UNIVERSITY OF AMSTERDAM

ASSESSING AUTONOMOUS ALGORITHMIC COLLUSION: Q-LEARNING UNDER SHORT-RUN PRICE COMMITMENTS

Timo Klein

Amsterdam Law School Legal Studies Research Paper No. 2018-15

Amsterdam Center for Law & Economics Working Paper No. 2018-05

Assessing Autonomous Algorithmic Collusion: Q-Learning Under Short-Run Price Commitments*

Timo Klein†

September 2018

Abstract

A novel debate within competition policy and regulation circles is whether autonomous machine learning algorithms may learn to collude on prices. We show that when firms face short-run price commitments, independent Q-learning (a simple but well-established self-learning algorithm) learns to profitably coordinate on either a fixed price or on asymmetric price cycles – although convergence to rational and Pareto-optimal collusive behavior is not guaranteed. The general framework used can guide future research into the capacity of more advanced algorithms to collude, also in environments that are less stylized or more case-specific.

JEL-codes: K21, L13, L49
Keywords: pricing algorithms, algorithmic collusion, machine learning, reinforcement learning, Q-learning, sequential pricing

*I am grateful for valuable comments and suggestions by Joe Harrington, Harold Houba, Michael Kaisers, Maarten Pieter Schinkel, Ulrich Schwalbe and Leonard Treuren on earlier versions. Errors remain my own.

†Amsterdam School of Economics, University of Amsterdam. E-mail: t.klein@uva.nl

"We will not tolerate anticompetitive conduct, whether it occurs in a smoke-filled room or over the internet using complex pricing algorithms"

– US Department of Justice Assistant Attorney General Bill Baer (6 April 2015)

"I think we need to make it very clear that companies can't escape responsibility for collusion by hiding behind a computer algorithm"

– EU Competition Commissioner Margrethe Vestager (16 March 2017)

"The [Autonomous Algorithmic Collusion] literature is the closest ever our field came to science-fiction"

– Nicolas Petit, Professor of Law at Liege University (2017)

1 Introduction

The growing prominence of digitization, big data and artificial intelligence in commercial activities has given rise to several novel debates within competition policy and regulation circles. One prominent concern is that intelligent, self-learning pricing algorithms may at some point work out that the best thing for them to do is to refrain from aggressive pricing, keeping prices high (Ezrachi and Stucke, 2016; Mehra, 2016). This would be akin to collusion, but without any overt act of communication required to establish a competition law infringement, preventing competition authorities from doing anything about it (Harrington, 2017; Gal, 2018).[1] The debate has received extensive press coverage[2] as well as increasing attention from policymakers – such as the European Commission[3] and the OECD[4] – and economic consultancy firms[5].

The concerns on autonomous collusion appear to be mostly based on a loose and intuitive interpretation of artificial intelligence only (Ittoo and Petit, 2017; Schwalbe, 2018). This has led several commentators to conclude that the debate is overblown. Substantially, the main critique is that self-learning algorithms would be ill-equipped

[1] Other prominent debates include the use of personalized pricing based on online behavior, the market power of large digital platforms and competitive risks to online privacy.

[2] These include, amongst others, "When Bots Collude", in: *The New Yorker* (25 April 2015); "How Pricing Bots Could Form Cartels and Make Things More Expensive", in: *Harvard Business Review* (27 October 2016); "Policing the Digital Cartels", in: *Financial Times* (8 January 2017), "Price-Bots Can Collude Against Consumers", in: *The Economist* (6 May 2017), "The Algorithms Have Landed!", in: *Antitrust Chronicle* (May 2017), "When Margrethe Vestager Takes Antitrust Battle to Robots", in: *Politico* (28 February 2018) and "Kartellbildung Durch Lernende Algorithmen?", in: *Frankfurther Allgemeine Zeitung* (13 July 2018)

[3] See in particular the speech by EU Commissioner Vestager "Algorithms and Competition" (16 March 2017), at the Bundeskartellamt 18th Conference on Competition, Berlin, https://ec.europa.eu/commission/commissioners/2014-2019/vestager/announcements/bundeskartellamt-18th-conference-competition-berlin-16-march-2017_en.

[4] OECD (June 2017) "Algorithms and Collusion: Competition Policy in the Digital Age", http://www.oecd.org/competition/algorithms-and-collusion.htm.

[5] Including Oxera (2017) and RBB Economics (2018).

to coordinate on any one out of a (possibly multidimensional) continuum of subgame perfect equilibria, at least in absence of illegal communication (Kühn and Tadelis, 2017; Schwalbe, 2018). This critique is supported with references to the experimental economics literature, where tacit collusion by humans fails to occur in reasonably realistic oligopoly settings. This would, however, presume equivalence between humans and learning algorithms in action-selection and learning, which is generally not the case. And explicit attempts to assess the capacity of different autonomous algorithms to collude in various oligopoly environments remain scarce.

This paper discusses the capacity of reinforcement learning – a type of machine learning in which agents learn from interacting autonomously with their environment – to collude in an oligopoly environment. More specifically, we assess the capacity of independent Q-learning (a simple but well-established reinforcement learning algorithm) to collude in a dynamic oligopoly environment with short-run price commitments. The results show that when firms face short-run price commitments, competing Q-learning algorithms profitably coordinate on either a fixed price or on Edgeworth price cycles. Under Edgeworth price cycles, large periodic price increases reset a gradual downward price spiral. This produces the kind of asymmetric price cycles that are similarly observed in other markets often suspected of tacit collusion – most prominently gasoline markets (Noel, 2011; Eckert, 2013; Byrne and de Roos, 2018).

There is a general absence of continuous online price adjustments in many e-commerce outlets, where prices instead follow stepwise adjustments. This provides suggestive evidence that firms are short-run price committed. To capture this dynamic we use the sequential pricing environment of Maskin and Tirole (1988), in which firms set prices sequentially and profits are realized after each turn. Following Maskin and Tirole we also impose the Markov assumption: firms only condition their strategy on state variables that are directly payoff-relevant. This includes demand estimation, marginal cost and current competitor price but excludes, for instance, communication and the history of prices. Maskin and Tirole show that in their environment firms are able to charge higher prices and earn higher profits in equilibrium provided firms value future profits sufficiently high. They interpret this as tacit collusion (p. 592).

The learning algorithm applied is a staightforward adaptation of independent Q-learning to sequential interaction. After choosing a price given current competitor price, it observes the realized profit and subsequent competitor response and updates the expected optimal long-run profit from choosing the price it did given the competitor price. By interacting autonomously with its environment, it has to make a continuous trade-off between exploitation (choosing the currently perceived optimal price) and exploration (choosing perhaps another price, to see what happens and improve precision). Q-learning is particularly suitable for studying autonomous pricing behavior, because it does not require any prior input, data mining or model of the environment (such as a demand function or competitor profit function). Additionally,

Q-learning is relatively straightforward and one of the most well-established methods within reinforcement learning. Finally, various types of Q-learning algorithms are being applied in real-world dynamic pricing application, including airline fares and wholesale electricity markets (Ittoo and Petit, 2017).

Three challenges remain unresolved in guaranteeing convergence to rational and Pareto-optimal collusive behavior: independent Q-learning is restricted to deterministic, pure strategy learning while in case of short-run price commitments equilibrium behavior involves *mixing strategies*; agents face a *moving target learning problem* in which their best response changes as others changes their response, which may result in endless recursive adaptation; and the existence of a set of multiple equilibria (albeit limited and discrete) provides an *equilibrium selection problem* in which Pareto-optimality is not guaranteed. Note, however, that despite these challenges the algorithm does not have to behave badly in practice. In absence of theoretical guarantees we provide an empirical understanding through simulations. An appendix is provided that discusses how developments in multi-agent reinforcement learning could resolve the remaining challenges, but also shows why they still lack practical applicability to oligopoly environments.

There are only a few papers that look at algorithmic oligopoly collusion beyond an iterated 2-by-2 prisoner's dilemma.[6] Looking at quantity competition, Huck, Normann and Oechssler (2003) find that a "win-continue-lose-reverse" rule provides joint-profit maximizing convergence and Waltman and Kaymak (2008) that independent Q-learning may collude on lower quantities. Convergence is however not robust to small fluctuations in the payoff function (Izquierdo and Izquierdo, 2015). Taking a novel experimental approach, Zhou *et al.* (2018) propose an ex ante restricted algorithm capable of extorting a human rival to collude. Looking at price competition, Tesauro and Kephart (2002) show in an environment similar to ours how independent Q-learning can converge on profitable asymmetric price cycles – with cycles becoming shorter and profits increasing if products are more differentiated or consumers less informed. They assume however full knowledge of the environment, which allows for calculating optimal behavior using dynamic programming. Finally, Salcedo (2015) shows that under certain sufficient conditions collusion is inevitable when firms adopt a fixed-strategy pricing algorithm that periodically 'decodes' the other algorithm and subsequently adjusts itself. The proposed conditions may however not hold in practice (Harrington, 2017) and may even be framed as explicit collusion by communicating your pricing strategies through decoding (Kühn and Tadelis, 2017; Schwalbe, 2018).

