TI 2018-056/VII
Tinbergen Institute Discussion Paper

Autonomous Algorithmic Collusion: Q-Learning Under Sequantial Pricing

Revision: January 2019

Timo Klein[1]

[1] University of Amsterdam

Autonomous Algorithmic Collusion: Q-Learning Under Sequantial Pricing*

Timo Klein†

January 2019

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 compete sequentially, independent Q-learning (a simple but well-established self-learning algorithm) learns to approximate profitable fixed price or asymmetric price cycle equilibria – although convergence to optimal collusive behavior is not guaranteed. The general framework used and limitations identified 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: L13, L41, D43, D83
Keywords: artificial intelligence, machine learning, reinforcement learning, Q-learning, pricing algorithms, algorithmic collusion, sequential pricing

*I am grateful for valuable comments and suggestions on earlier versions by Joe Harrington, Harold Houba, Michael Kaisers, Maarten Pieter Schinkel, Ulrich Schwalbe and Leonard Treuren, as well as audiences at Tinbergen Institute Jamboree 2018 in Amsterdam and ESWM 2018 in Naples. Errors remain my own.

†Tinbergen Institute and 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; 2017). 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, 2018; Gal, 2018).[1] The debate has received extensive press coverage[2] as well as increasing interest from competition authorities[3], 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. This has led several commentators to conclude that the debate is overblown. Substantially, the main critique is

[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 for instance speeches by EU Commissioner Vestager ("Algorithms and Competition", March 2017) and Acting FTC Chairman Ohlhausen ("Should We Fear The Things That Go Beep In the Night?", May 2017), as well as a recent report by the UK's Competition and Markets Authority (CMA) entitled "Pricing Algorithms: Economic working paper on the use of algorithms to facilitate collusion and personalised pricing" (October 2018).

[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).

that self-learning algorithms would be similarly ill-equipped as humans to coordinate on one out of many possible 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 already fails to occur in moderately complex oligopoly settings. Another critique is that there have not yet been any cases of truly autonomous algorithmic collusion.[6] This does not mean, however, that it cannot become feasible in the near future. 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 – the 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 sequential interaction. The results show that when firms behave sequentially, competing Q-learning algorithms learn to approximate profitable fixed price or Edgeworth price cycle equilibria. 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).

Using the same learning algorithm but in a simultaneous move environment, Calvano *et al.* (2018b) show how independent Q-learning is able to learn collusive strategies, provided it can condition its prices on its own pricing history. We differ from them in that we do not allow for such self-reactive conditioning. Instead, we do assume an environment of sequential move. This assumption is not uncontroversial, as it suggests a short-run price commitment that appears at odds with the fact that algorithms can adjust prices very quickly in response to each other. At the same time however, reinforcement learning algorithms are conventionally programmed in discrete time and it may not be obvious that competing algorithms are indeed updated simultaneously.

To capture the dynamics of sequential pricing we use the 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,

[6] One prominent cartel involving algorithms is the 2015 US case of Topkins, an online poster retailer that used algorithms to coordinate prices with competitors. This is a case, however, of humans using algorithms to execute their collusive agreement and not of algorithms learning autonomously to collude. Another illustration often referred to is "The Making of a Fly", a biology textbook sold on Amazon. In 2011, one seller used an algorithm that always priced 25% above its competitor, while its competitor used a price-matching algorithm. This of course caused prices to escalate (up to $23 million per copy). This is again, however, a case of humans using algorithms to execute a strategy and not any form of autonomous algorithmic collusion.

for instance, communication and the history of prices. Maskin and Tirole show that in their environment firms may charge higher prices and earn higher profits in equilibrium provided firms value future profits sufficiently high, which they interpret as tacit collusion (p. 592).

The learning algorithm applied is a staightforward adaptation of independent Q-learning to sequential interaction. Q-learning aims to maximize the present value of future rewards for environments with repeated choice. After choosing a price given current competitor price, it observes the realized profit and subsequent competitor response and updates recursively the expected optimal long-run profit from choosing the price it did. By interacting autonomously with its environment, it makes 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 starting to be applied in real-world dynamic pricing application, including airline fares and wholesale electricity markets (Ittoo and Petit, 2017).

Three theoretical challenges remain in guaranteeing convergence to optimal collusive behavior: independent Q-learning is restricted to pure strategies while in case of sequential pricing *mixed strategies* are required for subgame perfect equilibrium behavior; 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 profitability above static Nash 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.

Apart from Calvano *et al.* (2018b), there are only a few papers that look at algorithmic oligopoly collusion beyond an iterated 2-by-2 prisoner's dilemma.[7] Looking at quantity competition, Huck, Normann and Oechssler (2003) find that a "win-continue-lose-reverse" rule provides joint-profit maximizing convergence. Convergence is however not robust to small fluctuations in the payoff function (Izquierdo and Izquierdo, 2015). And Waltman and Kaymak (2008) show that independent Q-learning may collude on lower quantities, but these findings do not seem to be

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

based on equilibrium behavior (Calvano *et al.*, 2018b). 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 and rival learning, which allows for calculating optimal behavior using dynamic programming. 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, 2018) and may even be framed as explicit collusion by communicating your pricing strategies through decoding (Kühn and Tadelis, 2017; Schwalbe, 2018). Looking at the operations research and management science literature on revenue management algorithms, Cooper *et al.* (2015) show that the convention of estimating and optimizing monopoly models – ignoring in effect strategic considerations – may unknowingly lead to cooperative outcomes. Such model misspecifications are not based on equilibrium behavior, however.[8] Finally, there have recently been some interesting developments within so-called deep reinforcement learning that show the capacity to cooperate in several matrix games that model social dilemmas (Mnih *et al.*, 2015; Leibo *et al.*, 2017; Peysakhovich and Lerer, 2017). They are yet to show results in oligopoly environments however, in which firms have a larger action set and limited or no information on competitor profits.

