Balancing Anarchy and Central Control

Individual vs. Joint Action Reinforcement Learning

Daniel Claes

June 18, 2010

Abstract

There are more and more distributed systems in today's world, for which the control decisions can be taken centralized or decentralized. This article investigates the balance between central control and anarchy in multi agent cooperative games. Individual learning is compared to joint action learners using two different reinforcement algorithms - Q-learning and Cross-learning. While in individual learning each control decision is learned by exactly one learner, joint action learning controls a number of decisions that can range from a small partition of the decision to the full set.

For this purpose the the guessing game and a spatially constrained grid game are examined and a measure of central control is introduced. In the complete n-player guessing game it is shown that any form of central control yields convergence to suboptimal policies. In the spatially constrained grid game, a higher level of central control generally performs better, especially if the best actions are highly dependent on the agents' states. However, the maximal level of central control is very limited due to the exponential numerical explosion in the state and action space. Additionally, it is found that a stateless Nash equilibrium strategy is amongst the best performing policies.

Keywords: reinforcement learning, Q-learning, Cross-learning, cooperative games, central control

1 Introduction

Reinforcement learning in multi agent games has been studied extensively in the past [4, 6, 15]. Action selection policies are learned using a numeric reward which is observed through the environment. This feature makes reinforcement learning appealing, since as well in real life, we often only know if an action was good or bad, but not necessarily know the reasons behind it.

Many research is done in the domain of individual learning methods, since they are easy to implement and yield limited complexity. Additionally, no form of communication is needed, since each agent acts on its own. This assumption generally holds, since in real life, in many cases the acting agents are spatially distributed and have only limited communication. However, in some cases there is a central control, which takes decisions for a group of agents. The comparison between individual learning and central control will be the focus of this article.

The main research question is therefore how to balance central control and individual learning in multi agent cooperative games to maximize social welfare under constrained time.

To illustrate a situation in real life, where this research is applicable, imagine an economic market, where new industrial standards are introduced. Each company has to chose whether to adopt a certain standard or to keep their own system. These decisions have a big influence on the profit or loss of companies. If many other companies decide to adapt to the same standard, it is advisable to do so as well to get a bigger consumer base. Thus, the result of the company's decision is not only determined by its own action, but on the joint action of several companies. Additionally, we can have some controlling agencies, which regulate the actions for several companies. Thus, this research can be used to determine how many controlling agencies, if any, are advisable, or whether every company should act on its own.

This article evaluates two well-known learning methods, Cross-learning and Q-learning in two cooperative games. Special attention is paid on controlling multiple decisions at the same time, which means that a higher percentage of the reward is dependent on the learners' actions. This is introduced as percentage of central control. The concepts convergence, Nash equilibria and social welfare are used for evaluation, while in particular the maximum social welfare and the speed of convergence are two important performance indices for cooperative games. The games used as testbeds are the n-player guessing game and a spatially constrained grid coordination game.

The rest of this article is structured as follows: Section 2 introduces reinforcement learning in the single agent domain and presents the two learning algorithms, while Section 3 generalizes the algorithms to the multi agent domain. Section 4 gives a description the environment and the evaluation methods, and Section 5 and 6 present the conducted experiments and the discussion of the obtained results. Finally, Section 7 and 8 close with conclusions and an outlook on future work.

2 Reinforcement Learning in the single agent domain

Reinforcement Learning (RL) is a variant of machine learning, where an agent, for instance a computer program or a robot, is learning its behaviour only based on a signal of reward and punishment. The concept is useful to model difficult tasks, where it is easier to define the aim of the task instead of knowing how it is done exactly. The method of RL is imitating a way how an animal would learn certain behaviours. Likewise, we interact with our environment by performing actions after which the we observe the effects. This idea of *cause and effect* is used in our daily life for building up the knowledge of our world.

In general, in the mathematical RL model, the learner can be in a finite set of states S and for each state there is a set of possible actions A. Each action leads to a certain observable numeric reinforcement signal, which is the reward of the given actions. The task for the learner is to maximize the cumulative discounted future rewards. The RL problem for agents can be formulated as a Markov decision process with discrete time, finite states and finite actions, that is defined in the following [8]:

- Finite set of possible actions $a \in A$ and states $s \in S$
- Initial state: $s_0 \in S$
- Transition function: $T : S \times A \times S \to [0, 1]$, where T gives a probability distribution over S
- Reward function $R : S \times A \to \mathbb{R}$

In general, there are various methods in RL. In the next subsection, two methods will be introduced - examples of the policy and value iterator classes.

2.1 Learning behaviour in the RL environment

While this section considers single agent learning, this process can be generalized to a multi agent environment, as explained in Section 3. The task for an agent in the RL problem can be translated into certain steps for each learning step t:

- Observe the current state s
- Determine next action a based on a certain action selection policy $\pi : S \times A \to [0, 1]$
- Perform the selected action
- Observe the reinforcement signal (reward) gained from this action
- Use information about this state-action pair to update the current policy π, if necessary

Thus the agent needs an action selection policy π. In this article the notation of $\pi_s(t, a)$ is used, which gives the probability distribution over the actions a when being in state s and time step t. Thus, the following equation must hold for all states s and time steps t:

$$\begin{aligned} &\sum_a \pi_s(t, a) = 1 \\ &\pi_s(t, a) \in [0, 1] \end{aligned} \tag{1}$$

There are two principle ways to learn in the RL domain, namely policy and value iterating methods [13]. While policy iterators update the action selection policy directly, value iterators estimate the value of the given actions and deduce the optimal policy based on these.

Policy iterators only maintain an action selection policy to approximate the optimal behaviour. More specifically, a random initial policy is used to explore the environment, while it is continuously updated with the reinforcement signal (reward or punishment) observed [4].

Value iterators use the reinforcement signal to update a value estimation function for the value of each state and afterwards this function is used to determine a new action selection policy. In other words, an agent performs actions to estimate the gained value for each state-action pair and then chooses an action according to these estimation values [6, 13].

2.2 Cross-learning - A policy iterator

Cross-learning belongs to the class of finite action-set learning automata that were initially researched in the 1960's by Tsetlin to model empirical observations of learning behaviour [4, 14]. Later, learning automata became a topic in the engineering domain as adaptive decision makers and recently, they were used in the RL domain as a basis for multi-agent learning [10].

As described in Section 2, the set of possible actions A and states S is finite. The general update rule for policy $\pi_s(t, a)$ at each time step t is given below [10].

$$\begin{aligned}
&\text{If } a(t) = \hat{a} \text{ then}\\
&\pi_s(t+1, \hat{a}) = \pi_s(t, \hat{a}) + \alpha r(t)(1 - \pi_s(t, \hat{a}))\\
&\qquad\qquad - \beta(1 - r(t))\pi_s(t, \hat{a})\\
&\text{and } \forall a \neq a(t)\\
&\pi_s(t+1, a) = \pi_s(t, a) - \alpha r(t)\pi_s(t, a)\\
&\qquad\qquad + \beta(1 - r(t))[(k-1)^{-1} - \pi_s(t, a)]
\end{aligned} \tag{2}$$

Where s is the current state, $a(t)$ is the selected action out of k different actions in the set A, $r(t)$ is the observed reward and α and $\beta \in [0, 1]$ are the reward and penalty parameters respectively. Depending on α and β the update scheme is referred to as *linear reward-penalty* (L_{R-P}) if $\alpha = \beta$, if β is chosen to be small compared to α it is called *linear reward-ϵ-penalty* ($L_{R-\epsilon P}$) and for $\beta = 0$ it is called *linear reward-inaction* (L_{R-I}), which is also known as *Cross-Learning (CL)* after the author of [1]. Since for this article only CL is of interest, the equation can be simplified to the following:

$$\begin{aligned}
&\text{If } a(t) = \hat{a} \text{ then}\\
&\pi_s(t+1, \hat{a}) = \pi_s(t, \hat{a}) + \alpha r(t)(1 - \pi_s(t, \hat{a}))\\
&\text{and } \forall a \neq a(t)\\
&\pi_s(t+1, a) = \pi_s(t, a) - \alpha r(t)\pi_s(t, a)
\end{aligned} \tag{3}$$

Assuming that $r \in [0, 1]$ and continuous, Equations (2) and (3) maintain the probability distribution given in Equation (1) [9].

