Thesis submitted for the degree of

Master of Science in Informatics (Knowledge and Data Management)

Studying Cooperative Multi-Agent Reinforcement Learning in Networks

دراسة تعزيز التعلم التعاوني لعدة عملاء في الشبكات

By

Mayada Mohamed Oudah

Faculty of Engineering and Information Technology

Thesis Supervisor

Dr. Sherief Abdallah

May, 2012

Abstract

Cooperative Multi-Agent systems, where agents work together as one team to achieve a common goal, form the majority of real-life multi-agent applications. Therefore, it is important to find a suitable multi-agent reinforcement learning algorithm to help agents to achieve their goal through finding the optimal joint policy that maximizes the team's total reward. Since the last decade, several multi-agent learning algorithms have been proposed and applied to cooperative multi-agent settings. However, most of these learning algorithms do not allow agents to communicate with each other during the execution time, making it hard for agents to coordinate their actions especially in large-scale and partially observable domains. Thus, several coordinated learning algorithms which allow agents to communicate during the execution time have been applied to large cooperative multi-agent domains and proved to be efficient and effective in such domains.

Nonetheless, to the best of our knowledge, there is no work that studied the characteristics of such learning algorithms under different network structures. The work done in this thesis aims to study and analyze the characteristics of one of the recent coordinated multi-agent learning approaches, the coordinated Q-learning algorithm, in two-player two-action cooperative and semi-cooperative games under random and scale-free network structures. Also, this thesis conducts a comparison between the original Q-learning algorithm and the coordinated Q-learning algorithm to better understand the difference between both of these algorithms. A simulator has been built in order to conduct experimental analyses.

Experimental results verify the robustness, effectiveness and efficiency of the coordinated Q-learning algorithm. The coordinated Q-learning algorithm converges faster and performs better than the original Q-learning algorithm due to its distributive nature and its communication feature which do not exist in the original Q-learning algorithm. Also, the performance of the coordinated Q-learning is not affected by the network structures of random and scale-free networks. Such characteristics can be utilized in future works to further improve the performance of different coordinated learning algorithms in different cooperative multi-agent domains.

خُلاصة

تشكل الأنظمة التعاونية متعددة العملاء (أو متعددة المستخدمين)، حيث يقوم جميع العملاء بالعمل معاً كفريق واحد لتحقيق هدف مشترك، معظم التطبيقات الواقعية متعددة العملاء. لذلك، يعتبر إيجاد خوارزمية مناسبة لتعزيز تعلم العملاء لمساعدتهم في تحقيق هدفهم من خلال إيجاد السياسة المشتركة الأمثل التي إذا طُبِّقت في النظام فإنها ترفع من القيمة الكلية للمكافأة المكتسبة للفريق. تم اقتراح و تطبيق العديد من الخوارزميات لتعزيز التعلم منذ العقد الماضي، ومع ذلك فإن معظم هذه الخوارزميات لا تسمح للعملاء بالتواصل مع بعضهم البعض أثناء وقت التنفيذ، مما يجعل من الصعب على العملاء تنسيق حركتهم و أعمالهم، خصوصا في المجالات واسعة النطاق و التي يمكن معاينتها جزئياً فقط. و بالتالي، تم تطبيق عدة خوارزميات جديدة التي تحفز التعلم المنسق و تسمح للعملاء بالتواصل خلال فترة التنفيذ في عدة مجالات، و تم إثبات كفاءة و فعالية هذه الخوارزميات في هذه المجالات.

إلى حد علمنا، لا يوجد أي عمل بحثي لدراسة خصائص مثل هذه الخوارزميات عندما يتم تطبيقها في عدة شبكات مختلفة. العمل المنجز في هذه الرسالة يهدف إلى دراسة و تحليل خصائص واحدة من أحدث الخوارزميات التي تنسق التعلم في الأنظمة متعددة العملاء، و تعرف هذه الخوارزمية باسم "Coordinated Q-learning". يتم تطبيق هذه الخوارزمية في هذه الرسالة في عدة ألعاب تعاونية و شبه تعاونية و يتم مقارنة أداء الخوارزمية الأصلية Q-learning و الخوارزمية المنسقة Coordinated Q-learning لتحديد الفروقات بين هذه الخوارزميات. و قد تم بناء جهاز محاكاة من أجل تنفيذ و تحليل التجارب. تُثبِت النتائج التجريبية متانة و فعالية و كفاءة الخوارزمية المنسقة Coordinated Q-learning التي تؤدي أفضل من الخوارزمية الأصلية نظراً لطبيعتها التوزيعية و ميزة الاتصال بين العملاء التي لا توجد في الخوارزمية الأصلية Q-learning. تم ملاحظة أن أداء الخوارزمية المنسقة لا يتأثر باختلاف نوع الشبكة، و يمكن استخدام هذه الخصائص في الأعمال المستقبلية لتحسين أداء الخوارزميات المختلفة في مختلف المجالات متعددة العملاء.

Acknowledgements

I would like to thank Allah for providing me the enough strength to finish this important work and to thank my parents for their continuous support and encouragement and for providing the best environment for me to do the best I can. Also, I would like to thank my thesis supervisor Dr. Sherief Abdallah for his support and guidance throughout the thesis and for always being there whenever I needed help. Finally, I would like to thank my brother for always reminding me that there is a time when the person must take a break and enjoy life!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides a general overview about cooperative multi-agent systems, their importance and how to solve the potential problems presented in them through using coordinated multi-agent reinforcement learning. In addition, this chapter presents the motivations of this thesis, its main contributions to the current knowledge and the research questions it aims to answer.

1.1 General Overview

Cooperative multi-agent systems form the majority of real-life multi-agent systems where agents act as one team and try to jointly solve a common problem and maximize the team total reward (Panait & Luke, 2005). Several multi-agent reinforcement learning frameworks have been proposed to model cooperative systems to help agents achieve their common goal by finding the best joint action that maximizes the team total reward through applying a multi-agent learning algorithm. However, as the number of agents increases, the complexity of the agents' joint interactions gets higher resulting in a poor modeling of the cooperative system. Learning algorithms applied to poor models achieve poor solution and agents may not able to reach their goal. Networked Distributed Partially Observable Markov Decision Process (ND-POMDP) model has been proposed to solve this complexity problem (Nair *et al.*, 2005). However, most of the learning algorithms applied to ND-POMDPs are offline algorithms (i.e. does not learn during execution time) and need a very accurate model which is very expensive and hard to be obtained in large-scale, partially observable domains.

Coordinated multi-agent reinforcement learning is a new multi-agent reinforcement learning approach that has been proposed to coordinate the interactions between agents in cooperative systems by allowing online communication between agents during the execution time. In a recent work carried out by Zhang and Lesser (2011) a model-free and scalable coordinated multi-agent learning approach (will be known from now on as the Coordinated Q-learning approach) has been

proposed and applied to ND-POMDPs to compute the optimal joint action that maximizes the group total reward. Experimental results, achieved in the same work, verify the effectiveness and efficiency of the proposed approach when applied in the distributed sensor networks domain.

1.2 Motivations and Objectives of the Thesis

Since many multi-agent applications in real life are based on cooperative multi-agent systems, that can be solved using coordinated multi-agent reinforcement learning approaches, and as the coordinated Q-learning approach does not need a specific model to be applied in and requires low communication and computation complexity compared to other coordinated approaches, it is important to study agents' interactions in such systems and to solve the potential problems which can be faced when searching for the optimal joint action for agents deployed in these cooperative systems in order to improve the performance of real-life multi-agent systems that are built based on cooperative multi-agent systems. Therefore, this thesis aims to study and analyze the performance of the coordinated Q-learning approach in two-player two-action cooperative and semi-cooperative games, which are commonly used as a framework to best represent agents' interactions in a cooperative domain, under different network structures such as random and scale-free networks.

The main contribution of this thesis to the current knowledge is to specify and understand the characteristics of the coordinated Q-learning approach in order to improve its performance in different cooperative multi-agent systems and to state under what situations and networks the coordinated Q-learning approach performs better. A simulator of the tested games, networks and learning algorithms has been built to conduct several experiments in order to accomplish the objectives of the thesis. While the original work of the coordinated Q-learning algorithm (Zhang & Lesser, 2011) tests the performance of the algorithm in the sensor network domain and focuses on regular grids, here we study the performance of the coordinated Q-learning algorithm under random and scale-free networks when applied in games which, due to its simplicity, enable us to get more insights about the algorithm performance. Also, this thesis provides a comparison between the coordinated Q-learning algorithm and the original Q-learning algorithm to understand the main difference between both algorithms. Finally, since the coordinated Q-learning approach depends on distributing the learning agents among a number of groups, the original work

of the coordinated Q-learning algorithm used a hand-coded grouping of agents, while a simple grouping algorithm has been proposed in this thesis to perform the grouping process and to ensure cycle-free grouping.

1.3 The Thesis Research Questions

This thesis aims to achieve its main contribution, mentioned in the above section, by answering the following research questions:

- **Does the coordinated Q-learning approach help in improving the performance of multi-agent learning algorithms in networks when applied in two-player two-action cooperative/semi-cooperative games?**
- **Is the performance of the coordinated Q-learning approach affected by different network structures such as random and scale-free networks?**
- **Is there a simple grouping methodology to cluster agents in a network automatically? Can such methodology ensure cycle-free clustering?**
- **Is the performance of the coordinated Q-learning approach affected by some of its parameters?**

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 provides a comprehensive review about Multi-Agent Reinforcement Learning (MARL), Game Theory and Coordinated MARL. Chapter 3 presents a detailed description of the adopted coordination approach, the coordinated Q-learning approach, by presenting the main assumptions of the approach, how the learning process is carried out, how the global joint policy is computed and how we adjust this approach to work in different domain than the one it was designed for. The settings of the experiments conducted to evaluate and compare the performance of both the original Q-learning and the Coordinated Q-learning algorithms in two-player two-action games under different network structures are provided in Chapter 4. Chapter 4 also presents the experimental results, investigates the possible impact of modifying some parameters on the performance of the coordinated Q-learning algorithm and answers the thesis research questions. Finally, Chapter 5 concludes the work done in this thesis and provides a set of possible works to be done in the future.

Chapter 2

Background

This chapter provides a comprehensive review about Multi-Agent Reinforcement Learning (MARL), its problem definition, the proposed models for representing these problems and some examples of the learning algorithms used to solve these models. It also reviews the main concepts of Game Theory, its main types and several examples of each. Finally, this chapter defines Coordinated MARL, its importance and reviews several related works about it.