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 tit-for-tat competitor and where two Q-learning algorithms are set to compete with each other. In both cases profits exceed their static level, but only in the first case joint-profit maximization is always achieved. Finally, Section 4 provides 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 sequential pricing we take the environment as described by Maskin and Tirole (1988). There are two identical firms $i \in \{1, 2\}$ with unrestrictive capacity. Sequentially, each firm sets an integer price $p_t^i \in P = \{0, 1, 2, ..., k\}$, where in

[8] See Den Boer (2015) for a survey of studies on dynamic pricing and learning within operations research and management science, as well as some related fields. These studies generally involve estimating profit or revenue as a stochastic function of prices and maximizing this, without autonomous exploration. An exception is Nambiar *et al.* (2018), which adds a randomness to the perceived optimal price in order to stimulate exploration and learning.

odd-numbered periods t firm 1 chooses its price while firm 2 keeps its price unchanged and vice versa in even-numbered periods. Instantaneous profit of firm i is derived as $\pi^i = q^i\left(p_t^i, p_t^j\right)\left(p_t^i - c^i\right)$, where $q^i\left(p_t^i, p_t^j\right)$ is the demand for firm i and c^i its marginal cost. Fixed costs are assumed zero and profits are strictly concave.

In our case, we look at the simple setting of homogeneous goods with linear demand. Taking a as the demand intercept and normalizing the slope to 1, demand is defined as

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

Finally, 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

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

In principle, firms only observe realized profit given their own price and the price of their competitor; demand and competitor profit functions may be unknown. We assume a stationary environment and identical marginal cost. A discussion on how for instance product differentiation, non-stationary or asymmetry may affect results is included in Section 4. Finally, we impose similarly as Maskin and Tirole 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.

One Nash equilibrium here is the static Nash outcome in which firms always price at or one increment above marginal cost. While this holds for any discount factor, more Nash equilibria exist for δ sufficiently large. In particular, Maskin and Tirole define the concept of a Markov perfect equilibrium (MPE), which is a subgame perfect equilibrium under the Markov assumption. They 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 marginal cost 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. Figure 1 illustrates this for the monopoly price for the case where $a = 6$ 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 shown in Figure 1.

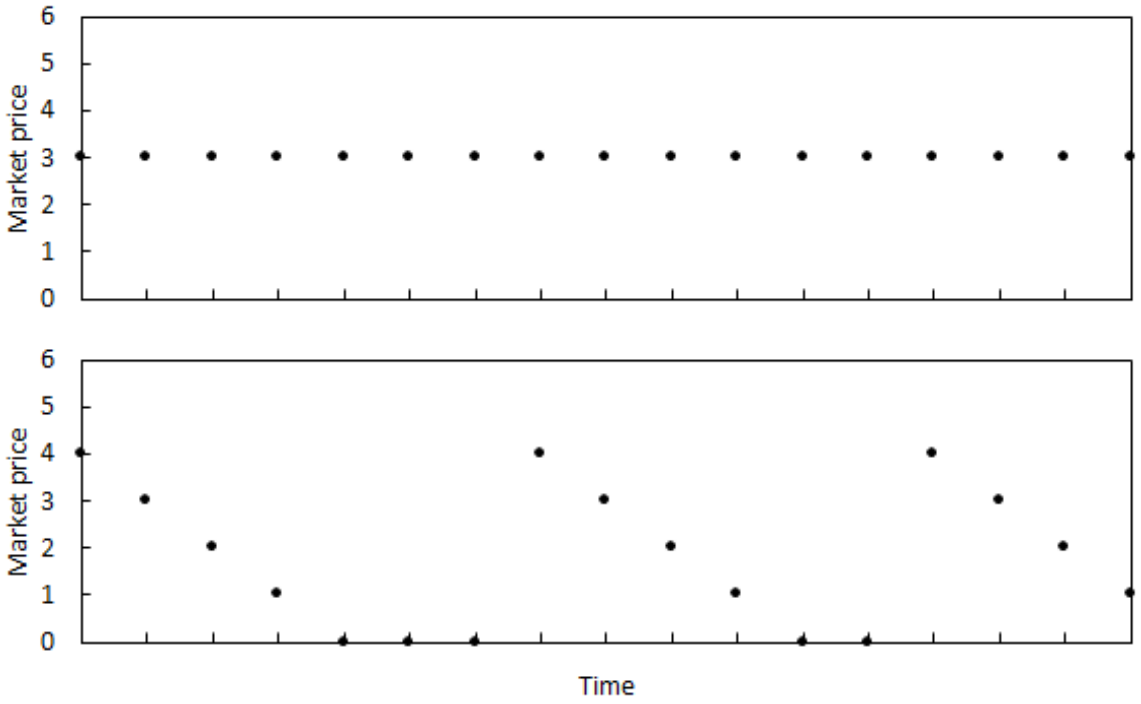


Figure 1: Price dynamics in fixed pricing (top) and Edgeworth cycles (bottom)

2.2 Algorithm: Sequential Independent Q-Learning

The learning algorithm applied is a straightforward adaptation of independent Q-learning to sequential interaction. Q-learning is a simple but well-established single-agent reinforcement learning model that aims to maximize the present value of future rewards for environments with repeated choice.[9] By interacting with its environment, the algorithm learns recursively a so-called Q-function that matches the optimal long-run value to setting any price given any competitor price. As such, the Q-function represents the "Quality" of choosing a particular action within a certain state. 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, followed by a note on its theoretical limitations and challenges in our context of oligopoly competition.