2.3 Q-Learning - A value iterator

As said before, the goal of the RL agent is to maximize the cumulative future rewards. While policy iterators maintain and update an estimation of the optimal policy, value iterators estimate the expected rewards for the current state-action pairs and derive a policy balancing exploration and exploitation from these. In order to understand Q-Learning (QL), the *value of a state* has to be explained in further detail. Additionally, the value of a state-action pair will be introduced as a Q-value. Afterwards, the Q-learning itself will be explained using an estimation function for these Q-values. Eventually, different action selection policies will be presented.

The value of a state equals its own reward plus the expected discounted reward of its successor states, when following policy π [11], as shown in Equation (4). In other words, it tells the agent how profitable it is, to be in this state. A utility function estimates the utility of a given state s under policy π.

$$V^\pi(s) = E[\sum_{t=0}^{\infty} \gamma^t r(t) | \pi, s_0 = s] \tag{4}$$

This includes a discount factor $\gamma \in [0, 1]$ that controls the agent's desire to achieve the goal quickly. Furthermore, it bounds the infinite sum. A low γ value leads to a setting where the agent becomes myopic, since the future rewards are heavily discounted.

If the value function is known, the problem is solved, since direct utility estimation could be used, which will not be explained here [11]. However, commonly, the function is part of the unknown environment that has to be explored. Hence, the values have to be estimated, since they are crucial to be able to accurately choose an action that maximises the total reward. As one solution for the estimation, the so-called temporal difference learning can be used. It takes the observed transitions to update the values of the observed states:

$$V^\pi_{t+1}(s) \leftarrow V^\pi_t(s) + \alpha(r(t) + \gamma V^\pi_t(s') - V^\pi_t(s)) \tag{5}$$

where $\alpha \in [0, 1]$ is the learning rate parameter and t the current time step. This learning rate should be a decreasing function in order to let $V^\pi_t(s)$ converge to the correct value [11]. In this way, an estimate of the final reward is calculated at each state and the state-action value is updated on every step t on the way. This method is often called *bootstrapping*, since it uses estimates to estimate the values. To link values of states to actions, the Q-values are introduced. The value of a state is directly related to Q-values as follows:

$$V^\pi(s) = \max_a Q(s, a) \tag{6}$$

Thus, a Q-value is the expected reward when starting at s and taking action a. If the transition model is known, it is possible to calculate all exact Q-values iteratively.

$$Q(s, a) = r(s) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q(s', a') \tag{7}$$

where $T(s, a, s')$ gives the probability of reaching state s', when taking action a in state s. However, most of the time, the transition model is not known in advance, and therefore, the Q-values have to be estimated.

A famous method of Q-learning was proposed by Watkins [16] in 1989. It iteratively approximates the Q-values by an estimation function $\hat{Q}_t$ for each time step t:

$$\hat{Q}_{t+1}(s, a) = \hat{Q}_t(s, a) + \alpha(r(s) + \gamma \max_{a'} \hat{Q}_t(s', a') - \hat{Q}_t(s, a)) \tag{8}$$

where α is the learning rate, which can be a constant or a decreasing function depending on visits of each state, and γ is the discount value. It only uses the current estimation $\hat{Q}_t$ and the observation of the current state's reward $r(s)$ to update the Q-values. This is a big advantage, since the transition function and reward function do not need to be known in advance and are not explicitly estimated. For this reason, it is called a model-free method [11].

Action selection policies
There exist several policies to decide which action should be taken next. However, there is the problem that the knowledge about these values is uncertain and incomplete. Hence, there has to be a tradeoff between exploration of the unknown and exploitation of the already learnt. Otherwise, if some decent action was found before, this will be chosen over and over again, although there may exist a better one. On the one extreme, there is greedy selection, which always takes the currently highest Q-value, and on the other extreme, there is uniform selection, where each action is selected randomly with equal probability.

Both extremes have shortcomings, which are the non existing exploration for greedy selection and the non existing exploitation in the uniform selection. Therefore, the commonly used policies are *soft*, meaning that every possible action has a non zero probability of being taken. Two common action selection policies are [13]:

- ε-greedy: The action with the highest estimated reward is chosen with probability $1-\varepsilon$, therefore it is called greedy. With a small and possibly decreasing probability ε, a random action will be performed, in order to provide the exploration, which is particularly important in the beginning. These random actions are selected uniformly and independent of the value estimations. If ε is decreasing to zero in the limit, this policy approaches the static greedy policy without any exploration. Therefore, the decrease has to be set carefully and it is not well suited for dynamic environments, which require ongoing exploration.
- softmax: One possible drawback of ε-greedy is that it selects the exploration actions uniformly. Therefore, the worst possible action will be selected with the same probability as the second best. The softmax policy uses a Boltzmann distribution to overcome this problem. Each action gets weighted, according to their action-value estimate $Q_t(s,a)$ at the current state:

  $$\pi_s(t,a) = \frac{e^{Q_t(s,a)/\tau}}{\sum_b e^{Q_t(s,b)/\tau}} \tag{9}$$

  were τ is the temperature. A high temperature scales the function to an almost uniform distribution, while a low temperature approaches a greedy selection. This allows a dynamic tradeoff between exploitation and exploration with possibly decreasing τ to converge to a static greedy strategy. This approach is favourable, when the action which is currently estimates with the lowest value should not be taken with a high probability and the other options with high Q-values have a good chance being the best option, since the estimation was not correct yet due to stochastic effects.

3 Reinforcement Learning in the Multi Agent Domain

The learning algorithms in the last section were introduced in a single agent environment. This section will generalize them to the multi agent domain. First, the differences between the single agent and multi agent domain will be discussed and afterwards possible learning methods in the multi agent domain will be presented.

In contrast to individual learning, there are multiple agents in the same environment. These can have a task which involves cooperation or competition. For instance many games are competitive where each agents tries to maximize his own reward regardless what the other agent does. In other games, you can have a cooperative part, where the agents need to collaborate to achieve a better reward. An example is the famous prisoners dilemma. When played iteratively, the players can cooperate to achieve the maximal reward.

For each game, there are a number of decision to be taken simultaneously, which are the possible actions of the players. To perform learning in this domain, there are several possibilities. On the one hand, each decision can be controlled by exactly one learner, or on the other hand one agent can control all the decisions. Additionally, the decisions can be partitioned into groups, where each group is controlled by a leaner.

The first option is the direct translation of the algorithms introduced and is called individual action learning. Each agent learns for himself and other agents are just seen as part of the environment. The second and third option are called Joint Action Learning (JAL), where the learning algorithm can be seen as an agent which controls multiple decisions. The JAL can access the information of each decision and tries to achieve a maximal joint reward.

3.1 Individual learning

Individual learning gives one decision to exactly one agent and applies the RL methods described in Section 2 directly. This implies that coordination of actions can only be learned based on interactions with the environment. More specifically, since there is no communication between the agents, there is no real form of cooperation.

The acting agents perceive the other agents only as part of the dynamic environment and ignore their presence or treat their impact as environmental effects.

