# Can Decentralized Q-learning learn to collude?

**Janusz M. Meylahn**
Department of Applied Mathematics
University of Twente
Enschede, 7522 NB
j.m.meylahn@utwente.nl

## Abstract

The possibility of algorithmic collusion between pricing algorithms and the necessary antitrust legislation to regulate against it are hotly debated among academics and policymakers. However, none of the algorithms shown to collude have theoretical convergence guarantees and no theoretical framework exists for characterizing an algorithm's likelihood to collude. In this article, we summarize recent work which provides tools for quantifying the likelihood of collusion for a provably convergent algorithm and applies the results to two simple pricing environments.

## 1 Introduction

As reinforcement learning (RL) algorithms are increasingly used for real world applications, situations in which multiple algorithms learn in the same environment will become more prevalent. Algorithms should thus be designed to solve the inherent pathologies of multiagent reinforcement learning (MARL) [25]. The information that is shared among the algorithms can be used to categorize algorithms of this type. In the decentralized setting, almost no information is shared between the algorithm. Provably convergent algorithms in this setting are particularly difficult to design, but recent work has made some progress. In the class of zero-sum Markov games, convergence can be proved for a two-timescale Q-learning algorithm by [29], for a new algorithm called V-learning by [14] and for an Actor-Critic type algorithm in [26]. Markov potential games also allow for convergent decentralized MARL algorithms, as shown in [16, 18]. The case of convergence in continuous-space Markov games is tackled by a quantized Q-learning algorithm in [2]. Convergence to coarse correlated equilibria has been shown, for example, in general-sum Markov games by [17]. In this paper, we focus on the Decentralized Q-learning (DQ) algorithm of [3], which converges to a Nash equilibrium in the class of weakly acyclic games. The convergence proof relies on showing that DQ approximates a modification of the usual best-response dynamics called the best-response strategy adjustment process (BRSAP) arbitrarily closely.

Previous work has focussed on the convergence of MARL algorithms to an equilibrium, but in many cases there are *multiple* equilibria. The outcomes in the different equilibria may be drastically different in terms of the payoff the agents receive. An example of this is algorithmic collusion, which will be discussed below. To improve performance, it then becomes necessary to analyze the likelihood of observing different outcomes, with the objective of designing algorithms that select preferable outcomes with high probability.

The case of algorithmic collusion between pricing algorithms is particularly pressing in this regard. Recent simulation-based research by [10, 9] has concluded that basic RL algorithms can lead to supracompetitive prices. In addition, classical dynamic pricing algorithms can provably learn supracompetitive prices under self-play [21, 15]. The occurrence of algorithmic collusion in the real world has been strongly suggested by empirical studies of gas prices [4] and multifamily rental markets by [8]. This is a cause for concern for competition authorities, as current legislation in, for example, the United States and Europe may be insufficient for prosecuting companies employing such

17th European Workshop on Reinforcement Learning (EWRL 2024).

algorithms (see, for example, the discussion in [24]). Understanding when and how algorithms can learn to collude is essential for designing effective policy and designing efficient detection techniques.

The RL simulation studies showing that algorithms can learn to price supracompetitively use algorithms without convergence guarantees in multi-agent environments. The results may thus be due to insufficient learning [12]. Of the algorithms with convergence guarantees, only those convergent in weakly acyclic games such as in [3] or algorithms convergent in potential games such as in [16] are relevant here, as pricing environments are not zero-sum. We show that DQ converges in the simplest pricing environment making collusion possible.

Collusion[1] is already possible when the algorithms select one of two prices: the competitive or the collusive price. The resulting payoffs corresponds to the prisoner's dilemma for most demand models. In more realistic pricing environments, the algorithms will have to select prices from more than two prices. We investigate if the addition of prices increases or decreases the likelihood of collusion. To do this, we consider a generalized prisoner's dilemma, in which we add an intermediate price between the competitive and collusive prices.

Adding a third action to the prisoner's dilemma may affect the level of cooperation. A third irrelevant (because dominated) action can, for example, increase the chance of reaching sustainable cooperative outcomes, as shown in [30]. Similarly, the option of opting out of playing the prisoner's dilemma may also lead to an increase in cooperation [7]. The payoffs' ordering in [30] and [7], however, differs from the ordering we will consider here. A model with the same payoff structure as our, which leads to an increase of cooperation, is studied in [28].

In this article, we summarize the results of [20] and [19] with a focus on the relevance of the results for algorithmic collusion. We show that DQ is provably convergent in two simple pricing setting in which algorithmic collusion could be observed: the iterated prisoner's dilemma with a memory of one period and a generalization thereof with three actions. By characterizing the likelihood of observing different equilibria in these games when using DQ, we can conclude that DQ learns to collude with positive probability under conditions favorable for collusion, but does not exhibit a significant level of collusion as measured by the likelihood of learning collusive strategy equilibria under self-play. For the sake of brevity, many details (including proofs) have been omitted, but these can be found in [20, 19]. The text of the article is largely based on that of the aforementioned articles.