2.1 Multi-Agent Reinforcement Learning

Reinforcement Learning (RL) has been an interesting and challenging research topic to be studied and investigated in Machine Learning field for many years. Unlike "supervised learning", training examples for learning systems in RL are not provided in the shape of input-output pairs. Instead, learning systems must explore and search for the most suitable output for every input in which, for each input, there is a set of possible outputs each has a value for being chosen and the output with the highest value is considered to be the best output for that input. The goal of RL is to reinforce outputs which result in high values and to weaken outputs with small values (Sandholm & Crites, 1995; Sandholm & Crites, 1996). RL is carried out as follows: each learning system/agent observes the environment and identifies its current state in this environment, selects an action to perform on the environment (making it transition to a new state) and then receives a reward of performing that action at the previous state and then repeat from observing the new current state.

Machine Learning studies two settings of RL, Single-Agent RL (SARL) and Multi-Agent RL (MARL). In SARL settings, learning systems are agents who, consecutively and separately, interact with the environment they are deployed in and aim to learn the best action that affects the environment in a way which maximizes their utility function. While in MARL settings, learning systems are agents who, simultaneously, interact with both the environment they are deployed in

and other agents and aim to learn the best action which maximizes their utility function. See figures 2.1 and 2.2 for an illustration example of SARL and MARL respectively.

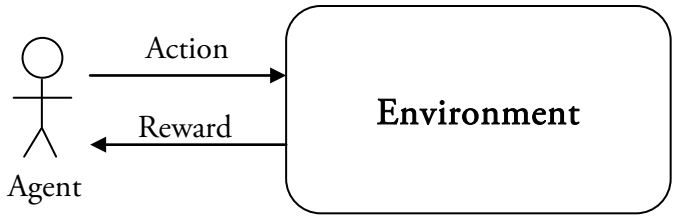

Figure 2.1: An illustration of a SARL setting. Each Agent, separately, executes an action that changes the state of the environment and receives a reward value for executing the action.

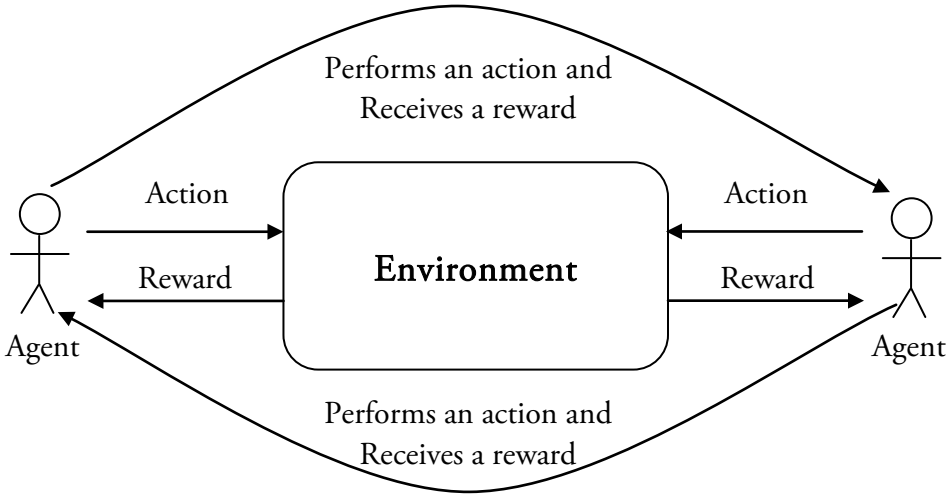

Figure 2.2: An illustration of a MARL setting with two agents. Both agents execute an action that affects the state of the environment and the other agent and receive a reward for executing the action

The main difference between SARL and MARL is that in the former, each agent learns separately from the other and its action has nothing to do with other agents, while in the latter, agents learn at the same time and the action of an agent affects the environment, limiting the possible set of actions of other agents, and can directly affect other agents (Shoham & Powers, 2010). Since in this thesis we are interested in MARL, the following sub-sections define the problem faced in MARL, its general representative models and provide several examples of learning algorithms which solve these models.

2.1.1 Problem Definition

The problem of MARL can be described as a decision problem where a multiple number of agents need to choose the best action that maximizes their utility function. The environment, where the agents are deployed, has a set of states in which each agent has a set of possible actions to choose from in each of these states. After executing actions, agents receive an immediate reward value for executing each action at each state. The challenge faced in MARL settings is that, since agents are learning simultaneously in the same environment causing it to be non-stationary and in realistic situations agents usually suffer from partial observability in which no agent has a full observation of the world states, each agent must take into its considerations the states and actions of other agents in order to learn effectively and efficiently (Abdallah & Lesser, 2007). However, in real-life situations agents suffer from limited communication during execution time in that they cannot communicate their states and actions to all other agents. The goal of MARL can be one for all agents (i.e. all agents aim to find the best joint actions to maximize their total expected reward) or one for each agent (i.e. each agent aims to maximize its own expected reward). Agents with the same goal are called cooperative agents, while agents with different goals are called self-interested agents (Mostafa, 2011).

There are two types of MARL decision problems: non-sequential decision making problems, where agents only care about finding the best policy (i.e. mapping states to actions) in order to maximize the immediate payoff (i.e. reward value), and sequential decision making problems, where agents are interested in finding the best policy that maximizes the future payoff. Sequential decision making problems are more difficult than non-sequential decision making problems in that agents interact with the environment for a longer period of time and each executed action in a state affects the set of possible actions for next states (Sandholm & Crites, 1995). Sequential decision making problems are used in most of MARL applications. MARL sequential decision problems can be formalized through using several representative models that simplify the underlying problem and provide an optimal solution to such problems. The following sub-section defines several

commonly used representative models which are used to model the domain of MARL problems where cooperative agents interact with each other.

2.1.2 General Representative Models

- **Markov Decision Process**

Several representative models have been designed to formalize the problem of sequential MARL. Marcov Decision Process (MDP) model (Howard, 1960) is considered to be the most commonly used model to represent the environment of RL problems in which it is originally designed for formalizing SARL problems but can be extended to formalize MARL problems too.

Definition 1. In its original form, an MDP is defined by a 4-tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R} \rangle$ where

- S is a finite set of world states.

- **A** is a finite set of actions.

- $\mathbf{P}: \mathbf{S} \times \mathbf{A} \rightarrow [\mathbf{0}, \mathbf{1}]$ is the transition probability function, where P ($s' \mid a, s$) is the probability of moving to a new state s' when performing action a in state s

- $\mathbf{R}: \mathbf{S} \times \mathbf{A} \rightarrow \mathfrak{R}$ is the reward function and $\mathrm{R}(a, s)$ specifies the expected immediate reward of choosing and executing action a in state s.

The above definition works only for SARL problems and does not suit MARL problems because besides assuming that only one agent interacts with the environment at a time (i.e. the environment is stationary), it assumes full observability of the world states and that each state has all the information needed by the agent. All these assumptions are unrealistic and impractical in Multi-Agent domains where the environment is non-stationary and suffer from uncertainty and partial observability. Therefore, a number of generalized forms of MDP has been proposed and successfully applied to MARL problems such as Multi-agent MDP (Boutilier, 1996), Partially Observable MDP (Kaelbling, Littman & Cassandra, 1998) and Decentralized Partially Observable

MDPs (Bernstein, Zilberstein & Immerman, 2000) . In this thesis, we are concerned about Decentralized Partially Observable MDPs (DEC-POMDPs) model which is one of the commonly used models for formalizing MARL domains.

- **Decentralized Partially Observable MDP**

DEC-POMDP is commonly used to reasonably and realistically model domains where cooperative agents need to coordinate their actions under uncertainty and partial observability. Unlike MDP, the environment modeled using DEC-POMDP is controlled by distributed multiple agents where each agent does not have full observability of the world states but obtains observations which help in forming a belief about the world states. However, the joint partial observations by all agents at a state do not necessarily completely determine that state.

Definition 2. An *n*-agent DEC-POMDP is defined by the tuple $\langle \mathbf{I}, \mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R}, \mathbf{\Omega}, \mathbf{O}, \mathbf{H} \rangle$ where

- $\mathbf{I} = \{0,\ldots,n\}$ is the set of agent indices.

- S is a finite set of world states with a unique initial state s_0

- $\mathbf{A} = \{\mathbf{A_1} \times \mathbf{A_2} \times \ldots \times \mathbf{A_n}\}$ is a finite set of joint actions, where A_i is the set of possible actions for agent i.

- $\mathbf{P}: \mathbf{S} \times \mathbf{A} \rightarrow [\mathbf{0}, \mathbf{1}]$ is the transition probability function, where $P(s' \mid a, s)$ is the probability of moving to a new joint state s' when performing joint action a in joint state s.

- $\mathbf{R}: \mathbf{S} \times \mathbf{A} \rightarrow \mathfrak{R}$ is the reward function and $R(a, s, s')$ represents the expected immediate reward of choosing joint action a in joint state s and transitioning to joint state s'.

- $\mathbf{\Omega} = \{\mathbf{\Omega_1} \times \mathbf{\Omega_2} \times \ldots \times \mathbf{\Omega_n}\}$ is a finite set of joint observations, where Ω_i is the set of observations for agent i.

- $\mathbf{O}: \mathbf{S} \times \mathbf{A} \times \mathbf{\Omega} \rightarrow [\mathbf{0}, \mathbf{1}]$ is the observation function, where $O(a, s, s', o)$ is the probability of all agents (from 1 to n) observing o when executing joint a at state s and transitioning to state s'.

- $\mathbf{H}$ is a positive integer that represents the horizon.

Since DEC-POMDP is a very general model in that it does not limit the interactions between agents in the model, it becomes very difficult and impractical to be solved and suffers from a very high computational complexity and a serious limitation in which, even with the total joint partial observations of all agents, the world state cannot be observed with certainty (Bernstein, Zilberstein & Immerman, 2000).

- **Networked Distributed POMDP**

An effective method to deal with the complexity problem of DEC-POMDP is to identify a specialization of the general model in a way that makes it easier to be solved. This can be carried out by making assumptions about the number of agents each agent interacts with. In real life situations, where there are large-scale domains, agents are organized into a network where an agent does not interact with all agents but interacts with its directly connected neighbors of agents (Abdallah & Lesser, 2007). This property is called *Locality of Interactions*. Networked Distributed POMDP (ND-POMDP) (Nair *et al.*, 2005) is a specialized model of DEC-POMDP that, besides being inspired by domains such as distributed sensor networks, combines the concept of modeling domains of MARL problems under uncertainty in DEC-POMDP with the concept of exploiting locality of interactions in Distributed Constraint Optimization (DCOP) (Modi *et al.*, 2003) to solve the complexity problem in DEC-POMDPs.

