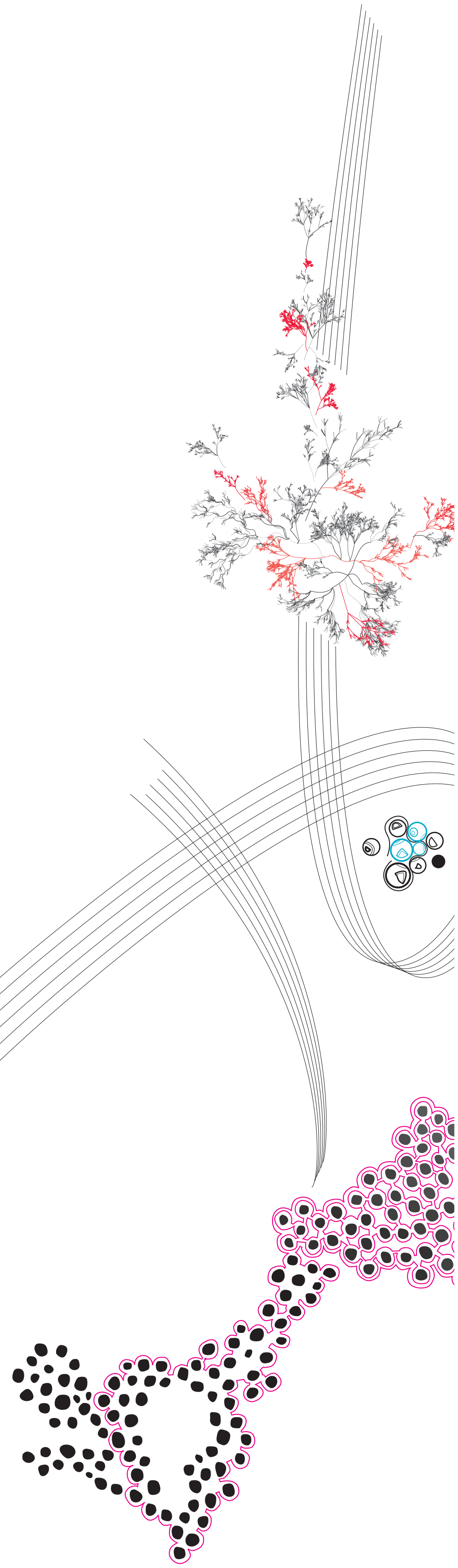


BSc Thesis Applied Mathematics

Average-reward reinforcement learning in two-player two-action games

Niels van Duijl

Supervisors: J. Meylahn and M. Hahn

June 28, 2024

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

This article was written for my Bachelor Assignment of Applied Mathematics and Technical Computer Science at the University of Twente.

I want to thank Janusz Meylahn for the idea of the thesis and his excellent supervision of the project. He was always very quick to respond to my emails, making meetings very easy to set up. He also provided good feedback very quickly and was always glad to help if I got stuck. I also want to thank Moritz Hahn for being willing to be a second supervisor for the project, and for his feedback on this article.

Average-reward reinforcement learning in two-player two-action games

Niels van Duijl

June 28, 2024

Abstract

Reinforcement learning (RL) algorithms are typically designed for stationary environments. However, as these algorithms are increasingly applied in real-world settings, more situations occur in which they interact with each other, such as algorithmic pricing. In this scenario, multiple agents try to learn optimal prices to maximise profit in a market. It is detrimental to consumers if these agents learn to cooperate by all setting high prices. Using average reward RL makes more sense in these settings than using the discounted counterpart.

We compare the use of average-reward methods to the discounted counterpart used in [11] in the prisoner's dilemma, stag hunt and snowdrift games with one-period memory. We analyse the differences in the individual best-response graphs, the basins of attraction of the Nash equilibria, and the effect of exploration. From our results, we conclude that average-reward RL gives very similar results to discounted-reward RL in these two-player two-action games. Therefore, it could be possible to apply average-reward RL in multiagent settings without much change in the Nash equilibria.

Keywords: average-reward, reinforcement learning, multiagent, individual best-response graphs,

Contents

1 Introduction

Reinforcement learning (RL) algorithms (See [18] for an introduction) are typically designed for stationary environments. However, as these algorithms are increasingly applied in real-world settings, more situations occur in which they interact with each other. However, the consequences of this interaction are hard to predict.

As for the theoretical work on multiagent RL, [1] introduces a decentralised Q-learning algorithm and proves convergence in the settings of weakly acyclic games. The algorithm introduced keeps the policies of the agents constant during periods of a fixed length, called batches, to simulate a stationary environment which ensures convergence. In [11], a similar idea is applied in a number of two-player two-action games with one-period memory to see under what circumstances the algorithm converges to different equilibria.

Specifically, the two-player two-action games considered are the prisoner's dilemma [13], the stag hunt [14] and the snowdrift (sometimes called hawk-dove or chicken) game [17]. These games are considered because of their relevance to algorithmic pricing [10]. In this scenario, multiple agents try to learn optimal prices to maximise profit in a market. It is detrimental to consumers if these agents learn to cooperate by all setting high prices. It has been shown that RL with one-period memory can eventually learn to do this in [6] and [5].

However, these algorithms use discounted rewards and don't discuss the potential of applying average-reward RL methods, such as those introduced in [15], [16] and [7]. Using such a method instead of a discount factor might lead to different conditions under which algorithms collude. This is also preferable in some real-world settings. For example, when the time between decisions is very little, a very high discount factor is needed to ensure convergence to the optimal strategy, but this can lead to numerical instability.
We will compare the use of average-reward methods to [11] in the same two-player two-action games with one-period memory. We will do this by analysing the differences in the individual best-response graphs and the basins of attraction of the Nash equilibria, as introduced in [9]. For a more thorough introduction to the setting, see [11].

From our results, we conclude that average-reward RL gives very similar results to discounted-reward RL in these two-player two-action games. Therefore, it could be possible to apply average-reward RL in multiagent settings without much change in the Nash equilibria.

In Section 2, we describe and define the notation and the model. In Section 3, we introduce the average-reward RL methods used to solve the model. In Section 4, we present the results. We first discuss the results and then compare them to the discounted reward equivalent that is used in [11].

2 Model

In this section, we introduce notation and the model of the environment. We first define the different two-player two-action games, and then define the model of the environment that we use in the subsequent sections.

2.1 Two-player two-action games

A two-player, two-action game is a game with two players, where each player can play one of two actions. We call the available actions *defect* (D) and *cooperate* (C). Depending on what action the players choose, they get the payoff shown in Table 1.

		Agent 2	
		D	C
Agent 1	D	p/p	t/s
	C	s/t	r/r

Table 1: Payoff matrix

If both players choose to cooperate, they both get the *reward* payoff (r). If both players choose to defect, they both get the *punishment* payoff (p). If one of the players chooses to defect, and the other player chooses to cooperate, the player that defected gets the *temptation* payoff (t), and the player that cooperated gets the *sucker's* payoff (s). The games are symmetric as classified in [4], which means the payoffs are not dependent on the identity of the agents. This means both agents can think of themselves as agent 1 without any issues.

The players are not able to communicate during the game. In this article, we only consider the situation where this game is repeated indefinitely. This allows the players to adapt their choice of action based on what their opponent has played in the previous rounds.

We can define different games of this form by changing the ordering of the payoffs t, r, p and s. We consider three examples of these games in this paper, as defined below. The definitions used are consistent with [11] to allow for a good comparison.

Prisoner's dilemma

In the prisoner's dilemma, first introduced in [13], the players represent two prisoners who are being interrogated. The police need more evidence to prosecute the pair for a major crime but do have enough evidence to convict them for lesser charges. The prisoners can choose to either testify against the other prisoner to reduce their sentence (D) or stay silent (C). If the prisoners both defect, they both get prosecuted but also have slightly reduced sentences for testifying. If they both cooperate, they both get prosecuted for the smaller crime. If prisoner 1 defects and prisoner 2 cooperates, prisoner 1 gets a very light sentence while prisoner 2 gets the heaviest sentence possible. We mathematically define this game by assuming the following ordering of the payoffs:

$$t > r > p > s.$$

Stag hunt

In the stag hunt, first introduced in [14], the players represent hunters who need to decide which animal to hunt. They can each choose to hunt either the stag (C) or a hare (D). However, the hunters need to work together if they want to successfully kill the stag. If only one hunter chooses to hunt the stag, that hunter will fail and receive nothing. If both hunters catch a hare, the supply is larger so the price of the hares drops. We mathematically define this game by assuming the following ordering of the payoffs:

$$r > t > p > s.$$

Snowdrift

In the snowdrift game, first introduced in [17], the players represent drivers on opposite sides of a snow-covered road. They need the road to be cleared of snow to continue driving, but they would rather stay in their warm car and let the other driver do the work in the cold. Thus, the drivers have the option to stay in the car (D) or shovel the snow (C). If both drivers shovel, the road is cleared quickly but both drivers will be cold. If both drivers stay in the car, they cannot drive on. If only one driver shovels the snow, they will remain in the cold for longer as it takes them longer to shovel the snow on their own. The other driver stays comfortably in their warm car. This game is also often called the chicken or hawk-dove game. The scenarios are different, but the mathematical definition is the same. We mathematically define this game by assuming the following ordering of the payoffs:

$$t > r > s > p.$$

2.2 Environment

We now introduce the model of the environment. It is very similar to the model used in [11].