The remainder is organized as follows. Section 2 defines the competitive environment and the algorithm used. Section 3 discusses the empirical results. We look at the case where a Q-learning algorithm faces a fixed-strategy competitor and where two Q-learning algorithms are set to compete with each other. In both cases prof-

[6] See for instance Harrington (2017) and Calvano *et al.* (2018) for a discussion on reinforcement learning and collusion in iterated prisoner's dilemmas and Schwalbe (2018) for a more general discussion on the relevant computer science and (experimental) economics literature.

its exceed their static level once firms price sequentially, but only in the first case joint-profit maximization is always achieved. In Section 4 we provide a discussion and possible extensions and Section 5 concludes.

2 Environment and Learning Algorithm

2.1 Environment: Sequential Pricing Duopoly

To capture the dynamics of short-run price commitments we take the sequential pricing environment as described by Maskin and Tirole (1988).

There are two identical firms $i \in \{1, 2\}$ with homogeneous goods and unrestrictive capacity. Sequentially, each firm sets an integer price $p_t^i \in \{0, 1, 2, ..., k\}$, where in odd-numbered periods t firm 1 chooses its price while firm 2 keeps its price unchanged and vice versa in even-numbered periods. Letting $D(\cdot)$ denote the market demand function and c marginal cost, define instantaneous profit of firm i at time t as

$$\pi\left(p_t^i, p_t^j\right) = \begin{cases} D\left(p_t^i\right)\left(p_t^i - c\right) & \text{if } p_t^i < p_t^j \\ \frac{1}{2} D\left(p_t^i\right)\left(p_t^i - c\right) & \text{if } p_t^i = p_t^j \\ 0 & \text{if } p_t^i > p_t^j \end{cases} \tag{1}$$

where total profit is assumed strictly concave. We restrict ourselves to $D(p) = k - p$ as in the illustrating example by Maskin and Tirole, with the joint-profit maximizing or monopoly price derived as $p^m = \frac{1}{2}(k + c)$. Firms discount future profits according to a discount factor $\delta \in [0, 1)$, where each firm has as objective to maximize at time t its cumulative stream of discounted future profits, so

$$\max \sum_{s=0}^{\infty} \delta^s \pi\left(p_{t+s}^i, p_{t+s}^j\right). \tag{2}$$

Note that firms only observe realized profit given their own price and the price of their competitor; demand and competitor profit functions are unknown. We assume a stationary environment and symmetric payoffs. A discussion of how non-stationaries and asymmetry may affect results is included in Section 4.

Similarly as in Maskin and Tirole we impose the Markov assumption: strategies only depend on state variables that are directly payoff-relevant. This includes demand estimation, marginal cost and current competitor price but excludes, for instance, communication and the history of prices. For this setting, Maskin and Tirole define the concept of a Markov perfect equilibrium (MPE), which is a subgame perfect Nash equilibrium with Markov strategies. Take $V^i(p^j)$ to capture the sum of all present discounted future profits of firm i given current state p^j when behaving optimally. From dynamic programming, any set of (random) reaction functions $\{R^i(p^j), R^j(p^i)\}$ is defined as an MPE if it satisfies the Bellman optimality equations

$$V^i(p^j) = \max_p \left(\pi\left(p, p^j\right) + \delta E_{R^j(p)}\left[\pi\left(p, R^j\left(p\right)\right) + \delta V^i(R^j\left(p\right))\right]\right) \tag{3}$$

for all prices p^j and for $i \in \{1, 2\}$ and $j \neq i$.

Maskin and Tirole show that if firms value future profits sufficiently high there are two types of MPE: fixed price equilibria and an Edgeworth price cycle equilibrium. Under a fixed price equilibrium both firms sustain a fixed price with the common believe that the other firm would follow if it were to decrease its price, but not if it were to increase it. Such believes are sustained by off-equilibrium price war punishments in case any firm undercuts, in which case prices drop towards 1 and firms mix between staying at the lower price and returning to the fixed price – with probabilities such that both firms are similarly indifferent between staying and returning. Table 1 and Figure 1 illustrate this for the monopoly price for the case where $k = 6$ and $c = 0$.

Under an Edgeworth price cycle equilibrium firms undercut each other until prices reach the lower bound and neither firm makes any profit. At this lower bound, both firms have an incentive to raise their price and reset the gradual downward spiral but prefer the other firm to do so. They then mix between maintaining zero profit to punish the other firm for not resetting the price cycle and resetting the price cycle itself – with probabilities such that both firms are similarly indifferent between staying and resetting. This is also illustrated in Table 1 and Figure 1.

p	$R(p)$, fixed pricing	$R(p)$, Edgeworth price cycles
6	3	4
5	3	4
4	3	3
3	3	2
2	1	1
1	$\begin{cases} 1 & \text{with prob. } \beta_1(\delta) \\ 3 & \text{with prob. } 1-\beta_1(\delta) \end{cases}$	0
0	3	$\begin{cases} 0 & \text{with prob. } \beta_2(\delta) \\ 5 & \text{with prob. } 1-\beta_2(\delta) \end{cases}$
Average profit	4.5	$2\frac{1}{3}$ for $\delta \to 1$

Table 1: Reaction functions for fixed pricing and Edgeworth price cycles, with $\beta_1 = (5+\delta)/\left(5\delta + 9\delta^2\right)$ and $\beta_2 = \left(3\delta^2 - 1\right)\left(1 + \delta^2 + \delta^4\right)/\left(8 + 7\delta^2 + 2\delta^4 + 3\delta^6\right)$

Leufkens and Peeters (2011) test experimentally whether humans are capable of coordinating on either fixed pricing or Edgeworth price cycles. Taking the illustrating example in Maskin and Tirole and shown here ($k = 6$, $c = 0$), they find that under a random ending rule, 13 out of 15 pairs end up coordinating on the joint-profit maximizing fixed price of $p = 3$, one pair on a fixed price of $p = 2$ and one pair on a pricing cycle. Compared to treatments with simultaneous moves, they find that sequential moves generate larger profits and more price stickiness.

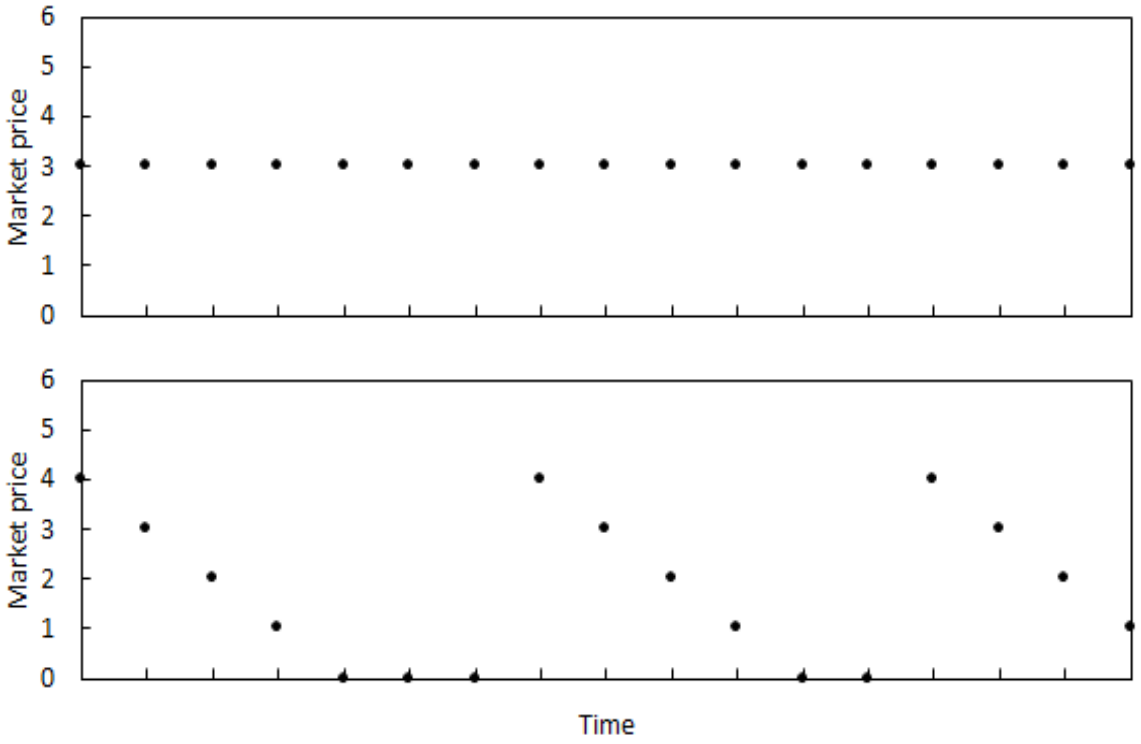


Figure 1: Price dynamics in fixed pricing (top) and Edgeworth cycles (bottom)

2.2 Algorithm: Sequential Independent Q-Learning

The learning algorithm applied is a straightforward adaptation of independent Q-learning to sequential interaction. Q-learning (Watkins, 1989) is a simple but well-established single-agent reinforcement learning model.[7] By interacting with its environment, the algorithm learns a so-called Q-function that matches the optimal long-run value to setting any price given any competitor price. During this interaction, the algorithm uses a dynamic action selection policy that balances exploitation (choosing the currently perceived optimal price) and exploration (choosing perhaps another price, to improve the precision of the Q-function). Below the specification is discussed in detail and a note is provided on its limitations and challenges in our context of oligopoly competition.