If two agents have to do the same to get a good reward, they will learn this eventually, since the reinforcement signal is largest when both choose this decision. However, this is hard to learn, since choosing the same action multiple times will lead to an ambiguous reinforcement signal based on the other agents' decisions. Additionally, if the environment is dynamic, it is hard to distinguish between mis-coordination and stochastic effects.

Some advantages of this approach are its simplicity, since no communication between the agents is needed, and furthermore, the action and state space is limited. Additionally, the implementation of this algorithm can be adapted straightforwardly from the single agent domain.

3.2 Joint action learning

In contrast to individual learning, the joint action learning gives multiple decisions to one JAL agent. This can be the full set of decision or only a partition. JAL strives to the best joint reward of the controlled group. This means for instance for CL and QL that the update functions get more complex, since the update functions need to be defined for each combination of the possible actions and states for all the possible decisions.

More specifically, the parameters for the state s and action a of the update function as introduced in Equation (3) and (8) for CL and QL become vectors of the combined states $s = (s_1, \dots, s_n)$ and the combined actions $a = (a_1, \dots, a_n)$, where each index i belongs to one controlled decision [5].

Since the learner controls multiple decisions, the joint action has to be based on the combined observed states. This implies that the action and state space grows exponentially to the number of controlled decision.

Let n_a and n_s be the number of possible actions and states for each decision and n the number of controlled decisions. This leads to ${n_a}^N$ possible actions and ${n_s}^N$ possible states. If $n_a = n_s = n$ the possible state-action pairs are n^{2N}. Since the growth is exponentially depending on the group size, this can lead to a huge space complexity for larger N.

However, the advantage is that JAL has more information which it can exploit to achieve higher rewards, especially in cooperative games. Coordination within a group should be trivial, hence only cooperation between the groups has to be learned.

4 The environment and the game

This section will introduce the environment in which the RL algorithm will be evaluated. Additionally, some terms from Game Theory (GT) will be presented that are used to analyze games.

4.1 A cooperation game without spatial constraints

Imagine the example of the introduction of new standards in the industry. If there are several standards, each company has to select one standard to support. However, its payoff is determined by how many other companies chose the same standard, since the consumer base grows with each additional company selecting the same standard. This can be modeled in the abstract *Guessing Game (GG)* [6, 12].

Let $n(a_i)$ be the number of agents choosing action a_i and let N be the total number of players. Then the reward for each action is the same for all agents and is defined as:

$$r_j(a_j) = \frac{n(a_j)}{N} \qquad (10)$$

In general terms, this means that the reward for each agent is fraction of the number of agents playing the same action as the agent itself. The GG considered in this article is the complete GG, which means that there are as many actions as players. The game is played repeatedly for several time steps. This kind of game is also called a repeated stochastic game.

4.2 A cooperation game with spatial constraints

In real life, not everybody interacts with everyone. For instance, the economic market is usually not as uniform and interconnected as assumed in the GG. First, the market is more distributed in segments which are overlapping to some extent and second, it is usually not equally easy for each company to adapt to a certain standard. These features are captured by the following game.

The game consists of a square grid of the size nxn. Each cell can have two different states (for visualization black and white), which are selected randomly with a certain state bias $p_{black} \in [0, 1]$ representing the probability of being in state black. For each cell, there is one decision to be taken which has effects on the neighbouring cells that are directly adjacent to it. Each decision can be the action black or white and a reward for the decision is given, if two adjacent decisions are the same. However, the reward for each decision depends on the state in which it is in. The payoff matrix is given below, where the rows represent the states and the columns represent the actions.

	A_B	A_W
S_B	0.2	1
S_W	1	0.2

Hence, when being in state black the agent gets a higher reward when playing white. However, if nobody is cooperating with it, the reward is 0, thus cooperation is a crucial factor in this game. The total reward for each individual cell is the averaged sum of all games with its adjacent cells. In terms of the market game, the states can be seen as how difficult it is to adapt to standard black or white and the actions to adapt to one of the standards.

Variants of the Game

To evaluate the effects of the state dependency, if the payoff matrix is changed to get different variants of the game. For instance, the payoff matrix can be as follows:

	A_B	A_W
S_B	0.5	1
S_W	1	0.5

In this case, the difference between the two actions is not as large as in the original case. Hence, the reward when choosing the inferior action of the cell's own state is still relatively high, which means that the state is less important. This setting is called *State Less Important Setting (SLI-Setting)*. Another variant is achieved with the following payoff matrix:

	A_B	A_W
S_B	1	0.2
S_W	1	0.2

The state is not important anymore for determining the decision, since in both states the payoff is the same. This will be called the *State Independent Setting (SI-Setting)* from now on.

4.3 Analyzing performances in games

For analyzing the games and the performances of the learners, there are some concepts from GT that aid in a more specific evaluation. A game has a number of players n, a set of states S in which the players can be in and a set of actions A the player can take. For competitive games, the most well known solution concept is the Nash Equilibrium (NE). A set of strategies $(\pi_1^*, \ldots, \pi_n^*)$ for all players is called a NE, if and only if there is no player that has an incentive for deviation, if all other agents keep their policies fixed [2, 3]. This yields stability, since if there is no incentive to deviate, the policies remain stable.

In the complete GG there are N pure NE, one for each action. This is when each player chooses the same action. Any deviating player would get much less reward, since he would be the only one in this group. Therefore, as long as there are two groups, you have always the incentive to join the other equally large or larger group to improve your own reward until all players take the same action [6].

Likewise, in the more complex game with spatial constraints, there are two pure NE when all agents play the same action regardless of the state. As a result, any deviating player would get no reward in this case, since nobody else is playing his action.

Social Welfare

For cooperative games, the goal is to maximize the the joint reward of all players. Hence, the social welfare ω of a strategy π is the total sum of the individual rewards:

$$\omega(\pi) = \sum_i r_i(\pi_i)$$

where $r_i(\pi_i)$ defines the reward for each individual i when following strategy π_i. Thus, for a cooperative game, the best strategy profile π^* is the one that maximizes the social welfare.

$$\pi^* = \arg\max_\pi \omega(\pi)$$

Since the games described above are cooperative games, the social welfare achieved by the learners will be used as the main measure for evaluating the performance.

Convergence and Confidence Intervals

Another measure to analyze performance in games is the time to converge for the learners. As distance metric the differences in average social welfare for subsequent time steps is chosen. This means that the achieved social welfare ω does not change more than a very small threshold value ε after some iteration T onwards, when compared to the mean of all social welfares after T time steps:

$$\begin{aligned} &\omega \textit{ converged to } \omega' \textit{ at } T \\ &\quad \leftrightarrow \ \forall t \ (t \geq T \rightarrow |mean(\omega_{T,\ldots,end}) - \omega_t| < \varepsilon) \end{aligned}$$

This definition can be interpreted intuitively as all values after convergence remain in a certain interval of length $2 * \varepsilon$. Once the specific T is known, a confidence interval for the mean convergence value is estimated using a t-distribution with $T_{max} - T - 1$ degrees of freedom. If the data is averaged over a large enough number of runs, the central limit theorem holds [7]. This confidence interval is used to compare the performance of the different learners.

4.4 Percentage of central control

Since this article examines the effect of Central Control (CC) vs. anarchy, a measure of CC has to be introduced. An intuitive definition will be used, which is the

1	2
3	4

1	1
2	2

1	1
1	1

Table 1: Distribution of the groups for the 2x2 grid. 4x1, 2x2 and 1x4 from left to right.

1	2	3
4	5	6
7	8	9

1	1	2
4	5	2
4	3	3

1	1	1
2	2	2
3	3	3

1	1	1
1	1	1
1	1	1

Table 2: Distribution of the groups for the 3x3 grid. 9x1, 4x2+1x1, 3x3 and 1x9 from left to right.

percentage of reward that is dependent on the decisions that the learner controls. Hence, in the GG this would mean an individual learner would have $\frac{1}{N}$ control over the reward it gets. The rest is determined by the actions of the other agents. In general, a group of size n, has control over $\frac{n}{N}$ of the reward.