Definition 3. An ND-POMDP is defined for a group of n-agents with the tuple$\langle \mathbf{I}, \mathbf{S}, \mathbf{A}, \boldsymbol{\Omega}, \mathbf{P}, \mathbf{O}, \mathbf{R}, \mathbf{H}, \mathbf{b} \rangle$ where

- **I** is the set of agent indices.
- $S = S_1 \times S_2 \times \dots \times S_n \times S_u$ is a finite set of world states (where S_i is the local state of agent i and S_u is the uncontrollable state which is independent of agents' actions).
- $\mathbf{A} = \{A_1 \times A_2 \times \dots \times A_n\}$ is a finite set of joint actions, where A_i is the set of possible actions for agent i.
- $\boldsymbol{\Omega} = \{\Omega_1 \times \Omega_2 \times \dots \times \Omega_n\}$ is a finite set of joint observations where Ω_i is the set of observations for agent i.
- $\mathbf{P}: S \times A \rightarrow [0,1]$ is the transition probability function, where $P(s' \mid a, s) = P(s'_u \mid s_u) \cdot \prod_{i \in I} P_i(s'_i \mid a_i, s_i, s_u)$ is the probability of moving to a new joint state $s' = \langle s'_u, s'_1, \dots, s'_n \rangle$ when performing joint action $a = \langle a_1, \dots, a_n \rangle$ in joint state $s = \langle s_u, s_1, \dots, s_n \rangle$.

- $\mathbf{R}: S \times A \rightarrow \Re$ is the reward function which is decomposable among sub-groups of agents, $R(a, s) = \sum_l R_l(a_l, s_l, s_u)$ represents the total expected reward of agents in sub-group l when choosing joint action a in joint state s and transitioning to joint state s'.
- $\mathbf{O}: S \times A \times \Omega \rightarrow [0,1]$ is the observation function in which $O(o|s, a) = \prod_{i \in I} O(o_i|s_i, s_u, a_i)$ where s is resulting state after performing action a and receiving joint observation o.
- **H** is a positive integer that represents the horizon and $\mathbf{b} = \langle b_u, b_1, \dots, b_n \rangle$ is the initial belief for joint state $s = \langle s_u, s_1, \dots, s_n \rangle$.

ND-POMDPs assume the independency of observations and state transitions (i.e. the observations and state transitions of an agent depend only on that agent actions and are independent of other agents' actions). The only dependent component in ND-POMDPs is the joint reward function is defined as the sum of local rewards of different sub-groups of agents $R(a, s) = \sum_{l \in E} R_l(a_l, s_l, s_u)$ where l is a sub-group of agents, s_l represents the state of the sub-group l and a_l defines the joint actions of agents in the sub-group l. The reward of an agent belongs to sub-group l depends only on the actions of agents in the same sub-group. An interaction hypergraph G = (I, E) is constructed based on the reward function where I represents the vertices (agents) and E represents a set of hyperlinks that connect agents in the same sub-group together. $l \in E$ is a hyperlink that connects agents which form the reward function R_l together.

All of the above models are designed for the case of having systems with cooperative agents where the goal is to find the best set of policies (one for each agent) which maximizes the total reward of all agents. However, for systems where self-interest agents exist in which each agent aims to maximize its own reward, very similar models have been proposed and applied to solve the problems of MARL with such agents. Markov Games (Littman, 1994), also called Stochastic Games, and Partially Observable Stochastic Games (POSGs) (Hansen, 2004) are two models that are defined exactly like MDPs and DEC-POMDPs respectively with the only difference in the reward function in which each agent has a reward function instead of one reward function for the whole set of agents (Mostafa, 2011).

Since the main contribution of this thesis is to study and analyze the performance of the coordinated Q-learning algorithm in cooperative/semi-cooperative two-player two-action games under different network structures, and games involve having self-interest agents, ND-POMDPs model is used to formalize our problem domain and is emerged with game theory concept to model the interactions between self-interest agents in this domain. Game The next section defines the concept of game theory, its problem definition and main types of games, and several examples of games from each type.

2.2 Game Theory

Game Theory is considered to be a very powerful framework that studies and represents the interactions between two or more learning agents when playing a game. An agent playing a game cares only about maximizing its expected reward which depends on the actions of other agents it is interacted with. Each agent has a strategy which is defined as a plan that tells the agent which action to choose at every possible situation faced when playing the game.

An agent's strategy can be either pure or mixed. Pure strategy states that the agent will choose this strategy with probability 1, while Mixed strategy states that the agent will choose this strategy with probability value that is less than 1 and larger than 0. If there are n agents playing a game where each agent has m actions to choose from at each game stage, then the possible combination of joint actions is m^n. The main object of game theory is the *Stage Game*. Stage Game is defined by a set of agents {1,2,…, n} where for each agent i there is a finite set of actions $\mathbf{A_i}$ and a reward function $\mathbf{R_i}: A_1 \times A_2 \times ... \times A_n \rightarrow \Re$ that depends on all agents' actions.

2.2.1 Game Representations

There are two representations of games in Game Theory, extensive form representation and normal form representation. Note that we use the terms agents and players interchangeably.

- **Extensive Form Representation**

Extensive form represents sequential games where agents do not play the game at the same time (i.e. play one after the other) and later agents know the actions of previously played agents (Shoham & Leyton-Brown, 2009). Games represented using this model are viewed as a decision tree, where nodes represents the agent's choice of actions and the player index (or name) is placed on top of the each node that represents its actions choice. The last numbers are defined as the payoffs (i.e. rewards) of the players. Figure 2.3 illustrates an example of a game, where two players, each with possibility of selecting from two actions, modeled using extensive form.

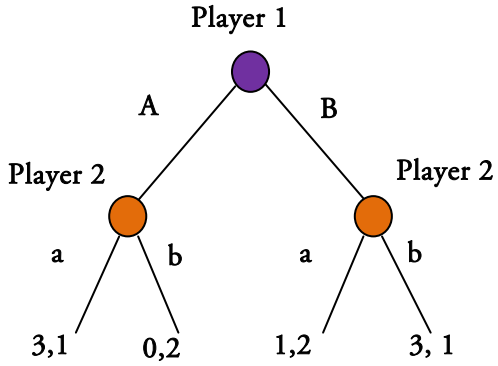

Figure 2.3: An illustration of an extensive form game

In the above example, player1 starts playing by choosing either of action "A" or Action "B". Then player2 observes the action played by player1 and depending on this action it selects either of action "a" or action "b". If player1 selects action "A" then player2 selects action "a", then player1 will receive a reward value of 3 and player2 will receive a reward value of 1, but if player2 selects action "b" then player1 will receive a reward value of 0 and player2 will receive a reward of 2. Same scenario works when player1 selects action "B" and player2 selects either of "a" or "b".

- **Normal Form Representation**

Unlike extensive form, that represents the game as a tree, normal form represents a game, where agents play simultaneously without knowing the action choice of each other or play consecutively but no agent knows the action(s) of previously played agents (Shoham & Leyton-Brown, 2009), as

a payoff matrix in which the players, their strategies and the reward value of each possible combination of actions (i.e. joint actions) are included. Table 2.1 illustrates an example of a game where two players, each with possibility of choosing one out of two actions, represented using normal form model.

Player 1/ Player 2	Left	Right
Up	(1,2)	(0,0)
Down	(0,0)	(4,3)

Table 2.1: An illustration of the payoff matrix of a normal form game

In the above example, player1 can choose either to move Up or move Down and player2 can choose wither to move to the Left or to move to the Right. The first number in every cell in the payoff matrix represents the reward value of the row player (i.e. player1) and the second number in every cell represents the reward value of the column player (i.e. player2). In this example if player1 chooses to move Up and player2 chooses to move to the left then player1 receives a reward value of 1 and player2 receives a reward value of 2, but if player1 chooses to move Down and player2 chooses to move to the Left, then both players will get a reward value of 0 (i.e. no reward).

Extensive form games can be transformed into normal form games but usually this conversion suffers from a high computational complexity in the representation (Leyton-Brown & Shoham, 2008). The work in this thesis focuses on using two-player two-action normal form repeated games where two agents play simultaneously with the goal of finding the best action to maximize their reward function value, with each agent has two actions to choose from.

A Repeated Game is defined as the repetition of the stage game for a number of times in which the current action of each agent affects the set of future possible actions of other agents (Conitzer & Sandholm, 2007). The games used in this thesis are the Coordination game and the iterated prisoner's Dilemma game which are discussed in the next section.

2.2.2 Types of Games

In game theory, games can be classified into three types: Cooperative, Competitive and Semi-Cooperative or Semi-Competitive Games.

- **Cooperative Games**

Agents playing cooperative games act like one group with a common goal, for each agent in the group, which is maximizing the expected payoff of the whole group by coordinating agents' strategies (Hoen *et al.*, 2006). Agents in this type of games prefer the group-interest over self-interest. A good example to illustrates cooperative games is the Battle of the Sexes game, a two-player two-action game where a couple prefer to spend the weekend together than spending it alone in one of two places: the cinema or a football match. However, the wife prefers to go to the cinema while the husband prefers to go to the football match. Table 2.2 demonstrates the payoff matrix of the Battle of the Sexes game where the wife is the row player (player1) and the husband is the column player (player2).

Player 1/ Player 2	Cinema	Football Match
Cinema	(3,2)	(0,0)
Football Match	(0,0)	(2,3)

Table 2.2: An illustration of the payoff matrix of the Battle of the Sexes game

Although each player gets a better payoff by choosing the opposite action of the other (i.e. player1 gets better payoff when choosing Cinema and player 2 gets a better payoff by choosing Football Match), they still prefer to choose the same action together than choosing their preferred actions. There are two pure Nash-Equilibrium strategies in this game: both players choose to go to the Cinema and both players choose to go to the Football Match.

Another good example of cooperative games is the Coordination game where 2 players must choose the same action in order to get the best joint reward. Table 2.3 illustrates the payoff matrix of a coordination game where two robots are trying to pass each other in a narrow pathway. If both robots choose the same action, then they will manage to pass each other successfully. But if each agent chooses a different action than the other agent, both agents will bump into each other (i.e. from each agent's perspective, if both agent chooses to move to their right hand side or both choose to move to their left hand side, then they will pass each other. Otherwise, a collision will occur.). Therefore, both agents must coordinate their actions in order to maximize their total payoff. Usually, this type of games requires communication between agents in order to coordinate their actions more efficiently. In this thesis, a similar Coordination game to this example is used and discussed in detail in the next Chapter.

