Artificial Intelligence, Algorithmic Pricing and Collusion

Vincenzo Denicolò
University of Bologna and CEPR

with E. Calvano, G. Calzolari, and S. Pastorello

FTC 12th Micro Conference, Washington D.C.
November 14-15, 2019

Algorithmic collusion

- How serious is the risk of collusion among AI pricing algorithms?
- Answer crucial for policy

  - Low risk $\longrightarrow$ lasseiz faire
  - Some risk $\longrightarrow$ *ex post* intervention (antitrust)
  - High risk $\longrightarrow$ *ex ante* intervention (regulation)

AI pricing algorithms

- Two vintages of software:
  1. Rule-based software
     - Similar to *Stockfish* in chess
     - Can collude only to the extent that they are designed or instructed to do so
       - No really new antitrust issues
  2. Reinforcement learning algorithms (based on Artificial Intelligence)
     - Similar to *AlphaZero*
     - Learn from scratch (experimentation)
     - Programmers just specify the objective function (e.g., profits) and what variables to condition strategies on (e.g., past prices)

Early debate

- Concerned
  - Algorithms can change prices very quickly
    - As if discount factor was close to one
- Skeptics
  - Price coordination is a very difficult task, especially in the presence of asymmetries, uncertainty, many players etc.
  - Early computer science literature finds that algorithms fail to learn optimal strategies

Method

- Theoretical
  - Unfeasible
- Empirical
  - Very hard
- Experimental (numerical simulations)

Experimental approach

- Build simple reinforcement learning algorithms
- Have them interact repeatedly over time in controlled economic environments
- Observe outcomes
- Challenges
  - Economic environments must be realistic
  - Algorithms must be representative of those used in practice

Findings

- We find that even relatively simple pricing algorithms (Q-learning) systematically learn to play sophisticated collusive strategies
  - Such strategies involve punishments that have a finite duration, with a gradual return to the pre-deviation prices
- The algorithms leave no trace of explicit collusion
  - They learn to play collusive strategies by trial and error, with no prior knowledge of the environment in which they operate
  - They have not been designed or instructed to collude
  - They do not communicate with each other

Findings

- Previous literature (in both computer science and economics) has sometimes found supra-competitive prices
- But high prices might be the result of the algorithms' failure to learn a Nash equilibrium
  - For example, Waltman and Kaymak (2008) find that prices are higher when algorithms are short-sighted and have no memory than when they are patient and can condition on past prices
- We document rational collusion, not simply high prices, among pricing algorithms

Q-learning

- We focus on Q-learning algorithms
- Q-learning is
  - designed expressly to maximize the present value of a ow of rewards in problems of repeated choice
  - guaranteed to deliver the optimal policy in single decision making (but not in games)
  - popular among computer scientists
  - simple so that can be fully characterized by few parameters
  - the building block of the more sophisticated programs

Q-matrix

	…	…	$p_{1,t}$=10	…
…	…	…	…	…
…	…	…	…	…
$\boldsymbol{p}_{\mathbf{1},\boldsymbol{t-1}}$=8 $\boldsymbol{p}_{\mathbf{2},\boldsymbol{t-1}}$=5			Q-value	
…	…	…	…	…

UPDATING

For $(a, s) = (a_t, s_t)$

$$Q_{t+1}(a, s) = (1-\alpha)Q_t(a, s) + \ \alpha\left[\pi(a, s) + \delta \max_a[Q_t(a, s')]\right]$$

For $(a, s) \neq (a_t, s_t)$

$$Q_{t+1}(a, s) = Q_t(a, s)$$

Q-learning

- State (past prices) and action (current prices) spaces must be discretized
- A value is attached to each possible action in each possible state
- Initial values may be arbitrary
- As the game unfolds, each Q-value is updated giving weight α to new information (α is the learning rate) and $1 - \alpha$ to old information
- The action with the highest Q-value is chosen with probability $1 - \epsilon$ whereas the algorithm randomizes uniformly across all possible actions (explores) with probability ϵ
- ϵ declines with speed β and eventually goes to 0

