collusion profit vs. Nash profit
- suppose $\alpha = 5\%$, $\delta = 95\%$ and $\varepsilon_t = \exp[-\beta t]$
- there is no collusive outcomes if $\beta$ is too large
Market Pricing Iterative Game

There is no collusive outcomes if $\beta$ is too large, cells are not visited enough (visits of the "collusion profit" cell)
Market Pricing Iterative Game

- if $\beta < \bar{\beta}$, iterations outcomes converge to (more) collusive ones, $n \to \infty$.

- if the companies can not agree on the duration of the training phase and on the $\epsilon$ value over the time there can be no collusion.
Market Pricing Iterative Game

- both agents should have similar exploration across time, otherwise...

Learning Curves for Different $\beta_1$ Values and $\beta_2 = 1e-05$

- $\beta_1 = 2.5e-06, \beta_2 = 1e-05$
- $\beta_1 = 1.0e-05, \beta_2 = 1e-05$
- $\beta_1 = 7.5e-05, \beta_2 = 1e-05$
"algorithm collusion" only occurs if companies first agree to set the same exploration parameters over time or if companies have a lot of time to explore.

It's in every agent's interest to lower his $\beta$ and increase his exploration phase.
one can notice that even if there is no collusion when agents don’t set the same level, prices are high due to the fact that the one which explore the most ‘undercut’ the other

Evolution of prices for Different $\beta_1$ Values and $\beta_2 = 1e^{-05}$
To go further...

- Calvano et al. (2020), *Artificial intelligence, algorithmic pricing, and collusion*

- What if the goal is not to collude (which we saw is difficult to prove/achieve etc.) but to profit from another agents behavior? Here, a participant in the market unilaterally exploits regularities in the pricing algorithms of competitors to achieve supracompetitive profits for all market participants.

- Rocher et al. (2023), *Adversarial competition and collusion in algorithmic markets*

- the Q-table is replace by a neural network that approximates the state-action value.

- Zheng et al. (2023), *Adaptive policy learning for offline-to-online reinforcement learning*

- Grondin et al. (2024) *Market pricing with reinforcement learning*