2.2.1 Sequential Q-Learning

Q-function $Q^i(p^j, p^i)$ maps for firm $i \in \{1, 2\}$ action p^i (new own price) into its optimal long-run value given current state p^j (current competitor price). In our approach,

[7] For a comprehensive introduction on single-agent reinforcement learning, see Sutton and Barto (2018).

$Q^i(p^j, p^i)$ is initialized as an empty matrix. After observing profits and competitor response, the algorithm updates the Q-function according to the following recursive relationship

$$Q^i(p^j, p^i) \leftarrow (1-\alpha)\, Q^i(p^j, p^i) + \alpha \left(\pi(p^i, p^j) + \delta \pi(p^i, p^{j'}) + \delta^2 \max_p Q^i(p^{j'}, p) \right), \quad (4)$$

where $p^{j'}$ is the subsequent price of firm j and $\alpha \in (0,1)$ a stepsize parameter that determines the weight ascribed to the observed value relative to its old value (assumed constant in our case but which may also be time-varying).

In balancing exploration and exploitation, the algorithm adopts a probabilistic action selection policy. Using a so-called ε-greedy exploration procedure, the Q-learning algorithm used here selects in each own turn as its action an integer price $p_t^i \in [0, k]$ following

$$p_t^i \begin{cases} \sim U[0,k] & \text{with probability } \varepsilon(t) \\ = \arg\max_p Q^i(p_t^j, p) & \text{with probability } 1-\varepsilon(t) \end{cases} \quad (5)$$

where exploration occurs with probability $\varepsilon(t) \in [0,1]$ and exploitation with probability $1-\varepsilon\,(t)$. In case multiple actions share the same highest Q-value under exploitation, the algorithm randomizes over these actions. The probability of exploration is in our case determined as

$$\varepsilon(t) = \varepsilon(0)(1-\theta)^t, \quad (6)$$

where $\varepsilon(0) \in [0,1]$ is the initial exploration probability and $\theta \in [0,1]$ a decay parameter. Whenever $\theta > 0$, the decay in exploration ensures convergence to a deterministic strategy.

An often-used alternative to the ε-greedy procedure is the so-called Boltzmann (or softmax) exploration procedure, which involves quantal responses: price p^i given state p^j is chosen with probability

$$\Pr\left(p^i | p^j\right) = \frac{\exp\left(Q^i(p^j, p^i)/\tau(t)\right)}{\sum_p \exp\left(Q^i(p^j, p)/\tau(t)\right)}, \quad (7)$$

with $\tau(t) > 0$ as a so-called temperature parameter. When $\tau \to \infty$, action selection is random and for $\tau \in (0, \infty)$ higher-valued actions are selected with a higher probability than lower-valued actions. Usually, τ is decreasing gradually towards 0 over time, to increase exploitation once precision improves. Because simulation results are qualitatively similar to ε-greedy for certain parameter settings, we restrict our further analysis to ε-greedy.

A pseudocode of the learning algorithm is provided below.

Pseudocode Independent Q-Learning in Sequential Pricing	
1	Set demand function and parameters δ, α, $\varepsilon(0)$ and θ
2	Initialize Q^1 and Q^2 as empty matrices
3	Initialize p_1^1 and p_2^2 randomly
4	Initialize $t = 3$, $i = 1$ and $j = 2$
5	**Loop over each period**
6	| Set price $p_t^j = p_{t-1}^j$
7	| Set price p_t^i according to (5)
8	| Update $Q^j(p_{t-2}^i, p_{t-1}^j)$ according to (4)
9	| Update $t \leftarrow t + 1$ and $\{i \leftarrow j, j \leftarrow i\}$
10	**Until** $t = T$ (specified number of periods)

In case of simultaneous move, step 7 applies to both agents and Q-value updates by both agents occur directly after moves are made and profits realized – with the observed value determined as $\pi(p^i, p^{j'}) + \delta \max_p Q^i(p^{j'}, p)$, where $p^{j'}$ is the simultaneous move of opponent j.

When a single Q-learning agent faces a fixed-strategy competitor, the sequence of Q^i in (4) provably converges to the values under the optimal (rational, best-response) strategy, given the general stepsize conditions that $\sum_{t=0}^{\infty} \alpha^2 < \infty$ and $\sum_{t=0}^{\infty} \alpha \to \infty$ and asymptotically all relevant state-action pairs $\{p^j, p^i\}$ are visited infinitely often (Watkins and Dayan, 1992; Tsitsiklis, 1994). The sequential Q-learning algorithm developed here would therefore converge to the optimal strategy if the opponent maintains a fixed strategy.

2.2.2 Limitations and Challenges

Two often used objectives for multi-agent learning algorithms include convergence and rationality (Bowling and Veloso, 2002; Busoniu *et al.*, 2008). Under convergence the algorithm converges to a stationary strategy when the other agents use a fixed strategy. Convergence is a desirable property in order to avoid endless recursive adaptation and perform analyses on eventual outcomes. And under rationality the algorithm adopts an optimal, best-response strategy in response to the other agents. This is a desirable property in order to preclude cases in which agents do not behave in their own self-interest – a common assumption in oligopoly analyses. A corollary of convergence and rationality is that the eventual outcome is a Nash equilibrium, in which no agent can be better off given the other strategies. A third desirable property in our context of autonomous collusion in oligopoly environments is that of Pareto-optimality, under which no agent can be made better off without making at least one agent worse off.

The independent Q-learning algorithm cannot convergence to rational and Pareto-optimal collusive behavior in our environment because it fails to address three remaining challenges. Firstly, our independent Q-learning algorithm is restricted to deterministic, pure strategy learning, while in our environment equilibrium behavior

involves *mixing strategies*. This means that given any currently-learned strategies, at least one agent is always better off adjusting its strategy. Secondly, even if it were capable of learning mixed strategies, the algorithm remains vulnerable to adaptation and experimentation by its opponent. More generally, agents that are simultaneously adapting to each others' behavior face a *moving target learning problem* (Bowling and Veloso, 2002; Busoniu *et al.*, 2008; Tuyls and Weiss, 2012), in which their best response changes as others changes their response. Convergence guarantees that exist for single-agent reinforcement learning algorithms then no longer hold and agents may end up in endless recursive adaptation. And thirdly, there exists a set of multiple equilibria (albeit limited and discrete) in our environment, involving fixed prices and Edgeworth price cycling. This provides an *equilibrium selection problem* in which it is *a priori* unclear whether the equilibrium that materializes is a Pareto-optimal equilibrium.

Despite these challenges the algorithm does not have to behave badly in practice. It only means that theory is unable to say how well it is expected to behave. In absence of theoretical guarantees we provide an empirical understanding through simulations in the next section.

3 Empirical Results

For the empirical exercise, we initially take the illustrating example in Maskin and Tirole as discussed in Section 2.1, with $k = 6$ and $c = 0$. We take an initial exploration probability $\varepsilon(0) = 1$, decay parameter $\theta = 0.001$ and stepsize parameter $\alpha = 0.5$, although results are robust to reasonable variations in these parameters as well as the use of Boltzmann exploration.

To assess the performance of the Q-learning algorithm, we simulate $1,000$ runs, each lasting $5,000$ periods.[8] Over these $5,000$ periods, the probability of exploration drops below 1%. For each period, we average over the $1,000$ simulated runs to see how on average market price and profit develop over time. We make a comparison between the static case ($\delta = 0$) and a dynamic case ($\delta = 0.95$) to see whether profits are above their static level. We also discuss how results would be when firms compete simultaneously instead of sequentially.

We first consider the case where the Q-learning algorithm faces a fixed-strategy agent with monopoly fixed price behavior as described in Section 2.1. Secondly, we consider the case where two Q-learning algorithms are set to face each other.

3.1 Q-Learning Versus Fixed-Strategy Behavior

We find that the dynamic Q-learning algorithm neatly converges to the monopoly price when faced with a fixed-strategy competitor with monopoly fixed price behavior.

[8] Simulations are programmed in MATLAB®.

This occurs even without any prior knowledge of the environment or competitor behavior.

More specifically, in Figure 2 the black curves show convergence in case of static optimization, providing a stable market price of 1 and average profits of 2.5.[9] The gray curves show what happens when the Q-learning algorithm takes into account the long-run effects of its pricing decisions. In this case, the Q-learning algorithm fully adapts to its fixed-strategy competitor and converges to the monopoly price of 3, providing a constant joint-profit maximizing profit of 4.5.

This result is not very surprising. Convergence to rationality is guaranteed here, because there is only a single Q-learning agent facing a fixed-strategy competitor – as discussed at the end of Section 2.2.1. Similarly, if firms would compete simultaneously, a fixed-strategy competitor would be able to equivalently "extort" a Q-learning algorithm to collude. Observe however that these outcomes, while resembling a monopoly price equilibrium, are not an MPE. This is because the Q-learning algorithm is unable to learn the off-equilibrium mixing strategy necessary for the fixed-strategy competitor to behave rationally. In response to the Q-learning algorithm, the fixed strategy agent can be better off.