Economic model

- An infinitely repeated Bertrand oligopoly game
- n firms, Logit demand and constant marginal costs c_i

$$q_i = \frac{e^{\frac{p_i a_i}{\mu}}}{\sum_{j=1}^{n} e^{\frac{p_j a_j}{\mu}} + e^{\frac{a_0}{\mu}}}$$

- Firms observe past prices and can condition current prices on them (however, finite memory)

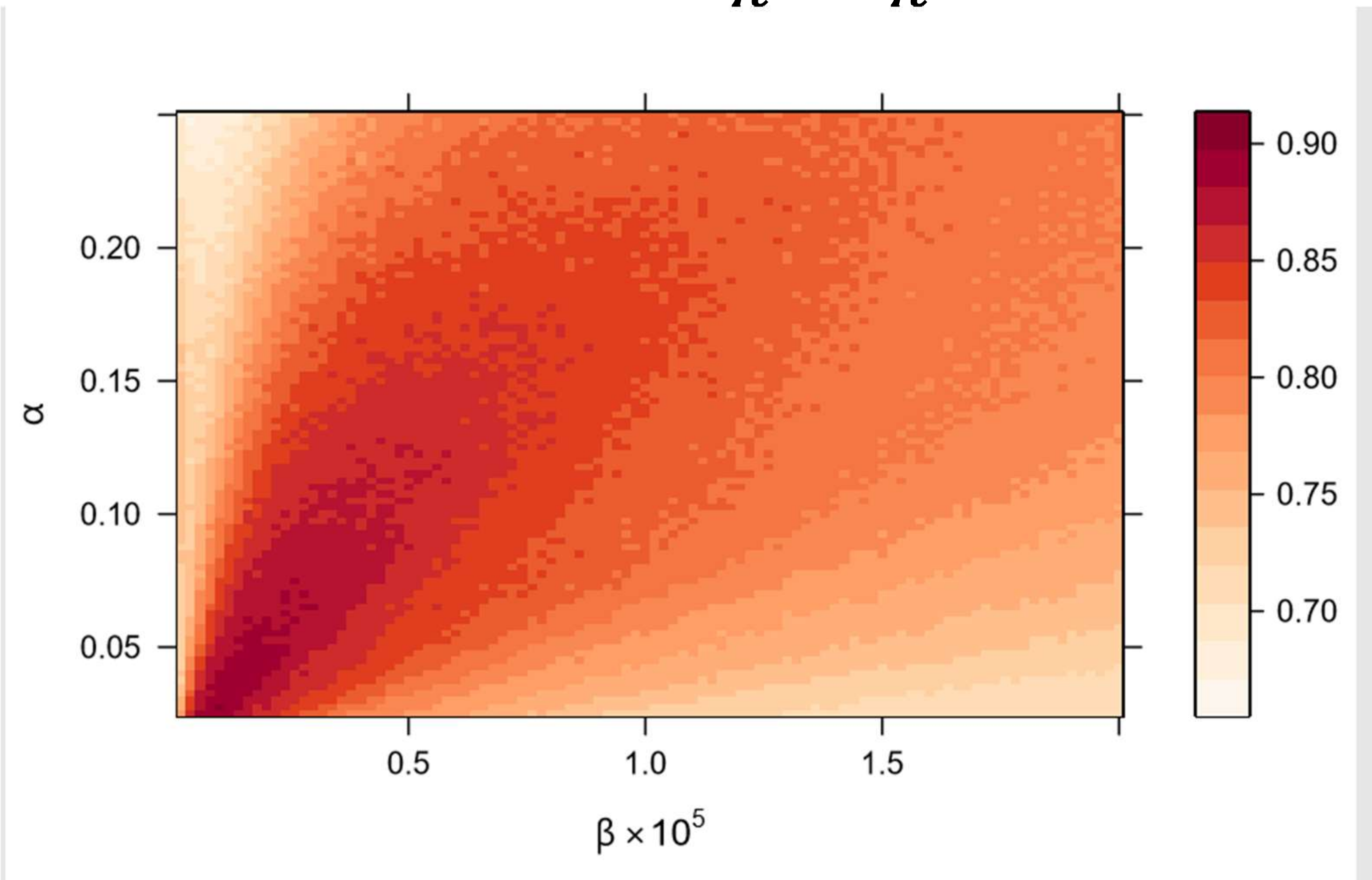
Baseline experiment

- $m = 15$
- $\xi = 10\%$
- $k = 1$
- $n = 2$
- $\delta = 0.95$
- $a_i = 2$
- $a_0 = 0$
- $c_i = 1$
- $\mu = \frac{1}{4}$