## 2 Setting

We will consider two-player finite discounted stochastic games. The strategies available to both players are taken from a common (finite) strategy space $\Pi$, and the best-response (BR) for each of the players given their opponent's strategy is unique. We will denote the BR of player $i$ by $\text{BR}_i : \Pi \to \Pi$. Our results will focus on the symmetric game setting where $\text{BR}_1(\cdot) = \text{BR}_2(\cdot)$, allowing us to drop the subscript and refer to the best-response function as $\text{BR}(\cdot)$. The function $\text{BR}(\cdot)$ induces a functional relation on the strategy space $\Pi$, which we will call the individual best-response (IBR) graph. The particular structure of functional relations (see Theorem 2.6) allows us to obtain our theoretical results. To define functional relations, we start with the definition of relations in Definition 2.1.

**Definition 2.1** (Relations). *Relations have the following primitives:*

1. *A set $V$ of elements called "vertices".*
2. *A set $E$ of elements called "edges".*
3. *A function $f$ whose domain is $E$ and whose range is contained in $V$.*
4. *A function $s$ whose domain is $E$ and whose range is contained in $V$.*

*and the following axioms:*

1. *The set $V$ is finite and not empty.*
2. *The set $E$ is finite.*

[1] A formal, agreed upon definition of algorithmic collusion is still lacking. A mathematical definition is proposed in [11] and an informal definition is proposed in [1]. For our purposes, we will define strategy equilibria as being collusive if they lead to stable supracompetitive prices (see Definition 4.1).

3. *No two distinct edges are parallel.*

Let $v_i v_j$ represent the edge from vertex $v_i$ to vertex $v_j$. This allows us to define paths and semipaths in Definition 2.2.

**Definition 2.2** (Path and Semipath)**.** *A (directed) path from $v_1$ to $v_n$ is a collection of distinct vertices, $v_1, v_2, \ldots, v_n$, together with the edges $v_1v_2, v_2v_3, \ldots, v_{n-1}v_n$. A semipath joining $v_1$ and $v_n$ is a collection of distinct vertices, $v_1, v_2, \ldots, v_n$ together with $n-1$ edges, one from each pair of edges, $v_iv_{i+1}$ or $v_{i+1}v_i$ for $i \in \{1, 2 \ldots, n-1\}$.*

A semipath between two vertices thus exists if and only if they are connected in the graph, where we replace all directed edges of the relation by undirected edges. We can use the definitions of paths and semipaths to define various notions of connectedness for relations in Definition 2.3.

**Definition 2.3** (Connectedness)**.** *A relation $R$ is strongly connected, if every two vertices are mutually reachable; $R$ is unilaterally connected, if for any two vertices at least one is reachable from the other. We say that $R$ is weakly connected, if every two vertices are joined by a semipath.*

The class of games we will consider has unique best-response functions, which motivates the definition of *functional* relations in Definition 2.4.

**Definition 2.4** (Functional relation)**.** *A functional relation is a relation in which every vertex has outdegree one. A relation $R$ is functional if and only if each of its weakly connected components is functional.*

**Definition 2.5** (IBR graphs)**.** *Given a symmetric, two-player game with unique BR function, the IBR graph consists of nodes representing all possible strategies $\pi \in \Pi$ and directed edges from each $\pi$ to its best-response $BR(\pi)$.*

The structure of functional relations is characterized by Theorem 2.6, which is adapted to relations from [13, Theorem 12.2].

**Theorem 2.6** (Structure of functional relations)**.** *The following statements are equivalent for a weakly connected relation $R$.*

1. *$R$ is functional.*
2. *$R$ has exactly one cycle or self-loop, and after deleting its edge(s), each weak component of the resulting relation consists of a tree toward a vertex*

*A relation $R$ is functional if and only if each of its weakly connected components is functional.*

Given a strategy profile $(\pi_1, \pi_2)$, where $\pi_1, \pi_2 \in \Pi$, the IBR dictates the strategy profiles reached when exactly one of the players plays a best-response to the current strategy of their opponent. This results in $(\text{BR}(\pi_1), \pi_2)$ and $(\pi_1, \text{BR}(\pi_1))$. Note that these strategy profiles do not depend on the initial strategy of the player playing a best-response. The BR graph is constructed by connecting every strategy profile with the profiles in which one player best-responds, leading to a directed graph on $\Pi \times \Pi$. Due to the assumption of uniqueness for the BR function, this will be a graph where each vertex has an out-degree of two.

Nash equilibrium are strategy profiles in which no player can improve their expected future discounted payoff by unilaterally changing strategy. In the BR graph, these appear as vertices with two self-loops.

**Definition 2.7** (Nash equilibrium)**.** *A Nash equilibrium is given by a strategy profile*

$$(\pi_1, \pi_2) \textit{ such that } \pi_1 = BR(\pi_2) \textit{ and } \pi_2 = BR(\pi_1). \tag{1}$$

For the DQ algorithm to converge, the BR graph must be weakly acyclic.

**Definition 2.8** (Weak acyclicity)**.** *A BR graph is weakly acyclic if there exists a finite directed path from any strategy profile to a Nash equilibrium.*

We are interested in understanding the dynamics of DQ of [3] in such a setting. Thus, we assume that the players do not know the payoffs of the game or the strategy employed by their opponent. The goal of the players is to learn a strategy that maximizes their expected discounted sum of future rewards

$$\mathbb{E}\Big[\sum_{t=1}^{\infty} \delta^t r_t^i\Big], \tag{2}$$

where $r_t^i$ is the reward received by the $i^{\text{th}}$ agent in the $t^{\text{th}}$ round of the game and $\delta$ is a common discount factor. They do this using the DQ as in Algorithm 1. The version given here is a slight variation of the original in [3].