In the spatially constrained game not only the size of the group is important but also the location of the controlled agents. For instance, as shown in Table 1, the individual learner 1x4 has no control over the reward, since it is completely dependent on the other agent's actions. Likewise, the groups of size two have 50% control, since for both agents, one adjacent agent is controlled within the group and another one outside. Of course, the full joint action learner has full control over the reward.

5 Experiments

This section explains the experimental setup regarding the two games introduced in Section 4 and the different learning algorithms. For the GG the performance of individual and joint action learning was evaluated for the sizes of $N = 4$, $N = 9$ and $N = 16$. The group sizes were 4x1, 2x2 and 1x4 for $N = 4$, 9x1 and 3x3 for $N = 9$ and 16x1, 8x2 and 4x4 for $N = 16$.

Since the possibilities for JAL rise exponentially, the full space joint action learning for $N \geq 9$ is not considered, since for $N = 9$ it already yields $9^9 \approx 3.87e^8$ possibilities. CL is tested with $\alpha = 0.1$. Since the game is played repeatedly and the chosen action does not have any influence on the future state, the discount value is not taken into account. Thus, QL has the same learning rate $\alpha = 0.1$ and additionally $\gamma = 0$. As action selection policy for the Q-learning, softmax was chosen with a constant temperature of $\tau = 0.1$. The temperature is chosen to provide a good balance of exploration and exploitation. Additionally, in a dynamic environment with stochastic influences a constant amount of exploration is desirable. Therefore, the temperature is not decreased

The experiments with the spatially constrained game were conducted on the following grid sizes: 2x2, 3x3 and 4x4 using Cross-learning and Q-learning. Furthermore, a random action selection policy is used as a baseline comparison. The groups for each grid size were distributed as seen in Table 1, 2 and 3. For the 4x4 grid, the full joint action learning has $2^{32} \approx 4.3e^9$ possibilities. Thus it is not feasible anymore, since the state and action space are too large to handle. Therefore, it is left out of the experiments. The resulting percentage in CC of the groups for 2x2 is 0%, 50% and 100% (see Section 4). For all other settings Table 4 can be consulted.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	1	2	2
3	3	4	4
5	5	6	6
7	7	8	8

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

1	1	1	1
1	1	1	1
2	2	2	2
2	2	2	2

Table 3: Distribution of the groups for the 4x4 grid. 16x1, 8x2, 4x4 and 2x8 from top-left to bottom-right.

3x3

Groups	% CC
9x1	0.00
4x2 (1-4)	41.67
+1x1 (5)	0.00
average	37.04
3x3 (1,3)	55.56
3x3 (2)	38.89
average	47.23
1x9	100.00

4x4

Groups	% CC
16x1	0.00
8x2 (1,2,7,8)	41.67
8x2 (3-6)	29.17
average	34.42
4x4	70.83
2x8	77.08
1x16	100.00

Table 4: Calculated percentage of CC depending on the group sizes for the 3x3 and 4x4 grid game. The numbers in brackets refer to the groups shown in Table 2 and Table 3 for 3x3 and 4x4 respectively.

Each setting was evaluated for each variant of the game explained in Section 4 with a constant state bias $p_{black} = 0.5$. Additionally, the influence of an abrupt change in the state bias p_{black} is analyzed. This is done by switching between the original value and the value of $p_{black} = 0.7$ after each quarter of the total time steps. In the real life application, this switch can be explained as for instance the tendency of being able to introduce a certain standard easier than the other one. The learning parameters for Cross-learning and Q-learning are set to the same values as in the GG.

For each setting, the social welfare is plotted over 10,000 time steps and averaged over 10 independent runs with a running average window of 200 to smooth the results. Additionally, the average number of time steps to converge and the 95% confidence interval of the achieved social welfare after convergence are calculated. The best possible actions are computed beforehand by exhaustive search to get a theoretical maximum expected social welfare as performance measure.

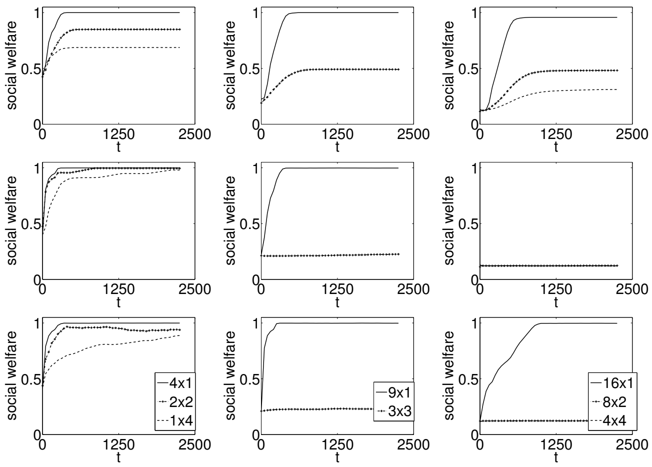

Figure 1: Results of the GG for $N = 4$ (left), $N = 9$ (center) and $N = 16$ (right). Top row: CL with $\alpha = 0.1$, central row: QL with $\alpha = 0.1$, $\tau = 0.1$ and bottom row: QL with $\alpha = 1$, $\tau = 0.1$.

6 Results & Discussion

This section presents the results obtained by the experiments introduced in Section 5. The analysis focuses on the performances of the different group sizes and the differences between Cross-learning and Q-learning.

6.1 Results for the GG

Figure 1 shows the results for the GG and the different settings. For visualization only the first 2500 time steps are plotted, since the learners converged before and the differences in the time to converge can be seen easier.

Since the difficulty to cooperate on a single action with more agents rises, it is expected that the larger N the less overall social welfare is achieved. Likewise, it is expected that convergence takes longer with more agents. This can be seen when comparing the graphs from left to right. However, it is striking that the separation into groups controlled by JAL, has a *negative* effect on the total social welfare. This means by facilitating coordination by CC the results deteriorate quite a lot. For instance, even for the simplest case of $N = 4$ the CL looses roughly 10% when splitting into groups of two or even 20% when having full joint control. If there are more decisions to be taken, JAL achieves even less that than 50% of the individual learner.

QL, on the other hand, performs decently in the smallest case of $N = 4$ for each group size. However, the individual learner still outperform the larger groups. Furthermore it is notable, that while for $N = 9$ the individual learners still reach the optimum, in $N = 16$ and $\alpha = 0.1$ QL algorithms converge at the very suboptimal setting in which pairs of agents choose the same action. When looking at $\alpha = 1$, the individual learner reaches optimum again, while the performance of the other groups stays equally bad. A high learning rate for QL is advisable in this game, since a change to a better group has to be rewarded that much that the learner stays in the same group.

It might be counterintuitive that JAL is outperformed by individual learning, since the coordination within the groups seems easier. However, it can be explained by the metaphor of finding the *needle in the haystack*. The more decisions are controlled by JAL, the harder it is to find the one action which leads to the highest reward. Especially, when actions of other JAL groups have a high influence on the reward, it is hard to decide if the low reward was due to mis-coordination within the group or mis-cooperation between the groups.

To sum up, for the GG it can be seen that CC is not desirable, since the individual agents outperform the joint action learners. Hence, the information gain by knowing the state of some other agents is outmatched by the rise in complexity to control more than one decision.