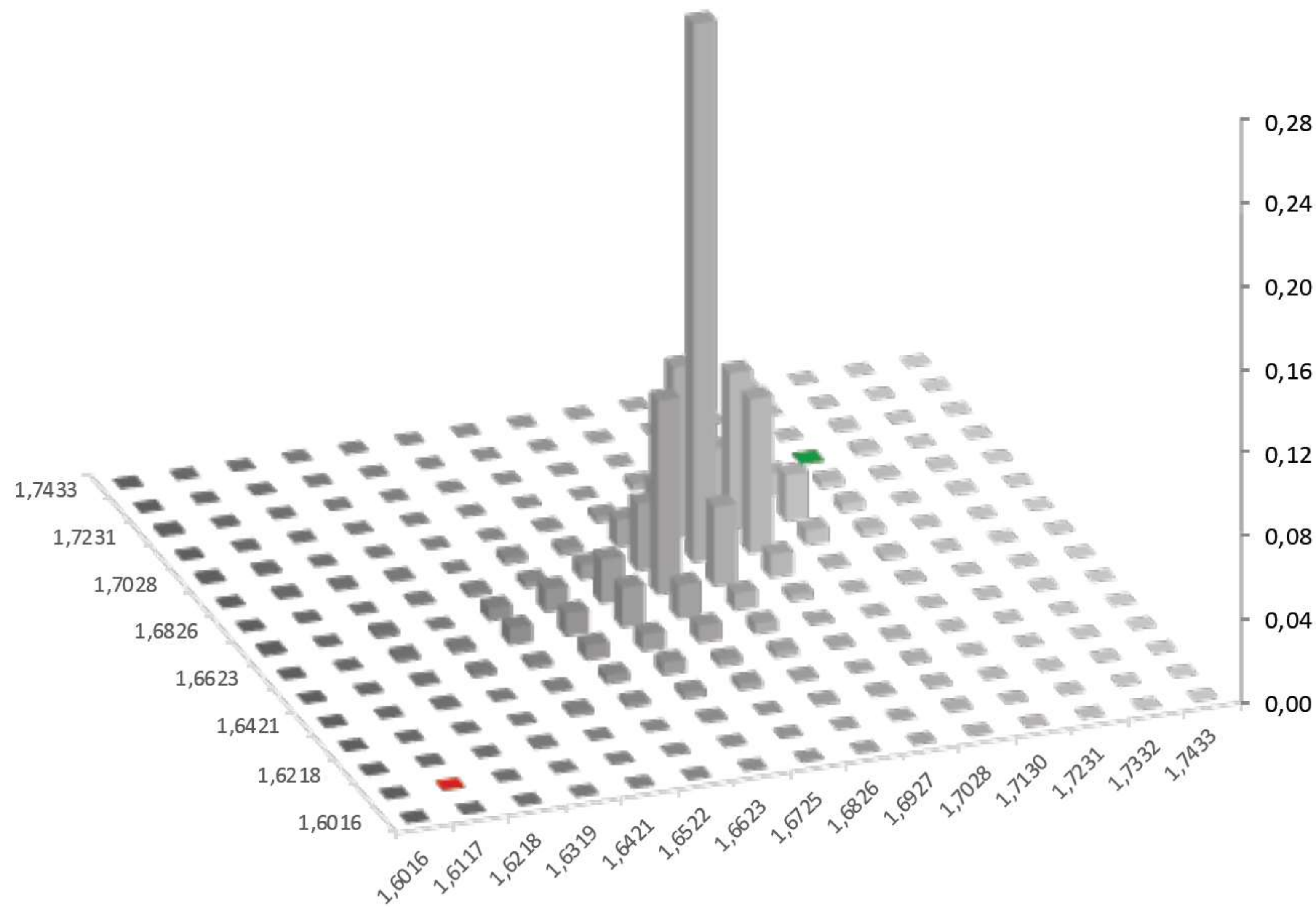
Convergence

- We let the algorithms interact and experiment until they settle to a constant pair of strategies
  - That is, until the perceived optimal strategy does not change for 100,000 periods in a row
- This typically requires that exploration has almost completely faded away
- We focus on outcomes upon convergence
  - Convergence is not guaranteed in theory but almost always achieved in practice