**Algorithm 1** Decentralized Q-learning

```
Given discount factor $\delta$, exploration rate $\epsilon$, batch size $K$, state space $\mathcal{S}$, action space $\mathcal{A}$ and inertia $\lambda$
begin
    Initialize $q_{\text{act}}(s,a) = q_{\text{env}}(s,a)$ randomly for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.
    Set $\pi_{\text{act}}(a|s)$ as $\epsilon$-greedy strategy from $q_{\text{act}}(s,a)$.
    Set $\pi_{\text{env}}(a|s)$ as $\epsilon$-greedy strategy from $q_{\text{env}}(s,a)$.
    Observe initial state $s$.
    repeat
        for $t = 1$ to $K$ do
            Execute action $a$ using $\pi_{\text{act}}(a|s)$;
            Observe reward $r$ and next state $s'$;
            Set $n(s,a) \leftarrow n(s,a) + 1$;
            Set $\tilde{\alpha} \leftarrow \frac{1}{n(s,a)+1}$;
            Set $q_{\text{env}}(s,a) \leftarrow (1-\tilde{\alpha})q_{\text{env}}(s,a) + \tilde{\alpha}\Big[r + \delta \sum_b \pi_{\text{env}}(b|s')q_{\text{env}}(s',b)\Big]$;
            Set $\pi_{\text{env}}(a|s)$ as $\epsilon$-greedy strategy from $q_{\text{env}}(s,a)$;
            Set $s \leftarrow s'$;
        end
        Generate random variable $u$ from $U[0,1]$;
        foreach $\hat{s}, \hat{a}$ do
            if $u > \lambda$ then
                Set $q_{\text{act}}(\hat{s},\hat{a}) \leftarrow q_{\text{env}}(\hat{s},\hat{a})$;
            end
            Set $\pi(\hat{a}|\hat{s})$ as $\epsilon$-greedy strategy from $q_{\text{act}}(\hat{s},\hat{a})$;
            Set $q_{\text{env}}(\hat{s},\hat{a}) \leftarrow q_{\text{act}}(\hat{s},\hat{a})$;
        end
    until done;
end
```

# 3 Theoretical results

IBR graphs of symmetric games with unique best-responses consist of weak relations, each containing a single cycle (from Theorem 2.6), referred to as the "eye" (following [27]). An IBR graph thus contains at least one eye and a maximum of $|\Pi|$ eyes. Besides the eye, the weak relations consist of trees connected to one of the vertices in the eye (from Theorem 2.6).

The IBR graphs contain all the information contained in the BR graphs. The Nash equilibria, which correspond to vertices with self-loops in the BR graphs, correspond to two possible structures in the IBR graphs: self-loops and 2-cycles. If the pair of vertices in the IBR graph corresponding to a vertex in the BR graph is given by the same vertex twice and this vertex has a self-loop, then both of the outgoing edges in the BR graph will also be self-loops. We refer to such equilibria as symmetric. Similarly, if the pair of vertices in the IBR graph have outgoing edges pointing to each other, i.e., they form a 2-cycle, then the pair corresponds to two different vertices in the BR graph which both have self-loops in the BR graph. Such equilibria are referred to as asymmetric. This leads us to our first result, Lemma 3.1.

**Lemma 3.1** (Nash equilibria in IBR graphs)**.** *Nash equilibria correspond to eyes in the IBR graph that are either self-loops or 2-cycles.*

The existence of inescapable cycle in the BR graph corresponds with the existence of cycle of length three or more in the IBR graph. We will refer to such eyes as "bad eyes".

**Definition 3.2** (Bad eye)**.** *For IBR graphs, a bad eye consists of a cycle of length three or more.*

**Theorem 3.3** (IBR graph conditions for weak acyclicity)**.** *A symmetric game with unique best-response function is weakly acyclic if and only if its IBR graph contains no bad eyes.*

We may be interested not only in whether a specific game is weakly acyclic, but if a class of games is weakly acyclic. As an example, consider the iterated prisoner's dilemma in which players employ strategies with a memory of one period and exploration as discussed in Section 4. Here, the structure of the IBR graph may depend on the combination of the parameters $T, R, P, S$ as well as $\epsilon$ and $\delta$. In the normalized case without exploration, i.e., with $T = 1, S = 0$ and $\epsilon = 0$, the parameter space is split into 12 regions each with its own IBR graph (see [22]). Including the exploration rate increases the number of regions significantly.

To check whether the game is weakly acyclic for all allowed parameter values, would require checking whether the game is weakly acyclic in all possible regions. Such a tedious calculation can be avoided by examining the graph of all possible edges of the IBR graph. Each of the vertices in the IBR graph has a unique BR, which may vary depending on the combination of environmental and algorithmic parameters. Calculating such a list of best-responses and the corresponding existence conditions can be automated following [22]. This leads to a list of all the edges that are possible in the IBR graph.

**Definition 3.4** (Maximal IBR graph)**.** *For a given class of symmetric games, the maximal IBR graph is the graph containing all edges that are possible in the IBR graphs of the class.*

The Maximal IBR graph is no longer a functional relation. Nevertheless, we can use it to determine sufficient conditions for the class of symmetric games under consideration being weakly acyclic.

**Corollary 3.5** (Sufficient conditions for weak acyclicity)**.** *If the Maximal IBR graph contains no cycles of length three or more, the corresponding class of games is weakly acyclic.*

Taking the limit of $K \to \infty$ of the DQ algorithm given in Algorithm 1, results in the BRSAP introduced in [3] and defined as follows.