Definition 2.2.1. *The state space is $S = \{(D, D), (D, C), (C, D), (C, C)\}$. A state σ is an element of S.*

The state space is based on the last action played by both players. One-period memory suffices for both agents to successfully compute the optimal responses to their opponent's strategy [3]. Initially, a random state is selected as the starting state. The current state is relative to the agent. The first symbol is the action the agent itself played last, and the second symbol is the action the other agent played last. This does mean the states are asymmetric, since whenever agent 1 is in state (D, C), agent 2 is in state (C, D).

Definition 2.2.2. *The action space is $A = \{D, C\}$. An action a is an element of A.*

The action space is defined by the actions that the players can take. Since both agents can choose to either defect or cooperate in every state, A does not depend on the current state.

As for the policies, we consider ϵ-greedy stationary policies. Here $\epsilon \in (0, 1)$ is the exploration parameter. That is, every policy $\pi \colon S \to A$ chooses an action deterministically in each state, not dependent on the time (number of rounds played) of the system, but the agent executes a random action instead of $\pi(\sigma)$ with probability ϵ. This means both agents play action $\pi(\sigma)$ with probability $1 - \frac{\epsilon}{2}$, and the other action with probability $\frac{\epsilon}{2}$. This is different from the policies used in [11], where no exploration parameter is used.

Definition 2.2.3. *A policy $\pi\colon S \to A$ is a function that selects an action to play in each state. We write $\pi = (a_{(D,D)}, a_{(D,C)}, a_{(C,D)}, a_{(C,C)})$ where $a_i = \pi(i)$.*

We use π_1 and π_2 to distinguish between the agent's own policy and the policy of the opponent. Common policies that we will often discuss in this article are all defect (D, D, D, D), all cooperate (C, C, C, C), grim trigger (D, D, D, C), introduced in [8] and win-stay, lose-shift (C, D, D, C), extensively analysed in [12].

For convenience, we will often use the binary encoding of the states, actions and policies by setting $D \equiv 0$ and $C \equiv 1$. We then get $S = \{0, 1, 2, 3\}$ and $\pi \in \mathbb{Z}$ with $0 \leq \pi \leq 15$. For example, policy 0 represents all defect (D, D, D, D), and policy 9 is win-stay, lose-shift (C, D, D, C).

Definition 2.2.4. *The transition probability function $P_{\pi_2}\colon S \times S \times A \to [0, 1]$ is the probability that the system will be in state σ' in the next round given that the system is now in state σ, $\pi_1(\sigma) = a$, and the opponent uses policy π_2. We write $P_{\pi_2}(\sigma' \mid \sigma, a)$.*

This is a stochastic function because of the policies being ϵ-greedy. For example, when the opponent plays all defect (policy 0), we have:

$$\begin{aligned} P_0(0 \mid 0,0) &= \left(1 - \frac{\epsilon}{2}\right)^2, \quad & P_0(1 \mid 0,0) &= \frac{\epsilon}{2}\left(1 - \frac{\epsilon}{2}\right), \\ P_0(2 \mid 0,0) &= \frac{\epsilon}{2}\left(1 - \frac{\epsilon}{2}\right), \quad & P_0(3 \mid 0,0) &= \left(\frac{\epsilon}{2}\right)^2. \end{aligned}$$

Definition 2.2.5. *The reward function $R_{\pi_2}\colon S \times A \to \mathbb{R}$ is the expected immediate reward when executing action a in state σ, given that the opponent chooses to play $\pi_2(\sigma)$. It is given by:*

$$R_{\pi_2}(\sigma, a) = P_{\pi_2}(0 \mid \sigma, a) \cdot p + P_{\pi_2}(1 \mid \sigma, a) \cdot s + P_{\pi_2}(2 \mid \sigma, a) \cdot t + P_{\pi_2}(3 \mid \sigma, a) \cdot r.$$

The transition probabilities and the expected immediate rewards are dependent on the policy that the opponent plays. However, we can fix the strategy of the opponent to find the best possible response to that strategy. When fixing the strategy of the opponent, the tuple $(S, A, P_{\pi_2}, R_{\pi_2})$ describes a Markov Decision Process (MDP). We can use well-known reinforcement learning methods to solve this MDP and find the best response to the strategy of the opponent.

3 Methods

3.1 Average reward learning

We now introduce the average-reward reinforcement learning method that we use to solve the MDP for a fixed policy of the opponent. Most of this section is based on [7]. We are looking to maximise the Bellman optimality equations. That is, we want to find the policy π_1^* that satisfies the following equation for every state σ:

$$v_b(\pi_1^*, \sigma) = \max_{a \in A}\{R_{\pi_2}(\sigma, a) + \sum_{\sigma' \in S} P_{\pi_2}(\sigma' \mid \sigma, a)\ v_b(\pi_1^*, \sigma')\} - v_g(\pi_1^*). \tag{1}$$

Here $v_b(\pi_1^*, \sigma)$ are the relative values of the states given π_1^*, and $v_g(\pi_1^*)$ is the average reward of the Markov Chain induced by π_1^*. However, solving these equations requires $v_g(\pi_1)$ to be independent of the starting state for any π_1. That is, for any policy pair (π_1, π_2), the resulting Markov Chain must be ergodic, meaning that every state can be reached from every other state with a positive probability. Due to the policies being ϵ-greedy, this condition is satisfied. Since both agents use this $\epsilon \in (0, 1)$, every state can be reached from every state regardless of the strategies of both players. This means we can solve (1) for any policy π_2.

What remains is how to compute v_g. It is possible to compute it exactly by taking the stationary distribution of the Markov Chain (as a function of ϵ) and setting $v_g = P(0) \cdot p + P(1) \cdot s + P_{\pi_2}(2) \cdot t + P(3) \cdot r$, where $P(\sigma)$ is the probability that the system is in state σ at any given time. However, we found that using the following simple estimation suggested in [7] gave the same results and faster computation time:

$$v_g(\pi_1) = \frac{1}{|S||A|} \sum_{(\sigma,a) \in S \times A} \left(R_{\pi_2}(\sigma, a) + \sum_{\sigma' \in S} P_{\pi_2}(\sigma' \mid \sigma, a)\ v_b(\pi_1, \sigma') \right). \tag{2}$$

Using this estimation has another advantage. In a simulation of this environment, the agents themselves cannot compute the exact value of the average reward as they don't know the policy of the opponent or the reward function. However, they can estimate (2).

3.2 Best response graphs

Using Mathematica, we can solve the equations of (1) algebraically to look at the individual best responses of each policy. By [1], the Q-values will eventually converge to these values with a long enough time period (batch size) during which both agents don't change their policy. Without loss of generality, we normalise the payoffs by setting the lowest reward in each of the games to 0, and the highest reward to 1.

Solving the equations for the Q-values allows us to find the conditions on the reward parameters and ϵ under which certain best response relations hold. We can analyse the possibilities by looking at the individual best-response graphs, introduced in [9].

Definition 3.2.1. *For a fixed set of parameters, the individual best-response (IBR) graph consists of vertices that correspond to the policies, with policy i having an arc to policy j whenever j is the best response to i.*

We can then look at the critical conditions under which the IBR graph changes. Then $\{(x, y, z) \in \mathbb{R}^3 \mid x, y, z \in [0, 1]\}$ is divided into different regions, where every region corresponds to a single graph. Here x, y and z represent the inner two reward parameters (different for each game) and ϵ respectively. We often analyse this space by choosing ϵ and looking at the 2D plot of the inner two reward parameters. We call this the **phase diagram** of the game, where we identify a phase with a particular IBR graph.

Definition 3.2.2. *A Nash equilibrium is a strategy pair* (π_1, π_2) *such that* π_1 *is the best response to* π_2*, and* π_2 *is the best response to* π_1*.*

In the IBR graphs, the Nash equilibria are either loops or 2-cycles. If the Nash equilibrium is a 2-cycle, we call this an asymmetric equilibrium.
For convergence of decentralised RL algorithms as in [1], the game needs to be weakly acyclic. Because the individual best responses are always unique, the games we consider are weakly acyclic as long as there is a directed path from every policy to a Nash equilibrium in the IBR graph [9].

Not all changes in the IBR graphs are interesting to analyse. For example, policy 0 and policy 1 might both be possible best responses to policy 2 for different values of the payoff parameters. However, if policy 0 is always the best response to policy 1, this change in the graph makes no practical difference. If we start with a random policy, the probability that that policy eventually leads to policy 0 is the same for both graphs. We are only interested in analysing when this probability changes for some equilibrium. For this, we define the basin of attraction.

Definition 3.2.3. *The basin of attraction* β *of a policy* π *is the set of policies that end in* π *in the IBR graph. That is:*
$\beta(\pi)$ *is the set of all policies that have a path leading to* π *if there exists a* π' *such that* (π, π') *is a Nash equilibrium.*
$\beta(\pi) = \emptyset$ *otherwise.*

In Section 4.2, we analyse the differences and similarities of the discounted and average reward approaches. Since the work in [11] is done without exploration, it is hard to know whether any changes come from average-reward learning, or only appear because of the introduction of the ϵ parameter. To partly resolve this, we let δ approach 1 in the discounted case, and ϵ approach 0 in the average-reward case.

4 Results

4.1 Best responses

We now look at the individual best responses for each of the three games. We combine all possible best-response graphs into one to show all the possible policy transitions. We then analyse under what conditions the basins of attraction of the equilibria change.