Average profit gain $\Delta = \frac{\bar{\pi} - \pi^N}{\pi^M - \pi^N}$



Prices


0,28
0,24
0,20
0,16
0,12
0,08
0,04
0,00
1,7433
1,7231
1,7028
1,6826
1,6623
1,6421
1,6218
1,6016
1,6016
1,6117
1,6218
1,6319
1,6421
1,6522
1,6623
1,6725
1,6826
1,6927
1,7028
1,7130
1,7231
1,7332
1,7433

Collusion?

- The key question is whether these high prices are the result of genuine collusion, or of the algorithms' failure to learn the static Nash equilibrium
- Policy implications would be radically different

Equilibrium play

- Do algorithms learn an optimal strategy (i.e., a Nash equilibrium)?
  - No theoretical guarantee
- Representative experiment ($\alpha = .15$; $\nu \approx 20$)
  - The algorithms play a Nash equilibrium about 50% of the times
  - When the algorithms do not play Nash, they play a strategy which is pretty close to a best response: the potential profit gain by playing a best response to the rival's strategy is , on average, less than 0.1%

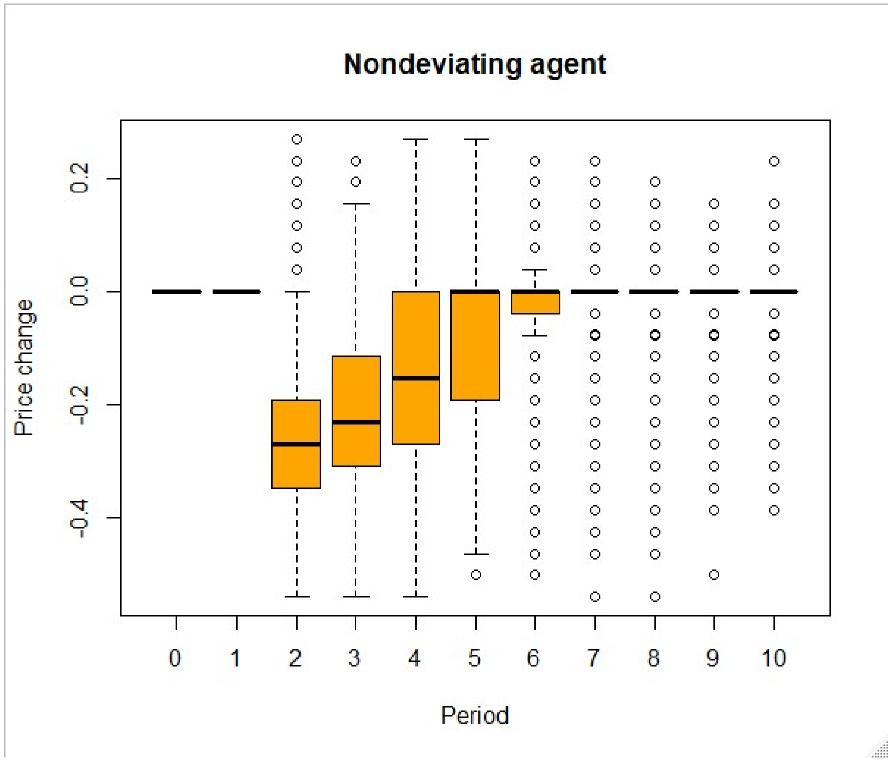
Tests of equilibrium play

- What do our algorithms learn when collusion cannot be an equilibrium phenomenon?
- Two cases:
  - $k = 0$ (no memory)
  - $\delta = 0$ (myopic behavior)
- In both cases, we find that the average profit gain tends to 0