**Definition 3.6** (BRSAP)**.** *At time $t = 0$, the players choose a strategy uniformly at random from the strategy space. Subsequently, they may change their strategy at discrete time points $t \in \{1, 2, \ldots\}$. We endow the players with an additional parameter $\lambda \in (0, 1)$, which we call inertia. At each point in time, a player plays the BR to their opponent's previous strategy with probability $1 - \lambda$ and continues playing their previous strategy with probability $\lambda$.*

If the BR graph of the game is weakly acyclic, the BRSAP must converge to a Nash equilibria. But how likely is the process is to end in the different equilibria when multiple equilibria are possible? To answer this, we construct the BRSAP graph. This is a weighted directed graph, with the edge weights corresponding to the likelihood of observing the represented transition.

**Definition 3.7** (BRSAP graph)**.** *To construct the BRSAP graph, weigh all edges in the BR graph by $\lambda(1 - \lambda)$, add self-loops to all vertices in the BR graph with weight $\lambda^2$, and add edges with weight $(1 - \lambda)^2$ between all strategy profiles $(\pi_i, \pi_j)$ and $(BR(\pi_j), BR(\pi_i))$ for $i, j \in \{1, \ldots, |\Pi|\}$.*

To quantify the likelihood of ending in an equilibrium, we introduce the concept of a stochastic basin of attraction (SBA). The BRSAP is an absorbing Markov Chain when the BR graph is weakly acyclic.

**Definition 3.8** (Stochastic Basin of Attraction)**.** *We define the SBA for a given Nash equilibrium of the BRSAP as the probability corresponding to that equilibrium in the stationary distribution of the process.*

Note that we will identify the SBA of asymmetric strategy equilibria with the likelihood of ending in either of the two asymmetric equilibria.

Using the transition matrix of the BRSAP graph, we can calculate the stationary distribution as the eigenvector corresponding to eigenvalue 1. The eigenvector will contain nonzero entries at the positions corresponding to the equilibria and these entries tell us the likelihood of learning that equilibrium.

**Theorem 3.9** (Relating BRSAP to IBR graph)**.** *Assume that the BR graph is weakly acyclic. The stationary distribution on the equilibria of the BRSAP is the same as the stationary distribution on the eyes of a random walk on the IBR graph with uniform initial conditions.*

Theorem 3.9 allows us to calculate the stationary distribution on the absorbing states using the IBR graph. The structure of the IBR graph consists of disconnected components, each consisting of a weakly connected functional relation. This means that the stationary distribution on the eye of a particular component is given by the number of vertices in that component divided by the total number of vertices. From Theorem 3.9, we can also conclude that the stationary distribution is independent of the inertia parameter, $\lambda$.

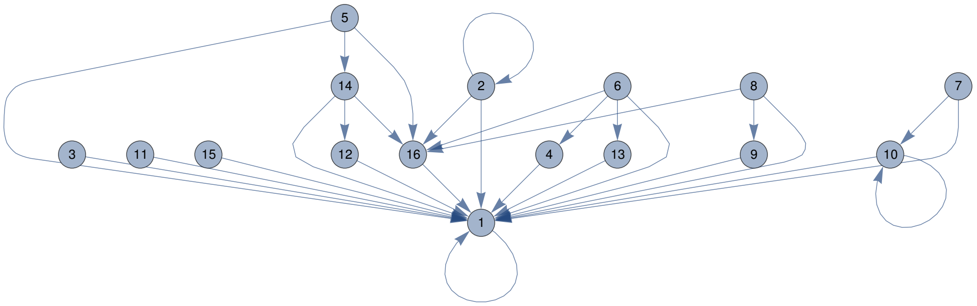

Figure 1: The maximal IBR graph of the iterated prisoner's dilemma with a memory of one period and $\epsilon$-greedy strategies. The strategies are enumerated from 1 to 16 with the AD, GT and WSLS strategies being given by nodes 1, 2 and 10 respectively.

# 4 Applications

## 4.1 Prisoner's dilemma

The simplest pricing environment in which collusion may be observed corresponds to the iterated prisoner's dilemma, with a memory of one period. The actions are either to cooperate (C) or to defect (D), where cooperation corresponds to charging the monopoly price and defection corresponds to charging the competitive price. The payoffs are given below.

| **1 \ 2** | D | C |
|---|---|---|
| D | $(P, P)$ | $(T, S)$ |
| C | $(S, T)$ | $(R, R)$ |

Here $T > R > P > S$. To reduce the number of parameters and without loss of generality, we will consider the normalized versions of this game, where we set $T = 1$ and $S = 0$.

Some of the most famous strategies played in the iterated prisoner's dilemma rely on conditioning the choice of action on the actions played in the previous round, as studied in [5]. An example of this is the win-stay, loose-shift (WSLS) strategy (see [23]), in which players cooperate if both players played the same action in the previous round and defect otherwise. For this reason, we allow the players to choose their one-period memory strategy from the set of 16 possible $\epsilon$-greedy strategies of this kind, i.e., $|\Pi| = 16$.[2]

Note that there are combinations of the parameters $T, R, P, S, \delta$ and $\epsilon$ that lead to a game in which the best-response function is not unique. This occurs at the critical conditions in the parameter space, and therefore only for a set of parameter values with measure zero. We will therefore exclude such parameter values in the following analysis.