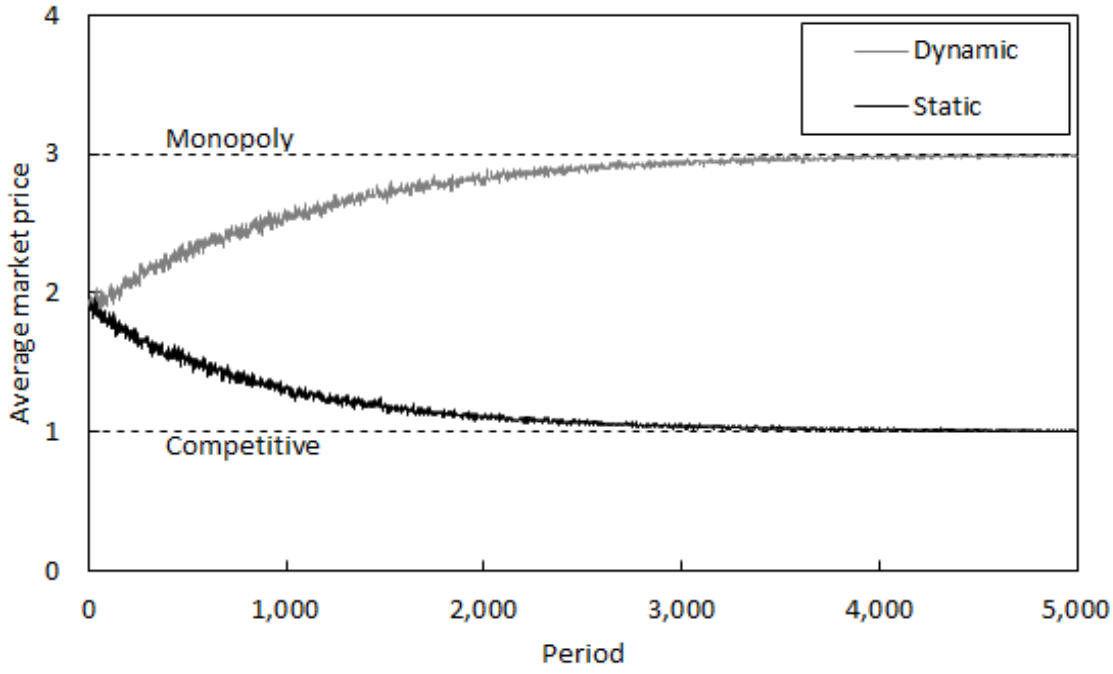
3.2 Q-Learning Versus Q-Learning

While the Q-learning algorithm performs well facing a fixed-strategy competitor, it remains to be shown how two competing Q-learning algorithms perform. We find that when two dynamic Q-learning algorithms face each other, they manage to profitably coordinate on either a fixed price or on asymmetric price cycles when k is low, and increasingly on asymmetric price cycles only when k is high.

Figure 3 shows for $k = 6$ that the Q-learning algorithms learn to maintain prices that are on average higher than their static levels. This occurs even though both algorithms have no prior knowledge of the environment and have to learn simultaneously. Average profits are around 3.5. This is above those in the Edgeworth price cycle MPE of around $2\frac{1}{3}$ and the static (or competitive) level of 2.5 but below monopoly profits of 4.5. While not shown here, prices and profits similarly converge to their static levels if firms compete simultaneously.

Table 2 shows the types of behavior the algorithms learn, as captured by the final 100 periods of all runs. In 349 out of $1,000$ runs, the algorithms converge to a single, stable fixed price, one-third of which at the monopoly level of $p = 3$. This is still well below the experimental results as found by Leufkens and Peeters (2011), in which 13 out of 15 human pairs manage to coordinate at the monopoly level. If the algorithms do not converge to a fixed price, they clearly display asymmetric pricing: decreases in the market price occur around twice as often as increases and the average time between market price increases is 4.0 periods. If a decrease occurs, this happens with

[9] In the figure average two-period profit is taken. This is done to include exactly one period in which the Q-learner moves and one in which it is price committed.



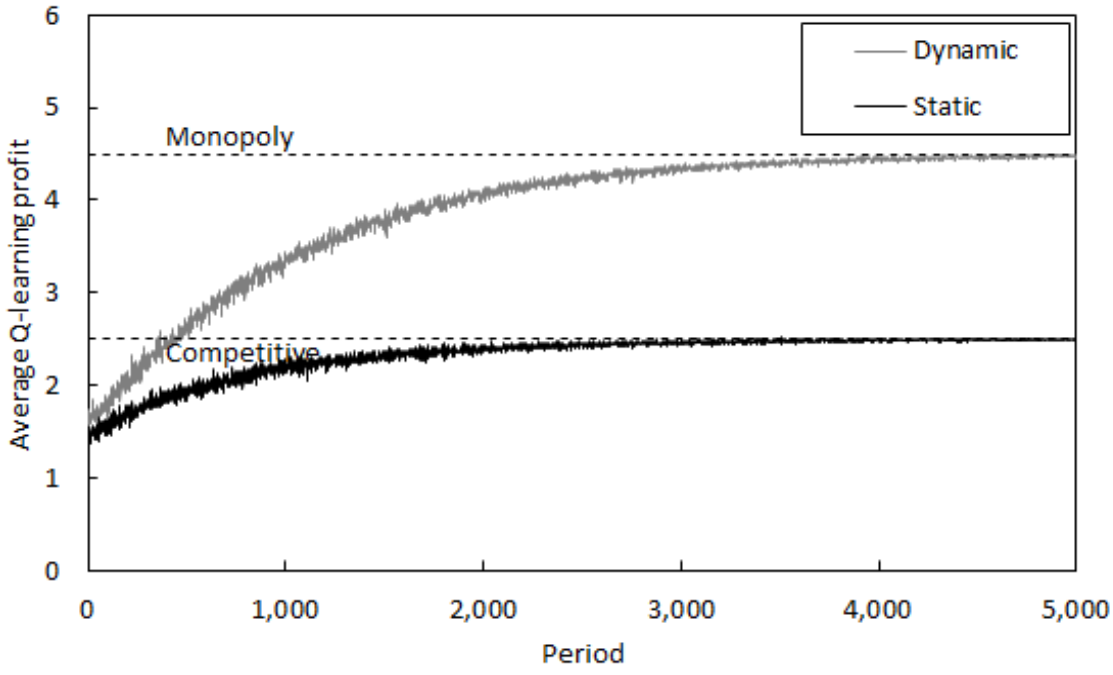
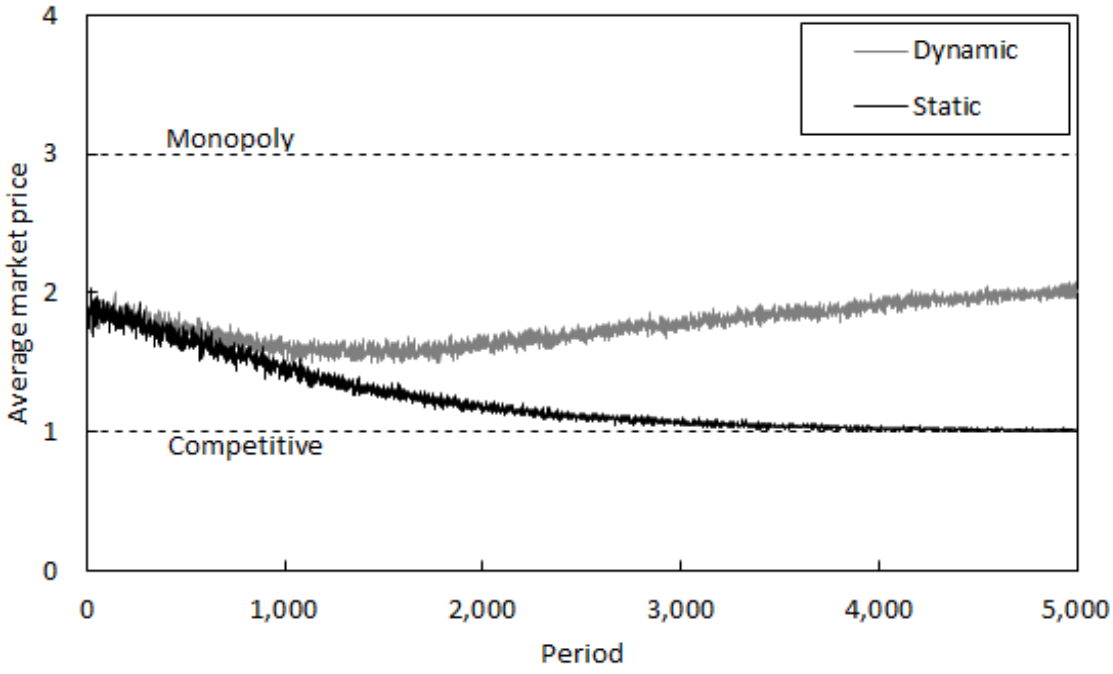