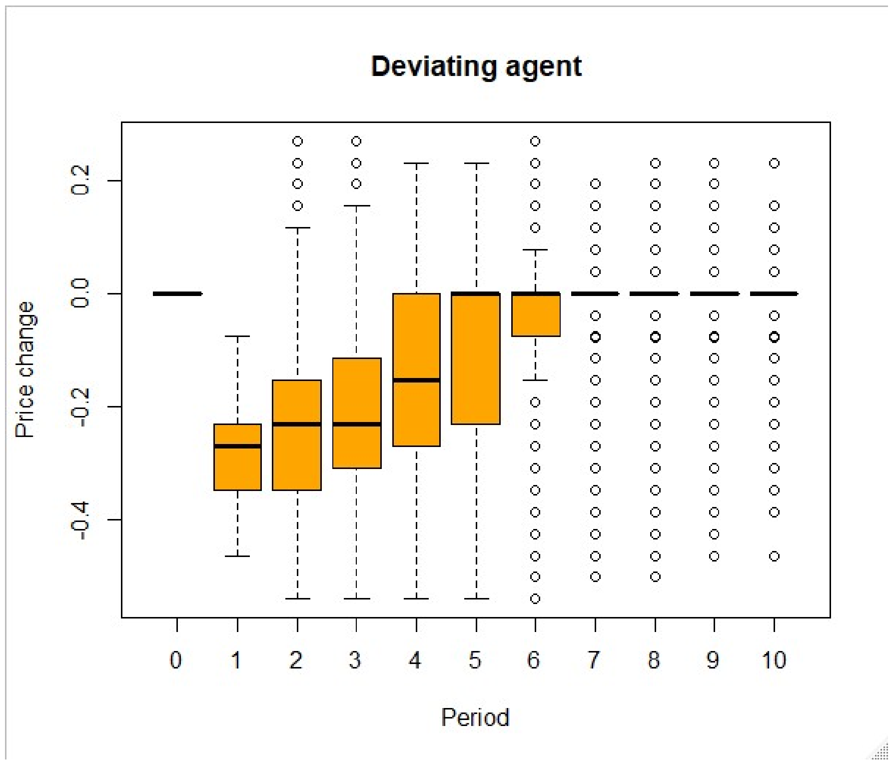
Impulse response

- Upon convergence, we force one algorithm to undercut
  - Deviation may last one or more period
  - Deviation price may be static best response to the opponent’s price, or different
- The other algorithm continues to play according to the learned strategy, and so does the deviating algorithms when it regains control of pricing
- We then look at what happens in the periods that follow
- In short, we derive "impulse-response" functions

Impulse response



Impulse response



Unprofitability of deviations

Colonna1	freq	1.43	1.47	1.51	1.54	1.58	1.62	1.66	1.70	1.74	1.78	1.82	1.85	1.89	1.93	1.97
1.62	0.01	0.96	0.95	0.93	0.89	0.9	0	NA	NA	NA	NA	NA	NA	NA	NA	NA
1.66	0.05	0.98	0.97	0.96	0.95	0.95	0.96	0	NA	NA	NA	NA	NA	NA	NA	NA
1.70	0.11	0.99	0.98	0.97	0.97	0.96	0.97	0.97	0	NA	NA	NA	NA	NA	NA	NA
1.74	0.16	0.99	0.99	0.98	0.98	0.97	0.97	0.97	0.98	0	NA	NA	NA	NA	NA	NA
1.78	0.19	0.99	0.99	0.98	0.98	0.97	0.97	0.97	0.97	0.98	0	NA	NA	NA	NA	NA
1.82	0.17	0.99	0.99	0.98	0.98	0.97	0.97	0.97	0.97	0.97	0.98	0	NA	NA	NA	NA
1.85	0.14	0.99	0.98	0.98	0.98	0.97	0.96	0.96	0.97	0.97	0.97	0.98	0	NA	NA	NA
1.89	0.09	0.99	0.98	0.98	0.97	0.96	0.96	0.96	0.95	0.96	0.96	0.97	0.98	0	NA	NA
1.93	0.05	0.99	0.98	0.97	0.97	0.95	0.95	0.94	0.94	0.94	0.95	0.96	0.97	0.98	0	NA
1.97	0.02	0.98	0.97	0.97	0.96	0.94	0.92	0.93	0.92	0.93	0.93	0.93	0.95	0.96	0.97	0

