

Does an Intermediate Price Facilitate Algorithmic Collusion?

JANUSZ M. MEYLAHN*, University of Twente, The Netherlands

Extensive simulation studies of reinforcement learning algorithms in pricing environments suggest that it may be possible for them to learn to collude when including a memory of one period. Theoretical analyses of this phenomenon have focused on the setting with two prices, while the simulations have been carried out with up to fifteen prices. Here, we address this gap by analyzing the probability of observing collusion by Decentralized Q-learning in a pricing duopoly with three possible prices and a memory of one period. Using the logit demand model for the payoffs leads to a payoff structure that is a three-action generalization of the iterated PD. Our central question is whether the addition of an intermediate action (or price) between “defection” and “cooperation” increases the likelihood of observing cooperation/collusion.

The Decentralized Q-learning algorithm is provably convergent in this environment, and the basins of attraction of its equilibria can be calculated explicitly using Individual Best-Response Graphs. We find that even though the additional action increases the number of collusive equilibria, the likelihood of collusion does not increase. In addition, we find common structure in the possible strategy equilibria, identify conditions for their existence, and refine the notion of collusive strategy equilibria. Finally, we show that introducing fluctuations to the learning process drastically increases the likelihood of collusion, but not to the levels observed when introducing fluctuations in the environment with two prices.

Designing effective antitrust policy and detection tools against the threat of algorithmic collusion requires understanding the mechanisms by which algorithms learn to collude. To this end, our results provide new insight into the drivers of collusion among reinforcement learners.

JAIR Associate Editor: Chris Beck

JAIR Reference Format:

Janusz M. Meylahn. 2026. Does an Intermediate Price Facilitate Algorithmic Collusion?. *Journal of Artificial Intelligence Research* 87, Article 14 (October 2026), 25 pages. DOI: [10.1613/jair.1.23820](https://doi.org/10.1613/jair.1.23820)

1 Introduction

The use of reinforcement learning (RL) algorithms in pricing environments can lead to collusive outcomes (Calvano, Calzolari, Denicoló, Harrington Jr., et al. 2020; Calvano, Calzolari, Denicoló, and Pastorello 2020). These initial findings have inspired a range of follow-up studies reporting analogous results in a variety of settings (Eschenbaum et al. 2022; Harrington Jr. 2022; Klein 2021). Similar concerns have also been raised for model-based and multi-armed bandit learning algorithms (Cooper et al. 2015; Hansen et al. 2021; Loots and Boer 2022; Meylahn and Boer 2022). The existence of such collusive algorithms would be detrimental to social welfare, as current antitrust legislation is unlikely to be well equipped to prosecute their use (Ezrachi and Stucke 2017, 2023; Harrington Jr. 2019; Mehra 2016). Developing effective antitrust legislation for algorithmic collusion therefore requires both a clear definition of what constitutes algorithmic collusion (Abada, Harrington Jr., et al. 2024; Boer and Meylahn 2024) and an understanding of whether and how algorithmic collusion can arise.

In the setting considered by (Calvano, Calzolari, Denicoló, Harrington Jr., et al. 2020; Calvano, Calzolari, Denicoló, and Pastorello 2020), the price space is discretized into 15 possible prices. By contrast, studies in

*Corresponding Author.

Author’s Contact Information: Janusz M. Meylahn, ORCID: [0000-0002-4388-805X](https://orcid.org/0000-0002-4388-805X), j.m.meylahn@utwente.nl, University of Twente, Enschede, Overijssel, The Netherlands.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).

DOI: [10.1613/jair.1.23820](https://doi.org/10.1613/jair.1.23820)

simplified environments with only two available prices have questioned claims regarding algorithmic collusion (Abada and Lambin 2023; Abada, Lambin, and Tchakarov 2024; Boer, Meylahn, and Schinkel 2026; Dolgoplov 2024; Lambin 2024; Xu and Zhao 2024). For example, it has been argued that the findings of (Calvano, Calzolari, Denicoló, and Pastorello 2020) are an artifact of insufficient learning (Boer, Meylahn, and Schinkel 2026). There is, however, a substantial difference between environments with two prices and those with 15 prices. When only two prices are available, charging a supracompetitive price may be unattractive because the cost of doing so without reciprocation is large. The availability of an intermediate price may facilitate collusion by allowing firms to signal a willingness to collude at a lower cost. Nevertheless, theoretical analyses of the emergence of collusive strategies in pricing environments with memory have so far been restricted to the two-price case (Bertrand et al. 2025; Meylahn 2025).

Analytical results for the prisoner’s dilemma (PD) have shown that introducing a third, irrelevant (because dominated) action can increase the likelihood of reaching sustainable cooperative outcomes (Shigenaka et al. 2017). Similarly, allowing agents to opt out of the game, as in the optional PD, can also promote cooperation (Batali and Kitcher 1995). The payoff structures considered in these studies differ, however, from those of the pricing environments examined here.

A model sharing the same payoff structure as ours, but characterized by only two model parameters, has been investigated through evolutionary simulations of populations of conditional strategy types (Saral 2020). In that setting, strategies are represented by a 4-tuple whose first entry specifies the initial action and whose remaining entries specify responses to each of the opponent’s possible previous actions. Surprisingly, cooperation is often supported by so-called “Hump-shaped” strategies (Saral 2020). These strategies prescribe a more cooperative action in the 3rd entry of the 4-tuple than in the 2nd and 4th entries. This finding suggests that the availability of an intermediate action may facilitate collusion, or more generally cooperation.

Repeated games resembling the three-action PD considered here have also been studied experimentally in the context of cooperation. Experiments in a game with the same payoff ordering as ours, but with sequential actions, indicate that conditionally cooperative strategies become less prevalent over time as participants learn to defect unconditionally (Andreozzi et al. 2020). The use of win-stay, lose-shift reasoning by human subjects has been investigated in two-player, three-action games (Alós-Ferrer and Ritschel 2018). A pricing-inspired generalization of the PD has also been studied experimentally by examining interactions between human managers and algorithmic strategies as well as interactions among managers themselves (Griffith and Rust 1997). One of the strategies employed in that study was a generalized tit-for-tat strategy that conditioned play on the opponent’s previous action.

Experimental evidence further suggests that achieving efficient outcomes in a weak-link coordination game has a strong positive effect on subsequent cooperation in a three-action PD (Knez and Camerer 2000). The relationship between stated beliefs and actual play has also been examined, revealing substantial inconsistencies between beliefs and observed behavior (Costa-Gomes and Weizsäcker 2008). In addition, cooperation under high discount factors has been associated with the emergence of predictable “Semi-Grim” strategies, in which players cooperate after mutual cooperation, defect after mutual defection, and randomize otherwise (Breitmoser 2015).

Competitive pricing has also received considerable attention in the dynamic pricing literature (see (Boer 2015; M. Chen and Z.-L. Chen 2015) for surveys and (Gerpott and Berends 2022) for a review focused on competitive environments). Recent studies have shown that coordinated pricing experimentation can lead to learning of the competitive equilibrium (Yang et al. 2024), that RL algorithms may also converge to competitive outcomes in oligopolistic markets (Kastius and Schlosser 2022), and that capacity constraints substantially affect competitive pricing behavior (Cesaret and Bayram 2024). Additional work has examined competitive pricing in the presence of reference prices (Famili Alamdar and Seifi 2023) and the impact of uncertainty regarding competitors’ capacities (Garcia et al. 2023).

Contributions In this paper, we study how the likelihood of collusion by a provably convergent multi-agent reinforcement learning algorithm in a pricing duopoly is affected by the introduction of a third, intermediate price. The algorithm considered is Decentralized Q-learning (DQ), which was designed for weakly acyclic games and shown to converge in that class of environments (Arslan and Yüksel 2016). Our analysis combines the self-consistent solution of Bellman’s optimality equations (Meylahn and Janssen 2022) with a characterization of the basins of attraction in symmetric games with unique best-response functions (Meylahn 2025).

We find that the addition of an intermediate price gives rise to a much larger number of absorbing strategy profiles or Nash equilibria (NE) for the limiting dynamics of DQ. Many of these strategy profiles are collusive. However, when considering the size of the basin of attraction for collusive strategies, we find that the likelihood of collusion is *not* increased by the addition of an intermediate price. Our results are confirmed by simulations of DQ.

To understand the absorbing strategy profiles better, we identify three primary strategy architectures that are present in almost all symmetric strategy equilibria. These correspond to the three absorbing strategy profiles found in the PD with a one-period memory (Meylahn and Janssen 2022; Usui and Ueda 2021). In addition, we calculate the critical conditions for the existence of all symmetric strategy equilibria. These results together show that the incentive structure underlying the strategy equilibria is the same in the three- and two-action pricing environments and points to a universal strategy equilibrium structure for all pricing environments.

Focusing on strategy equilibria that produce supracompetitive prices, we introduce the concept of stably collusive strategy equilibria. This definition differs from the characterization of collusive strategies based on reward-and-punishment (RAP) schemes (Calvano, Calzolari, Denicoló, Harrington Jr., et al. 2020) by emphasizing the role of forgiveness in re-establishing collusion following a defection. Simulations confirm that this definition accurately captures both the likelihood of collusion and its impact on consumers. These results suggest that reward-punishment-forgiveness schemes are a necessary component of algorithmic collusion by tabular reinforcement learning algorithms.

Finally, we demonstrate that the intrinsic fluctuations of DQ previously identified as conducive to cooperation (Barfuss and Meylahn 2023) also substantially increase the degree of collusion observed in the pricing environment studied here, although not to the same extent as reported by (Barfuss and Meylahn 2023). This effect appears to arise from differences in the stability of competitive and collusive strategy equilibria, suggesting that algorithms could be intentionally designed to learn collusive behavior through the control of fluctuations.

Overview. The rest of the article is structured as follows. In Section 2 we describe the environment we will consider. In Section 3 we introduce DQ and the inertial best-response dynamics (IBRD). Section 4 contains our results, and in Section 5 we conclude.

2 Setting

Inspired by the pricing setting, we will consider a generalized PD with three actions. We will base the payoff structure on the prices and revenues of a logit demand model. This choice is motivated by the recent studies of algorithmic collusion in such a setting (Boer, Meylahn, and Schinkel 2026; Calvano, Calzolari, Denicoló, and Pastorello 2020; Hettich 2021; Loots and Boer 2022; Wang et al. 2022). The realized demand of player i is

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

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

$$r_i(p_i, p_{-i}) = d_i(p_i, p_{-i})(p_i - c_i), \quad (2)$$

where $c_i \geq 0$ is the marginal cost. If both players simultaneously maximize their own revenue, we call the resulting price pair competitive. On the other hand, if both players maximize the joint revenue, we call the resulting price pair collusive. 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 $\mathcal{S} := \mathcal{A} \times \mathcal{A}$. The theory we will apply requires the game to be symmetric, which requires that $a_1 = a_2$ and $c_1 = c_2$. 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 1.