Figure 2: Q-learning versus fixed-strategy monopoly fixed price behavior



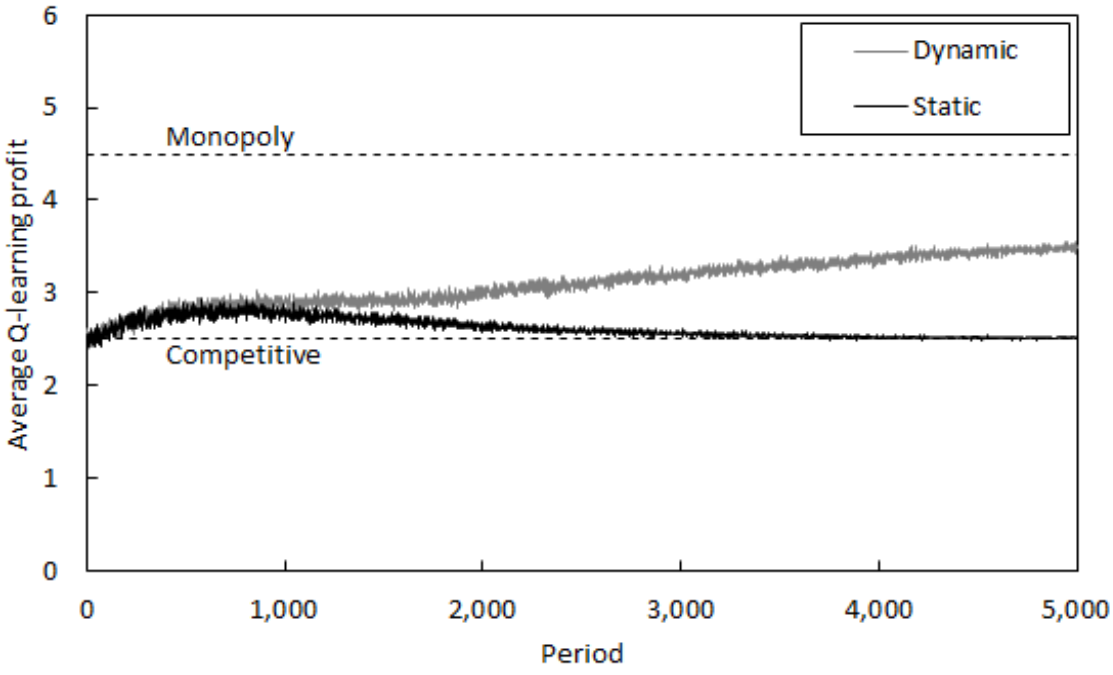


Figure 3: Q-learning versus Q-learning, $k = 6$

an average increment of 1.1, but increases occur with increments more than twice as much. The algorithms do not converge to perfect Edgeworth price cycles, which would involve price decreases of 1 and price increases of 4.

	$k = 6$	$k = 12$	$k = 100$
Average market price	1.9	4.4	41.7
Competitive level	1.0	1.0	1.0
Monopoly level	3.0	6.0	50.0
Average profit	3.5	14.5	1083
Competitive level	2.5	5.5	49.5
Monopoly level	4.5	18.0	1250
Runs with a fixed price	349/1,000	106/1,000	4/1,000
At monopoly price	116/1,000	23/1,000	0/1,000
Runs without a fixed price	651/1,000	894/1,000	996/1,000
Periods with a price decrease	40%	61%	74%
Average price decrease	−1.1	−1.3	−6.4
Periods with a price increase	20%	17%	13%
Average price increase	2.3	4.8	40.4
Average time until increase	4.0	4.9	7.5

Table 2: Market outcomes (top) and price dynamics (bottom), final 100 periods

Figure 4 shows that the Q-learning algorithms are similarly able to keep prices and profits above their static level even when extending the action set to $p_t^i \in \{0, 1, 2, ..., 12\}$, i.e. $k = 12$. To allow for sufficient state-action visits, runs are increased to 20,000 periods, with a learning decay of $\theta = 0.00025$ such that exploration again drops below 1% near the end. Average profits are now around 14.5. This is above the static level of 5.5 but below monopoly profits of 18. Table 2 shows that only in 106 runs the algorithms converge to a single, stable fixed price, 23 of which at the monopoly level of $p = 6$. When the action set is larger, the algorithm has increased difficulties to converge to the joint-profit maximizing monopoly price. In absence of a fixed price, the algorithms again clearly display an asymmetric pricing pattern: decreases in the market price occur almost four times as often as increases (61% versus 17%). Again, prices and profits would converge to the static level in case of simultaneous competition.

Finally, Figure 5 shows what happens when extending the action set to $p_t^i \in \{0, 1, 2, ..., 100\}$, i.e. $k = 100$. Runs are increased to 400,000 periods, with a learning decay of $\theta = 0.0000125$ such that exploration again drops below 1% near the end. Average profits are now around 1083, well above the static level of 49.5 but still below monopoly profits of 1250. Table 2 shows that only in 4 runs the algorithms