6.2 Results for the grid game

Before discussing the results on the various grid sizes some general observations can be made. First, a state unconscious learner can maximally achieve the following average social welfare independent from the grid size.

$$\begin{aligned}\omega_{stateless} =&\\ \max(p_{black} * r(A_B|S_B) \quad &+ \quad (1 - p_{black}) * r(A_B|S_W),\\ p_{black} * r(A_W|S_B) \quad &+ \quad (1 - p_{black}) * r(A_W|S_W))\end{aligned}$$

where $r(A_W|S_B)$ is the reward of cooperating on action W given state B and vice versa. This leads in the original setting to $\omega_{stateless} = 0.6$, and in the SLI-Setting to $\omega_{stateless} = 0.75$. These are the pure NE, when all agents coordinate on playing the same action regardless of their states.

Second, there is the strategy of always playing the action that yields highest reward given the current state if other surrounding decisions are the same. When examining the reward structure in the original game and the SLI-Setting, it is found that the highest reward for cooperation is 1, which is always the alternative action of its own state. Therefore, it will be called *alternating strategy*. The resulting average social welfare can be calculated, since in p_{black} of the cases any surrounding action will have the same color. Thus, it is $\omega_{alternate} = 0.5$ for $p_{black} = 0.5$.

Lastly, when looking at the results in Table 5, 6 and 7, it can be seen that in the SI-Setting most learners achieved a reward close to $\omega_{max} = 1$. Only the large group with JAL stayed well below the optimum, which can be explained by the exponentially increasing number of possible state-action pairs. Since there is not much else to derive, this setting is left out in the upcoming figures and discussion. The results of Table 5, 6 and 7

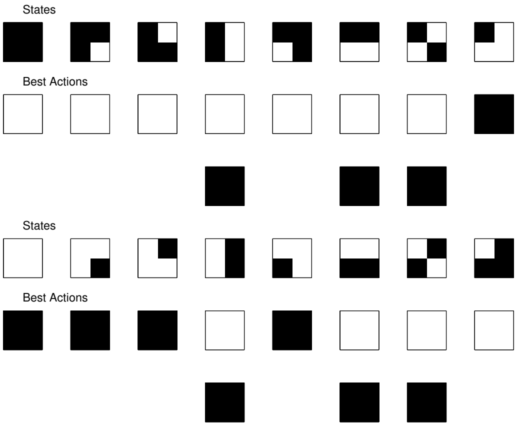

Figure 2: The best possible action(s) for each state of the 2x2 grid game with the original setting. The first and third row show all possibles states in the 2x2 grid game. Below each state the combined action yielding maximal social welfare is plotted.

Original Setting

	Time steps T		CI for ω	
Groups	CL	QL	CL	QL
4x1	5440	9274	0.4997±0.0069	0.4856±0.0081
2x2	1888	5961	0.6361±0.0066	0.5924±0.0059
1x4	2855	3709	0.7004±0.0061	0.7231±0.0068

SLI-Setting

4x1	5016	908	0.6244±0.0069	0.7448±0.0047
2x2	1151	7451	0.6790±0.0065	0.6608±0.0097
1x4	3331	6126	0.7771±0.0050	0.8036±0.0098

SI-Setting

4x1	274	209	1.0000±0.0008	0.9999±0.0004
2x2	461	272	1.0000±0.0010	0.9996±0.0010
1x4	2640	7230	0.9336±0.0056	0.9663±0.0070

Table 5: Results for the 2x2 grid. The time steps until convergence T with $\varepsilon = 0.025$ and the 95% confidence interval of the achieved social welfare ω for $p_{black} = 0.5$ are shown.

are summarized in Appendix B Figure 9, 10 and 11 respectively to get a more intuitive feeling about the mean convergence values depending on the different setups.

Results for the 2x2 grid game

As said in Section 5, to evaluate the results on the grid sizes, the best possible social welfare was calculated beforehand and then plotted with the achieved social welfare of the different learners.

In the smallest grid size that was examined, it is still feasible to plot all different states, since there are in total $2^4 = 16$ states. Additionally, there are for each state exactly the same amount of joint actions which can be taken. Therefore, the total number of possibilities is $2^8 = 256$, which can be searched exhaustively. The results are shown in Figure 2, where the best joint actions in the original setting are plotted to the corresponding state. The 2x2 grid game has only two possible best actions, which are all agents playing black or all agents playing white, depending on the state the majority of the agents is in. Likewise, since $p_{black} = 0.5$, this game is symmetric.

Original Setting

	Time steps T		CI for ω	
Groups	CL	QL	CL	QL
9x1	271	171	0.4997±0.0035	0.4910±0.0048
4x2+1x1	1195	255	0.5779±0.0047	0.5510±0.0046
3x3	2232	3868	0.5911±0.0056	0.5563±0.0068
1x9	4804	3	0.3208±0.0086	0.3004±0.0040

SLI-Setting

9x1	771	818	0.5791±0.0044	0.7444±0.0040
4x2+1x1	1172	2849	0.6473±0.0044	0.6862±0.0065
3x3	2165	1434	0.6390±0.0059	0.6157±0.0059
1x9	6087	29	0.3968±0.0079	0.3755±0.0036

SI-Setting

9x1	273	211	1.0000±0.0007	0.9999±0.0004
4x2+1x1	440	379	0.9999±0.0011	0.9996±0.0018
3x3	1298	5564	0.9748±0.0033	0.9718±0.0109
1x9	6783	18	0.3452±0.0084	0.3012±0.0048

Table 6: Results for the 3x3 grid. The time steps until convergence T with $\varepsilon = 0.025$ and the 95% confidence interval of the achieved social welfare ω for $p_{black} = 0.5$ are shown.

Original Setting

	Time steps T		CI for ω	
Groups	CL	QL	CL	QL
16x1	285	107	0.4996±0.0032	0.4915±0.0038
8x2	1462	1144	0.5660±0.0049	0.5326±0.0048
4x4	4165	3262	0.6046±0.0066	0.5969±0.0086
2x8	6922	2	0.3442±0.0094	0.3014±0.0025

SLI-Setting

16x1	725	3074	0.5863±0.0037	0.7448±0.0028
8x2	1031	3592	0.6341±0.0044	0.6782±0.0057
4x4	3982	4835	0.6633±0.0070	0.6721±0.0083
2x8	5557	1	0.4152±0.0119	0.3763±0.0037

SI-Setting

16x1	279	209	1.0000±0.0007	0.9999±0.0004
8x2	461	402	0.9999±0.0010	0.9995±0.0017
4x4	2732	7001	0.8956±0.0050	0.9287±0.0083
2x8	8488	12	0.4053±0.0116	0.3029±0.0046

Table 7: Results for the 4x4 grid. The time steps until convergence T with $\varepsilon = 0.025$ and the 95% confidence interval of the achieved social welfare ω for $p_{black} = 0.5$ are shown.

The results for the different learners are presented in Table 5, Figure 3 and Figure 4. When comparing Q-learning with Cross-learning, it can be seen, that QL converges slower. However, these results have to be treated with care. Especially, when comparing the calculated numbers with a visual inspection of the figures, the calculated Time steps to converge for the individual CL in original and the SLI-Setting and for QL in the original setting seem unlikely. The individual learning (4x1) and groups of size two (2x2), seem to converge well below 1000 iterations, although the calculated numbers are much higher.