[9] Q-learning has originally been proposed by Watkins (1989) as a way to solve Markov decision processes – which is a discrete time stochastic process in which the action by the agent affects the determination of both current reward and the next state in an otherwise stationary environment. For a valuable primer on Q-learning, see Calvano *et al.* (2018b). For a comprehensive introduction on single-agent reinforcement learning more generally, see Sutton and Barto (2018).

2.2.1 Sequential Independent Q-Learning

Reinforcement learning algorithms consist of two interacting modules: a learning module that processes observed information and an action-selection module that uses this processed information to map states into actions.

Learning Module Q-function $Q^i(p^i|p^j)$ is a matrix that 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). After observing profits and competitor response, the algorithm updates the Q-function according to the following recursive relationship

$$Q^i(p^i|p^j) \leftarrow (1-\alpha)\, Q^i(p^i|p^j) + \alpha \left(\pi(p^i, p^j) + \delta \pi(p^i, p^{j'}) + \delta^2 \max_p Q^i(p|p^{j'}) \right), \quad (3)$$

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). $\delta \in [0,1)$ is again the discount factor.[10] The Q-function is initiated as an empty matrix, although a prior may also be used to potentially speed up learning.

Note that each time only one cell within the Q-function matrix is updated. Such tabular learning leads to a mechanical and slow learning process. To speed up learning, function approximations and deep learning may additionally be used. These would however increase the complexity as well as the amount of learning parameters to be set and are left for future research.

Action-Selection Module In balancing exploration and exploitation, the algorithm adopts a probabilistic action-selection policy. We use a straightfoward procedure called ε-greedy exploration: with probability $\varepsilon_t \in [0,1]$ it selects any price randomly (exploration) and with probability $1-\varepsilon_t$ it selects the currently perceived optimal price (exploitation), so

$$p_t^i \begin{cases} \sim U\{P\} & \text{with probability } \varepsilon_t \\ = \arg\max_p Q^i(p|p_{t-1}^j) & \text{with probability } 1-\varepsilon_t \end{cases} \quad (4)$$

where $U\{P\}$ is a discrete uniform distribution over the action set P. In case of ties under exploitation, the algorithm randomizes over all optimal actions. The probability of exploration is in our case determined as

$$\varepsilon_t = \varepsilon_0 (1-\theta)^t, \quad (5)$$

[10] In case of very short periods, the discount factor would be very close to 1. In this case, however, sufficient learning may fail to occur because old Q-value estimates will get too much weight. It may then be required to set parameter δ lower than the actual discount factor.

where $\varepsilon_0 \in [0,1]$ is the initial exploration probability and $\theta \in [0,1]$ a decay parameter that regulates the degree to which the algorithm gradually explores less and exploits more as it learns.[11]

A pseudocode of the sequential learning algorithm is provided below.[12]

Pseudocode Sequential Independent Q-Learning

Set demand function and learning and exploration parameters α, δ, ε_0 and θ
Initialize Q^1 and Q^2 as empty matrices
Initialize p_t^i for $t = \{1,2\}$ and $i = \{1,2\}$ randomly
Initialize $t = 3$, $i = 1$ and $j = 2$
Loop over each period
  Set price $p_t^j = p_{t-1}^j$
  Set price p_t^i according to (4)
  Update $Q^j(p_{t-1}^j | p_{t-1}^i)$ according to (3)
  Update $t \leftarrow t+1$ and $\{i \leftarrow j, j \leftarrow i\}$
Until $t = T$ (specified number of periods)

When a single Q-learning agent faces a fixed-strategy competitor, it 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 $\sum_{t=0}^{\infty} \alpha \rightarrow \infty$ and asymptotically all relevant action-state pairs $\left\{p_t^i, p_{t-1}^j\right\}$ 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 and exploration occurs sufficiently often.

2.2.2 Theoretical Limitations and Challenges

The independent Q-learning algorithm cannot convergence to optimal collusive behavior in our environment because of three remaining theoretical limitations and challenges. Firstly, our independent Q-learning algorithm is restricted to pure strategy

[11] 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

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

with $\tau_t > 0$ as a so-called temperature parameter. When $\tau \rightarrow \infty$, 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.

[12] Our algorithm is equivalent to the one used by Calvano *et al.* (2018b) apart from three things: in our case action-selection and updating occurs sequentially instead of simultaneously, the state variable is defined as only the current competitor price instead of the history of all prices over some period and the learning decay function has a different specification.

learning, while in our environment *mixed strategies* are required for subgame perfect equilibrium behavior. This means that off equilibrium at least one agent is always strictly 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 strategy. 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, including MPE with 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 more profitable than static Nash.

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 Simulation Results

For the empirical exercise, we take an initial exploration probability $\varepsilon_0 = 1$ and stepsize parameter $\alpha = 0.5$. Firms can set a price $p_t^i \in P = \{0, 1, 2, ..., k\}$. To allow for a sufficiently wide scope of prices we set $k = a$, so equal to the demand intercept. We also extend the analysis to $a = 12$ and $a = 100$, to see the effect of increasing the set of possible prices. Marginal costs c is kept at 0.

To assess the performance of the Q-learning algorithm, we simulate 1,000 runs, each lasting 5,000, 20,000 or 400,000 periods depending on whether k is set at 6, 12 or 100.[13] We take as learning decay parameter θ respectively 0.001, 0.00025 and 0.0000125 such that the probability of exploration drops below 1% towards the end for each k considered. At 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 when firms have a high discount factor. We also provide comments on how simultaneous competition would affect results.