Player 1/ Player 2	Left	Right
Left	(5,5)	(0,0)
Right	(0,0)	(5,5)

Table 2.3: An illustration of the payoff matrix of an example of a Coordination Game

- **Competitive or Non-Cooperative Games**

Agents play this type of games as individuals, each aims to maximize its own expected payoff. In competitive games, agents prefer self-interest to group-interest and each agents chooses the action that maximizes its reward regardless of how this action may affect the agent's partner (Hoen *et al.*, 2006). A good example of competitive games is the matching pennies game, a two-player two-action game, where each agent can choose either to play Head or Tail. In this game, if both agents play the same action (i.e. either both agents choose Head or both agents choose Tail) then the row player receives a reward of 1 and the column player receives a negative reward (i.e. punishment) of 1 (i.e. -1). But if both players choose different actions (i.e. one agent chooses Head and the other chooses Tail) then the row player receives a punishment of 1 and the column player receives a reward of 1. Table 2.4 illustrates the payoff matrix of the matching pennies game.

Player 1/ Player 2	Head	Tail
Head	(1,-1)	(-1,1)
Tail	(-1,1)	(1,-1)

Table 2.4: An illustration of the payoff matrix of the Matching Pennies game

In the above example, each agent chooses its action without taking into consideration how this action affects its partner's reward. There is no pure Nash-Equilibrium strategy in this game in which agents choose actions with probability 1 to maximize their own reward. Instead, there is a mixed Nash-Equilibrium strategy in which agents choose any of their possible actions with the same probability value (i.e. both agents choose any action with probability $\frac{1}{2}$) (Abdallah & Lesser, 2006). Therefore, the mixed Nash-Equilibrium of this game is (0.5, 0.5). It is clearly observed that the value of the gain of an agent is exactly the value of the loss of another agent. Games with such property are called Zero-Sum Games in which the sum of total rewards of each agent playing such games is equal to zero (Shoham & Leyton-Brown, 2009).

- **Semi-Cooperative or Semi-Competitive Games**

In some cases, agents in competitive games exhibit a cooperative behavior in order to maximize their own payoff. Competitive games with cooperative behavior and cooperative games with competitive behavior are called Semi-Cooperative or Semi-Competitive games. A very good example of this type is the prisoner's dilemma game. As observed in table 2.5 which illustrates the prisoner's dilemma payoff matrix, it is clearly shown that by comparing the possible combinations of agents' actions, a player receives a better reward when choosing to "Defect", regardless of the action of the other player, than to choose "Cooperate" which is a more risky action in which if and only if both players choose to "Cooperate" they will get the best joint possible reward, but if only one player chooses to "Cooperate" then it will receive no reward and the other player with "Defect" as an action will receive the highest reward a single player can get.

Player 1/ Player 2	Defect	Cooperate
Defect	(1,1)	(5,0)
Cooperate	(0,5)	(3,3)

Table 2.5: An illustration of the payoff matrix of the prisoner's dilemma game

This comparison demonstrates that the action "Cooperate" is dominated by the action "Defect" and should not be used by agents when playing the prisoner's dilemma game and that the joint action (Defect, Defect) is the only *Nash Equilibrium* of this game (Osborne & Rubinstein, 1994; de Groot, 2008). However, although (Defect, Defect) is the only Nash Equilibrium of this game, it is clearly observed that it is not the optimal joint action that maximizes the reward value of agents in this game. On the contrary, the joint action (Cooperate, Cooperate) is the optimal joint action of this game as it maximizes the reward value of each player.

In this game, it is notable that both players suffer from a low reward value if they choose different actions from each other. A solution to this problem is to coordinate the agents' actions in a way that will avoid confliction in their action choice. Therefore, agents will have 2 possible strategies to play, which are (Cooperate, Cooperate) and (Defect, Defect). In this thesis, we are interested in studying the performance of a learning algorithm in cooperative and semi-cooperative (or semi-competitive) games. The next sub-section defines the best response and Nash-Equilibrium.

2.2.3 Nash-Equilibrium

Given the strategies of other agents, a strategy that selects the action which results in the highest payoff of an agent is called the *best response* of that agent. Nash-Equilibrium is the joint strategy where each agent plays a best response to the strategies of its opponent(s) (Babes, Wunder & Littman 2009). According to John Nash (1950), each game has at least one Nash-Equilibrium. A competitive game is considered to be solved once the Nash-Equilibrium is found since no agent can get better payoff by changing its strategy unilaterally. However, in cooperative games there is usually more than one Nash-Equilibrium. In this case, and when agents do not communicate with

each other, it is unclear which Nash-Equilibrium strategy is chosen by each agent. Nash-Equilibrium can be classified into two types: Pure Nash-Equilibrium, where all agents play pure strategies, and Mixed Nash-Equilibrium, where there is at least one agent playing a mixed strategy (de Groot, 2008).

Multi-Agent learning algorithms are used to help agents, when playing a game, to find their actions which maximize their payoff (i.e. helps solving the game by reaching Nash-Equilibrium). The next section defines MARL algorithms and provides several examples of their types.

2.3 Multi-Agent Learning Algorithms

Several MARL algorithms have been proposed and applied to many multi-agent domains (including games) to help agents learn their optimal action that will maximize their expected reward. In this section, MARL algorithms are classified into a number of families to simplify the discussion and comparison between each of them and the Q-learning algorithm is defined and discussed in detail.

2.3.1 MARL Algorithms Classification

In this thesis, MARL algorithms are classified in to three families: Q-learners, Equilibrium learners and Gradient Ascent learners.

- Q-learners

Algorithms of this class are designed based on the Q-learning algorithm which is the focus in this thesis and is discussed later in this section. Therefore, they can only learn deterministic policies. This property makes this class of MARL algorithms suffer from limitations when applied to competitive domains, where stochastic policies exist. Also, algorithms of this class require observing the actions of other agents which is impractical in large domains with partial observability. Independent Learners (ILs) and Joint Action Learners (JALs) are good example of algorithms of this class (Claus & Boutilier, 1998).

- **Equilibrium Learners**

Unlike previous class, algorithms of this class can learn stochastic policies. Examples of algorithms of this class are Nash-Q (Hu & Wellman, 2003), which assumes that both actions and rewards of other agents are observable, and AWESOME (Conitzer & Sandholm, 2007) that besides requiring each agent to know the actions of other agents assumes that the underlying game structure is known. After observing the actions of agents, each agent tries to compute the Nash-Equilibrium. These algorithms converge to Nash-Equilibrium in self-play (i.e. when all agents play using the same learning algorithm).

- **Gradient Ascent Learners**

Learning algorithms of this class learns a stochastic policy though following the gradient of the expected reward. The expected reward gradient is defined as a vector that points towards the highest value of the increase of the expected reward. Examples of algorithms of this class are: Infinitesimal Gradient Ascent (IGA) algorithm (Singh, Kearns & Mansour, 2000), Generalized IGA (GIGA) algorithm (Zinkevich, 2003), IGA-WoLF Algorithm (Bowling & Veloso, 2002), GIGA-WoLF algorithm (Bowling, 2005) and Weighted Policy Learner (WPL) algorithm (Abdallah & Lesser, 2008). While IGA and GIGA only converge to pure Nash-Equilibrium, GIGA-WoLF and WPL converge to both pure and mixed Nash-Equilibrium with WPL outperforms GIGA-WoLF in large-scale partially observable domains (Abdallah & Lesser, 2008).

2.3.2 Q-Learning Algorithm

Q-learning (Watkins & Dayan, 1992) is originally defined as a single-agent learning algorithms which helps agents to accomplish their goal, that is finding an optimal policy that maximizes their reward, in MDPs. However, Q-learning is proved to be working in Multi-agent settings too. In this algorithm, each agent has a Q-table where Q-values of each state action pair are stored. For each agent, each Q-value or state action value represents the expected payoff that the agent receives when choosing that action at that state. Let us consider a two-player two-action game where Q-learning

algorithm is used to find the optimal actions of each agent. The size of the Q-table of each agent is equal to the number of possible action combination for that agent. In case of two-player two-action games, the Q-table of each agent has $2^2 = 4$ Q-values. In repeated games, at the end of each stage the Q-value of the executed joint action is updated for each agent using the following Q-function:

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a')] \qquad (1)$$

Where s is the current state, a is the action chosen by the agent at state s and α is the learning rate whose value ranges from 0 to 1 in which as the value of α increases, the more important the learned information is. γ is the discount factor whose value ranges from 0 to 1 and as the value increases, the agent ignores the immediate high reward and focuses on the long-term high reward. If $\gamma \geq 1$ then there is a high possibility of a divergence in the learning algorithm performance. r is immediate reward received by the agent once choosing action a at state s. s' is the new state resulted after executing action a at state s and $\max_{a'} Q(s', a')$ is the maximum future expected reward value where a' is the best action that maximizes the expected reward value of the next state s'.

Let us consider the example of two-player two-action game where Q-learning algorithm is used, the learning process is carried out by each agent as follows

1) Initialize the Q-table randomly (e.g. (0, 0, 0, 0))
2) Let s represents the current state
3) Choose an action for the current state using an exploration method. (e.g. ε–greedy exploration):
   a ← $\text{argmax}_a Q(s,a)$ with probability (1- ε) and
   a ← random action with probability ε
4) Execute the action a
5) observe reward r and next state s'
6) Learn and update the Q-value of the executed action using rule (1)
7) Repeat 2-7

As it can be noted from the above process, an exploration method is used to explore possible actions to execute in the next state s'. In this thesis we use the ε–greedy exploration algorithm, where ε represents the exploration rate which is usually set to 0.1. Using this exploration algorithm, the agent will explore more than one action to execute and, for each new state, it will choose to execute the best action with the highest reward value 90% of the time and a random action 10% of the time. It is worthy to mention that Q-learning algorithm can only learn a deterministic policy which chooses an action at a state with a probability of 1 (i.e. Always chooses a particular action at a particular state).

2.4 Coordinated Multi-Agent Reinforcement Learning

2.4.1 Problem Definition

Cooperative Multi-Agent Systems are considered to be one of the most interesting perspectives in MARL field that received a lot of attention recently from many Artificial Intelligence (AI) researchers due to the fact that most real-world multi-agent applications behave as cooperative systems where all agents share common goal (Panait & Luke, 2005). To achieve the goal of such systems, several agents have to coordinate their actions in order to compute their optimal joint action which maximizes their global total reward. Several approaches have been proposed and applied to help coordinating agents' actions (e.g. MDP and DEC-POMDP). However, such approaches suffer from a very high communication and computation complexity, when trying to compute the exact optimal joint policy of agents, due to the large action space that scales exponentially with the number of agents.

Coordinating the actions of agents in cooperative multi-agent systems has been a common concept used by many researchers to reduce the computational complexity of computing the optimal solution (Weib, 1993; Boutilier, 1999). Using this concept, several coordination approaches are developed to allow agents to communicate with each other during execution time. Therefore, it solves the limited observability problem faced by agents in large-scale domains. The following works develop different coordination approaches using the coordination concept and apply these

approaches in several real-life domains and verify the effectiveness of their usage and show that they outperforms other multi-agent reinforcement learning approaches which do not use the coordination concept.