4.1.1 Prisoner's dilemma

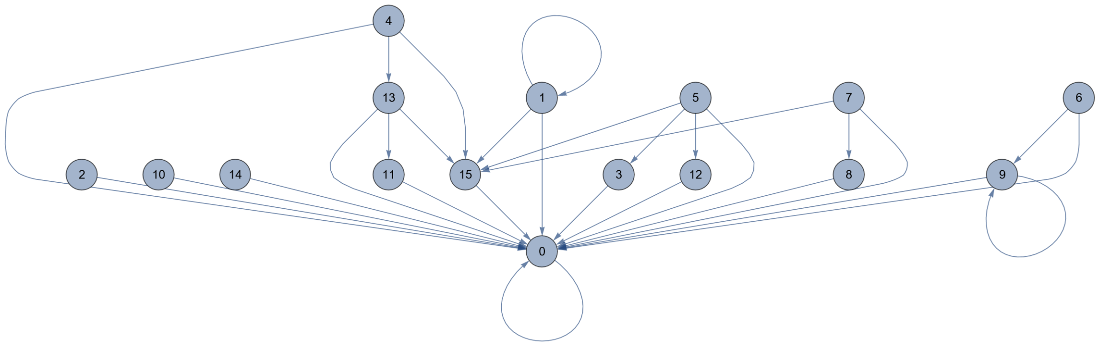


Figure 1: All possible transitions of the prisoner's dilemma.

Because the transition conditions, shown in Table 2, are much more complex than those found in [11], the amount of graphs also increased significantly. Instead of 12, there are 116 possible graphs for the prisoner's dilemma. However, this is no cause for concern, as we are not interested in what all possible graphs are, but rather when the basins of attraction of the equilibria change. By analysing the edges in Figure 1 we find the following interesting options:

Option 1: Policy 1 (D,D,D,C) is the best response to itself.

Option 2: Policy 9 (C,D,D,C) is the best response to itself.

Option 3: Policy 9 (C,D,D,C) is the best response to policy 6 (D,C,C,D).

All other policies eventually lead to 0 (D,D,D,D) no matter what the exact values of p, r and ϵ are. Under what conditions they immediately lead to 0 instead of needing 2 steps to do so is not interesting to analyse. We are only interested in finding the conditions under which the possible equilibria or the probability of converging to those equilibria change.

Taking that into account, note that the conditions under which Option 3 holds are only interesting when Option 2 holds as well. So instead of looking at all 116 possible graphs, we are only interested in the following 6 cases:

Case 1: Option 1 holds, and Option 2 does not.

Case 2: Option 1 and 2 hold, and Option 3 does not.

Case 3: Option 1, 2, and 3 all hold.

Case 4: Option 2 and 3 hold, and Option 1 does not.

Case 5: Option 2 holds, and Option 1 and 3 do not.

Case 6: Neither Option 1 nor Option 2 holds.

Thus, we want to find the critical conditions for these 6 cases and draw the graph for p and r while setting ϵ to a constant. For now, we set $\epsilon = 0.4$ and show the phase diagram. Here the numbering of the regions correspond to the case numbers.

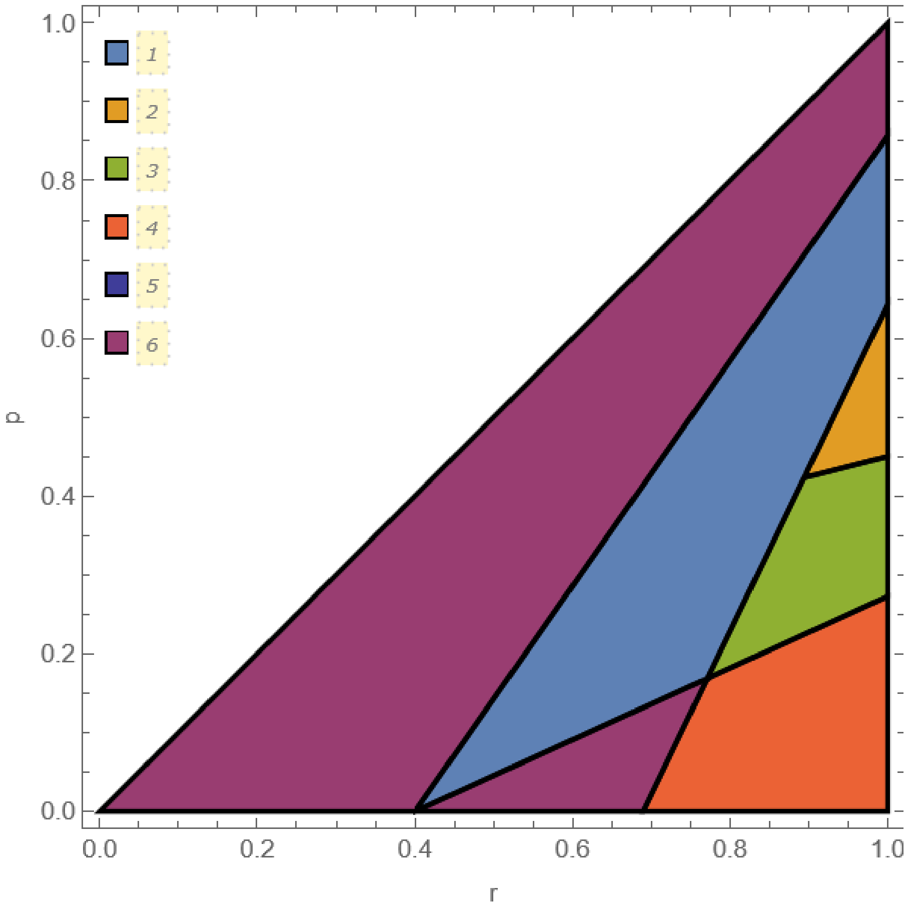


FIGURE 2: Phase diagram of the prisoner's dilemma with $\epsilon = 0.4$.

Since $r > p$, only the part of the graph below the line $r = p$ is relevant. We can see that for $\epsilon = 0.4$, every case can hold except for Case 5. It turns out Case 5 can only be true for very specific values of ϵ, approximately between 0.65 and 0.75.

With this constant ϵ, we can see that for most cases the equilibrium of all defect (policy $0 : (D, D, D, D)$) is less probable when r increases and p stays the same, and is more probable when p increases and r stays the same. It is interesting to see that this is not the case for grim trigger (policy 1: (D, D, D, C)). This policy is the best response to itself in the regions A, B and C. But when fixing r in any of these regions and letting p go to 0, starting with policy 1 leads to the equilibrium of policy 0. This is due to the fact that policy 15 (C, C, C, C) is the best response to policy 1 when r is large enough, but policy 15 always has policy 0 as its best response.

4.1.2 Stag hunt

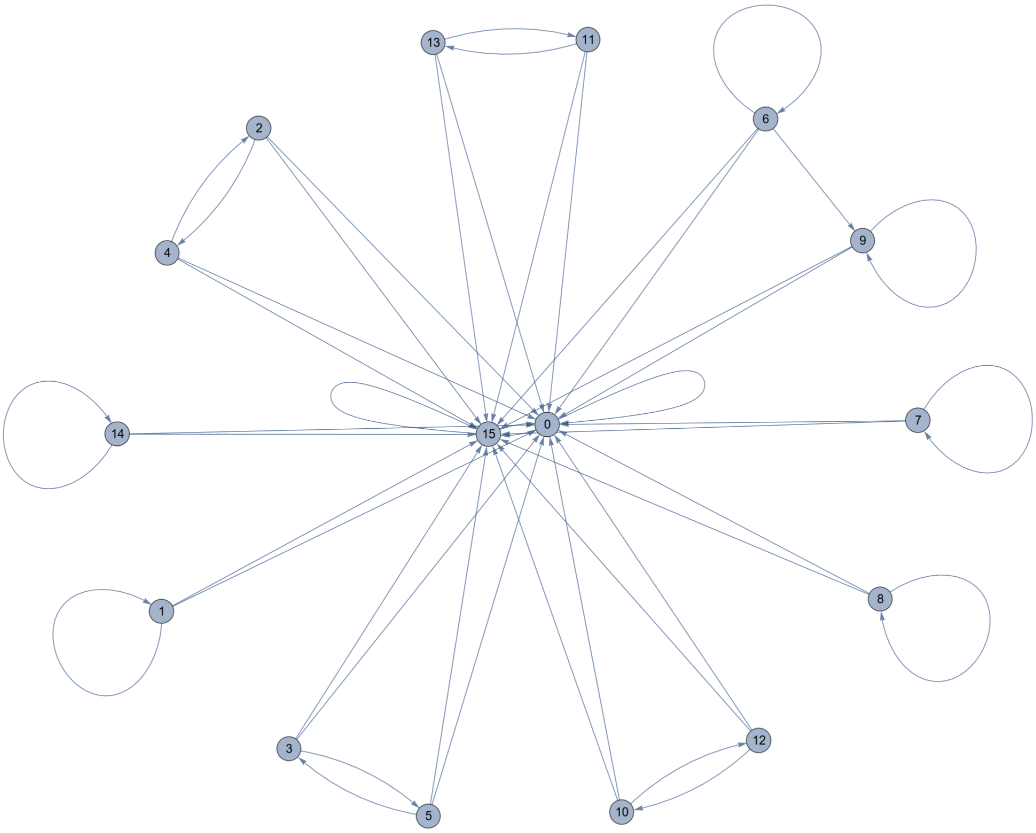


FIGURE 3: All possible transitions of the stag hunt.

Compared to the prisoner's dilemma, there are a lot more possible equilibria for the stag hunt, as shown in Figure 3. As such, there are also a lot more possible graphs that have different basins of attraction. It is not so easy to split these into different cases manually, and there are too many possibilities to get an interpretable phase diagram. However, since we are mostly interested in the comparison to the discounted rewards case, we can decrease the amount of possible transitions by looking at the possible transitions in the limit of $\epsilon \to 0$. This not only decreases the number of possible graphs significantly, but it also simplifies the equations, resulting in much faster computation of the graphs and a clearer view of the critical conditions. For the exact method used for this computation, see Appendix B.