Using Theorem 3.3, we find that the normalized iterated prisoner's dilemma with a memory of one period is weakly acyclic for all $R, P, \delta, \epsilon \in (0, 1)$ with $R > P$. The maximal IBR (Figure 1) shows that three Nash equilibria are possible in this setting, namely, both players playing all-defect (AD), grim trigger (GT) or win-stay, lose-shift (WSLS). Using Theorem 3.9, we find tight upper and lower bounds on the SBAs for all three Nash equilibria across all parameter values $R, P, \delta, \epsilon \in (0, 1)$ with $R > P$ as shown in Table 1. In Figure 2 we show two IBR graphs that realize all upper and lower bounds of Table 1.

## 4.2 Generalized prisoner's dilemma

Inspired by the pricing setting, we consider a generalized prisoner's dilemma with three actions. We will base the payoff structure on the prices and revenues of a logit demand model. This choice is

[2] We assume that the initial action is chosen uniformly at random by all players.

| **Equilibrium** | Maximum SBA | Minimum SBA |
|---|---|---|
| AD | 1 | 0.8125 |
| GT | 0.0625 | 0 |
| WSLS | 0.125 | 0 |

Table 1: Tight upper and lower bounds on the SBAs for the three Nash equilibria possible in the iterated prisoner's dilemma with a one period memory and $\epsilon$-greedy exploration under the BRSAP.

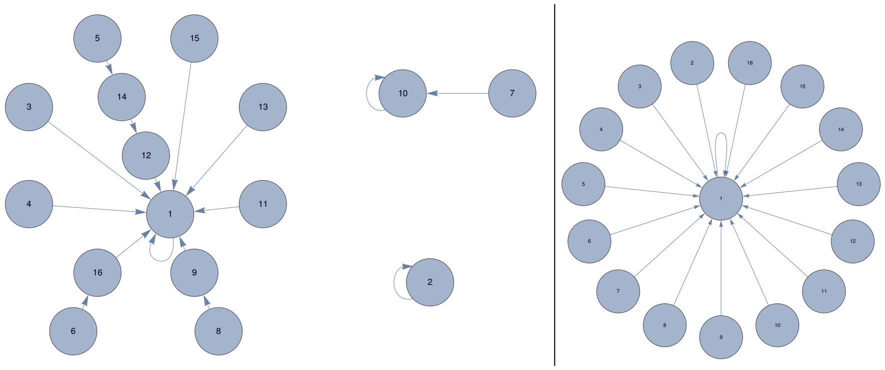

Figure 2: Examples of IBR graphs for the PD with $T = 1, R = 0.9, P = 0.3, S = 0, \epsilon = 0.1, \delta = 0.9$ on the left and $T = 1, R = 0.9, P = 0.3, S = 0, \epsilon = 0.1, \delta = 0.1$ on the right.

motivated by the recent application of Q-learning in the pricing setting [10]. The realized demand of player $i$, is

$$d_i(p_i, p_{-i}) = \frac{\mathrm{e}^{\frac{a_i + p_i}{\mu}}}{1 + \mathrm{e}^{\frac{a_1 + p_1}{\mu}} + \mathrm{e}^{\frac{a_2 + p_2}{\mu}}}. \tag{3}$$

Here $p_i \in \{p_L, p_M, p_H\}$ is the price chosen by player $i$ and $p_{-i} \in \{p_L, p_M, p_H\}$ is the price chosen by player $-i = \{1, 2\} \backslash i$. $a_1$, $a_2$ and $\mu$ are model parameters, where $a_1$ and $a_2$ capture vertical differentiation and $\mu$ captures horizontal differentiation. Given a price pair $(p_1, p_2)$, player $i$ receives reward

$$r_i(p_i, p_{-i}) = d_i(p_i, p_{-i})(p_i - c_i), \tag{4}$$

where $c_i \geq 0$ is the marginal cost. The three possible prices we will consider are the competitive price, the collusive price and the average of the competitive and collusive price, which we will refer to as $p_L$, $p_H$, and $p_M$ respectively. There are thus nine possible price pairs. To illustrate the payoff structure, we consider the case where $a_1 = a_2 = 1$, $\mu = 4$ and $c_1 = c_2 = 0$. This leads to the payoffs given in Table 2.

| **1 \ 2** | $p_L$ | $p_M$ | $p_H$ |
|---|---|---|---|
| $p_L$ | (1.06336, 1.06336) | (1.09345, 1.05559) | (1.12102, 1.03431) |
| $p_M$ | (1.05559, 1.09345) | (1.08634, 1.08634) | (1.11455, 1.06522) |
| $p_H$ | (1.03431, 1.12102) | (1.06522, 1.11455) | (1.09361, 1.09361) |

Table 2: Payoffs for each of the nine possible price pairs in the case of logit demand with parameters $a_1 = a_2 = 1$, $c_1 = c_2 = 0$ and $\mu = 4$. For each pair of payoffs, the first is the payoff to player one and the second is the payoff to player two.