$1 \setminus 2$	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 1. 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 2. To match the ordering of the payoffs in Table 1 we must have

$$s_3 < s_1 < p < s_2 < m < t_1 < r < t_2 < t_3. \quad (3)$$

$1 \setminus 2$	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 2. General payoff matrix for three action PD based on logit demand.

The goal of the agents is to maximize their discounted sum of future rewards

$$\sum_{t=1}^{\infty} \delta^t r_i(p_i, p_{-i}), \quad (4)$$

where $\delta \in (0, 1)$ is the discount factor, capturing the extent to which players discount revenue received in the future.

3 Methods

In this section, we introduce the algorithm we will be analyzing and an idealized process used for the analysis.

3.1 Decentralized Q-learning

We will consider two agents using DQ (Arslan and Yüksel 2016). The state space for the algorithm is determined by a memory of one period, where each possible price pair is considered a separate state. This allows the agents to condition their choice of price on the actions taken in the previous period. Such conditioning is necessary to implement strategies such as win-stay, lose-shift (WSLS) or tit-for-tat (TFT) in the standard iterated PD and this in turn leads to the existence of collusive strategy equilibria (Meylahn and Janssen 2022).

The DQ algorithm makes use of a batch structure. This means that it fixes a strategy for executing actions for a fixed number of interactions K (called a batch), while collecting samples on the opponent's behavior and the environment. This information is then used between batches to update the strategy.

Each player i randomly initializes a Q-matrix which determines their behavior during the first batch

$$\mathbf{q}_i^{\text{act}} = \begin{pmatrix} q_{i,1,1}^{\text{act}} & q_{i,1,2}^{\text{act}} & q_{i,1,3}^{\text{act}} \\ q_{i,2,1}^{\text{act}} & q_{i,2,2}^{\text{act}} & q_{i,2,3}^{\text{act}} \\ q_{i,3,1}^{\text{act}} & q_{i,3,2}^{\text{act}} & q_{i,3,3}^{\text{act}} \\ q_{i,4,1}^{\text{act}} & q_{i,4,2}^{\text{act}} & q_{i,4,3}^{\text{act}} \\ q_{i,5,1}^{\text{act}} & q_{i,5,2}^{\text{act}} & q_{i,5,3}^{\text{act}} \\ q_{i,6,1}^{\text{act}} & q_{i,6,2}^{\text{act}} & q_{i,6,3}^{\text{act}} \\ q_{i,7,1}^{\text{act}} & q_{i,7,2}^{\text{act}} & q_{i,7,3}^{\text{act}} \\ q_{i,8,1}^{\text{act}} & q_{i,8,2}^{\text{act}} & q_{i,8,3}^{\text{act}} \\ q_{i,9,1}^{\text{act}} & q_{i,9,2}^{\text{act}} & q_{i,9,3}^{\text{act}} \end{pmatrix}, \quad (5)$$

and transfers this to a Q-matrix for learning during the batch $\mathbf{q}_i^{\text{crit}} = \mathbf{q}_i^{\text{act}}$. Here $q_{i,s,a}$ represents the current estimate player i has of the value of taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, where we have encoded the states as follows: $1 \rightarrow (p_L, p_L)$, $2 \rightarrow (p_L, p_M)$, $3 \rightarrow (p_L, p_H)$, $4 \rightarrow (p_M, p_L)$, $5 \rightarrow (p_M, p_M)$, $6 \rightarrow (p_M, p_H)$, $7 \rightarrow (p_H, p_L)$, $8 \rightarrow (p_H, p_M)$, $9 \rightarrow (p_H, p_H)$. The first entry indicates the action of firm one and the second entry the action of firm two. We also use the shorthand notation for actions that $1 \rightarrow p_L$, $2 \rightarrow p_M$ and $3 \rightarrow p_H$.

DQ proceeds by observing the state the system is in (starting with a random initial state) and selecting an action according to the ϵ -greedy action selection mechanism based on $\mathbf{q}_i^{\text{act}}$. This prescribes that the action with the maximal Q-value is taken with probability $1 - \epsilon$ and an action is taken uniformly at random with probability ϵ . The new state is observed and the Q-values $\mathbf{q}_i^{\text{crit}}$ are updated according to

$$q_{i,s,a}^{\text{crit}}(t+1) = (1 - \alpha_{n(s,a)})q_{i,s,a}^{\text{crit}}(t) + \alpha_{n(s,a)} \left[r_i(s(t+1)) + \delta \max_{a \in \{1,2,3\}} \{q_{i,s(t+1),a}^{\text{crit}}\} \right]. \quad (6)$$

Here

$$\alpha_{n(s,a)} = \begin{cases} \frac{1}{1+n(s,a)} & \text{when } (s,a) = (s(t), a(t)) \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

and $n(s,a)$ is a local time which keeps track of the number of times the state-action pair (s,a) has been visited during the current batch. This choice of $\alpha_{n(s,a)}$ can be replaced by any choice satisfying the usual conditions for convergence of stochastic approximation algorithms (Borkar 2022). In the limit as $K \rightarrow \infty$, the updates in (6) are guaranteed to converge to the Q-values solving the Bellman equations given the current strategy of the opponent and so will converge to Q-values that prescribe a strategy that maximizes the discounted sum of future rewards, i.e., a best response to the opponent's current strategy. This convergence holds under the assumption that all state-action pairs are visited infinitely often, which is the case due to the ϵ -greedy action selection mechanism.

For finite but large K , the learned Q-values closely approximate the optimal Q-values (see (Arslan and Yüksel 2016) for a precise statement). At the conclusion of the k^{th} batch, the learned Q-values are transferred to the Q-values for the behavior in the next batch for each player i with probability $1 - \lambda$

$$\mathbf{q}_i^{\text{act}}(k+1) = \begin{cases} \mathbf{q}_i^{\text{crit}}(k) & \text{with probability } 1 - \lambda \\ \mathbf{q}_i^{\text{act}}(k) & \text{with probability } \lambda \end{cases}, \quad (8)$$

where λ is called the inertia. This ensures that the algorithm is able to escape limit cycles of incompatible NE strategies.

3.2 Inertial Best-Response Dynamics

The multi-agent learning dynamics resulting from DQ can be made to follow the inertial best-response dynamics (IBRD) arbitrarily closely by increasing K . This process takes place on the discrete strategy space Π , containing all possible ϵ -greedy strategies on the state and action space. The IBRD can be interpreted as a random walk on the IBRD graph. Here, $\text{BR}(\pi)$ represents the best response to strategy π .

Definition 3.1 (IBRD Graph). Identify each possible pair of strategies $(\pi_i, \pi_j), i, j \in \{1, 2, \dots, |\Pi|\}$ with a node in the graph. Edges of the graph are

- (1) two directed edges with weight $\lambda(1 - \lambda)$ from (π_i, π_j) to $(\text{BR}(\pi_j), \pi_j)$ and to $(\pi_i, \text{BR}(\pi_i))$,
- (2) a directed edge with weight $(1 - \lambda)^2$ from (π_i, π_j) to $(\text{BR}(\pi_j), \text{BR}(\pi_i))$,
- (3) a self-loop at (π_i, π_j) with weight λ^2 .

Using this relationship between DQ and the IBRD, it can be shown that the DQ dynamics converge when the best-response (BR) graph of the game is weakly acyclic (Arslan and Yüksel 2016). The BR graph is a subgraph of the IBRD graph, containing all the edges that have a weight of $\lambda(1 - \lambda)$ and it is weakly acyclic if there exists a finite path from every strategy profile to a Nash equilibrium.

The IBRD graph may have multiple absorbing nodes, which correspond to NE of the game. An important aspect of the learning dynamics then becomes the likelihood with which the dynamics converges to the different equilibria. To quantify this, we use (Meylahn 2025, Theorem 3.8) as explained below. We begin by defining the individual best-response (IBR) graph in Definition 3.2.

Definition 3.2 (IBR Graph). In the individual best-response (IBR) graph, each possible strategy $\pi_i \in \Pi$ is identified with a node in the graph, and a directed edge is added from node π_i to π_j if $\text{BR}(\pi_i) = \pi_j$.

All absorbing states of the IBRD appear as nodes with self-loops (symmetric NE) and 2-cycles (non-symmetric NE) in the IBR graph, and the likelihood of observing a given equilibrium or the size of its basin of attraction is given by the number of nodes in its weakly connected component of the IBR graph divided by $|\Pi|$. This measure is appropriate for comparing the two- and three-price settings, as it takes care of the normalization required due to the drastic difference in the size of the strategy space between the two settings. To establish that the DQ algorithm converges in our environment, i.e., that the game is weakly acyclic, we can show that the IBR graph contains no cycles of length three or more (Meylahn 2025, Theorem 3.3).

The IBR graphs thus contain all the necessary information for the results we aim to establish in Section 4. Using the IBR graphs instead of analyzing the IBRD process directly is advantageous in our setting, as the IBRD graph contains close to 4×10^8 nodes while the IBR graph contains approximately 2×10^4 nodes. In addition, it characterizes the basins of attraction of the Nash equilibria as the size of the weakly connected component containing the equilibrium, which has complexity $O(|\Pi|)$. Calculating the basins of attraction directly as the stationary distribution of the IBRD requires calculating the eigenvector for the largest eigenvalue of the normalized Laplacian, which is a matrix of size $|\Pi|^2 \times |\Pi|^2$.

4 Results

In this section, we present results on the idealized process, a classification of equilibria, conditions for the existence of equilibria and simulation results.

4.1 IBR Graph Measures

By self-consistently solving Bellman's optimality equations given the strategy of the opponent (Meylahn 2025; Meylahn and Janssen 2022; Usui and Ueda 2021), we can construct the IBR graph for the payoffs given in Table 1 and different values of δ (see Figure 11 for an example). For each IBR graph, we calculate a variety of measures,

which we summarize in Table 3. Before doing so, we verify that the graphs are weakly acyclic, ensuring that the DQ algorithm converges. We proceed by calculating the number of components in the IBR graph. This is equivalent to the number of absorbing states or NE in the IBRD, as each component contains exactly one self-loop or 2-cycle (see (Meylahn 2025, Lemma 3.1)). All NE which correspond to a self-loop in the IBR graph are symmetric, meaning that both players play the same strategy. By summing the sizes of all components ending in a self-loop, we calculate the size of the basin of attraction of symmetric NE. By contrast, we will call NE corresponding to 2-cycles in the IBR graph non-symmetric. Due to the symmetry of the game, each 2-cycle corresponds to two non-symmetric equilibria of the game. In counting NE and calculating basins of attraction, we merge these pairs of equilibria.