We first consider the case where the Q-learning algorithm faces a fixed-strategy tit-for-tat opponent that sets the monopoly price if the Q-learning does as well, but undercuts otherwise. Secondly, we consider the case where two Q-learning algorithms are set to face each other.

[13] Simulations are programmed in MATLAB®.

3.1 Q-Learning Versus Fixed-Strategy Tit-For-Tat

We find that Q-learning neatly converges to the monopoly price when faced with the fixed-strategy competitor with monopoly fixed price equilibrium behavior. The fixed-strategy competitor behaves according to the monopoly MPE as identified by Maskin and Tirole: it sets the monopoly price if the Q-learning does, but undercuts with one increment otherwise. If prices are at marginal cost it mixes between maintaining this price and returning to the monopoly price.

The black curves in Figure 2 show convergence in case of static optimization, providing a stable market price of 1 and average profits of 2.5 – in line with static Nash.[14] 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 adheres to the monopoly price of 3, providing a constant joint-profit maximizing profit of 4.5. While not shown here, equivalent results are found when extending the action set to $k = 12$ or $k = 100$.

This result is not very surprising. Convergence to optimality is theoretically 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. If firms would compete simultaneously, a fixed-strategy competitor would therefore be equivalently able to "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 subgame perfect. 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 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 only on asymmetric price cycles when k is high.

Figure 3 shows for $a = 6$ and $p_t^i \in P = \{0, 1, 2, ..., 6\}$ that the Q-learning algorithms learn to maintain prices and profits that are on average higher than their static level. Average profits are around 3.5, which is above static but below monopoly level. While not shown here, prices and profits would have converged to their static levels if firms compete simultaneously, for any discount factor.

Table 1 shows the type 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 $p = 3$. If the

[14] 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 does not.