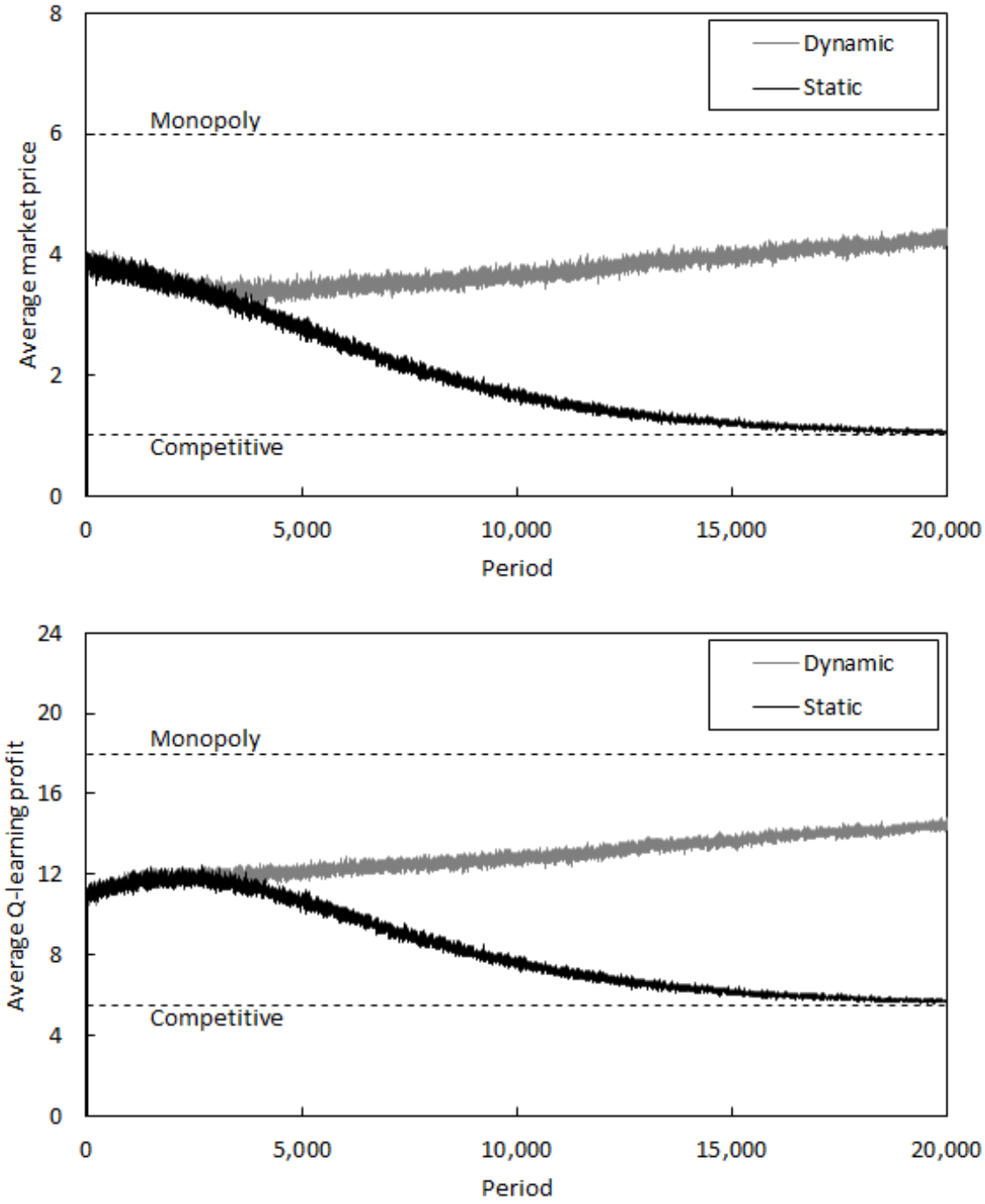


Figure 4: Q-learning versus Q-learning, $k = 12$

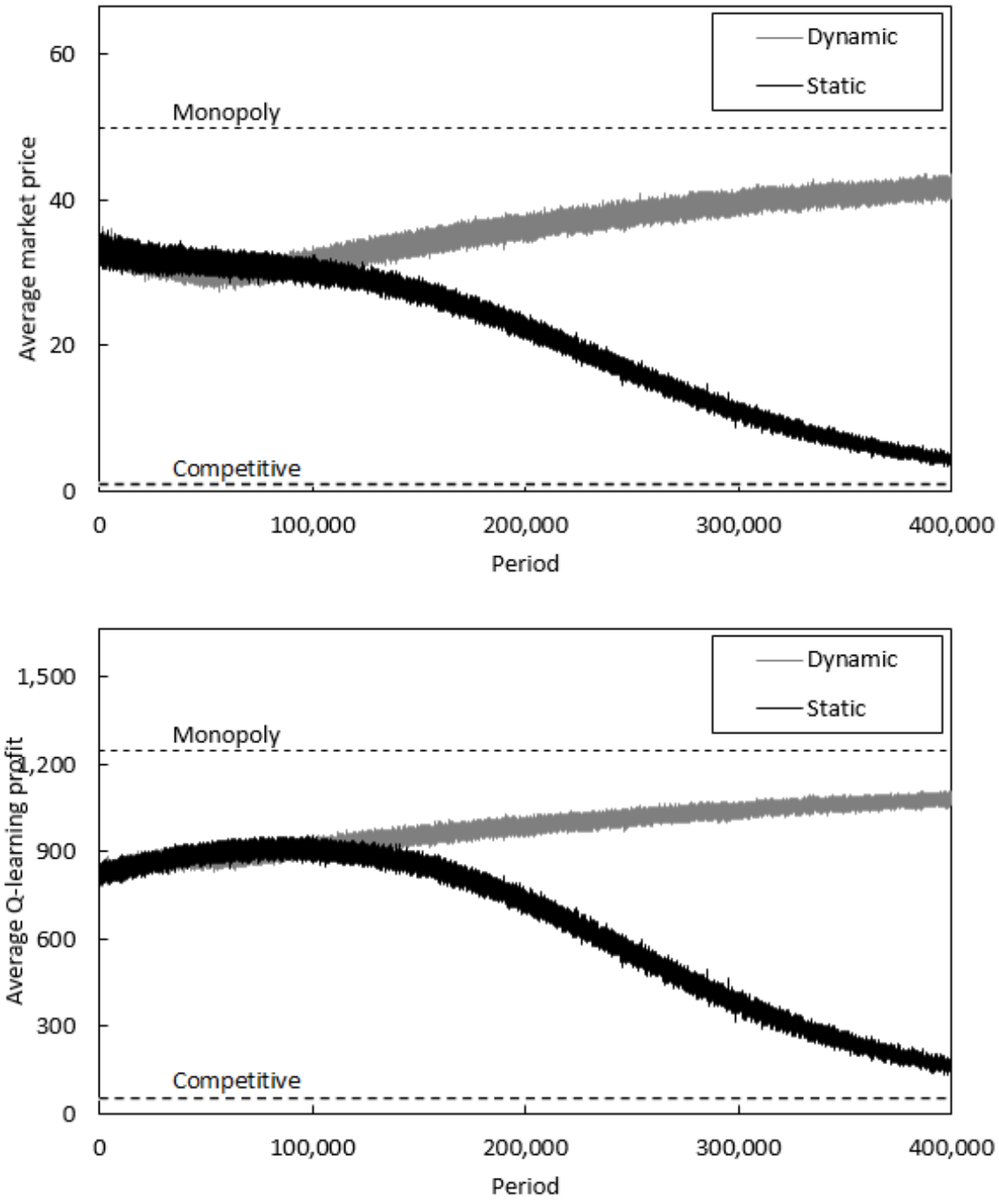


Figure 5: Q-learning versus Q-learning, $k = 100$

converge to a fixed price, which occurs at $p = \{32, 47, 53, 59\}$ (although this would be more when price destabilizations due to last-minute exploration are not taken into account). The algorithms again clearly display an asymmetric pricing pattern: decreases in the market price occur almost six times as often as increases (74% versus 13%) and the average time between market price increases is 7.5 periods. If a price decrease occurs, this happens with an average increment of 6.4, but when a price increase occurs the increment is around 40.4. Prices and profits would again converge to the static level in case of simultaneous competition.

4 Discussion and Extensions

This section provides a discussion on the appropriate collusive benchmark and valuable extensions to the learning algorithm and environment considered.

4.1 Appropriate Benchmark

Maskin and Tirole (1988, p. 592) argue that their theory "underscore(s) the relatively high profits that firms can earn when the discount factor is near 1" and that it therefore "can be viewed as a theory of tacit collusion". Harrington (2017) on the other hand characterizes collusion as a situation in which firms use a reward-punishment scheme to coordinate their behavior for the purpose of producing a supracompetitive outcome. As discussed in Section 2.1, both fixed price and Edgeworth price cycle MPEs are sustained by some threat of punishment. However, the question remains relative to what the outcome can be considered as "supracompetitive". In particular, when the static outcome of prices at or one increment above marginal cost is itself not an MPE (because of an absence of subgame perfection), is may not be reasonable to consider this as an appropriate competitive benchmark.

Even when an autonomous algorithm can be shown to outperform an appropriate competitive benchmark, a subsequent question remains whether it also outperforms humans. If humans can be shown to be (weakly) better at colluding that autonomous algorithms (as in Leufkens and Peeters (2011) for $k = 6$), the risk of autonomous algorithmic collusion would not add anything above and beyond any already existing risk of human collusion. The competitive edge of algorithms relative to humans would therefore have to be made explicit.

4.2 Extensions to the Learning Algorithm

Several valuable extensions to the learning algorithm itself could be developed. In particular, more advanced multi-agent reinforcement learning algorithms may be able to deal with the challenges that remain in guaranteeing convergence to rational and Pareto-optimal collusive behavior. However, key developments in multi-agent rein-

forcement learning still lack practical applicability to oligopoly environments. A discussion on such algorithms is provided in the appendix.

In our analysis, prices (and thus states and actions) are considered to be discrete. This allows for a tabular Q-function that matches a value to each unique state-action combination. Whenever the state-action set is limited this provides a convenient approach, but becomes intractable when this set becomes very large. Additionally, updates only occur in the exact state-action combination visited, while observed reward and opponent behavior may also be informative on neighboring state-action combinations. Function approximation (or differential games systems) can then be used, in which the reinforcement learning algorithm assumes a parametric model of the environment and observed rewards and state transitions provide updates of the parameters of the model (Schwartz, 2014; Sutton and Barto, 2018).

No domain knowledge or prior input is considered in the learning process above. However, previous experiences in comparable learning processes may contain valuable information to kick-start the new learning process. In such cases, transfer learning can be considered, where knowledge learned in one task domain is transferred to another, related domain (Pan and Yang, 2010). Similarly, human feedback through policy shaping may be used to provide outside guidance to a learning algorithm (Griffith *et al.*, 2013).

Finally, evolutionary game theory has recently been proposed as a framework for analyzing the learning dynamics in multi-agent learning (Tuyls *et al.*, 2006; Tuyls and Parsons, 2007; Bloembergen *et al.*, 2015). Evolutionary game theory concepts like replicator dynamics and evolutionary stable strategies allow for several novel and valuable ways to look at multi-agent learning. In particular, they can shed light into the black box of reinforcement learning by providing qualitative insights into its transient dynamics and subsequently guidance on parameter tuning and algorithm selection and development.

4.3 Extensions to the Environment

In addition to extensions to the learning algorithm, future research may also consider extensions to the environment considered. These may be aimed at making the environment less stylized or more case-specific.

To account for short-run price commitments, we have adopted the sequential pricing framework proposed by Maskin and Tirole (1988). Here, firms are exogenously restricted to respond sequentially. However, Maskin and Tirole also consider the case where firms face short-run price commitments but are not restricted to sequential behavior. They show that sequential pricing also occurs endogenously. Similar endogenization of sequential pricing can be considered in the case of reinforcement learning, where a restriction on the action set to the current price only occurs when the agent adopted a price change in the previous period.

Using a full-information environment and dynamic programming, Tesauro and

Kephart (2002) show that under independent Q-learning, the duration of the Edgeworth price cycles decreases and average prices and profits increase once products become more differentiated (either vertically or horizontally) or when consumers are less informed. It would be interesting to see to what degree these results are maintained when agents do not possess full information and have to learn while simultaneously interacting.

Throughout we have assumed that the environment itself remains stationary and agents are symmetric. The only non-stationarity that has been considered so far is opponent-induced non-stationarity. However, in oligopoly environments payoffs are rarely stationary and firms rarely symmetric. In particular, demand may fluctuate independently from firm behavior and marginal costs can be different and varying idiosyncratically. Robustness of multi-agent reinforcement learning algorithms applied to oligopoly environments would then also have to be evaluated in terms of these non-stationarities. For instance, firms may require some persistent degree of exploration in order to observe any changes in the environment or apply some recency-weighting to the observed state transitions and rewards. Additionally, it may be interesting to consider the case beyond two firms, possibly with entry and exit as well.

Finally, in the environment considered here, consumers are modelled as exogenous. Noel (2011) argues, however, than in the presence of Edgeworth price cycles, consumers may be better off when they are capable of shifting consumption to different periods. While a downwards sloping demand curve already accounts for the fact that more demand occurs if prices are lower and vice versa, it does not take into account any dynamic optimization – e.g. even higher demand during low prices if previous periods experienced high prices, especially if this is a recurrent pattern.