2.4.2 Related Work

Guestrin, Lagoudakis and Parr (2002) proposed a new approach, the Coordinated Reinforcement Learning, which computes an approximation of the joint policy as a linear combination of local policies through using a coordination graph as a message passing scheme that helps agents in choosing the optimal joint policy. Experimental evaluation shows that the proposed approach achieves policies of high quality and greatly reduces the computation and communication complexity unlike other reinforcement learning approaches.

Yagan and Tham (2007) have used the concept of Coordinated Reinforcement learning to propose a novel online model-free reinforcement learning approach that computes the approximation of the optimal joint policy to reduce time complexity, communication and computation resources in ND-POMDPs. In addition, this work proposes a distributed coordination technique which is based on exploiting interactions among neighboring agents to optimize the global system performance. Experimental results verify the effectiveness of the proposed approach and show that unlike other approaches, the proposed approach saves computational and communication time and cost.

Another work (Stranders et al., 2009) uses the concept of coordinated reinforcement learning approach to propose an on-line decentralized coordination approach for addressing the limitations presented by previous off-line algorithms proposed to find the optimal joint action in Disaster Response domains. The proposed approach uses the Max-Sum algorithm, a DCOP algorithm, as a negotiation mechanism among agents so that an optimal joint action can be achieved. In Addition, it uses a powerful Bayesian tool, Gaussian Processes, to model the spatial phenomena dynamics. Furthermore, the paper proposes two pruning techniques to speed up the Max-Sum algorithm and experimental evaluation verifies the effectiveness of the proposed algorithm by illustrating a reduction up to 50% in the root mean squared error compared to other approaches, and the

efficiency of the pruning techniques, which saved computational cost by pruning up to 95% of the joint actions in Max-Sum algorithm, resulting in speeding up the algorithm.

Zhang, Abdallah and Lesser (2010) proposed a decentralized self-organization approach that forms a hierarchically organizational control which evolves dynamically during the learning process. This proposed approach is based on the "nearly decomposable systems" concept, which states that interactions between sub-systems are weaker than interactions within sub-systems. Furthermore, this work proposes a new type of interactions, i.e. joint-event-driven interactions, and a measure of identifying the strength of these interactions. Based on these interactions, the proposed approach can dynamically form the supervisory organization by grouping strongly interacting agents together in one cluster/group. Experimental Evaluations show that the proposed approach is efficient and outperforms the pre-defined, fixed frameworks.

2.5 Summary

In this chapter, a comprehensive review about Multi-Agent Reinforcement Learning (its general representative models and algorithms), Game Theory (its problem definition, representations, types and several examples of each type) and Coordinated Multi-Agent Reinforcement Learning, which is the main focus in this thesis, have been provided. The main contribution in this thesis is to analyze and study the performance of a coordinated multi-agent reinforcement learning approach in cooperative and semi-cooperative two-player two-action games. This thesis adopts the coordination approach of (Zhang & Lesser, 2011) and applies it in the Coordination and Iterated Prisoner's Dilemma games. The adopted coordination approach is presented and discussed in more detail in the first section of the following Chapter (i.e. Section 3.1).

Chapter 3
Methodology

This chapter provides a detailed description of the adopted coordination approach, the coordinated Q-learning approach, which is proposed in one of the recent paper about coordinated multi-agent reinforcement learning (Zhang & Lesser, 2011) by presenting the main assumptions of the approach, how the learning process is carried out, how the global joint policy is computed and how we adjust this approach to work in different domain than the one it is designed for. It also defines several key elements that are important to the main contribution of the thesis, which is studying and analyzing the characteristics and performance of the coordinated Q-learning approach when applied in cooperative/semi-cooperative two-player two-action games under different network structures then compare the performance with that of the original Q-learning algorithm.

3.1 Coordinated Q-Learning Approach

Unlike most of the proposed coordination approaches (presented in the previous chapter) that need accurate, expensive to be obtained, models of the environment they are applied to and do not scale to large domains, Zhang and Lesser (2011) presented an online model-free coordinated multi-agent learning approach, the coordinated Q-learning approach, to solve ND-POMDPs more efficiently by utilizing both Multi-Agent Reinforcement Learning (MARL) and Distributed Constraint Optimization (DCOP).

This approach aims to maximize an approximation of the global expected reward function through optimizing a decomposable reward function that helps in distributing the learning of the optimal global joint policy among agents by exploiting their locality of interactions. This approximation is exact for ND-POMDPs with a special property called "*Groupwise Observability*". Then, using a DCOP technique (i.e. the Max-Sum algorithm), the optimal (or near-optimal) joint action which maximizes the global reward function is computed. This technique coordinates the distributed

learning to ensure the optimality (for ND-POMDPs with groupwise observability property) or near-optimality of the global learning performance. Experimental Results, carried out in the original work of the approach (Zhang & Lesser, 2011), verify the effectiveness of the proposed coordinated learning approach in solving ND-POMDP problems and show that the proposed approach significantly outperforms offline nearly-optimal with no-communication policies of other approaches. Therefore, the coordinated Q-learning approach is adopted in this thesis as the coordinated reinforcement learning approach to be studied and analyzed in two-player two-action cooperative/semi-cooperative games and compared later to the original learning algorithm without the coordination property. The following sub-sections provide a detailed description of the coordinated Q-learning approach, the coordinated learning process and how to compute the global optimal joint action.

3.1.1 The Approach Framework and Assumptions

A Q-function $\mathbf{Q}(\vec{h}, a)$ is used to represent the expected reward when executing joint action *a* at history of observations $\vec{h}$. The adopted coordination approach aims to optimize an approximation of the global reward function $\mathbf{Q}(\vec{h}, a)$. This approximation $\widehat{\mathbf{Q}}(\vec{h}, a)$ is defined as a decomposable reward function that is expressed as the sum of local reward functions of group(s) on hyperlinks in the interaction hypergraph constructed from the global reward function in ND-POMDPs (see equation 1):

$$\widehat{\mathbf{Q}}(\vec{h}^t, a^t) = \sum_{l \in E} \mathbf{Q}_l(\vec{h}_l^t, a_l^t) \qquad (2)$$

Where $\mathbf{Q}_l(\vec{h}_l^t, a_l^t)$ represents the expected reward for agents in group(s) defined on hyperlink l when executing joint action a_l^t at joint observation history $\vec{h}_l^t$. Optimizing the approximated reward function $\widehat{\mathbf{Q}}(\vec{h}, a)$ can be considered as maximizing the global reward function $\mathbf{Q}(\vec{h}, a)$ since the latter depends on the former to be computed in ND-POMDPs. However, this approximation turns to be exact when ND-POMDPs have a special property called "*Groupwise Observability*". This property indicates that, the local belief of agent i in group l can be stated

depending only on the initial local state of agent i and the joint observations and actions of other agents in group l (i.e. the group of agent$i$) and that the local belief of agent i is totally independent of the history of joint observations and actions of agents in other groups. The adopted approach is verified theoretically to learn the global optimal policy for ND-POMDPs with groupwise observability. Nevertheless, experimental results verify that, even when the groupwise observability property does not exist in the examined ND-POMDPs, the coordination approach learns a policy that optimizes the global reward function.

The adopted coordination approach assumes that agents are distributed among a predefined number of groups in which each group has a delegate agent that learns the group optimal policy on behalf of the group members. In this framework, each delegate agent has a local Q-function of the group it is in charge of. Figure 3.1 illustrates an example of the adopted approach framework when distributing 8 agents among 3 groups. While red nodes represent delegate agents, blue nodes represent normal agents (i.e. agents in the 1st level of the supervisory framework) and edges represent membership of different groups (where each delegate agent represents a group). Note that the number of delegate agents is equal to the number of learning groups and that an agent can belong to more than one group.

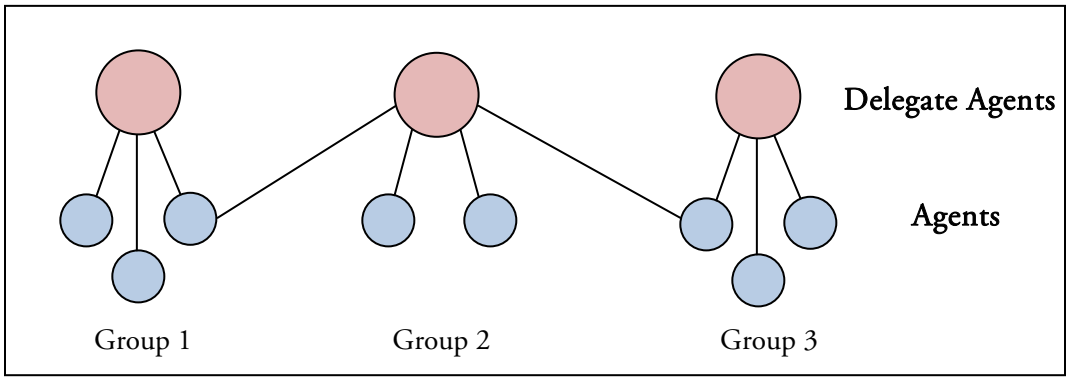

Figure 3.1: An example of the adopted coordination approach supervisory framework when having three groups of agents

3.1.2 The Coordinated Q-learning Process

The coordinated Q-learning approach learns the optimal $\widehat{\mathbf{Q}}(\vec{h}, a)$ that maximizes the global reward and distributes this learning among groups through updating the Q-function of each delegate agent using the following update rule:

$$\mathbf{Q}_l(\vec{h}_l^t, a_l^t) = (1-\alpha)\mathbf{Q}_l(\vec{h}_l^t, a_l^t) + \alpha[r_l^t + \gamma \mathbf{Q}_l(\vec{h}_l^{t+1}, a_l^*)] \qquad (3)$$

Where t is a learning cycle and α is the learning rate whose value ranges from 0 to 1 in which as the value of α increases, the more important the learned information is. γ is the discount factor whose value ranges from 0 to 1 and as the value increases, the agent ignores the immediate high reward and looks for the long-term high reward. If $\gamma \geq 1$ then there is a high possibility of a divergence in the learning algorithm performance. r_l^t is the total reward of group l that is defined as the sum of local rewards of each member in the group l. $\mathbf{Q}_l(\vec{h}_l^{t+1}, a_l^*)$ is the expected reward for agents in group l when executing the joint optimal action a_l^* at the updated observations history $\vec{h}_l^{t+1}$.