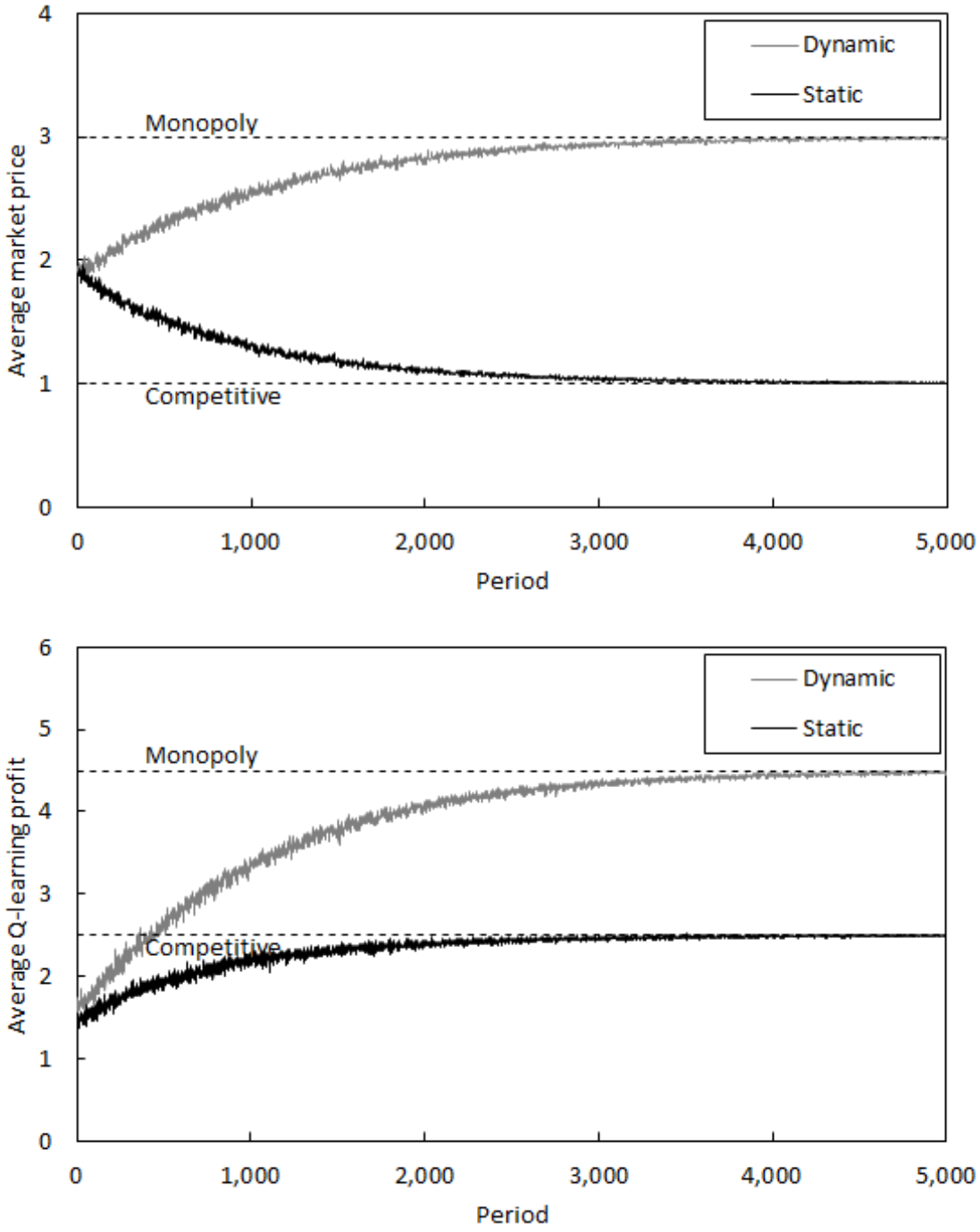


Figure 2: Q-learning versus fixed-strategy monopoly fixed price behavior

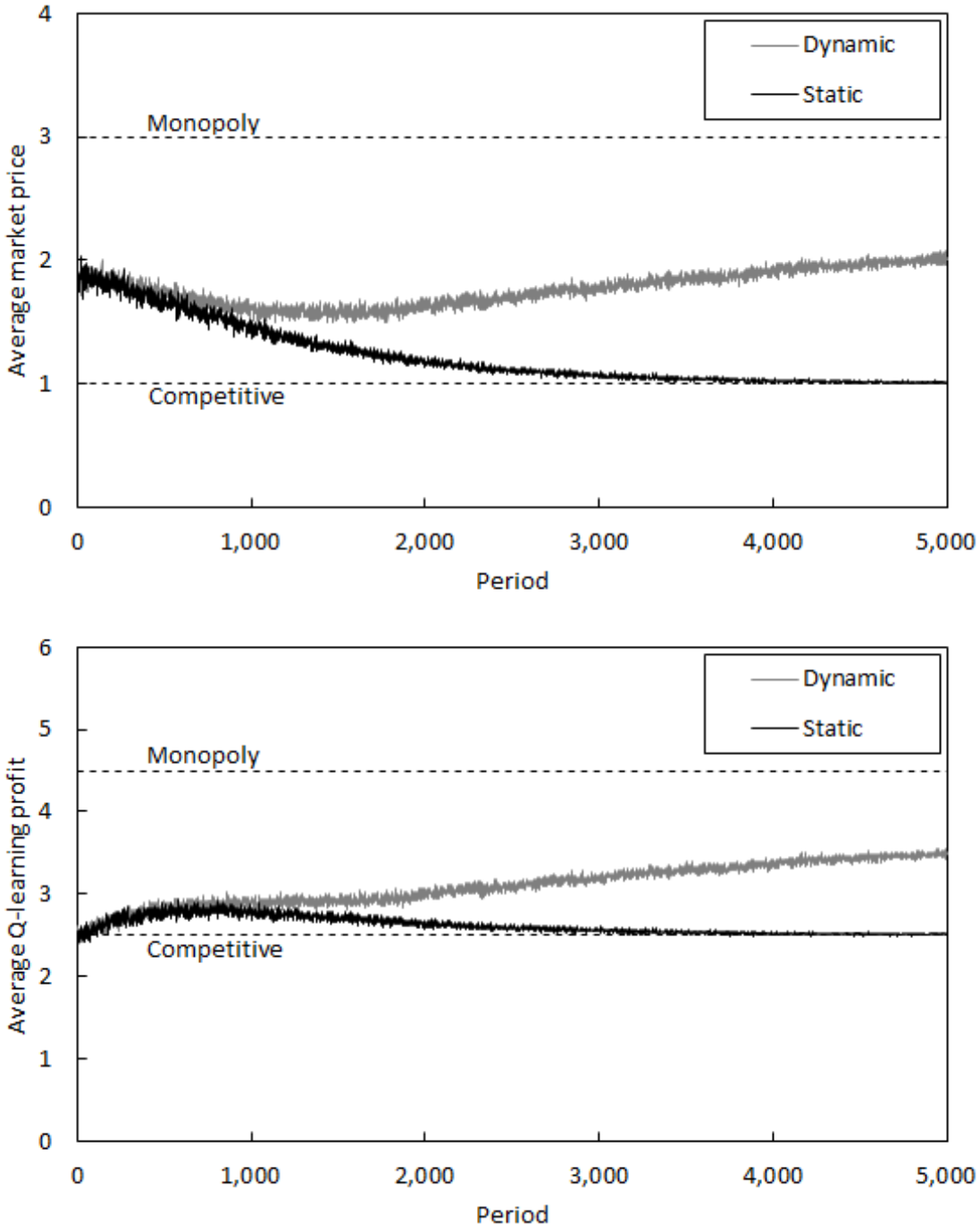


Figure 3: Q-learning versus Q-learning, $k = 6$

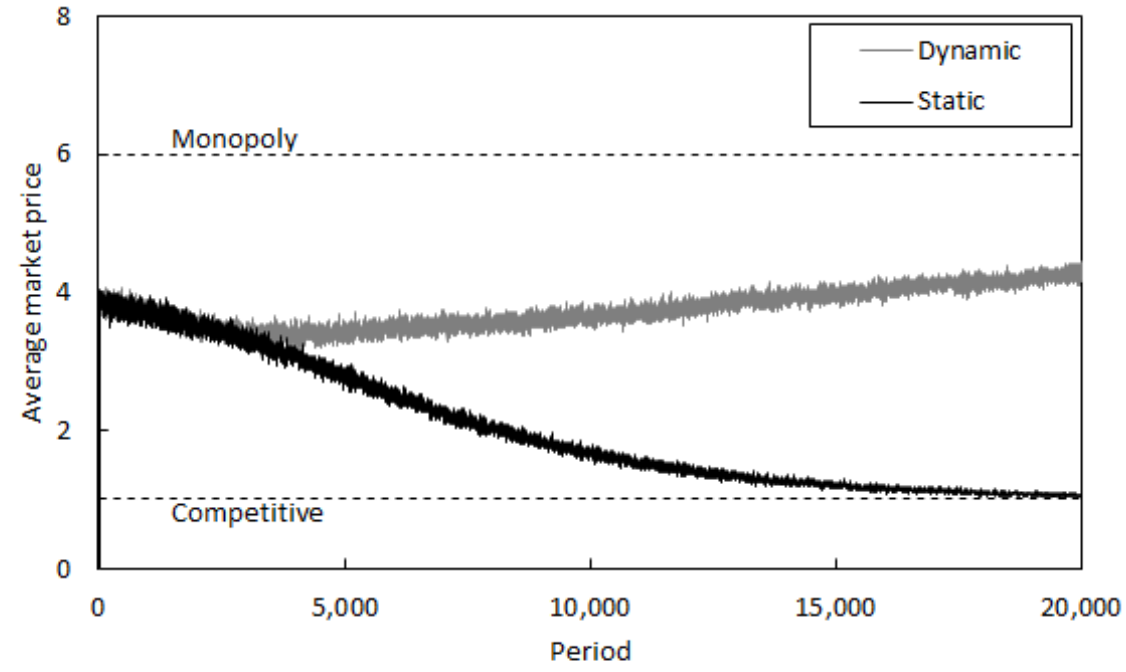
algorithms do not converge to a fixed price, they clearly display asymmetric pricing: decreases in the market price occur twice as often as increases and the average time between market price increases is 4.0 periods. If a decrease occurs, this happens with an average increment of 1.1, but increases 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	1,083
Competitive level	2.5	5.5	49.5
Monopoly level	4.5	18.0	1,250
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 1: 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 $p_t^i \in P = \{0, 1, 2, ..., 12\}$, with $a = 12$. Average profits are now around 14.5. This is again above static but below monopoly level. Table 1 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%).

Finally, Figure 5 shows what happens when extending the action set to $p_t^i \in P = \{0, 1, 2, ..., 100\}$, with $a = 100$. Average profits are now around 1,083, well above the static level of 49.5 but still below monopoly profits of 1,250. Table 1 shows that only in 4 runs the algorithms 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 now display