In general, we are thus interested in the payoff matrix in Table 3. To match the payoffs of Table 2 we must have

$$s_3 < s_1 < p < s_2 < m < t_1 < r < t_2 < t_3, \tag{5}$$

and to ensure that it is not profitable to alternate between off-diagonal states we also impose the following conditions:

$$2r > t_3 + s_3, \quad 2r > t_2 + s_2, \text{ and } 2m > t_1 + s_1. \tag{6}$$

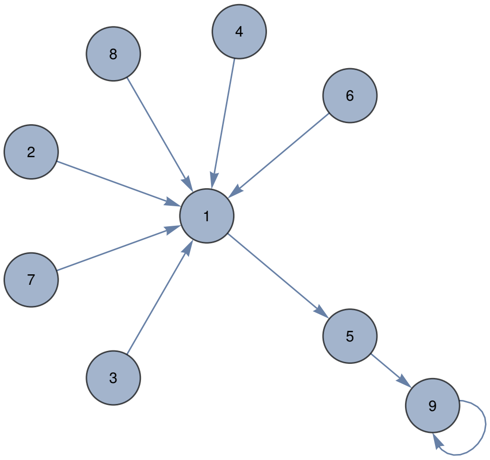

Figure 3: Example of a stably collusive strategy equilibrium. Here the nodes represent all possible one period histories, with nodes 1, 5 and 9 representing the histories $(p_L, p_L)$, $(p_M, p_M)$ and $(p_H, p_H)$, respectively.

These conditions are satisfied by the payoffs in Table 2.

| **1 \ 2** | $p_L$ | $p_M$ | $p_H$ |
|---|---|---|---|
| $p_L$ | $(p, p)$ | $(t_1, s_1)$ | $(t_3, s_3)$ |
| $p_M$ | $(s_1, t_1)$ | $(m, m)$ | $(t_2, s_2)$ |
| $p_H$ | $(s_3, t_3)$ | $(s_2, t_2)$ | $(r, r)$ |

Table 3: General payoff matrix for three action iterated prisoner's dilemma based on logit demand.

To quantify the likelihood of observing collusive strategy equilibria, we consider all Nash equilibria and determine whether they lead to stable supra-competitive prices. The rationale behind our definition of such strategies differs from that used in [10, 9] where the focus is on the existence of a reward-and-punishment scheme. In our case, we define the concept of stably collusive strategy equilibria in Definition 4.1 which focuses on the existence of a forgiveness mechanism restoring supracompetitive prices after a deviation.

**Definition 4.1** (Stably collusive strategy equilibria)**.** *We consider a strategy equilibrium stably collusive if it fulfills the following two conditions:*

*(1) Both players play the same action and an action other than $p_L$ in the $(p_L, p_L)$ state.*

*(2) Both players either play $p_M$ in the $(p_M, p_M)$ state or $p_H$ in the $(p_M, p_M)$ and the $(p_H, p_H)$ state.*

We identify all strategy equilibria fulfilling the conditions in Definition 4.1 and calculate the size of their joint SBA using the IBR graph. By taking the maximum value over $\delta$ of the collusive SBA, we obtain an upper bound for the likelihood of collusion by DQ in the large batch size limit in the pricing environment given by Table 2.

In addition, we calculate the size of the basin of attraction for all symmetric Nash equilibria and the AD strategy. This is the strategy in which both players play $p_L$ in all states and gives rise to competitive prices. There may be other strategies that lead to competitive prices, therefore the size of the basin of attraction of the AD strategy equilibrium gives a lower bound on observing a competitive strategy equilibrium in the BRSAP.

| **Measure** $\backslash$ $\boldsymbol{\delta}$ | 0.2 | 0.3 | 0.4 | 0.6 | 0.75 | 0.85 | 0.95 | 0.99 |
|---|---|---|---|---|---|---|---|---|
| NE | 1 | 21 | 41 | 33 | 41 | 53 | 65 | 69 |
| Symmetric NE | 1 | 9 | 17 | 18 | 21 | 24 | 29 | 31 |
| Symmetric SBA | 1 | 0.98 | 0.93 | 0.95 | 0.95 | 0.95 | 0.92 | 0.92 |
| Collusive NE | 0 | 0 | 20 | 20 | 23 | 31 | 37 | 41 |
| Collusive SBA | 0 | 0 | 0.07 | 0.07 | 0.06 | 0.06 | 0.08 | 0.08 |
| Sym. Coll. SBA | 0 | 0 | 0.02 | 0.02 | 0.02 | 0.02 | 0.03 | 0.03 |
| AD SBA | 1 | 0.97 | 0.90 | 0.92 | 0.92 | 0.92 | 0.88 | 0.88 |

Table 4: Measures of the IBR graph for different values of $\delta$. The first row gives the number of Nash equilibria (NE) in the game. The second row gives the number of symmetric Nash equilibria and the third row gives the SBA for the symmetric Nash equilibria. The fourth row gives the number of collusive Nash equilibria, the fifth row gives the SBA of all collusive Nash equilibria and the sixth row gives the SBA of symmetric and collusive Nash equilibria. The last row gives the SBA of the all defect strategy.