To quantify the likelihood of observing collusive strategy equilibria, we consider all NE and determine whether they lead to stable supracompetitive prices. The rationale behind our definition of such strategy equilibria differs from that used by (Calvano, Calzolari, Denicoló, Harrington Jr., et al. 2020; Calvano, Calzolari, Denicoló, and Pastorello 2020), where the focus is on the existence of a RAP scheme. In our case, we define the concept of stably collusive strategy equilibria in Definition 4.1, which focuses on the reestablishment of supracompetitive prices after playing (p_L, p_L) . This can be interpreted as mutual forgiveness.

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) states.

We identify all strategy equilibria fulfilling the conditions in Definition 4.1 and calculate the size of their joint basin of attraction. By taking the maximum value over δ of the collusive basins of attraction, 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 1.

Finally, we calculate the size of the basin of attraction for the all-defect (AllD) strategy. This is the strategy in which both players play p_L in all states, and it 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 AllD strategy equilibrium gives a lower bound on observing a competitive strategy equilibrium in the IBRD.

By examining Table 3, we reach the following conclusions. When δ is small enough, there is exactly one NE (cf. (Boer, Meylahn, and Schinkel 2026, Theorem 1)). This is the AllD strategy equilibrium and therefore leads to competitive prices. As we increase δ , the number of symmetric NE increases, but the size of their joint basin of attraction does not necessarily increase. The size of the joint basin of attraction of stably collusive NE is not monotonically increasing in δ , but reaches a maximum of 0.08 for δ close to one. This gives an upper bound on the likelihood of observing collusion in this environment, and is lower than the upper bound of 0.125 found for the setting with two prices (Meylahn 2025). The basin of attraction of the AllD strategy dominates the strategy space, as its size is between 0.88 and 1 for all values of δ we consider. To motivate our focus on the AllD strategy, note that the size of the second-largest basin of attraction for a single strategy never exceeds 0.011 for all values of δ we considered. The value of 0.88 can thus serve as a lower bound on the likelihood of observing a competitive NE in this setting, and is close to the lower bound of 0.875 found in the two-price setting (Meylahn 2025).

4.2 Classification of Strategy Equilibria

Given a strategy profile π , we can visualize the state transitions as a graph $G_\pi : \mathcal{S} \rightarrow \mathcal{S}$ called the strategy graph (Calvano, Calzolari, Denicoló, and Pastorello 2020; Eschenbaum et al. 2022). Here each of the nine states is identified with a node, and a directed edge is drawn from node i to node j when the greedy actions taken by the players at node i result in a transition to node j . This construction results in functional relations, meaning that each node has out-degree one, and we allow self-loops. Such graphs have the property that they consist of

Measure \ δ	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 BoA	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 BoA	0	0	0.07	0.07	0.06	0.06	0.08	0.08
Sym. Coll. BoA	0	0	0.02	0.02	0.02	0.02	0.03	0.03
AllD BoA	1	0.97	0.90	0.92	0.92	0.92	0.88	0.88

Table 3. Measures of the IBR graph for different values of δ . The first row gives the number of NE in the game. The second row gives the number of symmetric NE and the third row gives the basin of attraction for the symmetric NE. The fourth row gives the number of collusive NE, the fifth row gives the size of the basin of attraction of all collusive NE and the sixth row gives the basin of attraction of symmetric and collusive NE. The last row gives the size of the basin of attraction of the all defect strategy. All fractions are rounded to two decimal places.

components, where each of the components is composed of a tree towards a cycle or a self-loop (Harary et al. 1965). The resulting strategy graphs for the 31 symmetric strategy profiles found when $\delta = 0.99$ are shown in Figures 6, 7, 8 and 9 of the Appendix. In addition to these, there are four graphs that do not form an equilibrium when $\delta = 0.99$, but do when $\delta = 0.3, 0.4$. These are shown in Figure 10 in the Appendix. We will focus on the symmetric strategy equilibria, as their combined basin of attraction is above 92% for all values of δ we tested.

By inspecting these graphs, we can identify general characteristics of the symmetric NE. We first note that almost all the graphs comprise components leading to self-loops at the symmetric states: 1, 5 and 9. These are the states where both players play the same action. The exceptions are graphs 3 and 4. In graph 3 there are isolated self-loops at states 6 and 8, and in graph 4 there is a 2-cycle between nodes 6 and 8. The size of their combined basin of attraction is 0.0001. We will therefore also exclude these graphs from our classification. The remaining 33 graphs are all constructed of three primary strategy architectures, namely, the all defect (AllD) architecture, the grim trigger (GT) architecture and the win-stay, lose-shift (WSLS) architecture. To define these, we introduce some notation. First, we define the symmetric states to be

$$\Delta_a := \{(p, p) : p \in \mathcal{A}\}. \quad (9)$$

With $G_\pi(s)$ we mean the state that is reached from state s after one step in G_π .

Definition 4.2 (Primary strategy architectures). Consider $C \subseteq \mathcal{S}$ and a graph G_π . Then

- (1) C is an *AllD architecture* if there exists a symmetric state $\Delta_a \in C$ with $a = \min\{b \in \mathcal{A} : \Delta_b \in C\}$ and for all $s \in C$, $G_\pi(s) = \Delta_a$.
- (2) C is a *GT architecture* if $C \subseteq \{\Delta_{p_M}, \Delta_{p_H}\}$ and for all $s \in C$, $G_\pi(s) = s$. Furthermore, there exists no $s \in \mathcal{S} \setminus C$ for which $G_\pi(s) = s'$ with $s' \in C$.
- (3) C is a *WSLS architecture* if C contains symmetric states $\Delta_{a_1}, \dots, \Delta_{a_k}$ with $k \geq 2$, $a_1 < \dots < a_k$, and

$$G_\pi(\Delta_{a_j}) = \begin{cases} \Delta_{a_{j+1}} & \text{for } j < k \\ \Delta_{a_j} & \text{for } j = k \end{cases}.$$

In addition, for all non-symmetric states, i.e., $s \in C \setminus \{\Delta_{a_1}, \dots, \Delta_{a_k}\}$, $G_\pi(s) = \Delta_{a_1}$.

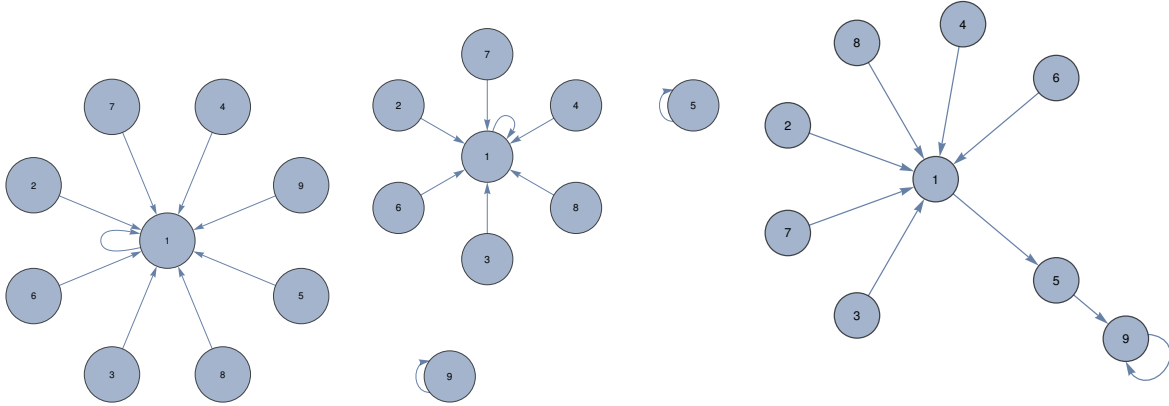


Fig. 1. Illustration of primary architectures. Strategy graphs 000000000 (left), 000010002 (middle) and 100020002 (right).

When we say that a graph consists of these three primary architectures, we mean that there exists a covering of the state space

$$\mathcal{S} = \bigcup_{l=1}^r C_l, \quad (10)$$

so that each $C_l \subseteq \mathcal{S}$ satisfies the conditions of one of the three primary architectures in Definition 4.2. Note that the C_l 's need not be disjoint and that the covering need not be unique.

To illustrate these primary architectures, we use graphs 1, 7, and 19 from Figure 6 and Figure 7, which we reproduce in Figure 1 for convenience. Graph 1 consists of all nodes having an edge to node 1 so that this is a pure ALLD architecture, i.e., here $C_1 = \mathcal{S}$ and this satisfies (1) in Definition 4.2. Graph 7 consists of an ALLD architecture between all nodes besides nodes 5 and 9 and a GT architecture on nodes 5 and 9, i.e., here $C_1 = \{1, 2, 3, 4, 6, 7, 8\}$ and $C_2 = \{5, 9\}$ with C_1 satisfying (1) in Definition 4.2 and C_2 satisfying (2) in Definition 4.2. Finally, graph 19 consists of a WSLS architecture between all the nodes, i.e., here, $C_1 = \mathcal{S}$ and C_1 satisfies (3) in Definition 4.2. As an example of a hybrid graph containing all three architectures, we use graph 18. Here, there is an ALLD architecture between nodes 5, 6, and 8, a GT architecture of node 9, and a WSLS architecture between nodes 1, 2, 3, 4, 5, and 7. In this case, node 5 forms part of two different primary architectures. Here we thus have $C_1 = \{5, 6, 8\}$, $C_2 = \{9\}$ and $C_3 = \{1, 2, 3, 4, 5, 7\}$ with C_1 , C_2 and C_3 satisfying properties (1), (2) and (3) of Definition 4.2, respectively.

Note that the architectures identified here by studying the symmetric strategy equilibria occur in almost all of the non-symmetric strategy equilibria as well. In these cases, they are, however, modified slightly. An example is the strategy shown in Figure 2. Here, the architecture between nodes 1–5 and 7 is close to being that of the WSLS architecture, with the difference that nodes 3 and 7 do not connect to node 1 directly but to nodes 2 and 4, which then connect to node 1. This captures the same kind of structure as the WSLS architecture in the sense that for the bulk of the nodes, the path to the absorbing state passes through node 1. Plots of all non-symmetric equilibria can be found in the shared code (see the data availability statement).