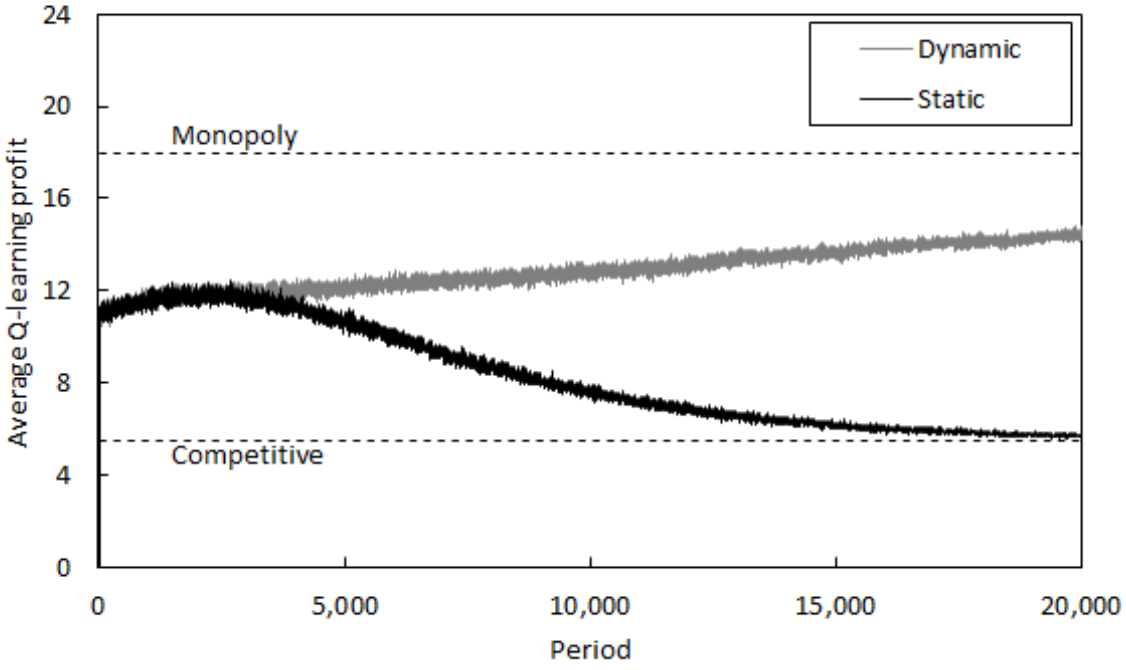


Figure 4: Q-learning versus Q-learning, $k = 12$

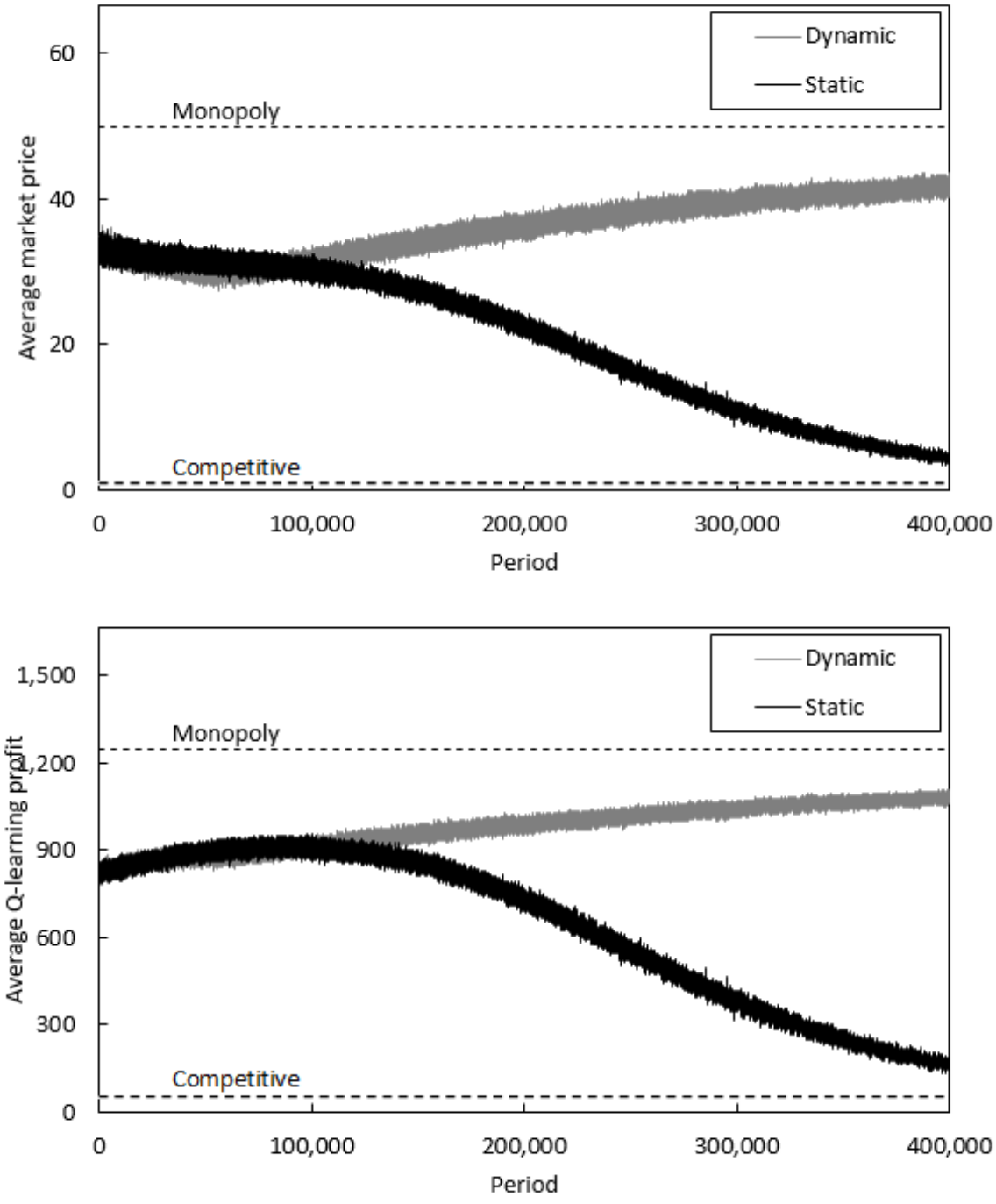


Figure 5: Q-learning versus Q-learning, $k = 100$

even more clearly 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. Figure 6 illustrates how the average pricing pattern looks like – based on the average market price, the average increase and decrease and the average time between increases. In case of simultaneous competition, prices would have converged to their static levels.

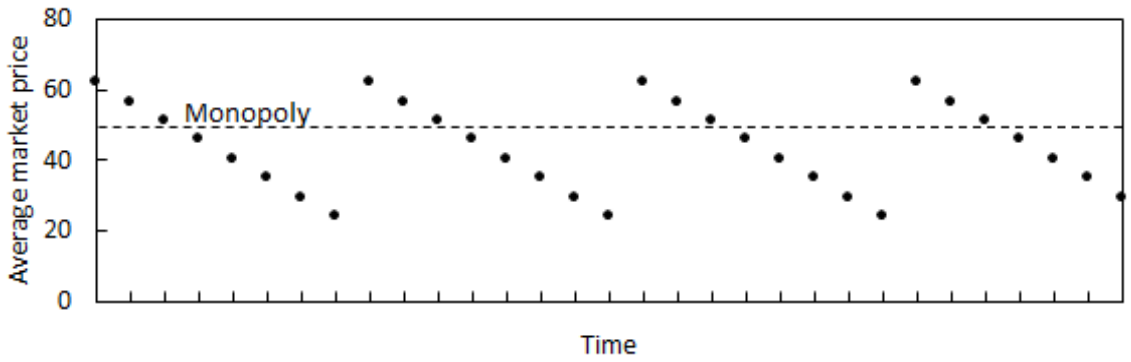


Figure 6: Average pattern over all 1,000 runs, during final 100 periods ($k = 100$)

3.3 Robustness

Our results already show that independent Q-learning can learn to price above static levels in case of sequential competition. Outcomes resemble theoretical equilibrium behavior, although more advanced algorithms appear necessary to ensure convergence to mutually optimal behavior. It would still be interesting to consider the robustness of our results when changing some of the underlying assumptions, such as the degree of product differentiation, more firms and different demand specifications, as well as possible non-stationarities in demand or marginal cost and asymmetries between firms. Additionally, we have taken fixed learning and exploration parameters. It may be valuable to develop more explicitly the effect of parameters α, δ and θ, as well as other settings.

4 Discussion and Extensions

We show how independent Q-learning learns to price above static levels in a homogeneous good sequential pricing duopoly with linear demand. This section provides a discussion on the appropriate benchmark to which to assess these results and possible 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 (2018) 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. In some way, both fixed price and Edgeworth price cycle MPEs are sustained by a threat of punishment, in the sense that rival response following deviation leads to lower profits. 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 than autonomous algorithms, the risk of autonomous algorithmic collusion may not add anything above and beyond any already existing risk of human collusion. For instance, 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 where $k = 6$ and $c = 0$ they find that under a random ending rule, 13 out of 15 human pairs end up coordinating on the joint-profit maximizing fixed price of $p = 3$. The problem with comparing algorithmic performance with the experimental economics literature, however, is that experiments with human subjects cannot replicate the speed at which algorithms can make decisions.