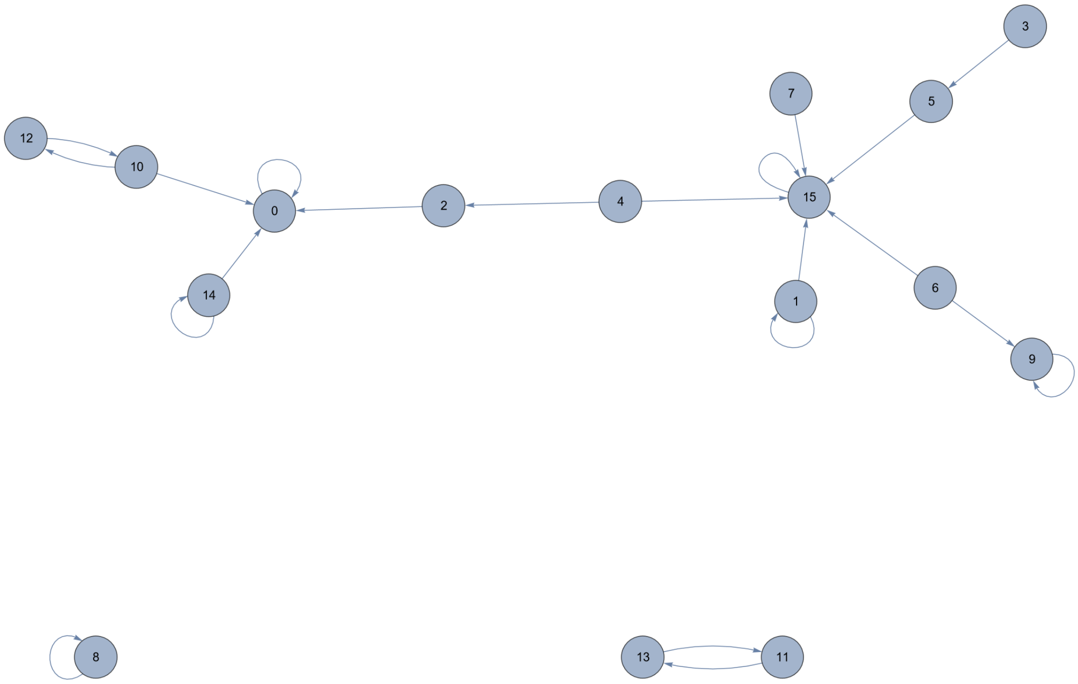


FIGURE 4: Possible transitions of the stag hunt when $\epsilon \to 0$.

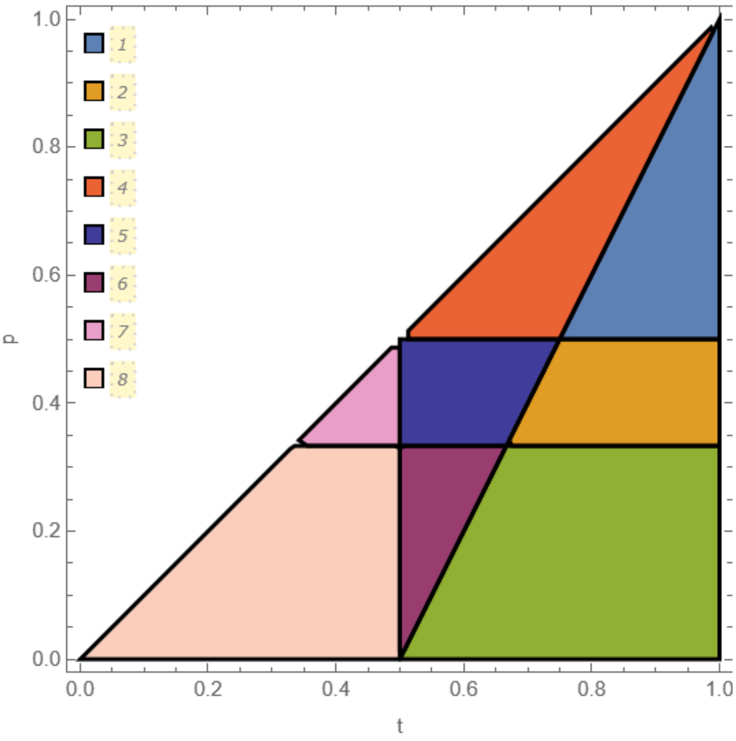


FIGURE 5: Phase diagram of the stag hunt when $\epsilon \to 0$.

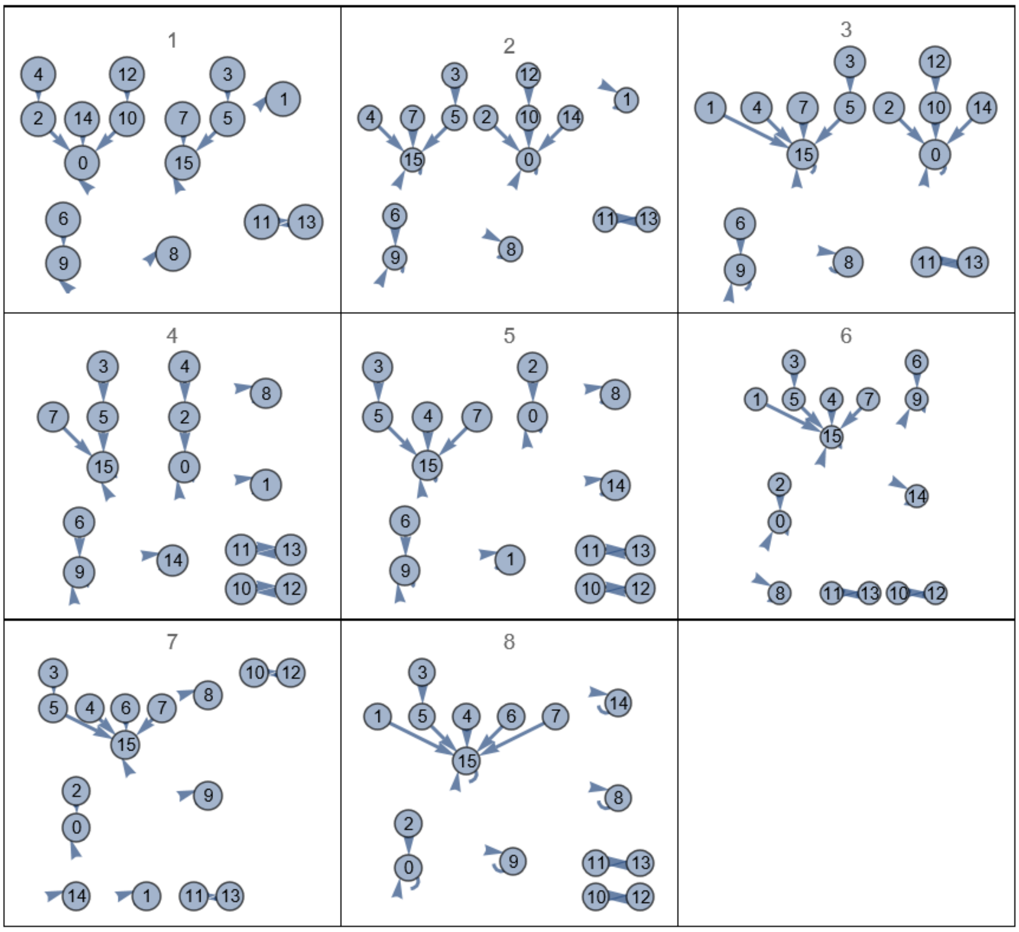


FIGURE 6: Individual best-response graphs of the stag hunt when $\epsilon \to 0$.

We find 8 possible graphs for the stag hunt. The numbers of the regions in Figure 5 correspond to the IBR graphs shown in Figure 6. Interestingly, the basins of attraction change in all of these graphs. That is, if we would define the graphs by the set of the basins of attraction of every policy, no two graphs would be equivalent. Policies 0 and 15 are always equilibria, but their basins of attraction vary significantly. Policies 11 and 13 are always an asymmetric together, with only themselves in their basin of attraction in every graph. It is the only possible equilibrium of which the basin of attraction never changes.

4.1.3 Snowdrift

As in the stag hunt, the snowdrift game has too many possible equilibria to consider the cases for all possible values of ϵ. Again, because we are mostly interested in the differences with the discounted case, we let $\epsilon \to 0$ and find the possible graphs with the reduced equations.