4.3 Critical Conditions for the Discount Factor

The results in Table 3 show how the number of NE changes as a function of the discount factor δ . This indicates that there are critical conditions on δ deciding when strategy profiles exist. Here, we calculate the critical conditions for the existence of the symmetric NE found in Figures 6, 7, 8, 9, and 10. The reasoning used for the



Fig. 2. Left: Illustration of a hybrid graph containing all three primary architectures using strategy graph 18 with strategy: 100011012. Right: Example of a non-symmetric strategy equilibrium, showing structures similar to the primary architectures.

calculation of all critical conditions is the same. We begin by assuming that both players play the strategy under consideration and then check whether a player has an incentive to unilaterally deviate from the strategy.

As an example, consider the strategy depicted by graph 5 in Figure 6. In this case, the maximal Q-value in all states besides state 5 is

$$q_{i,j,1} = \frac{p}{1-\delta} \quad \text{for } i \in \{1, 2\} \text{ and } j \in \{1, 2, 3, 4, 6, 7, 8, 9\}. \quad (11)$$

This is the case, as we know that the AllD strategy equilibrium is always stable. Now the Q-values for player 1 in state 5, assuming that the opponent continues to play the strategy, are

$$q_{1,5,1} = \frac{t_1 + \delta(p - t_1)}{1 - \delta}, \quad q_{1,5,2} = \frac{m}{1 - \delta}, \quad \text{and} \quad q_{1,5,3} = \frac{s_2 + \delta(p - s_2)}{1 - \delta}. \quad (12)$$

Given that $t_1 > s_2$, we know that $q_{1,5,1} > q_{1,5,3}$ always holds. To find the critical condition for δ , we solve $q_{1,5,1} = q_{1,5,2}$ for δ and find the critical condition

$$\delta_1 = \frac{t_1 - m}{t_1 - p}. \quad (13)$$

Graph 5 is thus stable when $\delta > \delta_1$. A similar calculation leads to the same critical condition for graphs 6, 8, and 9. Continuing in this way, we identify the critical conditions for all the symmetric NE found for the values of δ in Table 3. We collect the critical conditions in Tables 4, and Table 5. Many of the critical conditions are of the form also found in the PD with a memory of one period (Meylahn and Janssen 2022), but we also find more complex critical conditions, namely, δ_3 , δ_5 , δ_6 and δ_7 .

4.4 Simulation

We now simulate DQ in the three action environment of Table 1. To be able to sample all parts of the state space, the algorithm must be implemented with an exploration rate. Such an exploration rate was not taken into account when constructing the IBR graphs. Previous work on the iterated PD, however, suggests that the inclusion of a small constant exploration rate changes the IBR graphs only slightly. In the PD, a small exploration rate increases the basin of attraction of the AllD strategy and decreases the basin of attraction of the GT and WSLS strategies. The simulations in this section confirm that the inclusion of a small exploration rate does not drastically change the basins of attraction calculated above.

δ_c	Condition	Numerical Value	Graphs
δ_0	0	0	1
δ_1	$\frac{t_1-m}{t_1-p}$	0.236	5, 6, 8, 9
δ_2	$\frac{t_1-m}{m-p}$	0.310	13, 14, 16, 17, 20, 21, 23, 24
δ_3	$\frac{(t_2-p) - [4(t_1-p)(t_1-s_2) + (t_2-p)^2]^{1/2}}{2(p-t_1)}$	0.439	4
δ_4	$\frac{t_3-r}{t_3-p}$	0.475	2, 7, 12
δ_5	$\frac{(m-p) - [(p-m)^2 - 4(p-t_3)(t_3-r)]^{1/2}}{2(p-t_3)}$	0.518	11
δ_6	$\frac{(r-p) - [(p-r)^2 - 4(m-r)(t_2-r)]^{1/2}}{2(m-r)}$	0.604	19
δ_7	$\frac{(t_3-p) - [p^2 + 4r(m-p) + t_3^2 - 4mt_3 + 2pt_3]^{1/2}}{2(m-p)}$	0.637	15
δ_8	$\frac{t_2-r}{t_2-m}$	0.742	10, 18
δ_9	$\frac{t_3-r}{t_3-m}$	0.790	22, 25
δ_{10}	$\frac{t_3-r}{r-p}$	0.906	27, 28, 30, 31
δ_{11}	$\frac{t_1-s_2}{t_1-p}$	0.938	3
δ_{12}	$\frac{t_1-m}{r-m}$	0.978	26, 29

Table 4. Critical values of δ above which the symmetric NE found at $\delta = 0.99$ start to exist in the case of the logit demand model, with the three prices given in Table 1. The numbering of the graphs corresponds to the numbering as used in Figures 6, 7, 8 and 9.

Condition	Numerical Value	Graphs
$\frac{t_1-m}{t_1-p} < \delta < \frac{p-s_3}{m-s_3}$	$0.236 < \delta < 0.558$	32, 33, 34, 35

Table 5. Critical range of δ at which the additional symmetric NE of Figure 10 exist in the case of the logit demand model, with the three prices given in Table 1.

We show two simulations in Figure 3 for two different environments. We use the standard DQ algorithm with a batch size of 2 million. This is sufficiently large for the theoretical results on the basin of attraction to be valid. We see that the simulation matches the theoretical basin of attraction for the AllD, the symmetric, and the collusive symmetric NE profiles closely. The observed level of collusion, calculated as the fraction of time spent in the (p_M, p_M) and (p_H, p_H) states during a batch, is slightly lower than the theoretical basin of attraction of collusive NE. The deviation from the basin of attraction is explained by the inclusion of a small exploration rate in the simulation and concurs with the effect of exploration observed previously (Meylahn 2025). A greedy, stably collusive strategy equilibrium would spend all of its time in a supracompetitive price pair, but the inclusion of exploration leads to occasional deviations from this that are quickly restored.

These simulations affirm our definition of stably collusive strategies. The level of collusion is a good measure of the effect of collusion on the consumers, as it reflects the extent to which prices are above the competitive price. Crucially, this is only the case if the underlying strategies leading to the supracompetitive prices form an equilibrium. The fact that the DQ algorithm has convergence guarantees and that we observe a close match between the theoretical basin of attraction for stably collusive equilibria and the level of collusion suggests that supracompetitive prices are indeed driven by stably collusive strategies as defined in Definition 4.1.

In Figure 4, we repeat the simulations in Figure 3 with two differences: (1) we use a batch size that is an order of magnitude smaller than that used in the first simulation, and (2) we update the Q-values between batches using an inter-batch learning rate $\hat{\alpha} < 1$. The update in (8) becomes

$$\mathbf{q}_i^{\text{act}}(k+1) = \begin{cases} (1 - \hat{\alpha})\mathbf{q}_i^{\text{act}}(k) + \hat{\alpha} \mathbf{q}_i^{\text{crit}}(k) & \text{with probability } 1 - \lambda \\ \mathbf{q}_i^{\text{act}}(k) & \text{with probability } \lambda \end{cases}. \quad (14)$$

Changing the algorithm in this way has been shown to lead to a drastic increase in cooperation in the PD (Barfuss and Meylahn 2023). With this simulation, we see that a similar effect is possible in the three-action variant. We see that the level of collusion increases from approximately 0.06 to approximately 0.8, while the basin of attraction of the collusive and symmetric NE increases from approximately 0.03 to approximately 0.35. Both are thus increased tenfold. However, this is still less than the level of collusion observed when introducing fluctuations to the dynamics of DQ in the PD (Barfuss and Meylahn 2023), where for many parameter combinations DQ achieved collusion in all sample trajectories.

Note that Figure 4 implies that less than half of the observed collusion is driven by symmetric NE. This could be due to incomplete learning in states that are visited less frequently as a result of the smaller batch sizes. It is also notable that the AllD strategy equilibrium, which dominated when using a large batch size, is insignificant asymptotically when using a smaller batch size.

The increase in collusion may be the result of the typical architectures involved in competitive versus collusive strategy equilibria. The AllD architecture is such that a single exploration leads to a single sample of a state other than state 1. On the other hand, the WSL architecture is such that a single exploration leads to at least two states other than states 9 or 5 being sampled. Therefore, the effective exploration rate in collusive strategy equilibria is higher than in competitive equilibria. Collusive strategy equilibria thus lead to a more even distribution of the samples obtained during a batch over the states than competitive strategy equilibria. This in turn leads to more accurate Q-value estimates and thus increases the likelihood that a trajectory that has reached a collusive strategy equilibrium remains there, relative to a trajectory in a competitive strategy equilibrium.

This reasoning is supported by the simulation, in which we observe an initial increase in the number of trajectories in the AllD strategy equilibrium, driven by its large basin of attraction under the IBRD, followed by a decrease to zero, driven by its lack of stability. The collusive and symmetric NE, on the other hand, show a steady increase. In Appendix A, we discuss the possibility of this effect coming from the inertia parameter λ .

4.5 Robustness

In Table 6, we calculate the collusive basin of attraction and the basin of attraction of the AllD strategy for different choices of the parameters for the logit demand model. The table shows that the likelihood of colluding does not increase relative to the likelihood of colluding in the two price setting with the introduction of an intermediate price for all parameter values we tested. The resulting IBR graphs were all weakly acyclic. Notably, the collusive basin of attraction is of the same size in all environments we investigated when rounding to two decimal places.

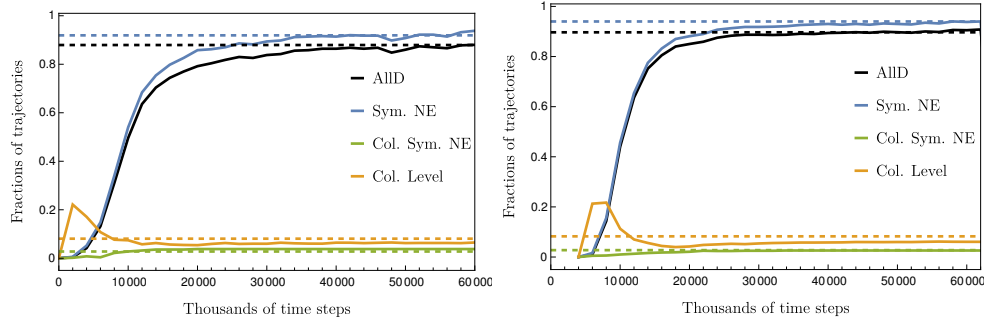


Fig. 3. Fraction of trajectories in the AllID strategy pair, symmetric NE, Collusive and symmetric NE, and the level of collusion of the DQ algorithm. Here, we define the level of collusion to be the fraction of times during a batch that the players played (p_M, p_M) or (p_H, p_H) . For the simulation on the left, we use the payoffs as defined in Table 1, 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 $\delta = 0.95, \epsilon = 0.1, \lambda = 0.1, K = 2 \times 10^6$ and simulate 500 trajectories. The dashed lines indicate the theoretical basins of attraction in the large batch size limit, as in Table 3 and Table 6.