5 Concluding Remarks

Fully autonomous algorithmic collusion remains elusive. On the one hand, an intuitive interpretation of the capabilities of artificial intelligence may suggest that increasingly more sophisticated pricing algorithms will at some point, inevitably, learn to undermine competitive pressures and achieve higher profits – at the expense of consumers. Such an outcome would be akin to collusion, but without the overt act of communication currently necessary to establish a competition law infringement. On the other hand, it remains unclear exactly how such autonomous algorithms would work.

We show how in a stylized oligopoly environment with repeated sequential price competition, independent Q-learning algorithms are able to achieve higher-than-static prices and profits. It provides ground for competition authorities and regulators to remain vigilant when observing the rise of autonomous pricing algorithms in the market place, in particular in cases where firms may be short-run price committed. Additionally, the general framework used here may be used to similarly assess the capacity of other, perhaps more advanced algorithms to collude in various environments.

Finally, note that only a diagnostic tool is provided here. It suggests a way in

which the collusive capabilities of autonomous algorithms may be assessed. It does not prescribe any particular medicine in response to a positive diagnosis. In any case, banning autonomous algorithms altogether would be a very clear overreaction, as this would disregard any social gains algorithms may bring in for instance clearing markets, reducing costs or increasing competitive pressures (Oxera, 2017). Additionally, it may not be reasonable to condemn firms for any increased margins when this is merely the result of an intelligent unilateral adaptation to oligopolistic markets – as may arguably be the case with Edgeworth price cycles. The appropriate policy response therefore seems to involves a case-by-case investigation into the capacity of various types of algorithms to achieve supracompetitive profits in certain environments. The conclusion will then most likely be that only in specific cases regulation will be required to prevent autonomous robots from getting together and fixing prices.

References

[1] Abdallah, S. and Lesser, V. (2008) "A Multiagent Reinforcement Learning Algorithm with Non-Linear Dynamics", *Journal of Artificial Intelligence Research*, 33(1), pp. 521-549

[2] Awheda, M. and Schwartz, H.M. (2013) "Exponential Moving Average Q-Learning Algorithm", In: *Proceedings of the IEEE Symposium Series on Computational Intelligence*

[3] Albrecht, S.V. and Stone, P. (2018) "Autonomous Agents Modelling Other Agents: A Comprehensive Survey and Open Problems", arXiv:1709.08071v2

[4] Bloembergen, D., Tuyls, K., Hennes, D. and Kaisers, M. (2015) "Evolutionary Dynamics of Multi-Agent Learning: A Survey", *Journal of Artificial Intelligence Research*, 53, pp. 659-697

[5] Bowling, M. and Veloso, M. (2002) "Multiagent Learning Using a Variable Learning Rate", *Artificial Intelligence*, 136(2), pp. 215-250

[6] Busoniu, L., Babuska, R., and De Schutter, B. (2008) "A Comprehensive Survey of Multiagent Reinforcement Learning", *IEEE Transactions on Systems, Man, and Cybernetics*, Part C 38(2)

[7] Byrne, D.P. and de Roos N. (2018) "Learning to Collude: A Study in Retail Gasoline", working paper

[8] Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. (2018) Algorithmic Pricing and Collusion: What Implications for Competition Policy?, working paper

[9] Eckert, A. (2013) "Empirical Studies of Gasoline Retailing: A Guide to the Literature", *Journal of Economic Surveys*, 27, pp. 140-166

[10] Ezrachi, A. and Stucke, M.E. (2016) *Virtual Competition: The Promise and Perils of the Algorithm-Driven Economy*, Harvard University Press, Cambridge, Massachusetts

[11] Gal, M.S. (2018) "Algorithms as Illegal Agreements", *Berkeley Technology Law Journal,* forthcoming

[12] Greenwald, A. and Hall, K. (2003) "Correlated Q-Learning", In: *Proceedings of the 22nd Conference on Artificial Intelligences*, pp. 242-249

[13] Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L. and Thomaz, A. L. (2013) "Policy Shaping: Integrating Human Feedback with Reinforcement Learning", In: *Advances in Neural Information Processing Systems*, pp. 2625-2633

[14] Harrington, J.E. (2017) "Developing Competition Law for Collusion by Autonomous Price-Setting Agents", working paper

[15] Hernandez-Leal, P., Kaisers, M., Baarslag, T. and Munoz de Cote, E. (2017) "A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity", arXiv:1707.09183

[16] Hu, J. and Wellman, M.P. (2003) "Nash Q-Learning for General-Sum Stochastic Games", *Journal of Machine Learning Research*, 4, pp. 1039-1069

[17] Huck, S., Normann, H.T. and Oechssler, J. (2003) "Zero-Knowledge Cooperation in Dilemma Games", *Journal of Theoretical Biology*, 220, pp. 47-54

[18] Ittoo, A. and Petit, N. (2017) "Algorithmic Pricing Agents and Tacit Collusion: A Technological Perspective", working paper

[19] Izquierdo, S. S. and Izquierdo, L. R. (2015) "The "Win-Continue, Lose-Reverse" Rule in Cournot Oligopolies: Robustness of Collusive Outcomes", In: Amblard, F., Miguel, F.J., Blanchet, A. and Gaudou, B. (Eds) *Lecture Notes in Economics and Mathematical Systems,* Volume 676, Springer, Berlin, Heidelberg

[20] Könönen, V. (2003) "Asymmetric Multiagent Reinforcement Learning", In: *Proceedings IEEE/WIC International Conference on Intelligent Agent Technology*, pp. 336-342

[21] Kühn, K.U. and Tadelis, S. (2017) "Regulating the Internet Economy: Policy Issues and Economic Analysis", presentation prepared for CRESSE 2017

[22] Leufkens, K. and Peeters, R. (2011) "Price Dynamics and Collusion Under Short-Run Price Commitments", *International Journal of Industrial Organization*, 29, pp. 134-153

[23] Maskin, E. and Tirole, J. (1988) "A Theory of Dynamic Oligopoly II: Price Competition, Kinked Demand Curves and Edgeworth Cycles", *Econometrica*, 56(3), pp. 571-599

[24] Mehra, S. K. (2015) "Antitrust and the Robo-Seller: Competition in the Time of Algorithms", *Minnesota Law Review*, 100, pp. 1323-1375

[25] Noel, M.D. (2011) "Edgeworth Price Cycles", In: Palgrave Macmillan (Eds) *The New Palgrave Dictionary of Economics*, Palgrave Macmillan, London

[26] Oxera (2017) "When Algorithms Set Prices: Winners and Losers", Oxera Discussion Paper, June 2017

[27] Pan, S. J. and Yang, Q. (2010) "A Survey on transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, 22(10), pp. 1345-1359

[28] Petit, N. (2017) "Antitrust and Artificial Intelligence: A Research Agenda", *Journal of European Competition Law and Practice*, 8(6), p. 361

[29] RBB Economics (2018) "Automatic Harm to Competition? Pricing Algorithms and Coordination", *RBB Brief 55*, February 2018

[30] Salcedo, B. (2015) "Pricing Algorithms and Tacit Collusion", Manuscript, Pennsylvania State University

[31] Schwalbe, U. (2018) "Algorithms, Machine Learning, and Collusion", working paper

[32] Schwartz, H.M. (2014) *Multi-Agent Machine Learning: A Reinforcement Approach*, Wiley, Hoboken, New Jersey

[33] Singh, S., Kearns, M. and Mansour, Y. (2000) "Nash Convergence of Gradient Dynamics in General-Sum Games", In: *Uncertainty in Artificial Intelligence Proceedings*, pp. 541-548

[34] Sutton, R.S. and Barto, A.G. (2018) *Reinforcement Learning: An Introduction*, 2nd Edition, The MIT Press, Cambridge, Massachusetts

[35] Tesauro, G. (2003) "Extending Q-Learning to General Adaptive Multi-Agent Systems", In: *Advances in Neural Information Processing Systems*, pp. 871-878

[36] Tesauro, G. and Kephart, J.O. (2002) "Pricing in Agent Economics Using Multi-Agent Q-Learning", *Autonomous Agents and Multi-Agent Systems*, 5, pp. 289-304

[37] Tuyls, K., 't Hoen, P. J. and Vanschoenwinkel, B. (2006) "An Evolutionary Dynamical Analysis of Multi-Agent Learning in Iterated Games", *Autonomous Agents and Multi-Agent Systems*, 12(1), pp. 115-153

[38] Tuyls, K. and Parsons, S. (2007) "What Evolutionary Game Theory Tells Us About Multiagent Learning", *Artificial Intelligence*, 171(7), pp. 406-416

[39] Tuyls, K. and Weiss, G. (2012) "Multiagent Learning: Basics, Challenges, and Prospects", *AI Magazine*, 33(3), pp. 41-52

[40] Tsitsiklis, J.N. (1994) "Asynchronous Stochastic Approximation and Q-Learning", *Machine Learning*, 16(3), pp. 185-202

[41] Waltman, L. and Kaymak, U. (2008) "Q-Learning Agents in a Cournot Oligopoly Model", *Journal of Economic Dynamics & Control*, 32, pp. 3275-3293

[42] Watkins, C.J.C.H. (1989) *Learning from Delayed Rewards*, PhD Thesis, University of Cambridge, England