Robustness

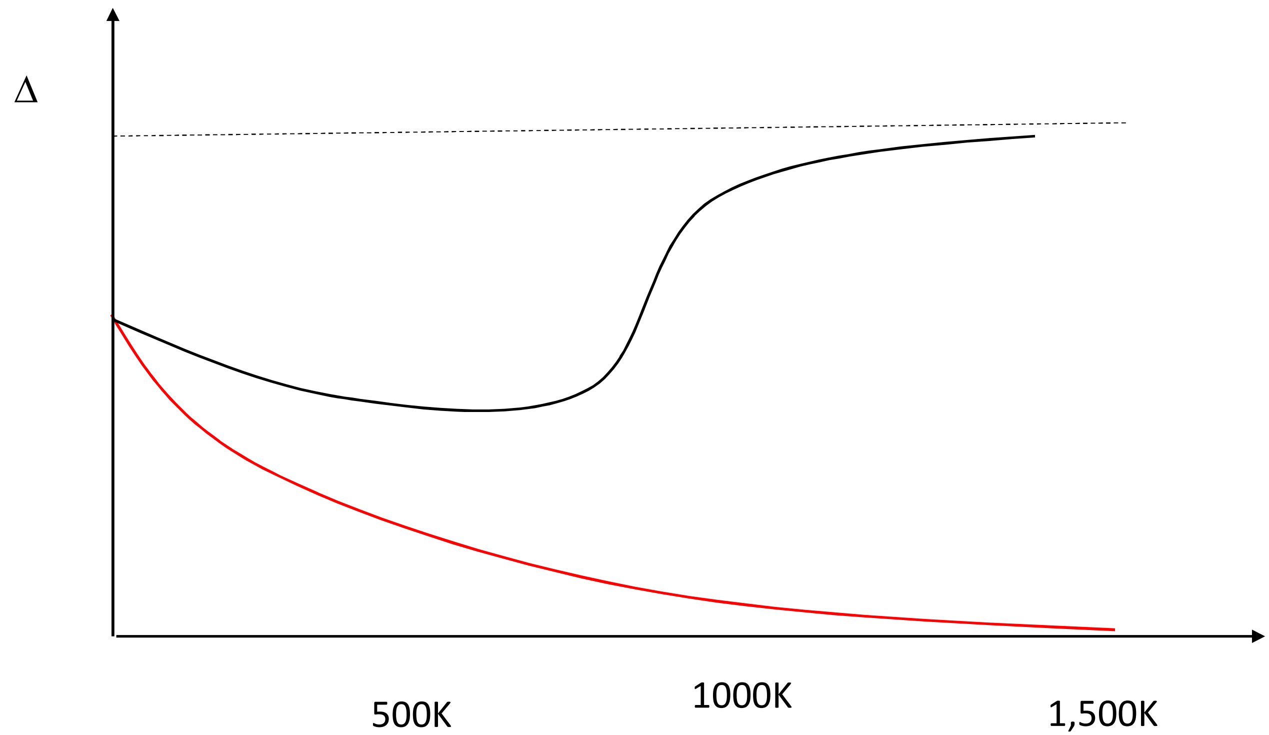
- Change in δ
- Asymmetric α and β
- Change in demand level
- Change in horizontal differentiation
- Stochastic demand
- Stochastic entry and exit
- More actions ($m = 30,50,100$)
- Longer memory ($k = 2$)
- Asynchronous learning

Time to convergence

- The algorithms do converge but convergence is slow
- For example, with $\alpha = 0.125$ and $\beta = 10^{-5}$ (the mid-point of our grid) convergence takes on average 850,000 periods
- We give the algorithms all the time that is needed to complete they learning