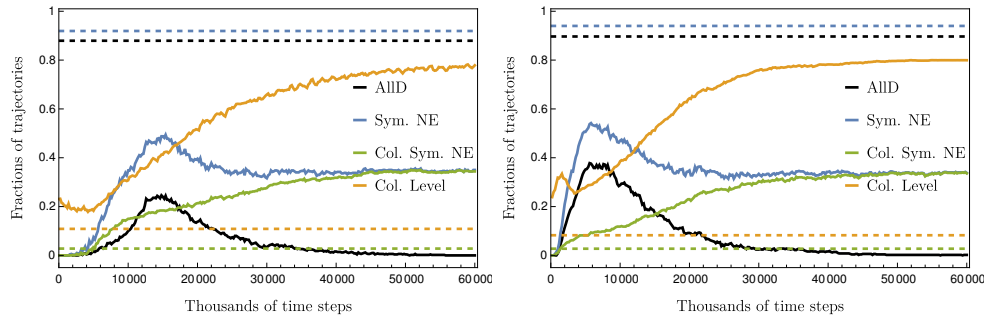


Fig. 4. Fraction of trajectories in the AllID strategy pair, symmetric NE, Collusive and symmetric NE and the level of collusion of the DQ algorithm. Here, we define the level of collusion to be the fraction of times during a batch that the players played (p_M, p_M) or (p_H, p_H) . For the simulation on the left, we use the payoffs as defined in Table 1, 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 $\delta = 0.95, \epsilon = 0.1, \lambda = 0.1, K = 2 \times 10^5$ and simulate 500 trajectories. The inter-batch learning rate is $\hat{\alpha} = 0.4$. The dashed lines indicate the theoretical basin of attraction in the large batch size limit, as in Table 3 and Table 6.

5 Conclusion

Analysis of the IBR graphs for a pricing-inspired three-action generalization of the PD indicates that the addition of an intermediate action does not increase the likelihood of observing collusive NE when a provably convergent multi-agent reinforcement learning algorithm is used. This contrasts with previous findings obtained from evolutionary simulations (Saral 2020). Surprisingly, the upper bound on collusive NE and the lower bound on competitive NE are close to those previously reported for the two-price setting (Meylahn 2025). Although most of the possible NE are stably collusive, their combined basin of attraction is approximately an order of magnitude smaller than that of the AllID strategy equilibrium. Given the dominance of the AllID basin of attraction, introducing additional intermediate prices, or prices below the competitive price and above the collusive price as

Parameters \ Measure	Collusive NE	Collusive BoA	AllD BoA
$a = 1, c = 0, \mu = 4$	37	0.08	0.88
$a = 1, c = 0, \mu = 3$	37	0.08	0.88
$a = 1, c = 0, \mu = 2$	37	0.08	0.88
$a = 1, c = 0, \mu = 1$	41	0.08	0.88
$a = 2, c = 1, \mu = \frac{1}{4}$	69	0.08	0.90
$a = 2, c = 1, \mu = \frac{1}{3}$	55	0.08	0.90
$a = 2, c = 1, \mu = \frac{1}{2}$	52	0.08	0.87
$a = 2, c = 1, \mu = 1$	41	0.08	0.88
$a = 3, c = 2, \mu = 2$	37	0.08	0.90
$a = 3, c = 2, \mu = 3$	37	0.08	0.88
$a = 3, c = 2, \mu = 4$	37	0.08	0.88
$a = 3, c = 2, \mu = 5$	37	0.08	0.88

Table 6. Measures of the IBR graph for different combinations of $a_1 = a_2 = a, c_1 = c_2 = c$ and μ with $\delta = 0.95$. The first column specifies the values of the logit demand parameters. The second column gives the number of collusive NE in the game. The third column gives the size of the basin of attraction of all collusive NE, and the fourth column gives the basin of attraction of the all defect strategy. All fractions are rounded to two decimal places.

in (Calvano, Calzolari, Denicoló, and Pastorello 2020), is unlikely to increase the basin of attraction of collusive equilibria.

We identify three primary architectures among the equilibrium strategies that arise in the three-action PD. These architectures are based on the three equilibria found in the standard PD with one-period memory, namely AllD, GT, and WSLS. These strategies also appear frequently in experimental studies of cooperation (Alós-Ferrer and Ritschel 2018; Andreozzi et al. 2020; Breitmöser 2015). Moreover, the collusive strategy profiles that have been identified as driving collusion are reminiscent of the WSLS architecture identified here (Calvano, Calzolari, Denicoló, and Pastorello 2020, Figures 8 and 9). The existence of a common architecture across strategy equilibria may provide opportunities for improving learning efficiency through methods such as adaptive state-space discretization (Sinclair et al. 2023) or the identification of analogy classes (Jehiel 2005).

The addition of memory to Q-learning in pricing settings has introduced the possibility of RAP scheme-type strategies. It has been speculated that the use of RAP schemes is necessary for algorithmic collusion (Harrington Jr. 2019). Such schemes are defined by players rewarding each other for adhering to the collusive price and punishing each other for deviating from it. Both the GT and WSLS architectures identified here would satisfy this description. We find that the crucial aspect of stably collusive equilibrium strategies for RL algorithms, in addition to the existence of a classical RAP scheme, is a mechanism that restores collusive behavior after a deviation from collusive behavior. This is exemplified by the WSLS architecture. Part of this strategy can be interpreted as consisting of a RAP scheme (both players pricing at the low price given a unilateral deviation from a collusive price), but it also consists of a part that prescribes the coordinated increase of prices after both players have

used the competitive price in the previous period. This provides a refinement of RAP schemes for algorithmic collusion.

Simulations of DQ in the three-price environment considered here confirm that the theoretically predicted level of collusion is accurate. Introducing fluctuations into the learning process by reducing the batch size and employing a learning rate between batches leads to a substantial increase in the observed level of collusion. Nevertheless, this result remains consistent with the conclusion that an intermediate price does not facilitate collusion, since even higher levels of collusion are observed in the two-price setting when fluctuations are present. The influence of fluctuations demonstrates that algorithms may be designed in ways that increase the likelihood of learning collusive behavior. Such design choices are not, however, obviously anticompetitive. Algorithms incorporating these features therefore occupy an intermediate position between algorithms that learn to collude autonomously (Calvano, Calzolari, Denicoló, and Pastorello 2020) and algorithms that are explicitly designed to collude (Meylahn and Boer 2022).

Future work. The results presented here suggest several promising directions for future research. Throughout the analysis, we have assumed that the game is symmetric. Extending the framework to asymmetric settings would require a corresponding extension of IBR graph theory. In this context, it is useful to distinguish between asymmetries arising from the profit function and asymmetries arising from algorithmic hyperparameters. The former is likely to be more challenging, whereas in the latter case both players would share the same space of IBR graphs while potentially possessing different IBR graphs within that space.

We have also assumed that the intermediate price lies exactly halfway between the competitive and monopoly prices. This assumption is natural because pricing environments are typically based on uniformly discretized price grids. Nevertheless, it would be valuable to investigate how the basin of attraction of collusive strategy equilibria changes as the intermediate price moves closer to, or further away from, the competitive price.

Our state space is based on a memory of one period, which has become standard in the algorithmic-collusion literature. However, alternative state-space specifications may be more conducive to collusive behavior. Examples include state spaces with longer memories or state spaces constructed from partitions of longer histories.

Finally, the most natural extension of the present work is to investigate how the basins of attraction change as additional intermediate prices are introduced. Although a fully algebraic analysis is unlikely to be feasible, estimating properties of the IBR graph could be tractable for four prices, resulting in approximately 4 billion nodes. At five prices, however, the graph expands dramatically to roughly 300 quadrillion nodes, making further analysis of this kind substantially more challenging. This is predominantly caused by the growth of the state space, which could be counteracted by considering state-space abstractions.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability

Code to reproduce the results is available at: <https://github.com/JanuszMeylahn/Intermediate-price-facilitate-collusion>.

References

- I. Abada, J. E. Harrington Jr., X. Lambin, and J. M. Meylahn. 2024. "Algorithmic Collusion: Where Are We and Where Should We Be Going?" Available at SSRN 4891033.
- I. Abada and X. Lambin. 2023. "Artificial intelligence: Can seemingly collusive outcomes be avoided?" *Management Science*, 69, 9, 5042–5065. doi:10.1287/mnsc.2022.4623.