[43] Watkins, C.J.C.H. and Dayan, P. (1992) "Q-Learning", *Machine Learning*, 8(3), pp. 279–292

[44] Zhang, C. and Lesser, V. (2010) "Multi-Agent Learning with Policy Prediction", In: *Proceedings of the 24th National Conference on Artificial Intelligence*, pp. 746-752

[45] Zhou, N., Zhang, L., Li, S. and Wang, Z. (2018) "Algorithmic Collusion in Cournot Duopoly Market: Evidence from Experimental Economics", working paper

[46] Zinkevich, M. (2003) "Online Convex Programming and Generalized Infinitesimal Gradient Ascent", In: *Proceedings 20th International Conference on Machine Learning*, pp. 928-936

Appendix: Multi-Agent Reinforcement Learning

Any direct application of single-agent reinforcement learning algorithms to multi-agent environments can be problematic, because they do not account for any non-stationarity in the environment caused by the adaptation of other agents. Additionally, single-agent reinforcement learning learns deterministic strategies, while often mixing is required in response to a strategic opponent. And even if opponent-induced non-stationarity is taken into account and agents manage to converge to behavior which is a mutual best response (possibly involving mixed strategies), there is no guarantee that the achieved equilibrium is a Pareto-optimal equilibrium. Developments in multi-agent reinforcement learning are aimed at resolving the issue of opponent-induced non-stationarity and mixing strategies. This section discusses several key developments, but also shows why they still lack practical applicability to oligopoly environments. For a general introduction on multi-agent reinforcement learning see Tuyls and Weiss (2012) and for an overview of the literature see Busoniu *et al.* (2008), Hernandez-Leal *et al.* (2017) and Albrecht and Stone (2018) in particular.

Nash-Q Learning and Hyper-Q Learning

The main limitation of using independent Q-learning in multi-agent environments is that both exploration and adaptation by an opponent can have a major impact in the Q-value updates. Hu and Wellman (2003) propose Nash-Q learning as an extension of independent Q-learning to multi-agent environments. Under Nash-Q, agents maintain Q-functions over joint actions and perform updates based on assuming Nash equilibrium behavior over current Q-values. Specifically, each agent $i \in \{1, ..., n\}$ takes in state s an action a^i based on some probability distribution $\rho^i(\cdot|s)$. Take r^i as its subsequent reward. Q-value updates now occur following

$$Q^i\left(s, a^1, ..., a^n\right) \leftarrow (1-\alpha)\, Q^i\left(s, a^1, ..., a^n\right) + \alpha\left(r^i + \delta Nash Q^i\left(s'\right)\right), \tag{8}$$

where $NashQ^i(s')$ is the present discounted profit in a selected equilibrium given the currently learned Q-values. $NashQ^i(s)$ and $\rho^i(\cdot|s)$ are subsequently updated using quadratic programming. Extensions include Correlated-Q learning (Greenwald and Hall, 2003), which instead looks for a more general correlated equilibrium, and Asymmetric-Q learning (Könönen, 2003), which deals with leader-follower stage games.

Nash-Q is guaranteed to converge to a Nash equilibrium (given certain technical conditions), but suffers from several practical limitations. Firstly, it requires full observability of opponent rewards in order to update the Q-functions. For environments where this is not feasible (such as in oligopoly competition), an observable proxy of opponent rewards (profits) would have to be used. Final results will then depend on how closely this proxy relates to actual rewards. Secondly, Nash-Q requires an

appropriate search algorithm to obtain at each step the values for $NashQ^i(s)$, which is non-trivial and may lead to a slow learning process. Finally, in case multiple Nash equilibria exist, it remains unclear whether the equilibria identified in each step for each state are Pareto-optimal equilibria.

As an alternative, Tesauro (2003) propose Hyper-Q learning, which learns the Q-values associated with mixed instead of pure strategies and uses estimated opponent strategies as additional state variables – i.e.

$$Q^i\left(s,\hat{\rho}^{-i},\rho^i\right) \leftarrow (1-\alpha)\,Q^i\left(s,\hat{\rho}^{-i},\rho^i\right) + \alpha\left(r^i + \delta \max_{\rho} Q^i\left(s',\hat{\rho}^{-i'},\rho\right)\right), \qquad (9)$$

where $\hat{\rho}^{-i}$ are estimates of all the competitor probability distributions given each state – based on (for instance) Bayesian inference or exponential moving average estimation. In theory, Hyper-Q is able to deal both with non-stationary opponents and mixing strategies, while only having to observe joint actions and own rewards. However, maintaining tabular Q-functions requires discretization of the probability distributions, which would increase the size of the Q-function exponentially. Function approximation may then have be used to allow for continuous state and action spaces.

Gradient Ascent Algorithms

Under gradient ascent, the algorithm increases or decreases the probability of selecting an action based on some gradient: increase the probability of an action when it is expected to increase the sum of all present discounted future profits (positive gradient) and decrease otherwise (negative gradient).

Singh *et al.* (2000) first proposed infinitesimal gradient ascent (IGA) for the simple two-agents, two-actions stateless game – later generalized by Zinkevich (2003) as generalized infinitesimal gradient ascent (GIGA) for two-agent stateless games with more than two actions. Take α and β as the probabilities that the first out of the two actions is chosen by agent 1 and 2 respectively and $V^i(\alpha,\beta)$ as the associated present discounted future profits of firm $i \in \{1,2\}$. Probabilities are updated based on the gradients following

$$\alpha \leftarrow \alpha + \eta\frac{\partial V^1(\alpha,\beta)}{\partial \alpha} \quad \text{and} \quad \beta \leftarrow \beta + \eta\frac{\partial V^2(\alpha,\beta)}{\partial \beta}. \qquad (10)$$

Taking an infinitesimal stepsize $\eta \to 0$ when the amount of steps goes to infinity, competing algorithms will display a weak form of convergence: average rewards converge to Nash rewards, but strategies might still display endless recursive adaptation in case of a mixed-strategy Nash equilibrium. To achieve convergence in strategies as well, Bowling and Veloso (2002) suggest the win-or-learn-fast (WoLF) heuristic, in which the gradient stepsize is small (learn cautiously) when the agent is winning but large (learn quickly) when losing, where winning or losing is defined relative

to an equilibrium strategy. This heuristic stimulates convergence without giving up rationality.

The above gradient ascent algorithms require full information on current opponent strategies and in case of the WoLF heuristic also prior knowledge on existing equilibria. Additionally, the game is assumed stateless. Bowling and Veloso (2002) propose win-or-learn-fast policy hill climbing (WoLF-PHC) as a practical algorithm that can be applied in cases when agents do not possess such information and the environment may display different states. WoLF-PHC uses an exogenous learning rate instead of the actual gradient and an approximate notion of winning. Taking $\rho(a|s)$ as the strategy, capturing the probability action a is taken in state s, updates occur following

$$\begin{aligned} \rho(a|s) &\leftarrow \rho(a|s) + \begin{cases} \eta & \text{if } a = \arg\max_{a'} Q(s,a') \\ \frac{-\eta}{A-1} & \text{otherwise} \end{cases} \\ \text{where } \eta &= \begin{cases} \eta_w & \text{if } \sum_a \rho(a|s)\, Q(s,a) > \sum_a \bar{\rho}(a|s)\, Q(s,a) \\ \eta_l & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

and A is the size of the action set, $\eta_w < \eta_l$ and $\rho(\cdot|s)$ is restricted to a legal probability distribution. $\bar{\rho}(\cdot|s)$ is the probability distribution of the average strategy over time and updates of Q-function $Q(s,a)$ occur conventionally. Abdallah and Lesser (2008) propose weighted policy learner (WPL) as an extension that uses a continuous spectrum of learning rates and Zhang and Lesser (2010) propose policy gradient ascent with approximate policy prediction (PGA-APP), which uses an approximation of the opponent strategy and gradient to estimate its own gradient with respect to the opponent's forecasted (instead of current) strategy. Finally, Awheda and Schwartz (2013) propose a more straightforward exponential moving average Q-learning (EMA-Q) algorithm that is comparable to WoLF-PHC, WPL and PGA-APP but is claimed to converge in a wider variety of situations. Under EMA-Q, strategy updates occur following

$$\rho(a|s) \leftarrow \begin{cases} (1 - k\eta_w)\,\rho(a|s) + k\eta_w & \text{if } a = \arg\max_{a'} Q(s,a') \\ (1 - k\eta_l)\,\rho(a|s) + k\eta_l \frac{1}{A-1} & \text{otherwise} \end{cases} \tag{12}$$

where A is again the size of the action set, $\eta_w < \eta_l$ and k a constant gain – with $k\eta_l \in (0,1)$. $\rho(\cdot|s)$ is again restricted to a legal probability distribution.

The above gradient ascent algorithms have the main advantages that they can deal with opponent-induced non-stationarity, can learn continuous mixing strategies and do not require any model of the environment. In the application of for instance WoLF-PHC or EMA-Q to oligopoly environments, however, several practical problems arises: it may take a (very) long time for the algorithm to converge; it is not obvious how the exploration and learning rates and their decay should be set; and even if convergence to a (possibly mixed) equilibrium occurs, it is not obvious that this is a Pareto-optimal equilibrium.