The following is a description of the learning process of the coordinated Q-learning approach. In each learning cycle t, each agent in group l executes its action a^t and receives a reward r^t of executing that action. Agents send their observations to their delegate agent whose action and observation history are the set of joint actions a_l^t and joint observations history $\vec{h}_l^t$ of agents in group l and its reward is the group total reward r_l^t (i.e. the sum of local rewards of all agents in group l). Then, using a DCOP algorithm, the group optimal action a_l^* for the updated history of observations $\vec{h}_l^{t+1}$ is computed. The delegate then updates its Q-function $\mathbf{Q}_l(\vec{h}_l^t, a_l^t)$ using rule (3) and distributes the next actions (that can be either the best actions a_l^* or exploration actions) to each agent in the group.

3.1.3 Choosing the Optimal Joint Action

- **Problem Definition**

The optimal joint action a^* maximizes the global reward function $\mathbf{Q}(\vec{h}, a)$, that is the sum of all local reward functions $\mathbf{Q}_l(\vec{h}_l, a_l)$ in which:

$$a^* = argmax_a \sum_{l \in E} \mathbf{Q}_l(\vec{h}_l, a_l) \qquad (4)$$

Where a^* is used for updating local reward functions $\mathbf{Q}_l(\vec{h}_l, a_l)$ and as a possible action to be executed by agents during execution time. The adopted approach represents the problem of computing the optimal joint action as a Distributed Constraint Optimization Problem (Nair *et al.*, 2005). DCOP is a multi-agent paradigm where agents try to find the optimal joint action that maximizes the total reward obtained by them. It is expressed as a set of variables $x = \{x_1, \dots, x_n\}$, where x_i represents a possible action for agent i, and a set of functions $\mathrm{Q} = \{\mathrm{Q}_l | l \in E\}$, where Q_l is the reward function for group(s) defined on hyperlink l.

- **Max-Sum Message Passing Algorithm**

DCOP is solved through applying message-passing algorithms which help agents to communicate with each other during execution time, enabling them to get better observability of the world state and other agents' actions. Despite the fact that a lot of message-passing algorithms have been proposed to solve DCOP, most of these algorithms either suffer from high computational complexity even though it computes/finds the optimal solution or compute an approximate solution with less complexity. Generalized Distributive Law (Aji & McEliece, 2000) contains a class of algorithms which require much less communication and computation complexity and compute a very good approximation of the optimal solution. The Adopted coordination approach uses the Max-Sum message passing algorithm (Stranders *et al.*, 2009) which is considered to be one of the pretty good algorithms in that class to compute the optimal joint action and can be directly used to coordinate interactions of more than two agents.

The Max-Sum algorithm operates on a factor graph which is an undirected bipartite graph, in which there is a node created for each variable x and for each function Q in the DCOP. Each variable node is connected to a function node if and only if the corresponding function depends on the corresponding variable. Figure 3.2 illustrates an example of a factor graph where 7 agents (i.e. variable nodes) are distributed among 3 groups in which each delegate agent has the Q function of the group it is in charge of (i.e. function nodes).

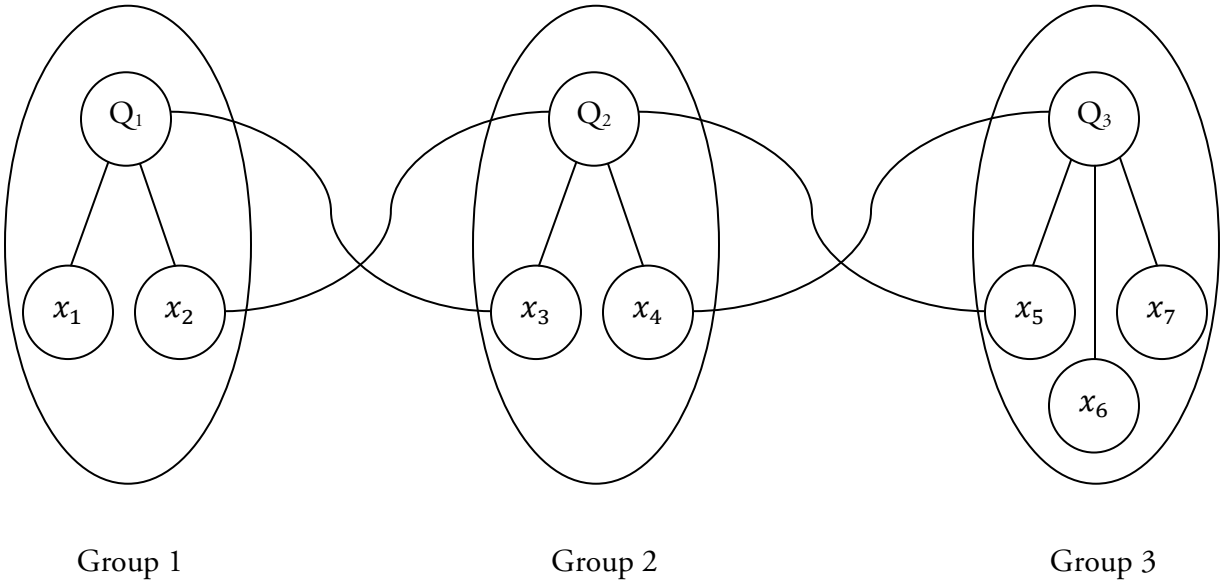

Figure 3.2: An example of a factor graph with 7 agents and 3 groups

In the above figure, the reward function of group 1 (Q_1) depends on the action of each of agent1, agent2 and agent3 (x_1 , x_2 and x_3 respectively). The reward function of group 2 (Q_2) depends on the action of each of agent2, agent3, agent4 and agent5 (x_2, x_3, x_4 and x_5 respectively). Finally, the reward function of group 3 (Q_3) depends on the action of each of agent4, agent5, agent6 and agent7 (x_4, x_5, x_6 and x_7 respectively). The Max-Sum algorithm provides a way for agents to communicate in the factor graph with each other during execution time by identifying the messages passed from variable nodes to function nodes and the messages passed from function nodes to variable nodes. The following are the rules for computing the mentioned messages.

- **Message from variable node i to function node l:**

$$q_{i\to l}(x_i) = \sum_{g\in F_i\setminus l} r_{g\to i}(x_i) + c_{il} \qquad (5)$$

Where F_i is a vector of function indices that specifies which function nodes are connected to the variable node i and c_{il} is a normalizing constant which prevent messages from increasing continuously in cyclic factor graphs. Using the above rule, each agent who belongs to more than one group sends to each of its delegate agents the sum of messages it received from other delegate agents it is connected to when performing action x. However, $q_{i\to l}(x_i)$ = zero if agent i does not belong to more than one group.

- **Message from function node l to variable node i:**

$$r_{l\to i}(x_i) = \max_{a_l\setminus x_i} [\mathrm{Q}_l(\vec{h}, a_l) + \sum_{g\in V_l\setminus i} q_{g\to l}(x_g)] \qquad (6)$$

Where V_l is a vector of variable indices that specifies which variable nodes are connected to the function node l and $a_l\setminus x_i = \{x_g : g \in V_l\setminus i\}$. Using the above rule, the delegate agent of group l computes the reward function of the group when the group members/agents execute joint action a_l at the joint history of observations $\vec{h}$. Then, for agent i in group l, the delegate agent sends a message to agent i which contains the value of the group Q-function when agent i chooses action x added to the sum of the messages sent to the delegate agent from other agents in the same group l (except agent i) when they execute the same action x as agent i.

The Max-Sum algorithm computes the optimal action a_i^* of each agent i using rule (7) only when the factor graph is acyclic. Otherwise, the algorithm is not guaranteed to compute the optimal action of each agent although previous experimental analyses verify that the max-sum algorithm provides pretty good solution quality even when the factor graph is cyclic.

$$a_i^* = argmax_{a_i} \sum_{g\in F_i} r_{g\to i}(x_i) \qquad (7)$$

In our work we ensure the optimality of the joint action computed using the Max-Sum algorithm by ensuring that all factor graphs induced from DCOP are acyclic. This feature is ensured using a grouping technique which distributes agents over a number of groups. Figure 3.3 provides an acyclic version of figure 3.2 using our grouping technique.

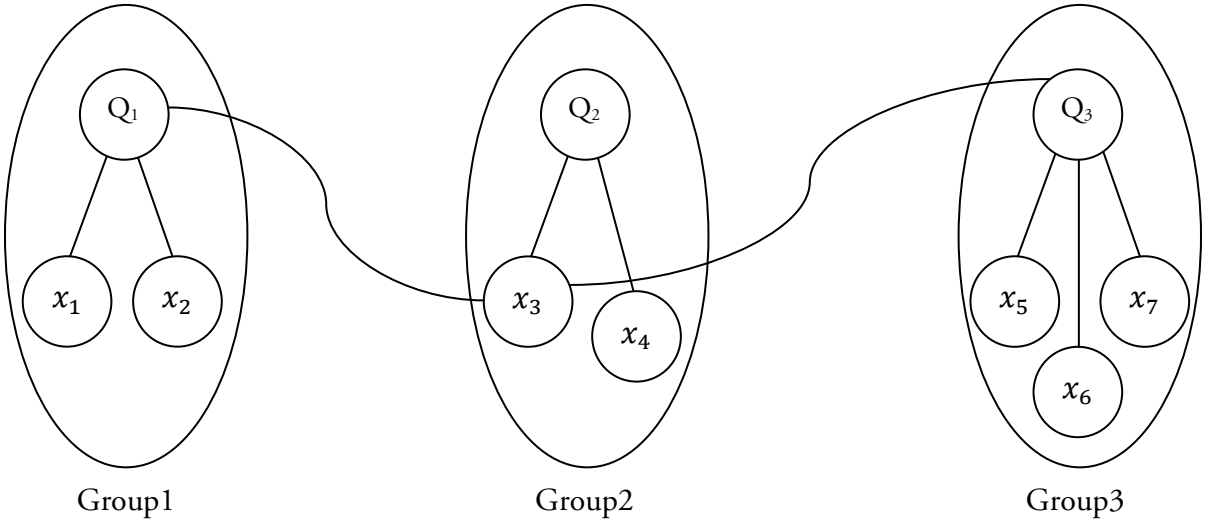

Figure 3: An example of an acyclic factor graph with 7 agents and 3 groups

3.2 Agents Grouping Mechanism

Since the work on the coordinated Q-learning approach (Zhang & Lesser, 2011) did not state a technique to group agents (they assumed that the grouping is given using a hand-crafted grouping), we propose our own grouping mechanism that is used to distribute agents among a pre-defined number of groups and to ensure that the induced factor graph is acyclic. The agents grouping process is carried out over two distributions as follows: the number of agents is divided by the number of groups and then the Floor of the resulted number represents the number of agents per group. After the first distribution process takes place, all agents that got no group will be added to the latest group (e.g. if there are 5 agents and 2 groups then group1 will take 2 agents and group2 will take 3 agents). For the second distribution, if an agent *x* has a neighbor *y* which is in different group than agent *x* group, then the set of groups agent *x* belongs to will be extended to have the group of agent *y* too.