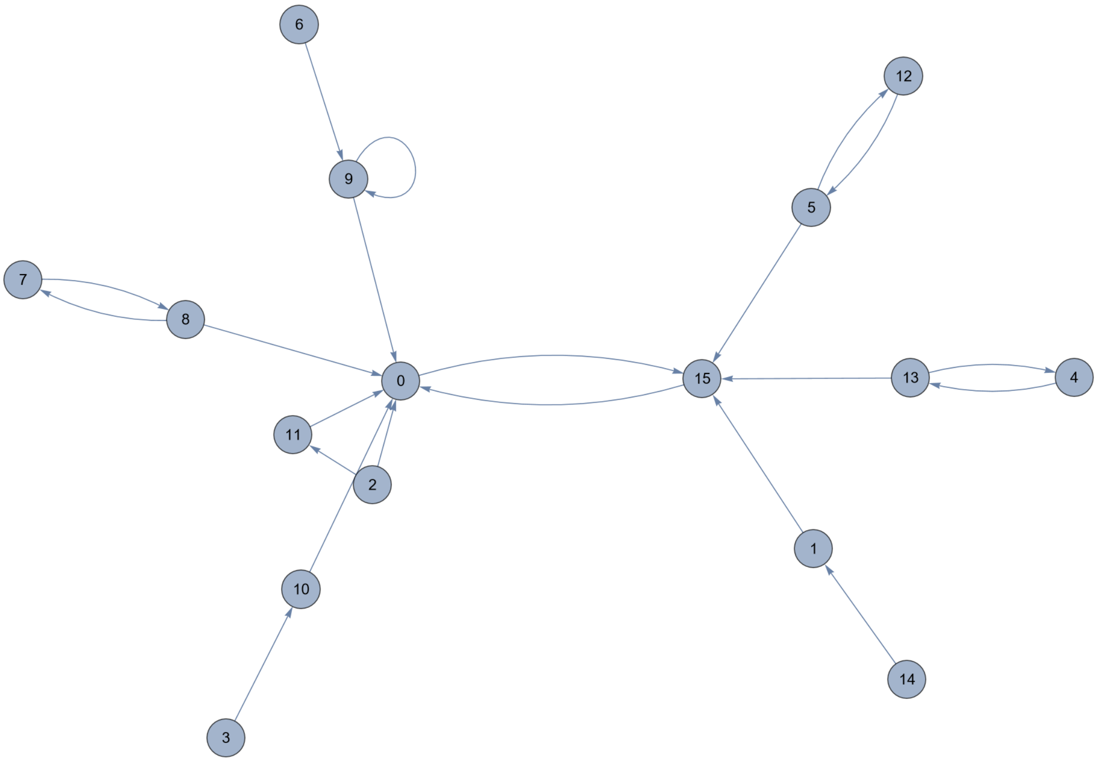


FIGURE 7: Possible transitions of the snowdrift when $\epsilon \to 0$.

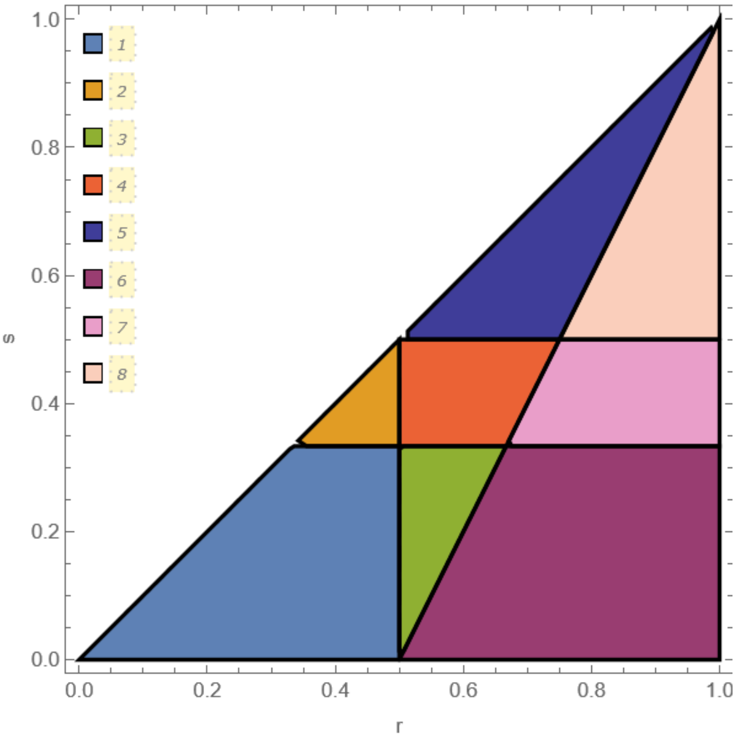


FIGURE 8: Phase diagram of the snowdrift when $\epsilon \to 0$.

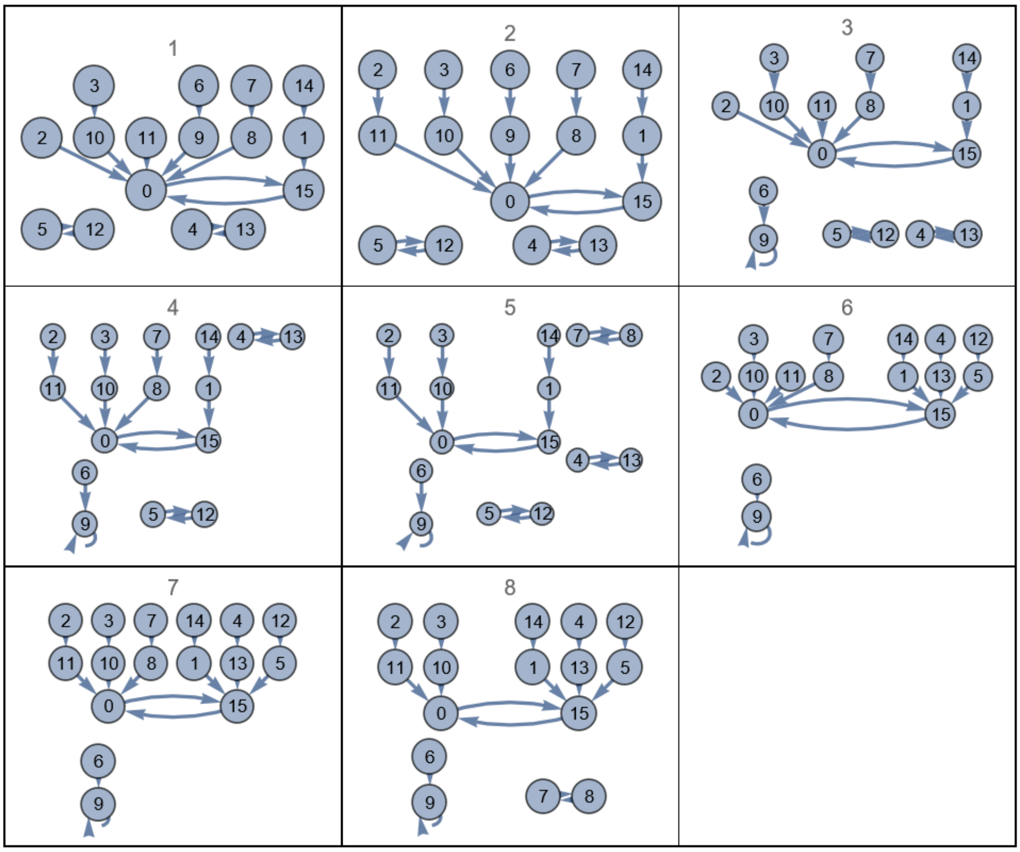


FIGURE 9: Individual best-response graphs of the snowdrift when $\epsilon \to 0$.

It is interesting to see that the phase diagrams of the snowdrift and the stag hunt (Figure 8 and Figure 5) look exactly the same. The only difference is the graphs that the regions correspond to. So, again there are 8 possible IBR graphs. However, this time there are graphs with the same basins of attraction. For example, the only difference between graphs 1 and 2 is whether policy 2 leads directly to 0 or first to 11 and then to 0, meaning the basins of attraction don't change. The cycle between policy 0 and policy 15 is the only equilibrium that appears for all values of s and r.

4.2 Average-reward versus discounting

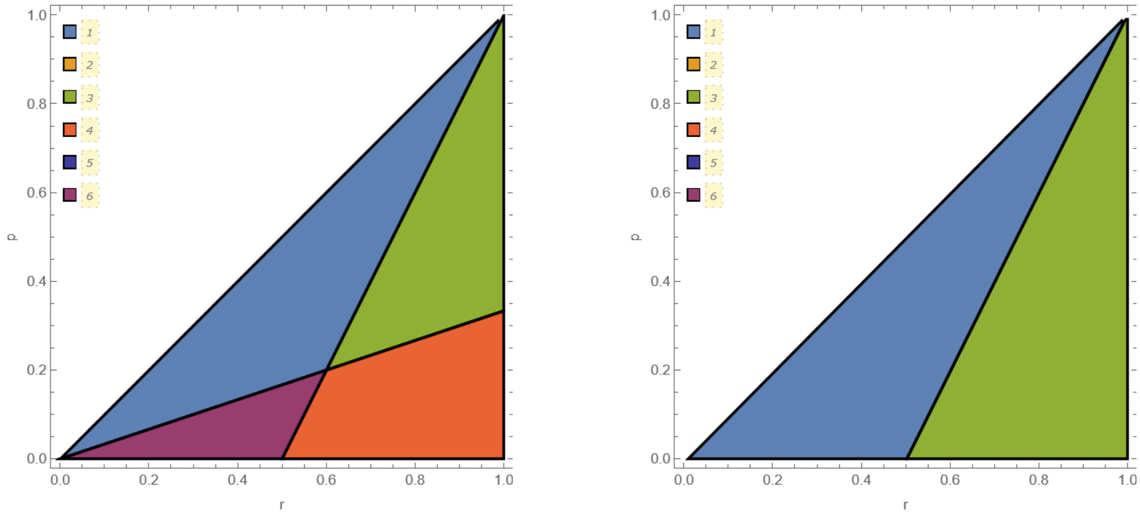
We now analyse the differences and similarities of the discounted and average reward approaches for the three games. The results of the discounted reward approach come from [11], where no exploration parameter is used ($\epsilon = 0$). The results of the average reward approach come from the previous section. We start by comparing the individual best-response networks and the transition condition tables. However, the main thing to analyse are the conditions under which the basins of attraction change. We do this by comparing the different phase diagrams. We let δ approach 1 in the discounted case, and ϵ approach 0 in the average-reward case, and compare the resulting phase diagrams of the two approaches.

4.2.1 Prisoner's dilemma

By comparing Figure 1 to Figure 3 of [11], we can see that there are more transitions possible, but the possible equilibria are the same. This can also be seen by comparing Table

2 to Table 2 of [11]. Note that the conditions in [11] are shown without the assumption of $s = 0$ and $t = 1$, and still the conditions using average-reward and exploration are a lot more complex. This leads to many more possible best-response graphs, with a total of 116, whereas there are only 12 in [11].