This can be explained by the high variance in average social welfare, when the decisions are taken according to the alternating strategy. For instance, when regarding Figure 2, the first state of each row will lead to an average social welfare of 1 and the second last state in the row will yield zero reward, since there are not any two adjacent

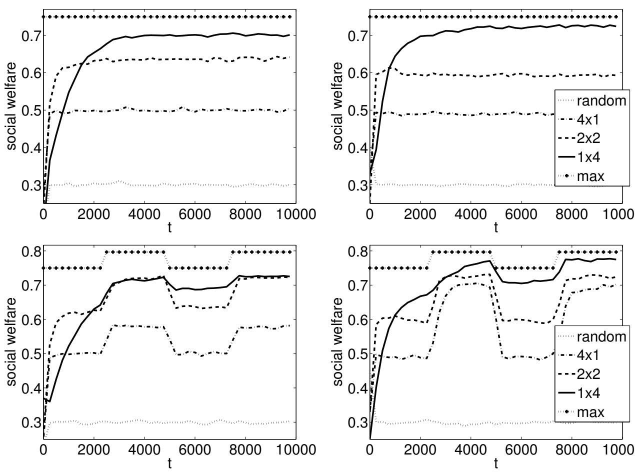

Figure 3: Plots for the 2x2 grid using the original setting. The top row represents the results with constant state bias, while the bottom row represents the switching state bias. Left: Cross-learning, right: Q-learning

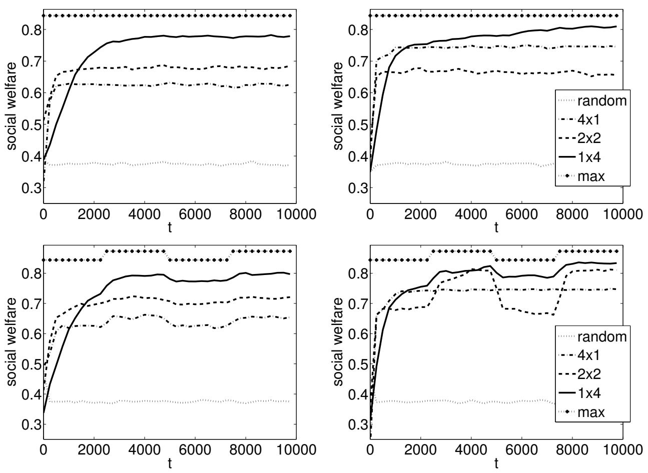

Figure 4: Plots for the 2x2 grid using the SLI-Setting. The top row represents the results with constant state bias, while the bottom row represents the switching state bias. Left: Cross-learning, right: Q-learning

decisions equal. With a large enough number of runs this effect will be equalled out, but with averaging over 10 runs, there is a high possibility that a single run induces a deviation that is larger than ε.

Only the full space joint action learning achieves a value higher than the $\omega_{stateless}$ for the original setting and the SLI-Setting. Likewise, for the original setting, the more actions are controlled by the learner, the more reward can be achieved.

Another interesting result is that while individual QL and individual CL both converge in the original setting to the strategy $\omega_{alternate}$, individual QL converges to the $\omega_{stateless}$ in the SLI-Setting. This might be since the probability of exploration is high enough and the value achieved by the NE-strategy is a stronger incentive in the SLI-Setting. In the original setting the differences between $\omega_{stateless}$ and $\omega_{alternate}$ is 0.1, while in the SLI-Setting it is 0.25, which is 50% more for the NE-strategy in comparison to the alternating strategy. Hence, QL adapts to the better strategy only when the incentive is high enough.

For switching state bias in the original setting, the general order of performances does not change much. When the state bias p_{black} rises, the black states are visited more often. Therefore, there has to be a difference in the expected social welfare, but for the learning process as a whole it does not change. Since full joint action learning has complete information, it is expected to be less effected by the change. This can be confirmed by the results as shown in Figure 3. Additionally, the individual and 2x2 CL are effected by the change, but their differences in reward remains roughly the same.

In contrast, the individual and 2x2 QL achieve quite similar rewards as soon as the state bias rises and drop back to the old levels again as soon as the state bias drops again. That dramatic impact on QL can be explained by the constant exploration, i.e., by fixing the temperature. Therefore, the individual learner and the groups of 2x2 have the chance to adapt to the changes while the CL converges to a fixed strategy. Additionally, the nature of smaller groups is more flexible to adapt to changes in the environment, while the larger groups need longer to take notice that the environment has changed.

In the switching SLI-setting, most of the CL and QL groups converge to a mixed strategy of preferring the alternate action of the dominant state even when being in the other state. This means, if p_{black} rises, action white is chosen with a higher probability, even when being in state white. Individual Q-learning, on the other hand, converges to $\omega_{stateless}$ with equal probability of selecting white or black as action. Therefore, the average over all runs results in the straight line as seen in Figure 4 bottom right.

Conclusively, in the 2x2 grid game, full joint action learning is advisable, since it yields the highest rewards for each setting. However, when the state is less important, individual QL is quite close to the performance of the centrally controlled group at a much higher speed of convergence.

Results for the 3x3 grid game

Coming to the larger grid size of 3x3 the resulting plots are shown in Figure 5 and Figure 6. Additionally, Table 6 gives an overview on the calculated confidence intervals and number of time steps to converge. When examining the figures, it is evident that the overall performance has decreased when comparing to the theoretical maximum value. For every case, the simple NE strategy would outperform all of the learners. That said, it can

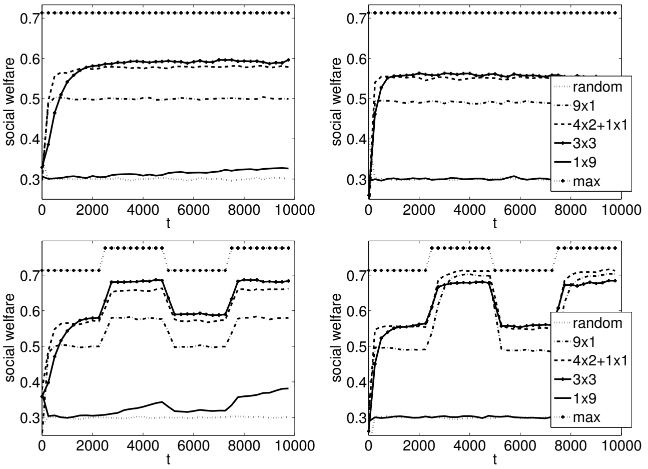

Figure 5: Plots for the 3x3 grid using the original setting. The top row represents the results with constant state bias, while the bottom row represents the switching state bias. Left: Cross-learning, right: Q-learning

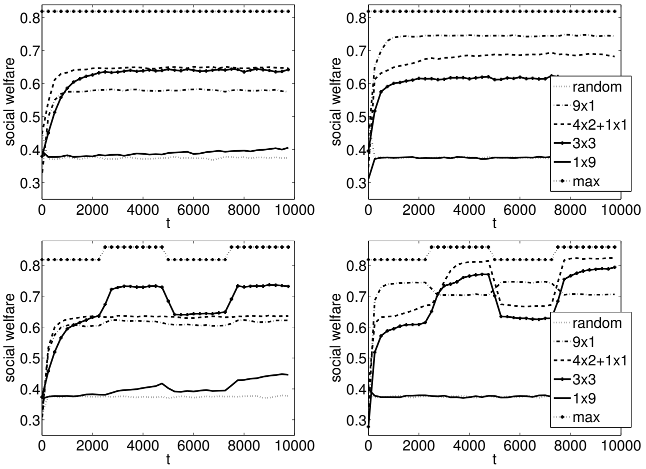

Figure 6: Plots for the 3x3 grid using the SLI-Setting. The top row represents the results with constant state bias, while the bottom row represents the switching state bias. Left: Cross-learning, right: Q-learning

be seen that the general trends are the same as in the smaller grid size, with a higher level of dependency a higher value is achieved, although in both settings 3x3 is only barely better than 4x2+1x1. This might relate to the small comparatively small increase in CC between the two settings as seen in Table 4. The setting of the groups at 3x3 mean that on average only roughly 50% are determined by decisions taken by JAL, the rest is influenced by the other decisions. Hence, taking the same action might lead to totally different rewards. For the setting of 4x2+1x1 the reward depends roughly 40% on the groups' own action, which is not that much less than in the 3x3 setting. Therefore, the rewards are similar.