Our grouping mechanism is designed in a way that forbids having two or more agents influencing the group of each other (i.e. if the group of agent i depends on action of agent j, then it is forbidden for the group of agent j to depend on any agent that belongs to the group of agent i). This prohibition ensures having acyclic factor graphs which ensure the optimality of the computed joint action using the Max-Sum algorithm. The following figures illustrate examples of the induced network of agents which are distributed among a number of groups. Note that label on each node/agent indicates the group of this node/agent. For example, nodes with a label of [0] means that these nodes/agents belong to group 0, while nodes with a label of [0,1] states that these agents belong to both group 0 and 1.

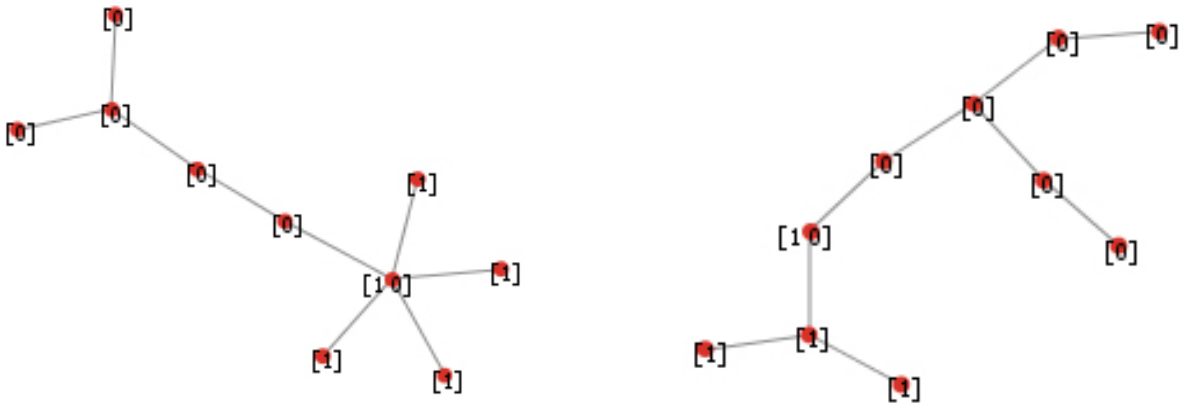

Figure 3.4: Two examples of using the grouping technique to distribute 10 agents among 2 groups

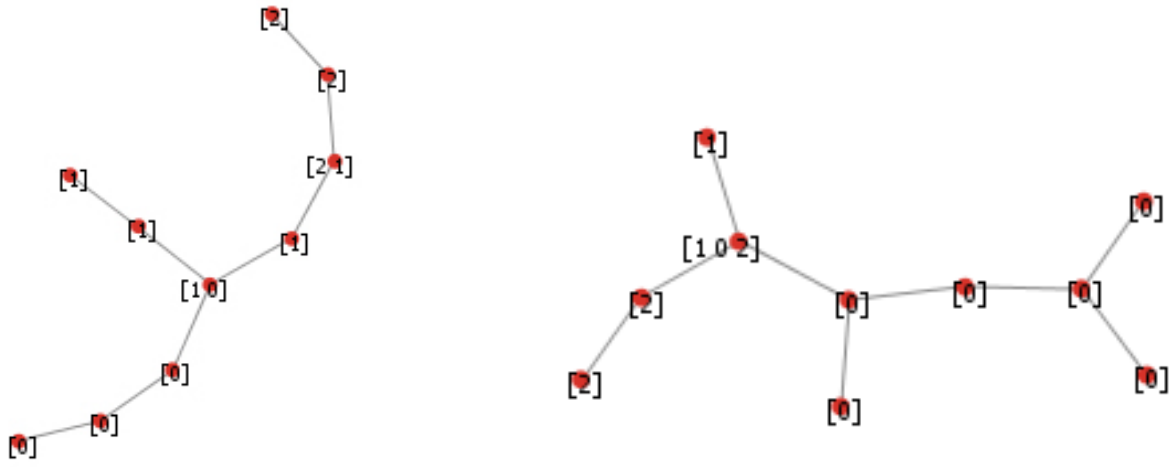

Figure 3.5: Two examples of using the grouping technique to distribute 10 agents among 3 groups

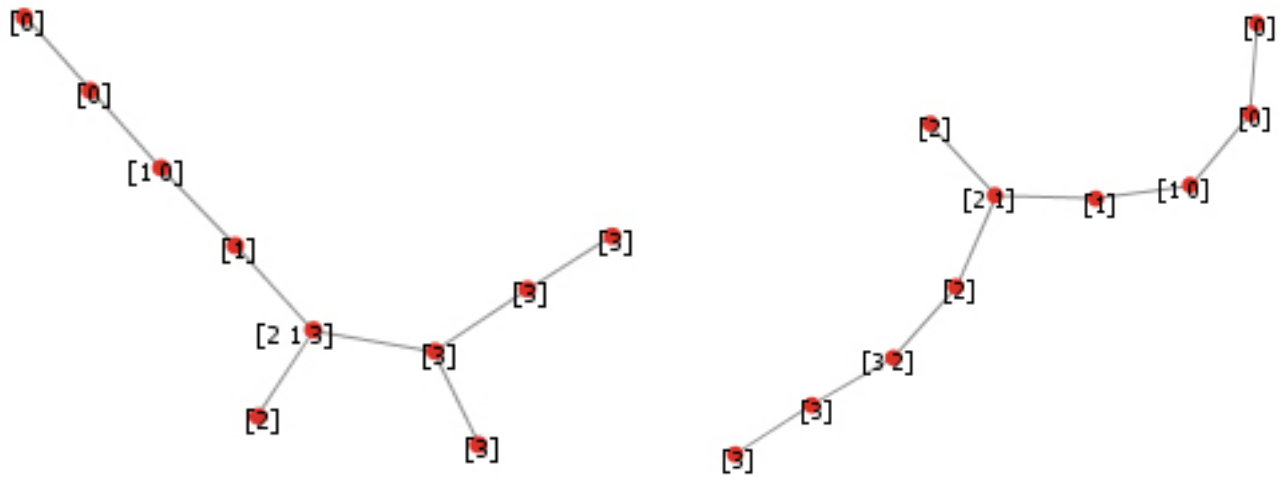

Figure 3.6: Two examples of using the grouping technique to distribute 10 agents among 4 groups

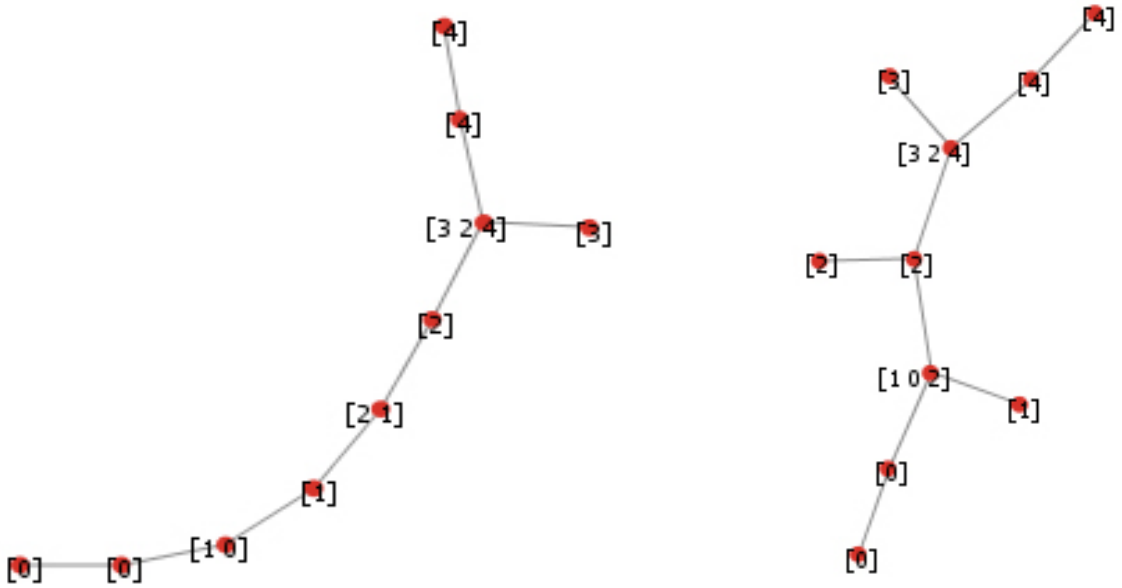

Figure 3.7: Two examples using the grouping technique to distribute 10 agents among 5 groups

3.3 Our Problem Domain and Network Structures

3.3.1 Cooperative and Semi-Cooperative Games

Unlike the domain used in (Zhang & Lesser, 2011) which is the distributed sensor networks domain, we are interested in cooperative/semi-cooperative two-player two-action games. In such games, agents maximize their reward function by coordinating their joint actions. We choose to study the performance of the adopted coordination approach in one of the most famous cooperative games, the Coordination Game, where two agents get the same high reward if both choose the same action and get the same negative or no reward otherwise. Table 3.1 illustrates the payoff matrix of both players in the coordination game where each player chooses one of two colors (i.e. Red or Blue). After choosing their action, each player receives a reward value, if both players choose the same color then each will get a reward of 1, otherwise both gets 0. One can note that this coordination game has two pure Nash Equilibriums: both agents choose red and both agents choose blue.

Player 1/ Player 2	Red (R)	Blue (B)
Red (R)	(1,1)	(0,0)
Blue (B)	(0,0)	(1,1)

Table 3.1: Payoff matrix of both players when playing the tested coordination game

We also study and analyze the performance of the Coordinated Q-learning Algorithm in one of the most famous semi-cooperative games, the Iterated Prisoner's Dilemma (IPD). This game illustrates a situation where it is hard for two players to coordinate their actions due to the imbalanced reward value received by agents if one chose a different action than its partner. As mentioned in chapter 2, semi-cooperative/semi-competitive games present a challenge for reinforcement learning algorithms because an agent's self-interest feature rises above the mutual interest. The following is a good example to simplify and illustrate the IPD:

Two men got caught while trying to rob a bank. However, the police have no evidence to convict the two men with the robbing crime and they can only prison them for a month for carrying guns. Therefore, the police separate each of those men and put them in different rooms then offer them the same deal: if one testifies against the other and betrays him (i.e. Defect) while the other remains silent (i.e. Cooperate), then the betrayer will go free while the other is sentenced to a one-year in jail. If both remain silent then both are sentenced to a one-month in jail for carrying guns with them, and if both testify against the other then both are sentenced to three-months in jail. Neither of the prisoners knows what the other chooses to do.