We will now compare the phase diagrams of both approaches, by letting δ approach 1 and ϵ approach 0. The phase diagram for the discounted approach is found by using the transition conditions of Table 2 of [11], instead of the transition conditions of Table 2.



(A) Average reward with $\epsilon = 0.001$

(B) Discounted reward with $\delta = 0.99$

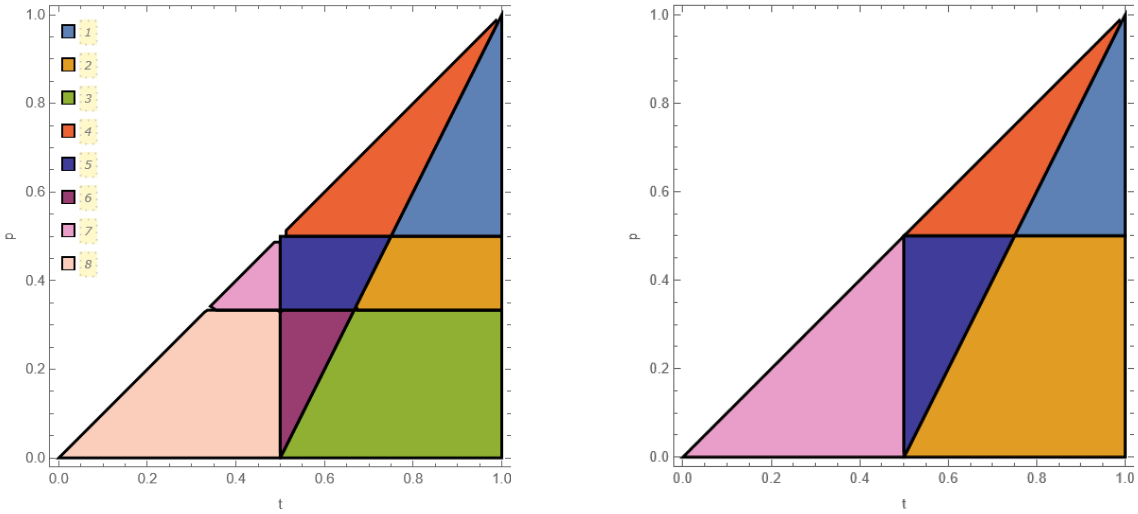
FIGURE 10: Phase diagrams for the prisoner's dilemma for both approaches

We can see that Cases 4 and 6 do not appear in the phase diagram of the discounted approach at all. In fact, Case 4 is not even possible for any value of δ. Case 6 can hold, but only when $\delta < p$, so not for high values of δ.

The difference we can see between the two approaches is the equilibrium of grim trigger. As previously discussed in Section 4.1.1, in the average reward approach it is possible for policy 15 to be the best response to policy 1. When ϵ is close enough to zero, this condition reduces to $r > 3p$, as can be seen in Table 2. This leads to policy 1 not being an equilibrium and instead being in the basin of attraction of policy 0, since policy 0 is always the best response to policy 15. In the discounted reward approach, this transition is not possible at all, which leads to grim trigger always being an equilibrium when δ is large enough. That grim trigger is not always an equilibrium for large enough δ when using an exploration parameter was previously found in [2]. It is shown there that the line separating the region where grim trigger is and is not an equilibrium intersects with the point $(\epsilon, \delta) = (0, 1)$, where any $\epsilon > 0$ results in grim trigger not always being an equilibrium if $\delta = 1$. This means the same case can still happen when implementing exploration in the discounted reward case and using a large enough δ and ϵ. However, only when using average-reward RL does this happen for any ϵ, no matter how close to 0 it is.

The conditions under which win-stay, lose-shift (policy 9) is an equilibrium is the same for the two approaches. With these conditions on ϵ and δ, policy 9 is always the best response to policy 6, but not always to itself. The line separating these equilibria is the same in both graphs, as it can be easily verified that both conditions reduce to $2r > 1 + p$.

4.2.2 Stag hunt



(A) Average reward with $\epsilon = 0.001$

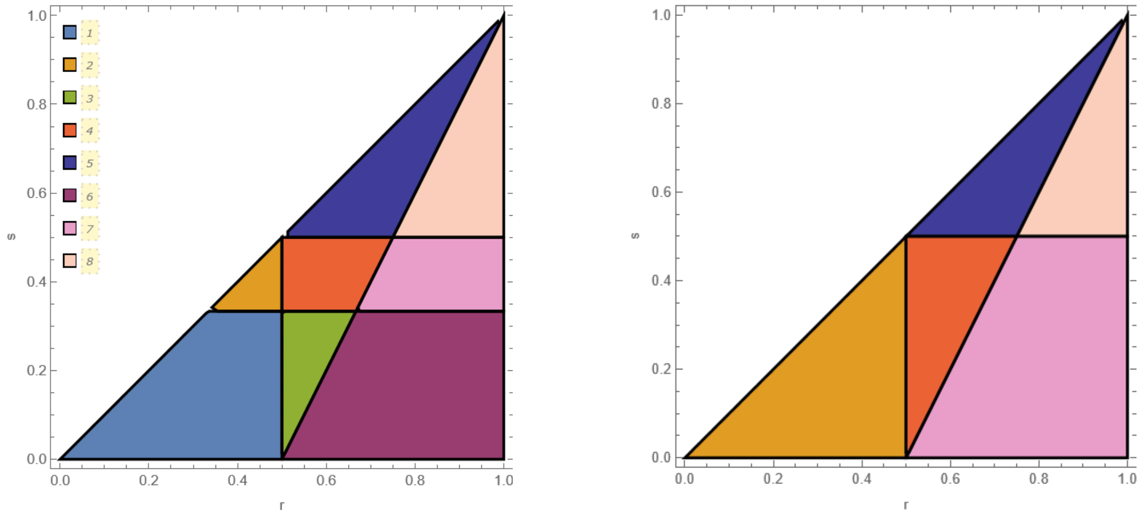
(B) Discounted reward with $\delta = 0.99$

Figure 11: Phase diagrams for the stag hunt for both approaches

The differences are quite similar to the prisoner's dilemma. Again, policy 1 (grim trigger) is always an equilibrium in the discounted case, but in the average-reward case, policy 15 is the best response to policy 1 when $p < \frac{1}{3}$. This is the exact same condition as found in the prisoner's dilemma, as $r = 1$ for the stag hunt.

The other transition conditions appear to all be the same in the phase diagram. This can also be verified by comparing Table 3a to Table 3b.

4.2.3 Snowdrift



(A) Average reward with $\epsilon = 0.001$

(B) Discounted reward with $\delta = 0.99$

Figure 12: Phase diagrams for the snowdrift for both approaches

For the snowdrift, the changes in the graphs that don't appear in the discounted case correspond to the case where policy 11 is not the best response to policy 2, but instead, policy 0 is. However, policy 0 is always the best response to policy 11, so these different graphs don't change any of the basins of attraction for the equilibria. It is interesting to see that this is the only game where the discounted and average-reward approaches lead to the exact same basins of attraction for the equilibria for every s and r. In the previous two games, we saw that grim trigger was the only difference between the two approaches. It was always an equilibrium in the discounted case, but not an equilibrium when $r > 3p$ in the average-reward case. Now, we have that $p = 0$, so $r > 3p$ is always true. Grim trigger is also never an equilibrium now. However, in the discounted case it is also never an equilibrium.

5 Conclusion

To conclude, average-reward RL gives very similar results to the discounted reward alternative in these specific two-player two-action games when δ is close to 1, and ϵ is close to 0. For the prisoner's dilemma and the stag hunt, we found that the only the conditions under which grim trigger is an equilibrium change. For the snowdrift, we found that all the basins of attraction are the same for the two approaches. It could therefore be possible to apply average-reward RL in multiagent settings where it is preferable to the discounted counterpart, without significant changes in the Nash equilibria. However, we only looked at the best responses given an infinite time for the Q-values to converge, and not at the time it takes for them to converge with this approach. We planned to do a simulation in Python of this environment with a finite batch size, but could not get it working in time.

Note that the average-reward approach is compared here with a discounted approach that does not use exploration. It remains to be seen if there are any differences between using discounted rewards with exploration and using average rewards with exploration. That comparison could lead to a significant difference in results, since the introduction of the exploration parameter has a massive effect on some strategies, even when ϵ is very small. Grim trigger is a good example of this, which is also the main difference we found between the two approaches. When not exploring, the system will never leave the cooperating state when both agents play grim trigger and start in that state. However, when ϵ is introduced, grim trigger will have very similar behaviour to all defect, especially with low ϵ. This happens since, for low values of ϵ, there is a probability of approximately ϵ that one (or both) of the agents explore out of state (C, C). Then, to get back into that state, both agents need to explore at the same time, which happens with probability $(\frac{\epsilon}{2})^2$. This causes the system to be in state (D, D) almost the entire time, which makes grim trigger almost the same as all defect.

Other possible extensions are to consider more complicated environments for this approach, such as asymmetric games, games with more players or games with a larger action space.

References

[1] Gürdal Arslan and Serdar Yüksel. Decentralized Q-Learning for Stochastic Teams and Games, May 2016. arXiv:1506.07924 [cs, math].

[2] Wolfram Barfuss and Janusz M. Meylahn. Intrinsic fluctuations of reinforcement learning promote cooperation. *Scientific Reports*, 13(1):1309, January 2023. Publisher: Nature Publishing Group.