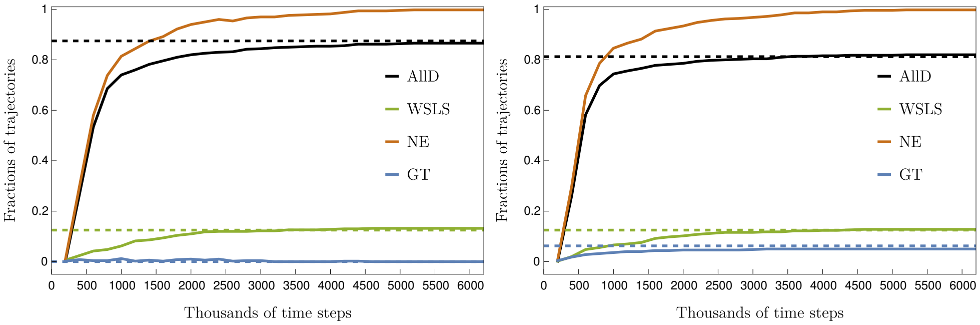

Figure 4: Fraction of trajectories in the equilibria of the prisoner's dilemma: WSLS (green), GT (blue) and AD (black) as well as the total fraction of trajectories in equilibria as a function of time. The dashed lines indicate the theoretical SBA based on Theorem 3.9. We set the discount factor at $\delta = 0.95$, the exploration rate at $\epsilon = 0.2$ and the inertia at $\lambda = 0.1$. The algorithm makes use of a batch size of $K = 200000$ and executes 30 batches. All simulations are based on 500 samples. The game parameters are set to $T = 1.0, R = 0.8, P = 0.2, S = 0.0$ (left) and $T = 1.0, R = 0.8, P = 0.3, S = 0.0$ (right).

## 5 Numerical simulation

Simulating the Decentralized Q-learning algorithm with a large batch size for both environments in Figure 4 and Figure 5 shows that the theoretical predictions are accurate when finite, but large batches are used. In both cases, we include simulations and theoretical predictions for two sets of payoff parameters. Our simulations show that our definition of collusive strategy equilibria in Definition 4.1 captures the extent to which consumers experience supracompetitive prices well. In the PD setting, supracompetitive prices are caused by the WSLS strategy equilibria which is the only strategy equilibrium that satisfies Definition 4.1 in this setting. In the generalized PD setting, this can be seen by the close match between the level of collusion and the theoretical basin of attraction of strategy equilibria satisfying Definition 4.1.

## 6 Conclusion

By examining Table 1 we conclude that DQ can learn to collude in the iterated prisoner's dilemma with a memory of one period with positive probability when the environmental and algorithmic

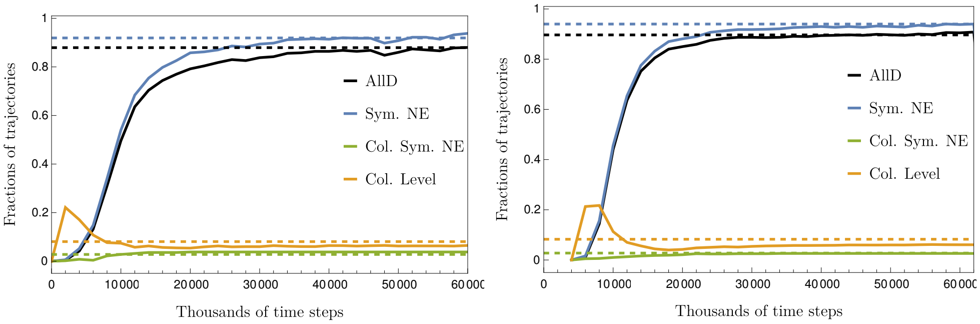

Figure 5: Fraction of trajectories in the AD strategy pair, symmetric Nash equilibrium, Collusive and symmetric Nash equilibrium and the level of collusion of the DQ algorithm. Here, we define the level of collusion to be the fraction of times during a batch that the players played $(p_M, p_M)$ or $(p_H, p_H)$. For the simulation on the left we use the payoffs as defined in Table 2 and for the simulation on the right we use the payoffs corresponding to a logit demand with $a_1 = a_2 = 2, c_1 = c_2 = 1$ and $\mu = 1/4$. In both cases, we set $\delta = 0.95$ and use 500 samples. For the algorithm, we use an exploration rate of $\epsilon = 0.1$, an inertia of $\lambda = 0.1$ and a batch size of $K = 2000000$. The dashed lines indicate the theoretical SBA based on Theorem 3.9.

parameters are conducive for collusion. The probability of collusion in the large batch size limit is bounded from above by $0.125$.

Table 4 shows that when $\delta$ is small enough, only the AD strategy is an equilibrium, leads to competitive prices. Increasing $\delta$ increases the number of symmetric Nash equilibria. Their joint SBA does not necessarily increase, however. Similarly, the size of the joint SBA for stably collusive Nash equilibria is not monotonically increasing in $\delta$, but reaches a maximum of $0.08$ for $\delta$ close to one. This gives an upper bound on the likelihood of observing collusion, and is lower than the upper bound of $0.125$ found in the setting with two prices. The SBA of the AD strategy dominates the strategy space, as its size is between $0.88$ and $1$ for all values of $\delta$ we consider. To motivate our focus on the AD strategy, note that the size of the second-largest SBA for a single strategy never exceeds $0.011$ for all values of $\delta$ we considered. The value of $0.88$ can thus serve as a lower bound on the likelihood of observing a competitive Nash equilibrium, and is close to the lower bound of $0.875$ found in the two-price setting.

Collusion is thus possible with DQ, but its likelihood is significantly lower than the level of collusion observed by Q-learning in [10]. We find that adding a third intermediate action to the prisoner's dilemma does not facilitate the learning of collusion, but actually leads to a decrease in the likelihood of collusion. The introduction of fluctuations as a result of using smaller batches may however drastically increase the occurrence of collusive outcomes, as observed in [6] and also shown in [19]. Characterizing the likelihood of learning different equilibria when smaller batches are used is a promising avenue for future research.