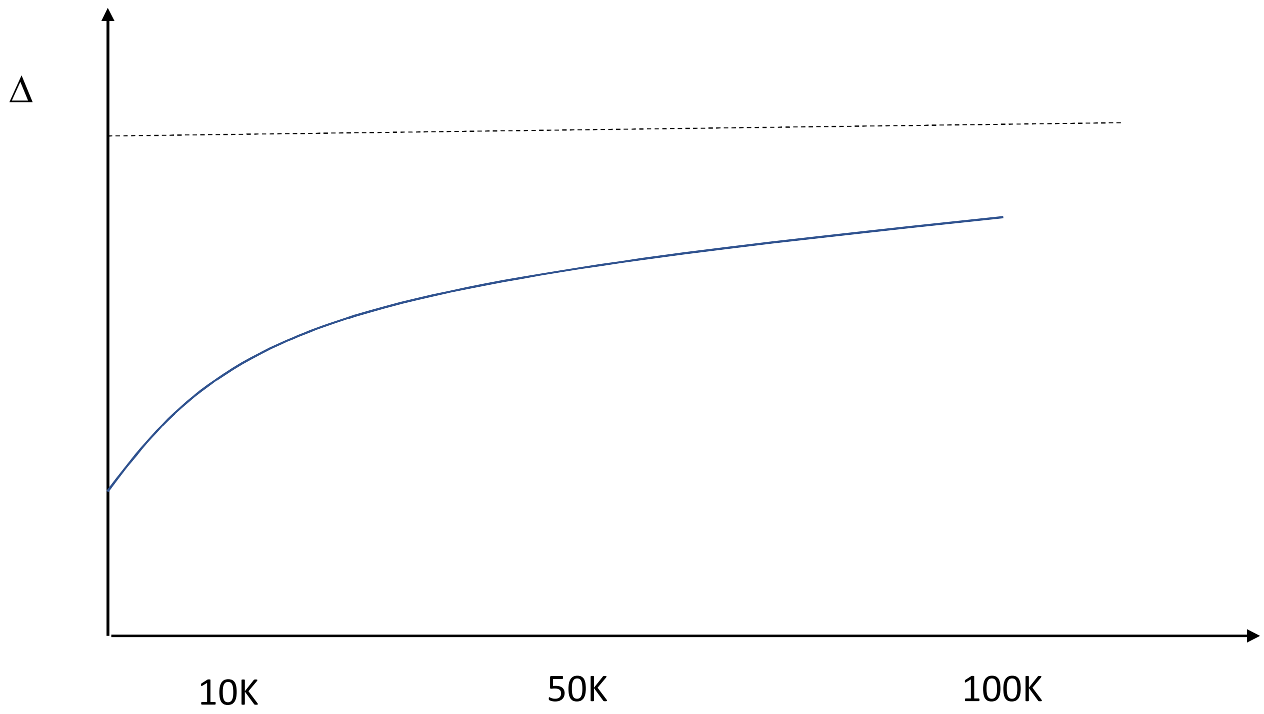
Transitional dynamics

- Algorithms may start to collude much before convergence is achieved



Off-line learning

- Algorithms may be trained in artificial environments (i.e., off-line) before being put to work in real market (i.e., on-line)



Off-line learning

- Algorithms may be trained in artificial environments (i.e., off-line) before being put to work in real market (i.e., on-line)

Faster learning

- More efficient algorithms exists and ought to be considered in future work
  - Value function approximation
  - Deep learning

Implications for policy

- Collusion among AI pricing algorithms would defy current policy
  - In most countries, tacit collusion is not regarded as illegal on the ground that
    - It is unlikely (few false negatives under lasseiz faire)
    - It would be hard to detect (many false positives with more active policy)
- Balance between type I and type II errors may change with pricing algorithms
  - More false negatives under lasseiz faire
  - When there are signs of algorithmic collusion, agencies may subpoena
    - unlike human decision-makers, algorithms can be seized and studied in artificial markets
  - This reduce the risk of false positives

More firms

- In the lab, supra-competitive prices disappear as soon as there are three or more competing firms
- We have looked at the case $n = 3$ and $n = 4$
- For $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$, results are reported below

	$\boldsymbol{n = 2}$	$\boldsymbol{n = 3}$	$\boldsymbol{n = 4}$
Δ	85%	64%	56%

Asymmetric firms

- Collusion is notoriously more difficult when firms are asymmetric
- We have considered both the case of cost and demand asymmetries
- Results are similar
- With $c_1 = 1$ and $c_2 = 0.75$ (which implies a market share for the more efficient firm of almost 60%), for $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$ we have

	Symmetric	Asymmetric
Δ	85%	81%