[3] Mehmet Barlo, Guilherme Carmona, and Hamid Sabourian. Repeated games with one-memory. *Journal of Economic Theory*, 144(1):312–336, January 2009.

[4] Bryan R. Bruns. Names for Games: Locating 2 × 2 Games. *Games*, 6(4):495–520, October 2015. Publisher: MDPI AG.

[5] Emilio Calvano, Giacomo Calzolari, and Vincenzo Denicolò. Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review*, 110(10):3267–3297, October 2020. Publisher: American Economic Association.

[6] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, Joseph E. Harrington, and Sergio Pastorello. Protecting consumers from collusive prices due to AI. *Science*, 370(6520):1040–1042, November 2020. Publisher: American Association for the Advancement of Science.

[7] Vektor Dewanto, George Dunn, Ali Eshragh, Marcus Gallagher, and Fred Roosta. Average-reward model-free reinforcement learning: a systematic review and literature mapping, August 2021. arXiv:2010.08920 [cs].

[8] James W. Friedman. A Non-cooperative Equilibrium for Supergames. *The Review of Economic Studies*, 38(1):1–12, January 1971.

[9] Janusz M. Meylahn. Weak Acyclicity in Games With Unique Best-responses and Implications for Algorithmic Collusion, October 2023.

[10] Janusz M. Meylahn and Arnoud den Boer. Learning to Collude in a Pricing Duopoly. *Manufacturing & Service Operations Management*, 24(5):2577–2594, September 2022.

[11] Janusz M. Meylahn and Lars Janssen. Limiting Dynamics for Q-Learning with Memory One in Symmetric Two-Player, Two-Action Games. *Complexity*, 2022:e4830491, November 2022. Publisher: Hindawi.

[12] 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, July 1993. Publisher: Nature Publishing Group.

[13] Anatol Rapoport and Albert M. Chammah. *Prisoner's Dilemma: A Study in Conflict and Cooperation.* University of Michigan Press, 1965. Google-Books-ID: yPtNnKjXaj4C.

[14] Jean-Jacques Rousseau. *Discours sur l'origine et les fondements de l'inégalité parmi les hommes.* Bordas, Paris, 1754.

[15] Anton Schwartz. A Reinforcement Learning Method for Maximizing Undiscounted Rewards. pages 298–305, December 1993.

[16] Satinder Singh. Reinforcement Learning Algorithms for Average-Payoff Markovian Decision Processes. November 1999.

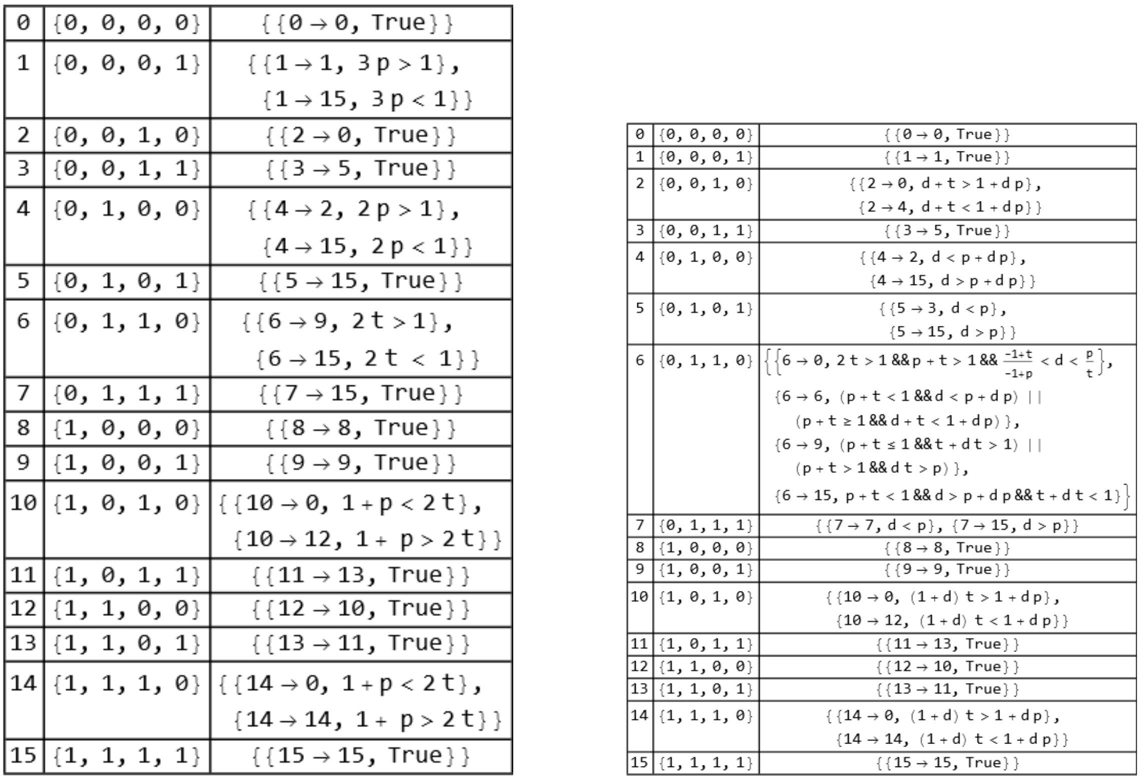
[17] John M. Smith and George R. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

[18] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction.* Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018.

A Transition conditions

0	{0, 0, 0, 0}	$\{\{0\to 0,\ \text{True}\}\}$
1	{0, 0, 0, 1}	$\{\{1\to 0,\ ((r<\frac{(-2+3\text{ epsilon}-2\text{ epsilon}^2)\,p}{-2+\text{epsilon}}+\text{epsilon}))\},$ $\{1\to 1,\ (\frac{(-2+3\text{ epsilon}-2\text{ epsilon}^2)\,p}{-2+\text{epsilon}}+\text{epsilon}<r<(3-2\text{ epsilon})\,p+\text{epsilon})\},$ $\{1\to 15,\ 3p<r\ \&\&\ 2\text{ epsilon}\,p+r>\text{epsilon}+3p\}\}$
2	{0, 0, 1, 0}	$\{\{2\to 0,\ \text{True}\}\}$
3	{0, 0, 1, 1}	$\{\{3\to 0,\ \text{True}\}\}$
4	{0, 1, 0, 0}	$\{\{4\to 0,\ 1+\text{epsilon}\,(-2+p+r)<2p\ \vert\vert\ 2p>1\},$ $\{4\to 13,\ (\text{epsilon}\,(p+r)+1>2\,(p+\text{epsilon})\ \&\&\ \frac{4r}{\text{epsilon}}+\text{epsilon}\,(p+r)+5<2\,(p+2r+\frac{2}{\text{epsilon}}+\text{epsilon}))\},$ $\{4\to 15,\ ((\frac{4r}{\text{epsilon}}+\text{epsilon}\,(p+r)+5>2\,(p+2r+\frac{2}{\text{epsilon}}+\text{epsilon})))\}\}$
5	{0, 1, 0, 1}	$\{\{5\to 0,\ (\text{epsilon}\,p+r\geq p+\text{epsilon}\ \vert\vert\ 2\text{ epsilon}\,(r-2)+2+\text{epsilon}^2\,(-r+1)<(-2+\text{epsilon})^2\,p)\ \&\&$ $(\text{epsilon}\,p+r<p+\text{epsilon}\ \vert\vert\ \text{epsilon}\,(-2+\text{epsilon})<(2-2\text{ epsilon}+\text{epsilon}^2)\,p+(-2+\text{epsilon}^2)\,r)\},$ $\{5\to 3,\ (r<p+\text{epsilon}\,r\ \&\&\ (-2+\text{epsilon}^2)\,p+(2-2\text{ epsilon}+\text{epsilon}^2)\,r<2\text{ epsilon}$ $\&\&\ (2-2\text{ epsilon}+\text{epsilon}^2)\,p+(-2+\text{epsilon}^2)\,r+2\text{ epsilon}<\text{epsilon}^2)\},$ $\{5\to 12,\ ((\sqrt{2}+\text{epsilon}<2\ \&\&\ (r>p+\text{epsilon}\,r\ \vert\vert\ 1>\frac{(-2+\text{epsilon})^2\,p-2\text{ epsilon}\,(r)+\text{epsilon}^2\,(r)}{2-4\text{ epsilon}+\text{epsilon}^2})$ $\&\&\ (r\leq p+\text{epsilon}\,r\ \vert\vert\ 2\text{ epsilon}\,(p+2r-1)+2\,(-2r+1)+\text{epsilon}^2\,(-p-r+1)>0))\ \vert\vert$ $(\sqrt{2}+\text{epsilon}>2\ \&\&\ r>p+\text{epsilon}\,r\ \&\&\ 4r+\text{epsilon}^2\,(p+r-1)<2\,(s+\text{epsilon}\,(p+2r-1)+1)$ $\&\&\ 1<\frac{(-2+\text{epsilon})^2\,p-2\text{ epsilon}\,(r)+\text{epsilon}^2\,(r)}{2-4\text{ epsilon}+\text{epsilon}^2})))\},$ $\{5\to 15,\ ((2\text{ epsilon}<(-2+\text{epsilon}^2)\,p+(2-2\text{ epsilon}+\text{epsilon}^2)\,r\ \vert\vert\ r>p+\text{epsilon}\,r)$ $\&\&\ (2\text{ epsilon}\,(p+2r-1)+2\,(-2r+1)+\text{epsilon}^2\,(-p-r+1)<0\ \vert\vert\ r\leq p+\text{epsilon}\,r))\}\}$
6	{0, 1, 1, 0}	$\{\{6\to 0,\ 4\text{ epsilon}+p+r+\sqrt{9+p^2-10r+r^2+2p\,(3+r)}>5\},$ $\{6\to 9,\ 4\text{ epsilon}+p+r+\sqrt{9+p^2-10r+r^2+2p\,(3+r)}<5\}\}$
7	{0, 1, 1, 1}	$\{\{7\to 0,\ (\frac{p+r-4+\sqrt{p^2+2pr+r^2-8r+8}}{-2}<\text{epsilon}<\frac{p+r+-4-\sqrt{p^2+2pr+r^2-8r+8}}{-2}\ \&\&\ 2r>1)\ \vert\vert$ $((\text{epsilon}<\frac{p-3r+2}{-2r+1}\ \vert\vert\ p+3r>1)\ \&\&\ 2r<1\ \&\&\ 4<p+r+2\text{ epsilon}+\sqrt{p^2+2pr+r^2-8r+8}$ $\&\&\ (p+r+2\text{ epsilon}<4+\sqrt{p^2+2pr+r^2-8r+8}\ \vert\vert\ p+3r\leq 1)))\},$ $\{7\to 8,\ (((4\text{ epsilon}<(-2+\text{epsilon})\,p+\text{epsilon}\,r+2+\text{epsilon}^2\ \&\&\ \text{epsilon}\,(p+r-2)+2+\text{epsilon}^2>2r)))\},$ $\{7\to 15,\ (((\text{epsilon}\,(p+r-2)+2+\text{epsilon}^2<2r)))\}\}$
8	{1, 0, 0, 0}	$\{\{8\to 0,\ \text{True}\}\}$
9	{1, 0, 0, 1}	$\{\{9\to 0,\ ((2-3\text{ epsilon}+2\text{ epsilon}^2)\,p+(-4+5\text{ epsilon}-2\text{ epsilon}^2)\,r+2>\text{epsilon})\},$ $\{9\to 9,\ ((-3+2\text{ epsilon})\,p<\frac{(-2+\text{epsilon})\,((-1+2\text{ epsilon})\,r)}{\text{epsilon}}+1$ $\&\&\ (2-3\text{ epsilon}+2\text{ epsilon}^2)\,p+(-4+5\text{ epsilon}-2\text{ epsilon}^2)\,r+2<\text{epsilon})\}\}$
10	{1, 0, 1, 0}	$\{\{10\to 0,\ \text{True}\}\}$
11	{1, 0, 1, 1}	$\{\{11\to 0,\ \text{True}\}\}$
12	{1, 1, 0, 0}	$\{\{12\to 0,\ \text{True}\}\}$
13	{1, 1, 0, 1}	$\{\{13\to 0,\ ((2-2\text{ epsilon}+\text{epsilon}^2)\,p+2>(-2+\text{epsilon})^2\,r+\text{epsilon})\},$ $\{13\to 11,\ (((\text{epsilon}\,p+2r<\frac{2p}{\text{epsilon}}+\text{epsilon}\,r+1\ \&\&\ (2-2\text{ epsilon}+\text{epsilon}^2)\,p+2<(-2+\text{epsilon})^2\,r+\text{epsilon})))\},$ $\{13\to 15,\ \frac{1}{2-\text{epsilon}}<r\ \&\&\ \text{epsilon}+2p+\text{epsilon}^2\,(-p+r)<2\text{ epsilon}\,r\}\}$
14	{1, 1, 1, 0}	$\{\{14\to 0,\ \text{True}\}\}$
15	{1, 1, 1, 1}	$\{\{15\to 0,\ \text{True}\}\}$