4.2 Extensions to the Algorithm

Several valuable extensions to the learning algorithm itself could be developed. In particular, deep reinforcement learning techniques may be used to speed up learning and deal with more complex environments (Mnih *et al.*, 2015; Leibo *et al.*, 2017; Peysakhovich and Lerer, 2017). Additionally, more advanced multi-agent reinforcement learning (or joint learning) may be able to deal with the theoretical challenges that remain in guaranteeing convergence to optimal collusive behavior. However, key developments in multi-agent reinforcement 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 for looking 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.

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. Additionally, time-varying capacity constraints such as inventory management may play a relevant role. 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.

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

On the one hand, an intuitive interpretation 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 prices and profits above their static level. While convergence to optimal behavior is not guaranteed, outcomes resemble equilibrium behavior. It is therefore unlikely that more advanced algorithms would instead lead to competitive behavior. This provides ground for competition authorities and regulators to remain vigilant when observing the rise of autonomous pricing algorithms in the market place. Additionally, the general framework may be used to similarly assess the capacity of other algorithms to collude – perhaps identifying unique properties that allow for such behavior. This research shows however that it is not only the properties of the algorithm that matter, but also the environment in which the algorithm is deployed – in this case whether competition occurs simultaneously or sequentially.

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", *American Economic Review*, forthcoming

[8] Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. (2018a) "Algorithmic Pricing and Collusion: What Implications for Competition Policy?", *Review of Industrial Organization*, forthcoming

[9] Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. (2018b) "Artificial Intelligence, Algorithmic Pricing and Collusion", CEPR discussion paper 13405

[10] Cooper, W.L., Homem-de-Mello, T., and Kleywegt, A.J. (2015) "Learning and Pricing with Models that do not Explicitly Incorporate Competition", *Operations Research*, 63(1), pp. 86-103

[11] Den Boer, A.V. (2015) "Dynamic Pricing and Learning: Historical Origins, Current Research, and New Directions", *Surveys in Operations Research and Management Science*, 20(1), pp. 1-18

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

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

[14] Ezrachi, A. and Stucke, M.E. (2017) "Artificial Intelligence & Collusion: When Computers Inhibit Competition", *University of Illinois Law Review*, p. 1775

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

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

[17] 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

[18] Harrington, J.E. (2018) "Developing Competition Law for Collusion by Autonomous Price-Setting Agents", *Journal of Competition Law and Economics,* forthcoming

[19] 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

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

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

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

[23] 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

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

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

[26] Leibo, J.Z., Zambaldi, V., Lanctot, M., Marecki, J., and Graepel, T. (2017) "Multi-Agent Reinforcement Learning in Sequential Social Dilemmas", In: *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*, pp. 464-473

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

[28] 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

[29] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015) "Human-Level Control Through Deep Reinforcement Learning", *Nature*, 518(7540), pp. 529-533

[30] Nambiar, M., Simchi-Levi, D. and Wang H. (2018) "Dynamic Learning and Pricing with Model Misspecification", working paper

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

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

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

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

[35] Peysakhovich, A. and Lerer, A. (2017) "Maintaining Cooperation in Complex Social Dilemmas Using Deep Reinforcement Learning", arXiv:1707.01068

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

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

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

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

[40] 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

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

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

[43] 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

[44] 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

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

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

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

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

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

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

[51] 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

[52] 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 more profitable. Developments in multi-agent reinforcement learning (also referred to as joint learning) are aimed at resolving the issue of opponent-induced non-stationarity and mixing strategies. This section discusses several prominent 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).

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

$$Q^i\left(s, a^1, ..., a^n\right) \leftarrow (1-\alpha)\, Q^i\left(s, a^1, ..., a^n\right) + \alpha\left(r^i + \delta Nash Q^i\left(s'\right)\right), \tag{6}$$

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.

$$Q^i\left(s,\hat{\rho}^{-i},\rho^i\right) \leftarrow (1-\alpha)\,Q^i\left(s,\hat{\rho}^{-i},\rho^i\right) + \alpha\left(r^i + \delta \max_{\rho} Q^i\left(s',\hat{\rho}^{-i'},\rho\right)\right), \quad (7)$$

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

$$\alpha \leftarrow \alpha + \eta\frac{\partial V^1(\alpha,\beta)}{\partial \alpha} \quad \text{and} \quad \beta \leftarrow \beta + \eta\frac{\partial V^2(\alpha,\beta)}{\partial \beta}. \quad (8)$$

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

$$\begin{aligned} \rho(a|s) &\leftarrow \rho(a|s) + \begin{cases} \eta & \text{if } a = \arg\max_{a'} Q(s,a') \\ \frac{-\eta}{A-1} & \text{otherwise} \end{cases} \\ \text{where } \eta &= \begin{cases} \eta_w & \text{if } \sum_a \rho(a|s)\, Q(s,a) > \sum_a \bar{\rho}(a|s)\, Q(s,a) \\ \eta_l & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

and A is the size of the action set, $\eta_w < \eta_l$ and $\rho(\cdot|s)$ is restricted to a 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

$$\rho(a|s) \leftarrow \begin{cases} (1 - k\eta_w)\,\rho(a|s) + k\eta_w & \text{if } a = \arg\max_{a'} Q(s,a') \\ (1 - k\eta_l)\,\rho(a|s) + k\eta_l \frac{1}{A-1} & \text{otherwise} \end{cases} \tag{10}$$

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 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.