- I. Abada, X. Lambin, and N. Tchakarov. 2024. "Collusion by Mistake: Does Algorithmic Sophistication Drive Supra-competitive Profits?" *European Journal of Operational Research*, 318, 3, 927–953. doi:10.1016/j.ejor.2024.06.006.
- C. Alós-Ferrer and A. Ritschel. 2018. "The reinforcement heuristic in normal form games." *Journal of Economic Behavior & Organization*, 152, 224–234. doi:10.1016/j.jebo.2018.06.014.
- L. Andreozzi, M. Ploner, and A. S. Saral. 2020. "The stability of conditional cooperation: beliefs alone cannot explain the decline of cooperation in social dilemmas." *Scientific Reports*, 10, 1, 13610. doi:10.1038/s41598-020-70681-z.
- G. Arslan and S. Yüksel. 2016. "Decentralized Q-learning for stochastic teams and games." *IEEE Transactions on Automatic Control*, 62, 4, 1545–1558. doi:10.1109/TAC.2016.2598476.
- W. Barfuss and J. M. Meylahn. 2023. "Intrinsic fluctuations of reinforcement learning promote cooperation." *Scientific Reports*, 13, 1, 1309. doi:10.1038/s41598-023-27672-7.
- J. Batali and P. Kitcher. 1995. "Evolution of altruism in optional and compulsory games." *Journal of Theoretical Biology*, 175, 2, 161–171. doi:10.1006/jtbi.1995.0128.
- Q. Bertrand, J. A. Duque, E. Calvano, and G. Gidel. 2025. "Self-Play Q-Learners Can Provably Collude in the Iterated Prisoner's Dilemma." In: *International Conference on Machine Learning*.
- A. V. den Boer. 2015. "Dynamic pricing and learning: historical origins, current research, and new directions." *Surveys in operations research and management science*, 20, 1, 1–18. doi:10.1016/j.sorms.2015.03.001.
- A. V. den Boer and J. M. Meylahn. 2024. "A (mathematical) definition of algorithmic collusion." Available at SSRN 5012923.
- A. V. den Boer, J. M. Meylahn, and M. P. Schinkel. 2026. "Artificial Collusion: Examining Supracompetitive Pricing by Q-Learning Algorithms." *Management Science*, 0, 0. eprint: <https://doi.org/10.1287/mnsc.2024.08557>. doi:10.1287/mnsc.2024.08557.
- V. S. Borkar. 2022. *Stochastic Approximation: A Dynamical Systems Viewpoint: Second Edition*. Texts and Readings in Mathematics. Hindustan Book Agency. ISBN: 9788195196111. https://books.google.nl/books?id=k_ChEAAQBAJ.
- Y. Breitmoser. 2015. "Cooperation, but no reciprocity: Individual strategies in the repeated prisoner's dilemma." *American Economic Review*, 105, 9, 2882–2910. doi:10.1257/aer.20130675.
- E. Calvano, G. Calzolari, V. Denicoló, J. E. Harrington Jr., and S. Pastorello. 2020. "Protecting consumers from collusive prices due to AI." *Science*, 370, 6520, 1040–1042. doi:10.1126/science.abe3796.
- E. Calvano, G. Calzolari, V. Denicoló, and S. Pastorello. 2020. "Artificial Intelligence, Algorithmic Pricing, and Collusion." *American Economic Review*, 110, 10, 3267–97. doi:10.1257/aer.20190623.
- B. Cesaret and A. Bayram. 2024. "Competitive Dynamic Pricing Under Capacity Constraints: An Experimental Study." *Production and Operations Management*, 10591478241255667. doi:10.1177/10591478241255667.
- M. Chen and Z.-L. Chen. 2015. "Recent developments in dynamic pricing research: multiple products, competition, and limited demand information." *Production and Operations Management*, 24, 5, 704–731. doi:10.1111/poms.12295.
- W. L. Cooper, T. Homem-de-Mello, and A. J. Kleywegt. 2015. "Learning and pricing with models that do not explicitly incorporate competition." *Operations research*, 63, 1, 86–103. doi:10.1287/opre.2014.1341.
- M. A. Costa-Gomes and G. Weizsäcker. 2008. "Stated beliefs and play in normal-form games." *The Review of Economic Studies*, 75, 3, 729–762. doi:10.1111/j.1467-937X.2008.00498.x.
- A. Dolgoplov. 2024. "Reinforcement learning in a prisoner's dilemma." *Games and Economic Behavior*, 144, 84–103. doi:<https://doi.org/10.1016/j.geb.2024.01.004>.
- N. Eschenbaum, F. Mellgren, and P. Zahn. 2022. "Robust Algorithmic Collusion." *arXiv preprint arXiv:2201.00345*. doi:10.48550/arXiv.2201.00345.
- A. Ezrachi and M. E. Stucke. 2017. "Artificial intelligence & collusion: When computers inhibit competition." *U. Ill. L. Rev.*, 1775. doi:10.2139/ssrn.2591874.
- A. Ezrachi and M. E. Stucke. 2023. "Algorithmic tacit collusion." In: *Research Handbook on Cartels*. Edward Elgar Publishing, 187–212.
- P. Famil Alamdar and A. Seifi. 2023. "Dynamic pricing of differentiated products under competition with reference price effects using a neural network-based approach." *Journal of Revenue and Pricing Management*, 1–13. doi:10.1057/s41272-023-00444-8.
- D. Garcia, M. C. Janssen, and R. Shopova. 2023. "Dynamic pricing with uncertain capacities." *Management Science*, 69, 9, 5275–5297. doi:10.1287/mnsc.2022.4613.
- T. J. Gerpott and J. Berends. 2022. "Competitive pricing on online markets: a literature review." *Journal of Revenue and Pricing Management*, 21, 6, 596. doi:10.1057/s41272-022-00390-x.
- D. E. Griffith and R. T. Rust. 1997. "The price of competitiveness in competitive pricing." *Journal of the Academy of Marketing Science*, 25, 109–116. doi:10.1007/BF02894346.
- K. T. Hansen, K. Misra, and M. M. Pai. 2021. "Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms." *Marketing Science*, 40, 1, 1–12. doi:10.1287/mksc.2020.1276.
- F. Harary, R. Z. Norman, and D. Cartwright. 1965. *Structural models: An introduction to the theory of directed graphs*. John Wiley & Sons, Inc.
- J. E. Harrington Jr.. 2019. "Developing competition law for collusion by autonomous price-setting agents." *Journal of Competition Law & Economics*, 14, 3, 331–363. doi:10.1093/joclec/nhy016.

- J. E. Harrington Jr.. 2022. "The effect of outsourcing pricing algorithms on market competition." *Management Science*, 68, 9, 6889–6906. doi:[10.1287/mnsc.2021.4241](https://doi.org/10.1287/mnsc.2021.4241).
- M. Hettich. 2021. "Algorithmic Collusion: Insights from Deep Learning." Available at SSRN 3785966. doi:[10.2139/ssrn.3785966](https://doi.org/10.2139/ssrn.3785966).
- P. Jehiel. 2005. "Analogy-based expectation equilibrium." *Journal of Economic Theory*, 123, 2, 81–104. doi:[10.1016/j.jet.2003.12.003](https://doi.org/10.1016/j.jet.2003.12.003).
- A. Kastius and R. Schlosser. 2022. "Dynamic pricing under competition using reinforcement learning." *Journal of Revenue and Pricing Management*, 21, 1, 50–63. doi:[10.1057/s41272-021-00285-3](https://doi.org/10.1057/s41272-021-00285-3).
- T. Klein. 2021. "Autonomous Algorithmic Collusion: Q-learning under Sequential Pricing." *RAND Journal of Economics*, 52, 3, 538–558. doi:[10.1111/1756-2171.12383](https://doi.org/10.1111/1756-2171.12383).
- M. Knez and C. Camerer. 2000. "Increasing cooperation in prisoner's dilemmas by establishing a precedent of efficiency in coordination games." *Organizational behavior and human decision processes*, 82, 2, 194–216. doi:[10.1006/obhd.2000.2882](https://doi.org/10.1006/obhd.2000.2882).
- X. Lambin. 2024. "Less Than Meets the Eye: Simultaneous Experiments as a Source of Algorithmic Seeming Collusion." Available at SSRN 4498926. doi:[10.2139/ssrn.4498926](https://doi.org/10.2139/ssrn.4498926).
- T. Loots and A. V. den Boer. 2022. "Data-driven collusion and competition in a pricing duopoly with multinomial logit demand." *Production and Operations Management*, 32, 4, 1169–1186. doi:[10.1111/poms.13919](https://doi.org/10.1111/poms.13919).
- S. K. Mehra. 2016. "Antitrust and the Robo-Seller: Competition in the Time of Algorithms." *Minnesota Law Review*, 100, 1323–1375. <https://api.semanticscholar.org/CorpusID:46992378>.
- J. M. Meylahn. 2025. "Quantifying the likelihood of learning collusive strategy equilibria." *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35, 8.
- J. M. Meylahn and A. V. den Boer. 2022. "Learning to collude in a pricing duopoly." *Manufacturing & Service Operations Management*, 24, 5. doi:[10.1287/msom.2021.1074](https://doi.org/10.1287/msom.2021.1074).
- J. M. Meylahn and L. Janssen. 2022. "Limiting dynamics for Q-learning with memory one in symmetric two-player, two-action games." *Complexity*, 2022, 4830491, 1–20. doi:[10.1155/2022/4830491](https://doi.org/10.1155/2022/4830491).
- A. S. Saral. 2020. *Evolution of Conditional Cooperation in Prisoner's Dilemma*. Tech. rep. Center for Open Science. doi:[10.31219/osf.io/wcpkz](https://doi.org/10.31219/osf.io/wcpkz).
- F. Shigenaka, T. Sekiguchi, A. Iwasaki, and M. Yokoo. Feb. 2017. "Achieving Sustainable Cooperation in Generalized Prisoner's Dilemma with Observation Errors." *Proceedings of the AAAI Conference on Artificial Intelligence*, 31, 1, (Feb. 2017). doi:[10.1609/aaai.v31i1.10577](https://doi.org/10.1609/aaai.v31i1.10577).
- S. R. Sinclair, S. Banerjee, and C. L. Yu. 2023. "Adaptive discretization in online reinforcement learning." *Operations Research*, 71, 5, 1636–1652. doi:[10.1287/opre.2022.2396](https://doi.org/10.1287/opre.2022.2396).
- Y. Usui and M. Ueda. 2021. "Symmetric equilibrium of multi-agent reinforcement learning in repeated prisoner's dilemma." *Applied Mathematics and Computation*, 409, 126370. doi:[10.1016/j.amc.2021.126370](https://doi.org/10.1016/j.amc.2021.126370).
- Q. Wang, Y. Huang, and P. V. Singh. 2022. "Algorithms, Artificial Intelligence and Simple Rule Based Pricing." Available at SSRN 4144905. doi:[10.2139/ssrn.4144905](https://doi.org/10.2139/ssrn.4144905).
- Z. Xu and W. Zhao. 2024. "On Mechanism Underlying Algorithmic Collusion." *arXiv preprint arXiv:2409.01147*.
- Y. Yang, Y.-C. Lee, and P.-A. Chen. 2024. "Competitive Demand Learning: A Noncooperative Pricing Algorithm with Coordinated Price Experimentation." *Production and Operations Management*, 33, 1, 48–68. doi:[10.1177/10591478231224912](https://doi.org/10.1177/10591478231224912).

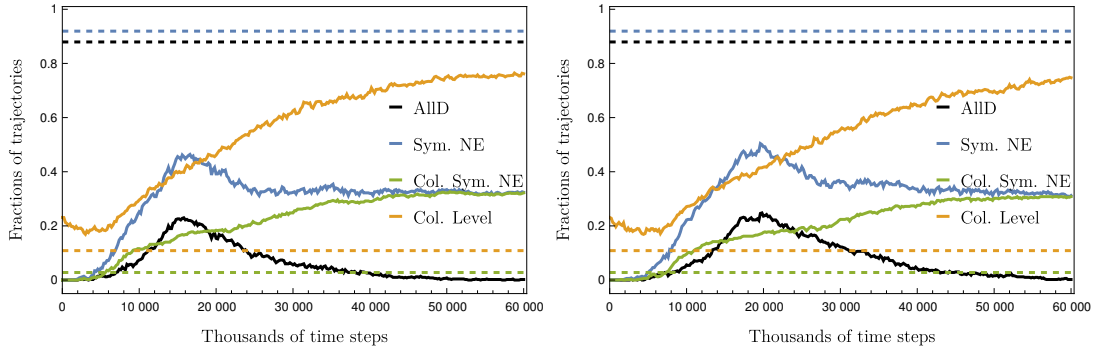


Fig. 5. Fraction of trajectories in the AllD strategy pair, symmetric NE, Collusive, and symmetric NE and the level of collusion of the DQ algorithm. Here, we define the level of collusion to be the fraction of times during a batch that the players played (p_M, p_M) or (p_H, p_H) . For both simulations, we use the payoffs as defined in Table 1, we set $\delta = 0.95$, $\epsilon = 0.1$, $K = 2 \times 10^5$ and simulate 500 trajectories. In the left plot, we use $\lambda = 0.2$ and in the plot on the right we use $\lambda = 0.3$. The inter-batch learning rate is $\hat{\alpha} = 0.4$. The dashed lines indicate the theoretical basin of attraction in the large batch size limit, as in Table 3 and Table 6.