Furthermore, CL with group size 3x3 in the original setting almost achieves the value of $\omega_{stateless} = 0.6$.

Remarkably, the achieved social welfare with full joint action learning and QL does not change at all over the whole 10,000 iterations. When controlling nine agents, with two actions and two states each, the total number of possible state-action pairs reach $2^{18} = 262144$, which is a multiple of the total number of time steps. Therefore, it is as good as playing random actions, since each state is just not visited enough to achieve a proper learning. The similar reasoning holds for CL, even though the value does rise a bit as seen in the top left of Figure 5 and Figure 6, but still it is nowhere near to convergence.

Interestingly, the order of the performance of the different group sizes in QL reverses from the original setting to the SLI-Setting. While in the original setting 3x3 scored highest and individual learning lowest, this is reversed in the SLI-Setting. This is the case, since the NE strategy is achieved easier with a lower level of CC. In the GG, which has not state, it is seen that it is advisable to use individual learning. This can be transferred to the SLI-Setting. The state is less important and the NE strategy yields higher reward than the alternating strategy. Thus that the individual learner adopts to a NE strategy easier than JAL is consistent with the results of the GG.

Additionally, it is notable that for CL with switching state bias in the SLI-Setting only the groups of 3x3 and the full space learning seem to be influenced by the change in state bias. This is probably due to the fact that the other settings converge equally likely to preferring black and to preferring white, which is on average equalled out to the straight line.

On the other hand, an interesting result is that the individual QL in the switching SLI-Setting drops as soon as the state bias changes, while all other settings stay the same or rise. This is probably due to stochastic effects, where it converged slightly more often to the action black before the state bias changed. With the increase in p_{black} the action black becomes inferior, and thus the average social welfare drops.

To sum up, already with a grid size of 3x3 full joint action learning is not advisable anymore, since it is just too complex to achieve good results in reasonable time. Furthermore, it can be seen that it is important how interconnected the groups are, since the groups of 3x3 do not perform a lot better than 4x2+1x1. Lastly, it can be concluded that with QL the less important the states are, the less % of CC is advisable, since it is easier to converge to a NE strategy.

Results for the 4x4 grid game

The results for the largest grid size are shown in Figure 7, Figure 8 and Table 7. In the original setting, the advantage of a higher percentage of CC can be seen

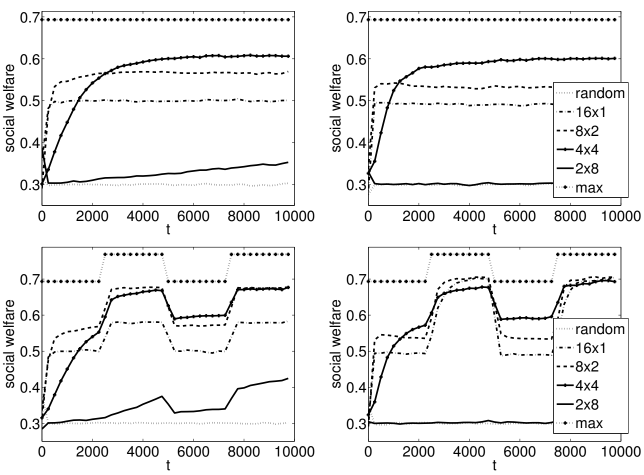

Figure 7: Plots for the 4x4 grid using the original setting. The top row represents the results with constant state bias, while the bottom row represents the switching state bias. Left: Cross-learning, right: Q-learning

again. However, for the largest group size of 8x2, the same applies as for the full joint action learning in the 3x3 grid. It has too many possibilities to evaluate with only 10,000 iterations. Individual learning converges at $\omega_{alternate} = 0.5$ and only the 4x4 CL and QL achieve the same as $\omega_{stateless} = 0.6$. These are similar results as found in the 3x3 setting.

Again, with switching state bias, the CL converges too early to really benefit of the better expected reward, while QL is able to adapt its strategy, yielding a much higher reward when $p_{black} = 0.7$ for all group sizes. Similar to the other grid sizes the social welfare drops again to the old level as soon as the state bias is changed back.

When regarding the SLI-Setting presented in Figure 8 the general trends are the same as for 3x3. For CL, the difference in CC is visible again. It takes much more time steps to converge for a higher level of CC, but eventually yielding a higher reward. However, in Q-learning the less controlled groups, especially the individual learner converge at the $\omega_{stateless} = 0.75$ and thereby outperform the groups of 4x4 with a much higher level of CC. This is also consistent with the results in the 3x3 grid game.

Regarding switching state bias, the results for CL are very similar to those at constant state bias. On the other hand, for QL the groups of 8x2 are able to adapt to the changed state bias and therefore outperform the other settings when the state bias has changed. Afterwards, it drops back to its old level, while the individual learner converges equally likely to one of the NE strategies, which averages to the line at $\omega_{stateless} = 0.75$.

Conclusively, in 4x4, the results are similar to the results seen in the other grid sizes. In general, in the original setting a higher level of CC performs better at the expense of high computational complexity. When the state becomes less important the incentives to adopt a NE strategy rise. However, only the individual QL achieves the $\omega_{stateless} = 0.75$. With changing state bias, the CL converges too early to benefit from the better expected rewards, while the smaller groups and QL are able to profit most from the change in the environment.

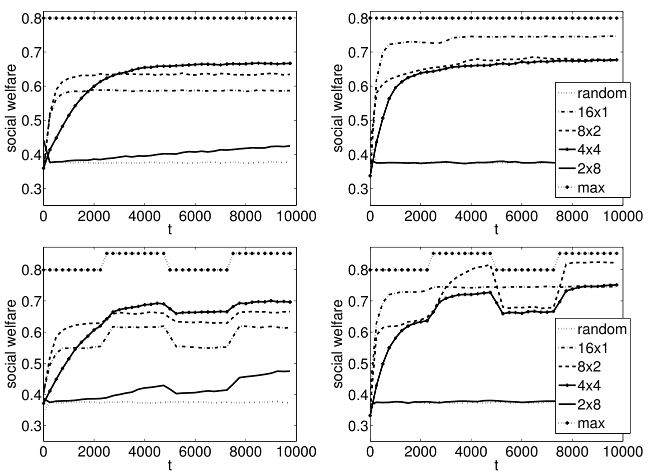

Figure 8: Plots for the 4x4 grid using the SLI-Setting. The top row represents the results with constant state bias, while the bottom row represents the switching state bias. Left: Cross-learning, right: Q-learning

7 Conclusion

The performances of the learners have been shown to be dependent on the different levels of CC and the correct parameters. Additionally in the GG, it has be shown that CC does not yield better rewards, in contrast, it resulted in much worse rewards. For the grid game it has been shown that the rise in CC, in general leads to better rewards in the original setting. However, the complexity rises exponentially when increasing the group sizes for JAL. Thus, the number of time steps has to be balanced with the number of possible states to facilitate convergence.

Additionally, it has been found that a setting where the controlled decisions are highly interconnected are preferable, since they yield higher rewards with only slowly increasing complexity. For instance the groups of 4x4 in the 4x4 grid game already have a CC of more than 70%, while still being reasonable complex in terms of possible state-action pairs.

Conclusively, it has been shown that as soon as the states become less important, a high level of CC actually hinders convergence to a higher NE strategy. Hence, the optimal balance of CC and anarchy is very dependent on the task.

With respect to the analogy of the industry introducing standards, it can be seen that if a controlling agency has to be deployed, it should be only on a highly interconnected number of companies. Furthermore, it is impossible to achieve the maximum social reward with full joint action learning in a realistic time frame. Hence, a certain level of self-control (anarchy) is desirable.