## References

[1] Ibrahim Abada, Joseph E Harrington Jr, Xavier Lambin, and Janusz M Meylahn. Algorithmic collusion: Where are we and where should we be going? *Available at SSRN 4891033*, 2024.

[2] Awni Altabaa, Bora Yongacoglu, and Serdar Yüksel. Decentralized multi-agent reinforcement learning for continuous-space stochastic games. In *2023 American Control Conference (ACC)*, pages 72–77. IEEE, 2023.

[3] Gürdal Arslan and Serdar Yüksel. Decentralized Q-learning for stochastic teams and games. *IEEE Transactions on Automatic Control*, 62(4):1545–1558, 2016.

[4] Stephanie Assad, Robert Clark, Daniel Ershov, and Lei Xu. Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *Journal of Political Economy*, 2023.

[5] Robert Axelrod and William D. Hamilton. The Evolution of Cooperation. *Science*, 211(4489):1390–1396, 1981.

[6] Wolfram Barfuss and Janusz M. Meylahn. Intrinsic fluctuations of reinforcement learning promote cooperation. *Scientific Reports*, 13(1):1309, 2023.

[7] John Batali and Philip Kitcher. Evolution of altriusm in optional and compulsory games. *Journal of theoretical biology*, 175(2):161–171, 1995.

[8] Sophie Calder-Wang and Gi Heung Kim. Coordinated vs efficient prices: The impact of algorithmic pricing on multifamily rental markets. *Available at SSRN: 4403058*, 2023.

[9] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicoló, Joseph E Harrington Jr, and Sergio Pastorello. Protecting consumers from collusive prices due to AI. *Science*, 370(6520):1040–1042, 2020.

[10] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicoló, and Sergio Pastorello. Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–97, 2020.

[11] Arnoud V den Boer. A (mathematical) definition of algorithmic collusion. *Available at SSRN 4636488*, 2023.

[12] Arnoud V. den Boer, Janusz M. Meylahn, and Maarten Pieter Schinkel. Artificial collusion: Examining supracompetitive pricing by Q-learning algorithms. *Amsterdam Law School Research Paper*, (2022-25), 2022.

[13] Frank Harary, Robert Zane Norman, and Dorwin Cartwright. *Structural models: An introduction to the theory of directed graphs*. John Wiley & Sons, Inc., 1965.

[14] Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent RL. *arXiv preprint arXiv:2110.14555*, 2021.

[15] Thomas Loots and Arnoud V. den Boer. Data-driven collusion and competition in a pricing duopoly with multinomial logit demand. *Production and Operations Management*, 32(4):1169–1186, 2022.

[16] Chinmay Maheshwari, Manxi Wu, Druv Pai, and Shankar Sastry. Independent and decentralized learning in Markov potential games. *arXiv preprint: 2205.14590*, 2022.

[17] Weichao Mao and Tamer Başar. Provably efficient reinforcement learning in decentralized general-sum Markov games. *Dynamic Games and Applications*, 13(1):165–186, 2023.

[18] Weichao Mao, Lin Yang, Kaiqing Zhang, and Tamer Başar. On improving model-free algorithms for decentralized multi-agent reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 15007–15049. PMLR, PMLR, 2022.

[19] Janusz M. Meylahn. Does an intermediate price facilitate algorithmic collusion? *Available at SSRN: 4594415*, 2023.

[20] Janusz M Meylahn. Weak acyclicity in games with unique best-responses and implications for algorithmic collusion. *ArXiv preprint*, (0), 2023.

[21] Janusz M. Meylahn and Arnoud V den Boer. Learning to collude in a pricing duopoly. *Manufacturing & Service Operations Management*, 24(5), 2022.

[22] Janusz M. Meylahn and Lars Janssen. Limiting dynamics for Q-learning with memory one in symmetric two-player, two-action games. *Complexity*, 2022:1–20, 2022.

[23] Martin Nowak and Karl Sigmund. A strategy of Win-Stay, Lose-Shift that outperforms Tit-for-Tat in the Prisoner's Dilemma game. *Nature*, 364(6432):56–58, 1993.

[24] OECD. Algorithmic competition: OECD competition policy roundtable background note, 2023.

[25] Gregory Palmer. *Independent learning approaches: Overcoming multi-agent learning pathologies in team-games*. The University of Liverpool (United Kingdom), 2020.

[26] Julien Perolat, Bilal Piot, and Olivier Pietquin. Actor-critic fictitious play in simultaneous move multistage games. In *International Conference on Artificial Intelligence and Statistics*, pages 919–928. PMLR, 2018.

[27] Houssem Sabri. An enumeration of distinct and non-isomorphic functional quasi-order relations. *Discrete Mathematics*, 345(11):113039, 2022.

[28] Ali Seyhun Saral. Evolution of conditional cooperation in prisoner's dilemma. Technical report, Center for Open Science, 2020.

[29] Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. Decentralized Q-learning in zero-sum Markov games. *Advances in Neural Information Processing Systems*, 34:18320–18334, 2021.

[30] Fuuki Shigenaka, Tadashi Sekiguchi, Atsushi Iwasaki, and Makoto Yokoo. Achieving sustainable cooperation in generalized prisoner's dilemma with observation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.