A Interplay of Inertia and Fluctuations

In Figure 4, we show that introducing fluctuations to the learning process drastically increases the level of collusion and offer a hypothesis regarding why this is the case. In this appendix, we discuss the possibility that the inertia parameter λ is a driver of this increase in collusion. Note that the IBR graph theory developed in (Meylahn 2025) shows that, at least in the large batch-size limit with an intra-batch learning rate of $\hat{\alpha} = 1$, the basins of attraction are independent of λ . However, introducing fluctuations and the intra-batch learning rate may change this. To investigate the effect of λ , we repeat the simulation of the left plot of Figure 4 with λ set to 0.2 and 0.3 in Figure 5. We find that the results are very similar to those in Figure 4, except that the time it takes for the dynamics to stabilize is longer as λ increases. This is natural, as a larger value of λ means that there are more batches after which one or both of the players do not update their strategies, which is bound to slow down the dynamics. A further argument for the independence of the basins of attraction from λ , is that λ changes the probability of an update occurring between batches but does not change the expected dynamics.

B Strategy Graphs

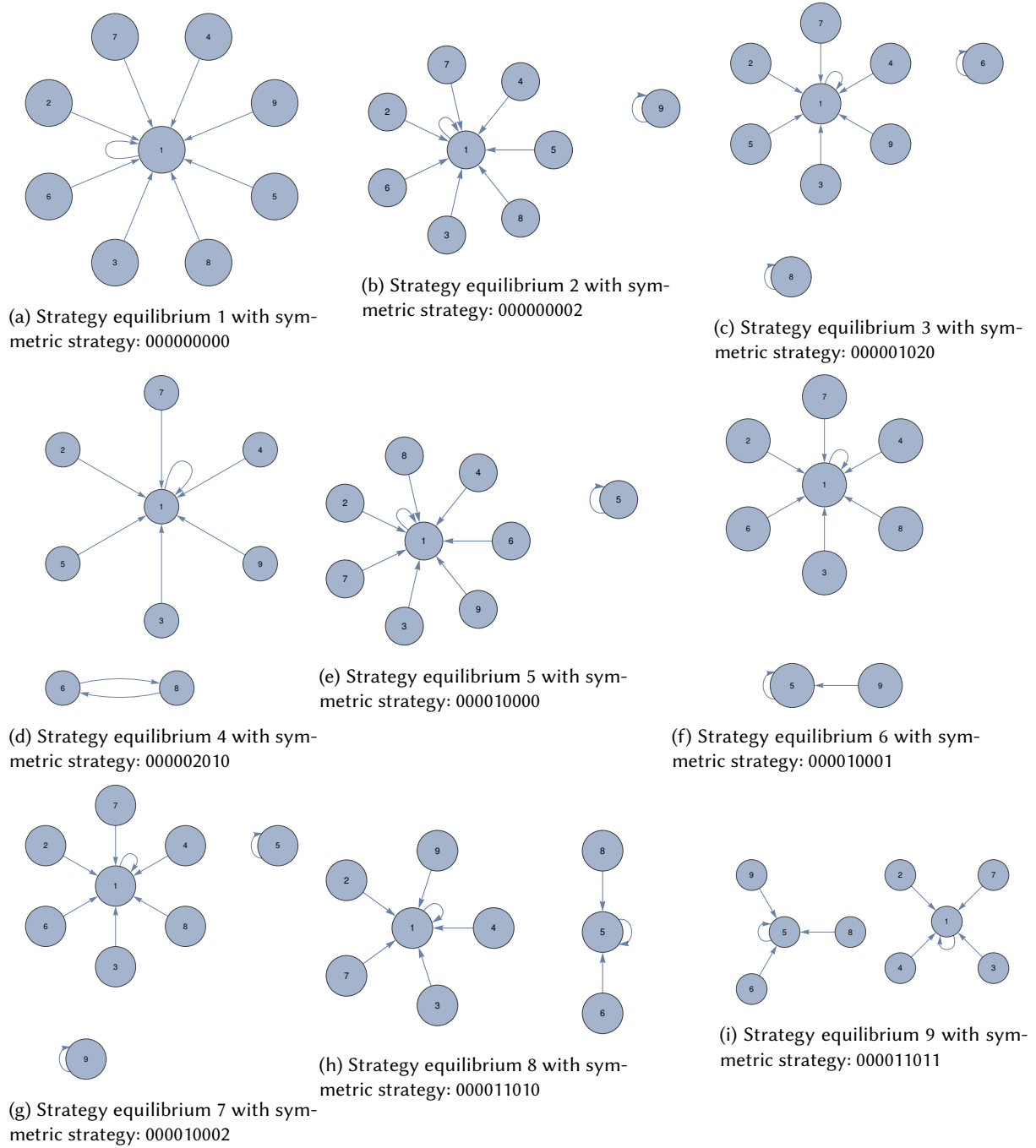


Fig. 6. Strategy equilibria 1–9.

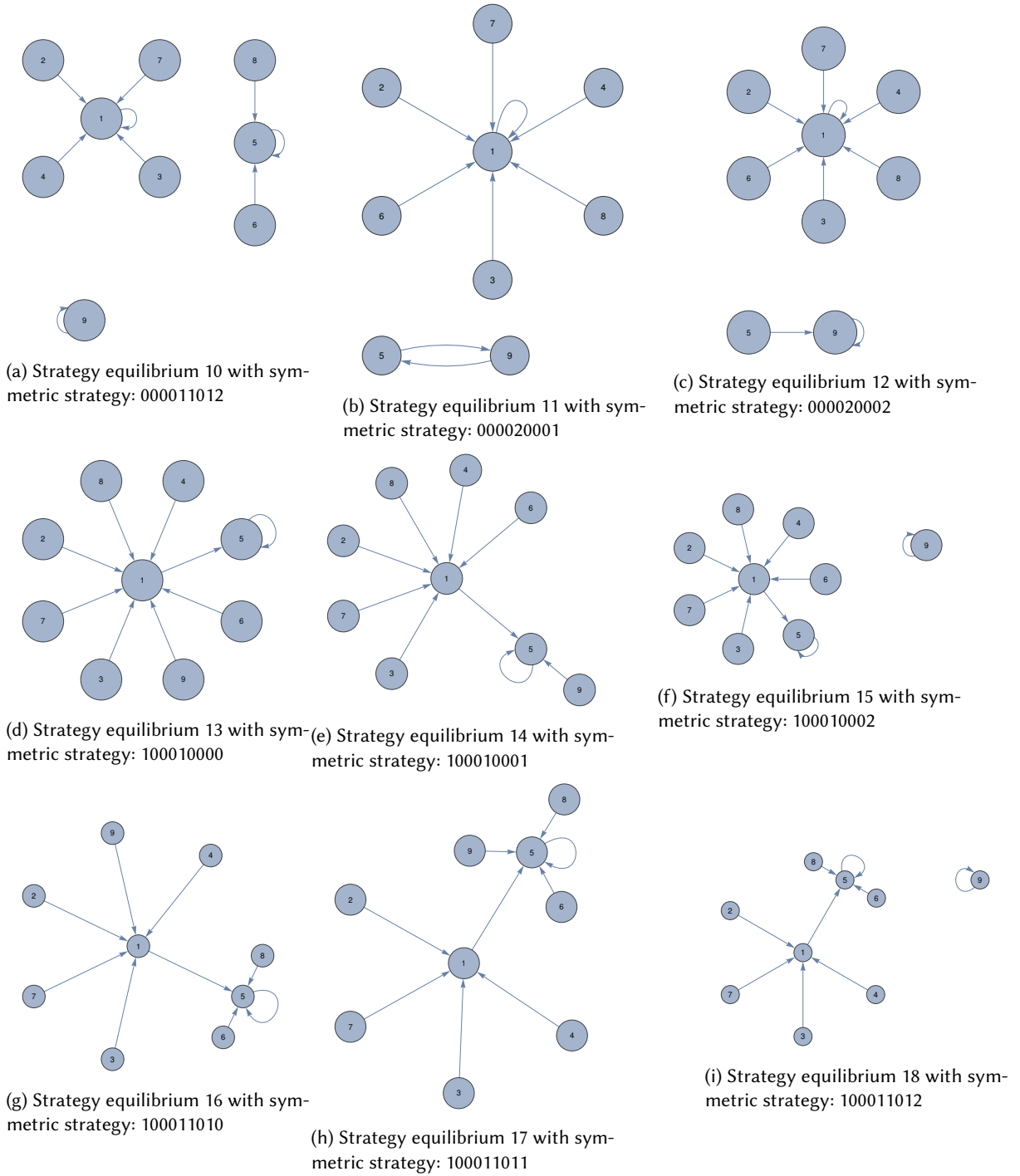


Fig. 7. Strategy equilibria 10–18.