On the other hand, it can be concluded that an agency, which would force to introduce a certain standard for everybody regardless of their states, can also be preferable. As seen in the experiments, the social welfare of $\omega_{stateless}$ is more than most learners achieved after convergence.

The contributions of this paper can be summarized as follows: the value iteration method of Q-learning has been compared with the policy iteration method of Cross-learning. Additionally, the consequences of different levels of CC where compared in the n-player Guessing Game and an abstract spatially constrained grid game. Furthermore, it has been shown that while full joint action learning yields the highest reward, it is infeasible for most settings due to its complexity.

Lastly, it has been shown that JAL is preferable when having easily separated highly interconnected groups and with games where the rewards are heavily state dependent. This is the case in many real life applications. Learning in the real life is heavily dependent on the individuals' states, since knowledge of each individual differs, and furthermore, it is generally possible to separate the population into highly interconnected groups. Hence, JAL with groups is the best choice, since this gives the benefits of coordination at a reasonable expense of complexity.

8 Further Outlook

This section will propose some ideas which could be researched in the future. The results can be evaluated on different games, for instance the proposed traffic simulation model could be used to confirm the results found in this paper.

Furthermore, the performance of an "environment conscious individual learner" could be tested in comparison to the introduced JAL. The environment conscious individual is aware of its adjacent states, but does not know or decide upon their actions. This could lead to some interesting results while not increasing the complexity too much, since each agent has at most four neighbours. Hence, the maximal number of states is 2^5, which can be handled easily, since there are still only two actions.

Another interesting option for a learner could be if each learner votes for all the decisions on which the reward of a certain cell depend upon. Hence, the CC is theoretically increased to 100%, while the actually performed actions are the outcome of the votes given by each learner. This would increase the complexity to at most 2^{10} for the central fields, since we have four neighbours and the own agent with two states and actions each.

References

[1] Cross, J.G. (1973). A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, Vol. 87, pp. 239–266.

[2] Gibbons, R. (1992). *A Primer in Game Theory*. Harvester Wheatsheaf.

[3] Gintis, H. (2000). *Game Theory Evolving*. Princeton University Press.

[4] Hennes, D. (2007). Reinforcement learning in multi-agent games - A policy based perspective. *BSc thesis, Universiteit Maastricht*.

[5] Hu, J. and Wellman, M. P. (2003). Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, Vol. 4, pp. 1039–1069.

[6] Kaisers, M. (2007). Reinforcement learning in multi-agent games - A value iteration perspective. *BSc thesis, Universiteit Maastricht*.

[7] Law, A.M. (2007). *Simulation Modeling and Analysis*. McGraw Hill, 4th edition.

[8] Mitchell, T. (1997). *Machine Learning*. McGraw Hill, 1st edition.

[9] Narendra, K. and Thathachar, M.A.L. (1989). *Learning Automata An Introduction*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

[10] Nowé, A., Verbeeck, K., and Peeters, M. (2006). Learning automata as a basis for multi agent reinforcement learning. *LAMAS 2005*, Springer LNAI 3898, pp. 71–85.

[11] Russell, S. and Norvig, P. (2003). *Artificial Intelligence A Modern Approach*. Pearson Education, Inc., 2nd edition.

[12] Schelling, T. (1960). *The Strategy of Conflict*. Harvard University Press.

[13] Sutton, R.S. and A.G.Barto (1998). *Reinforcement Learning: An Introduction*. MIT Press, 1st edition.

[14] Tsetlin, M.L. (1962). On the behavior of finite automata in random media. *Autom. Remote Control*, Vol. 22, pp. 1210–1219.

[15] Verbeeck, K. (2004). *Coordinated Exploration in Multi-Agent Reinforcement Learning*. Ph.D. thesis, Vrije Universiteit Brussel.

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

A Acronyms

CC Central Control

CL Cross-Learning

GG Guessing Game

GT Game Theory

JAL Joint Action Learning

NE Nash Equilibrium

QL Q-Learning

RL Reinforcement Learning

SLI-Setting State Less Important Setting

SI-Setting State Independent Setting

B Additional Figures

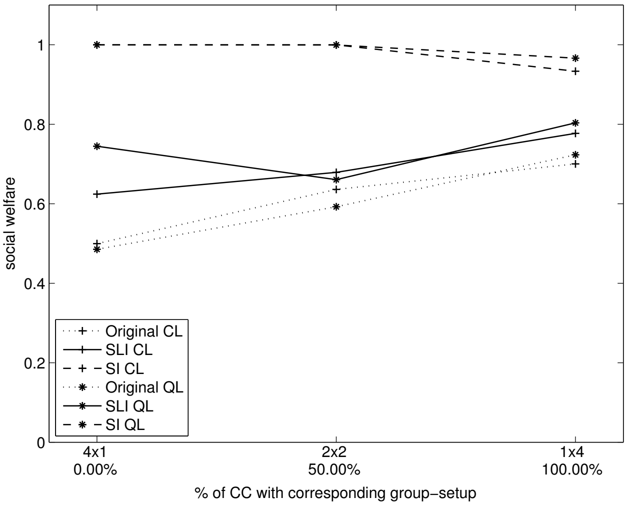

Figure 9: Summarized results for the 2x2 grid game. The mean convergence value of the social welfare is plotted depending on the percentage of CC.

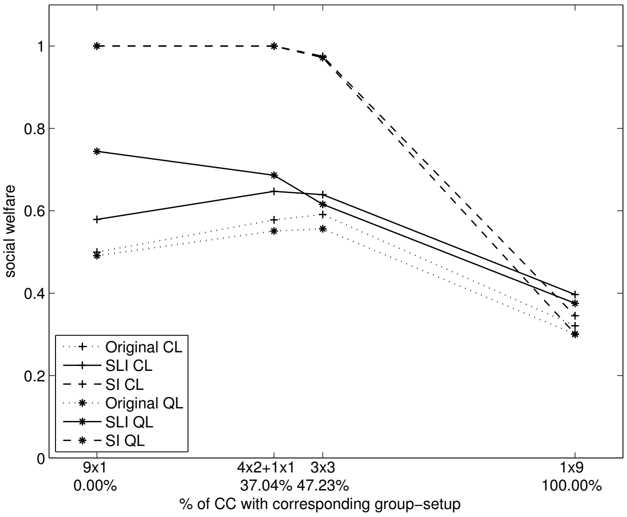

Figure 10: Summarized results for the 3x3 grid game. The mean convergence value of the social welfare is plotted depending on the percentage of CC.

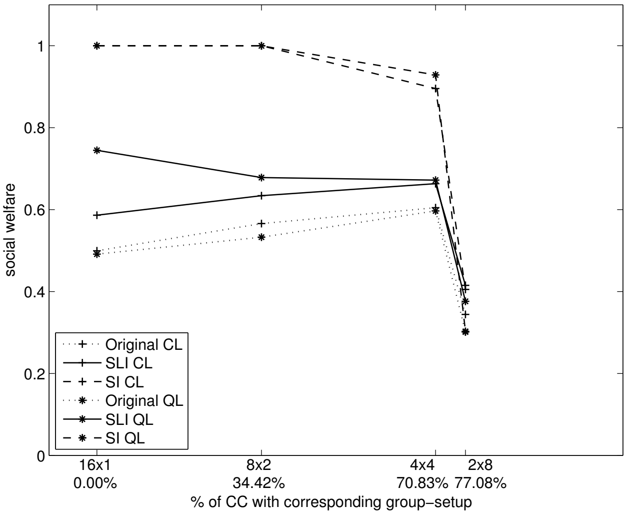

Figure 11: Summarized results for the 4x4 grid game. The mean convergence value of the social welfare is plotted depending on the percentage of CC.