Table 2: Transition conditions for the prisoner's dilemma



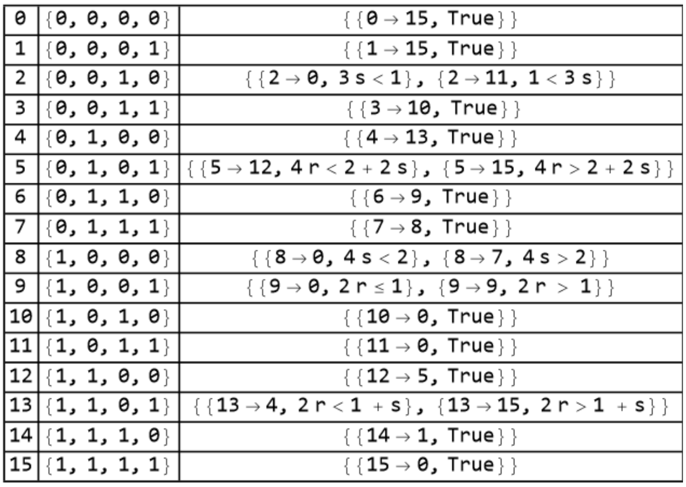
0	{0, 0, 0, 0}	{{0 → 0, True}}
1	{0, 0, 0, 1}	{{1 → 1, 3 p > 1}, {1 → 15, 3 p < 1}}
2	{0, 0, 1, 0}	{{2 → 0, True}}
3	{0, 0, 1, 1}	{{3 → 5, True}}
4	{0, 1, 0, 0}	{{4 → 2, 2 p > 1}, {4 → 15, 2 p < 1}}
5	{0, 1, 0, 1}	{{5 → 15, True}}
6	{0, 1, 1, 0}	{{6 → 9, 2 t > 1}, {6 → 15, 2 t < 1}}
7	{0, 1, 1, 1}	{{7 → 15, True}}
8	{1, 0, 0, 0}	{{8 → 8, True}}
9	{1, 0, 0, 1}	{{9 → 9, True}}
10	{1, 0, 1, 0}	{{10 → 0, 1 + p < 2 t}, {10 → 12, 1 + p > 2 t}}
11	{1, 0, 1, 1}	{{11 → 13, True}}
12	{1, 1, 0, 0}	{{12 → 10, True}}
13	{1, 1, 0, 1}	{{13 → 11, True}}
14	{1, 1, 1, 0}	{{14 → 0, 1 + p < 2 t}, {14 → 14, 1 + p > 2 t}}
15	{1, 1, 1, 1}	{{15 → 15, True}}

(A) Average reward when $\epsilon \to 0$

0	{0, 0, 0, 0}	{{0 → 0, True}}
1	{0, 0, 0, 1}	{{1 → 1, True}}
2	{0, 0, 1, 0}	{{2 → 0, d + t > 1 + d p}, {2 → 4, d + t < 1 + d p}}
3	{0, 0, 1, 1}	{{3 → 5, True}}
4	{0, 1, 0, 0}	{{4 → 2, d < p + d p}, {4 → 15, d > p + d p}}
5	{0, 1, 0, 1}	{{5 → 3, d < p}, {5 → 15, d > p}}
6	{0, 1, 1, 0}	{{6 → 0, 2 t > 1 && p + t > 1 && $\frac{-1+t}{-1+p}$ < d < $\frac{p}{t}$}, {6 → 6, (p + t < 1 && d < p + d p) || (p + t ≥ 1 && d + t < 1 + d p)}, {6 → 9, (p + t ≤ 1 && t + d t > 1) || (p + t > 1 && d t > p)}, {6 → 15, p + t < 1 && d > p + d p && t + d t < 1}}
7	{0, 1, 1, 1}	{{7 → 7, d < p}, {7 → 15, d > p}}
8	{1, 0, 0, 0}	{{8 → 8, True}}
9	{1, 0, 0, 1}	{{9 → 9, True}}
10	{1, 0, 1, 0}	{{10 → 0, (1 + d) t > 1 + d p}, {10 → 12, (1 + d) t < 1 + d p}}
11	{1, 0, 1, 1}	{{11 → 13, True}}
12	{1, 1, 0, 0}	{{12 → 10, True}}
13	{1, 1, 0, 1}	{{13 → 11, True}}
14	{1, 1, 1, 0}	{{14 → 0, (1 + d) t > 1 + d p}, {14 → 14, (1 + d) t < 1 + d p}}
15	{1, 1, 1, 1}	{{15 → 15, True}}

(B) Discounted reward

Table 3: Transition conditions for the stag hunt for both approaches



0	{0, 0, 0, 0}	{{0 → 15, True}}
1	{0, 0, 0, 1}	{{1 → 15, True}}
2	{0, 0, 1, 0}	{{2 → 0, 3 s < 1}, {2 → 11, 1 < 3 s}}
3	{0, 0, 1, 1}	{{3 → 10, True}}
4	{0, 1, 0, 0}	{{4 → 13, True}}
5	{0, 1, 0, 1}	{{5 → 12, 4 r < 2 + 2 s}, {5 → 15, 4 r > 2 + 2 s}}
6	{0, 1, 1, 0}	{{6 → 9, True}}
7	{0, 1, 1, 1}	{{7 → 8, True}}
8	{1, 0, 0, 0}	{{8 → 0, 4 s < 2}, {8 → 7, 4 s > 2}}
9	{1, 0, 0, 1}	{{9 → 0, 2 r ≤ 1}, {9 → 9, 2 r > 1}}
10	{1, 0, 1, 0}	{{10 → 0, True}}
11	{1, 0, 1, 1}	{{11 → 0, True}}
12	{1, 1, 0, 0}	{{12 → 5, True}}
13	{1, 1, 0, 1}	{{13 → 4, 2 r < 1 + s}, {13 → 15, 2 r > 1 + s}}
14	{1, 1, 1, 0}	{{14 → 1, True}}
15	{1, 1, 1, 1}	{{15 → 0, True}}

Table 4: Transition conditions for the snowdrift when $\epsilon \to 0$

B Code

The Mathematica code for the project can be found at `https://github.com/DeKleineKabouter/bachelorassignment`. It contains a README with an explanation of the structure of the code.