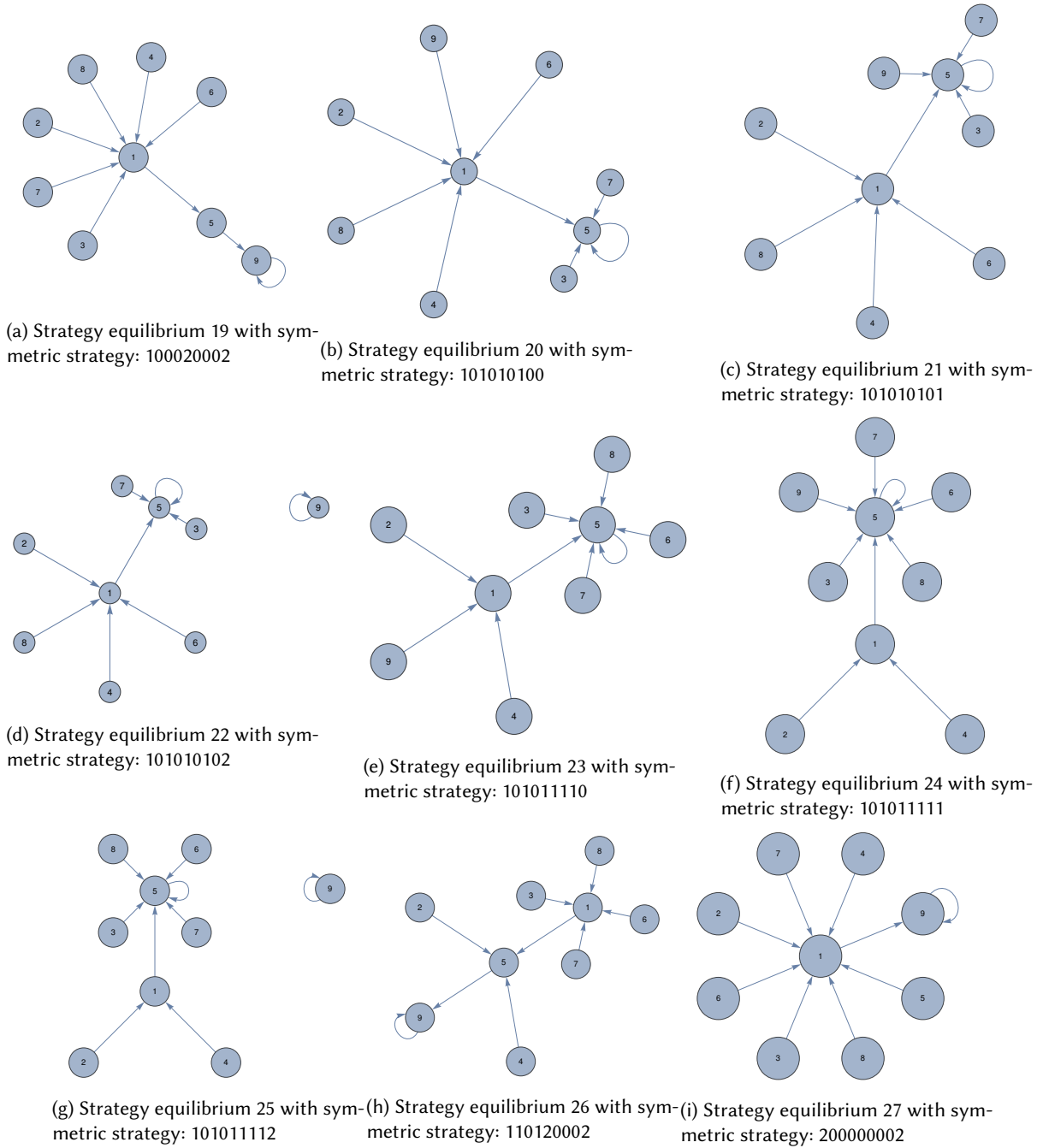
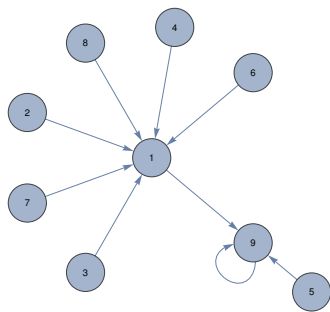
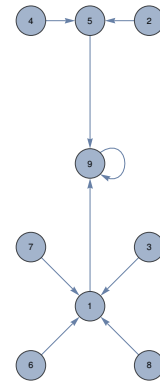


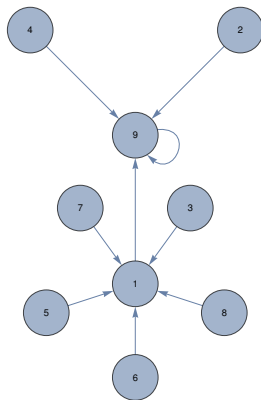
Fig. 8. Strategy equilibria 19–27.



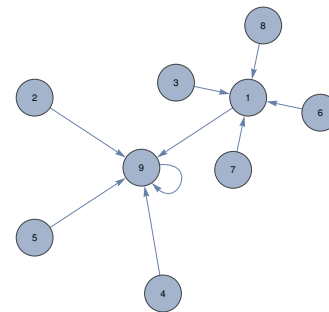
(a) Strategy equilibrium 28 with symmetric strategy: 200020002



(b) Strategy equilibrium 29 with symmetric strategy: 210120002



(c) Strategy equilibrium 30 with symmetric strategy: 220200002



(d) Strategy equilibrium 31 with symmetric strategy: 220220002

Fig. 9. Strategy equilibria 28–31.

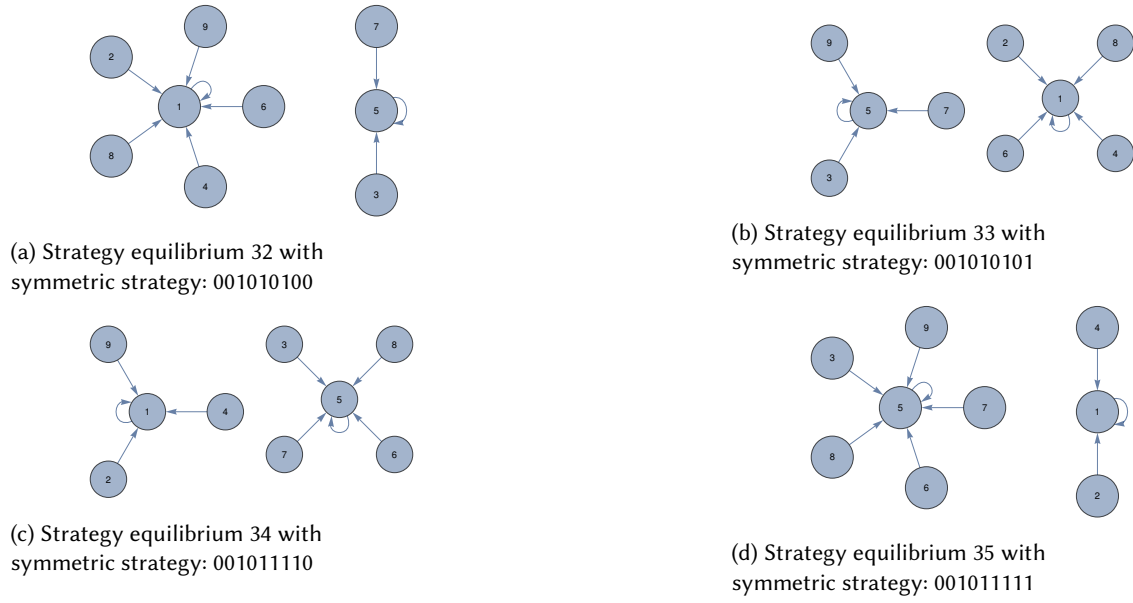


Fig. 10. Strategy equilibria 32–35.

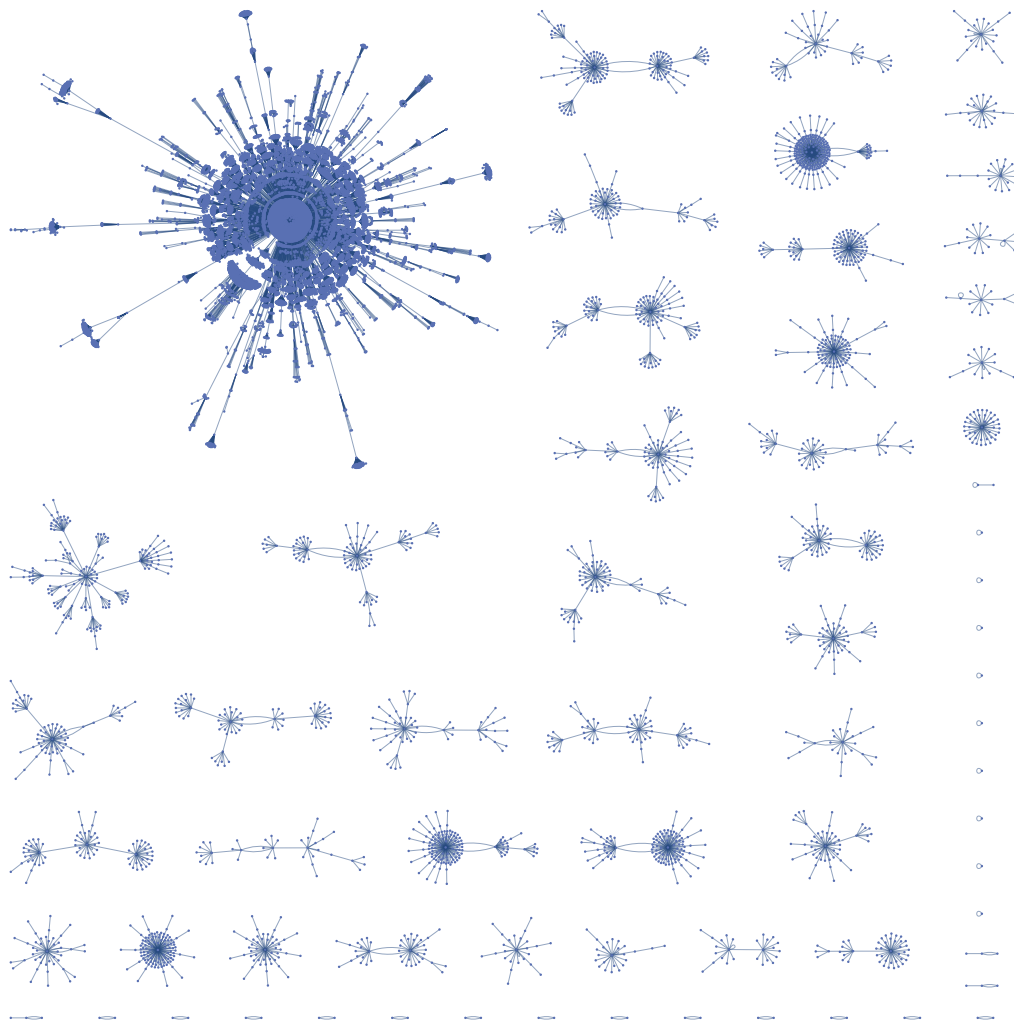


Fig. 11. IBR graph for the payoffs given in Table 1 with a discount factor of $\delta = 0.95$. The graph gives an impression of the typical distribution of component sizes, clearly showing the largest component corresponding to the ALLD strategy equilibrium (top left).

In Figures 6, 7, 8, 9 and 10, we show all symmetric strategy equilibria possible with the baseline parametrisation and $\delta = 0.99$. Figure 11 shows an example of an IBR graph.

Received 20 July 2026; accepted 18 September 2026