Each player is supposed to choose the action that lessens its prison time (represented as a high-reward action). Table 3.2 illustrates the payoff matrix of both players when playing IPD. Although it is clearly obvious that both players must choose to "Cooperate" as their optimal joint action that will maximize their reward function, they may not choose to do so since the cooperator is not rewarded if the other player chooses to defect (in some other payoff matrices of the IPD the cooperator receives a punishment, negative reward value, if the other player chooses to defect). As mentioned in Chapter 2, Defecting is a dominant action in game theory in which that agent is rewarded if it chooses to defect regardless of its partner action. Therefore, there is only one Nash Equilibrium presented in this game, that is, both players choose to Defect (Sandholm & Crites, 1995).

Player 1/ Player 2	Defect (D)	Cooperate (C)
Defect (D)	(1,1)	(5,0)
Cooperate (C)	(0,5)	(3,3)

Table 3.2: Payoff matrix of both players when playing the tested iterated prisoner's dilemma game

3.3.2 Network Structures

The main contribution of this thesis is to study and analyze the performance of the coordinated Q-learning approach in cooperative/semi-cooperative two-player two-action games under different network structures. Therefore, we choose to test the coordinated Q-learning approach in two different network structures, that is, Random and Scale-Free network structures. Erdos and Renyi (1959) are the first to define random networks. There assumption is that, in random networks, nodes are connected to another node(s) using random placement of edges/links and most nodes have approximately the same number of connected nodes (i.e. the same number number of edges/links), a unique characteristic of random networks in which it is considered to be very rare to find a node with extremely more or less number of edges/links than the average.

Unlike random networks, the distribution of links/edges connecting nodes in scale-free networks follows a power law in that most nodes have a low number of connecting edges/links and few nodes, called hubs, have an extremely large number of edges/links (Barabasi and Bonabea, 2003). Scale-free networks are strong and robust against random and accidental failures in which if a random failure occurred to some of the nodes with small degree, then the probability that a hub is affected by this failure is so low. Even if the failure happened to a hub-node, then the network will remain connected due to the rest of existing hubs. However, scale-free networks are weak against coordinated attacks that target hub-nodes. Figures 3.8 and 3.9 illustrate several examples of random networks and scale-free networks respectively.

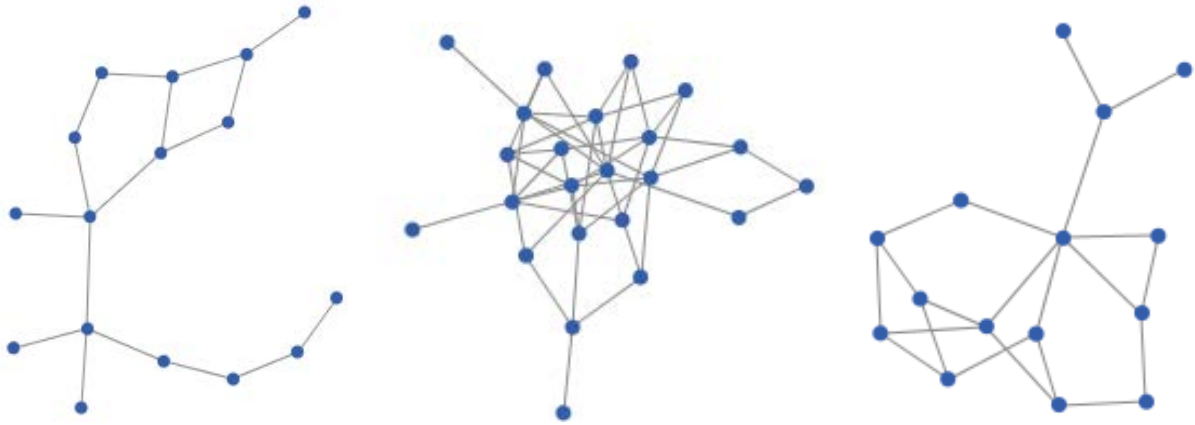

Figure 3.8: Examples of Random Networks

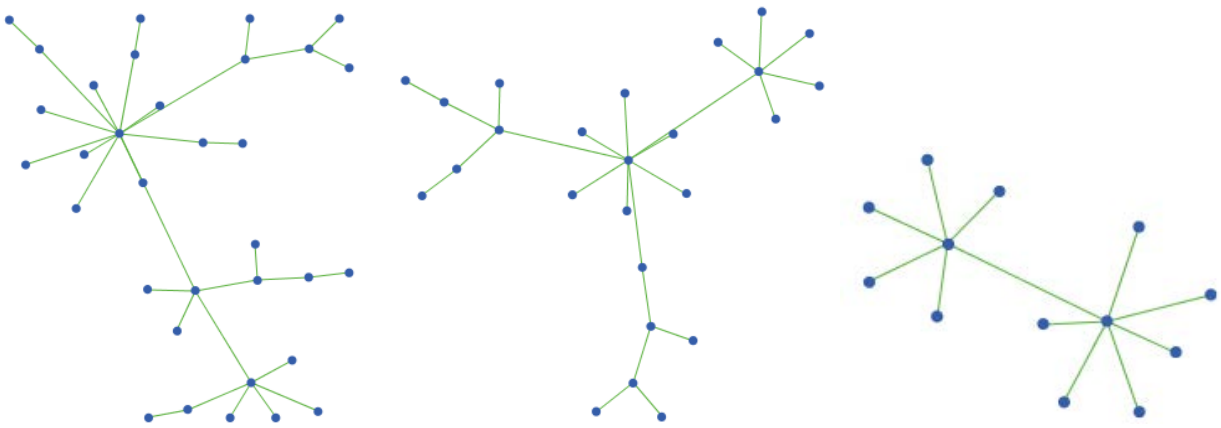

Figure 3.9: Examples of Scale-Free Networks

In the following chapter we apply the adopted coordinated Q-learning approach in both of the coordination game and the iterated prisoner's dilemma, each with its payoff matrix presented in this chapter (see table 3.1 and table 3.2). Both games are modeled using ND-POMDP model without the *groupwise observability* property. Our grouping mechanism which was discussed in this chapter is used to distribute agents among groups and to ensure that the induced factor graph of the DCOP is acyclic (which will ensure the optimality of the joint action computed by the Max-Sum algorithm). It is worthy to mention that we ignore the use of states in the original Q-learning algorithm in the following experiments to minimize the computational complexity (i.e. Q(a) is used instead of Q(s,a)).

Chapter 4

Experimental Analysis

This chapter presents the settings of the experiments conducted to evaluate and compare the performance of both the original Q-learning and the Coordinated Q-learning Algorithm in the Coordination and the Iterated Prisoner's Dilemma games under different network structures and settings. It provides the results achieved by conducting the experiments and investigates the possible effect of increasing the number of delegate agents and the horizon on the performance of the coordinated Q-learning algorithm. Finally, it presents a reasonable justification of the achieved results and answers the thesis research questions.

4.1 Experimental Setup

A simulator of the tested games, network structures and algorithms is built using NetLogo (Wilensky, 1999), an agent-based modeling environment and programming language, to evaluate the performance of the Coordinated Q-learning algorithm and compare it with that of the original Q-learning algorithm. The process of evaluating the performance of each algorithm for each tested game is carried out as follows. For each network structure, the average payoff is computed over 30000 simulation time steps and the results are averaged over 15 simulation runs. For the coordinated Q-learning, different number of learning groups of agents is used and the evaluation process is repeated for each horizon H, which ranges from 1 to 3, to measure the effect of both the number of learning groups and horizon on the performance of the algorithm (i.e. results are averaged over 45 simulation runs).

The learning rate α of both algorithms is set to 0.1, the exploration rate ε is set to 0.1(the ε-greedy exploration is used as the exploration algorithm) and the discount factor γ is set to 0.9 which are commonly used values for these learning parameters. As mentioned in the 3rd section of chapter 3, both original Q-learning and coordinated Q-learning are tested in 2 multi-agent games, the

coordination game and the iterated prisoner's dilemma game under two different network structures, random and scale-free networks. The scale-free network is created using the preferential attachment generative model. Section 3 in the previous chapter contains figures which illustrate some examples of the network structures generated by our simulator. Figure 4.1 presents an example of each network structure with a demonstration of each node degree.

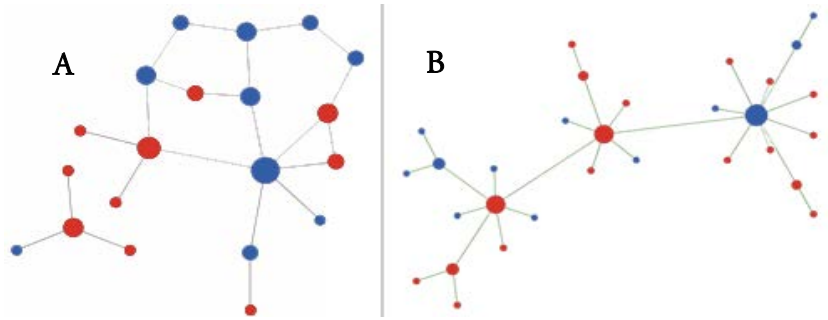

Figure 4.1: An example of the tested Random Network (A) and the tested Scale-Free Network (B). The node size represents the node degree (the higher the node degree, the larger the node size)

4.2 Experimental Results and Evaluation

4.2.1 Coordination Game Results

The following presents the results of implementing the tested algorithms in the Coordination game under two different network structures. As mentioned in the sub-section 4.1, the average payoff is computed over 30000 simulation time steps and then, for each network structure, the results are averaged over 15 and 45 simulation runs for the original Q-learning algorithm and the coordinated Q-learning algorithm respectively.

- **Original Q-learning Algorithm**

The algorithm converges to Nash Equilibrium in both network structures. However, it is noted that while the algorithm converges faster in scale-free networks, it performs better in random networks (see Figure 4.2). A reasonable justification is that, in POMDPs each agent has limited

observability of the state and actions of other agents in the same network. Agents are only aware of the actions and states of their connected neighbors and unlike random networks, the degree distribution of nodes in scale-free networks follows a power law in which a network of nodes with extremely high degree (i.e. hubs) followed by and connected to relatively smaller degree is created (see Figure 4.1 in the above sub-section). When the coordination game is played in scale-free networks, the network structure forces most of the learning agents to be connected to only one agent (i.e. most likely a hub). Therefore, agents that have a hub agent as their only possible partner (i.e. only connected to a hub agent) choose their action only based on the action of this hub agent even if this action is not globally optimal (i.e. sub-optimal) causing the average payoff to be lower than that of random networks, where most agents have more than one possible partner giving them more action choices than in scale-free networks, and resulting in a better performance.

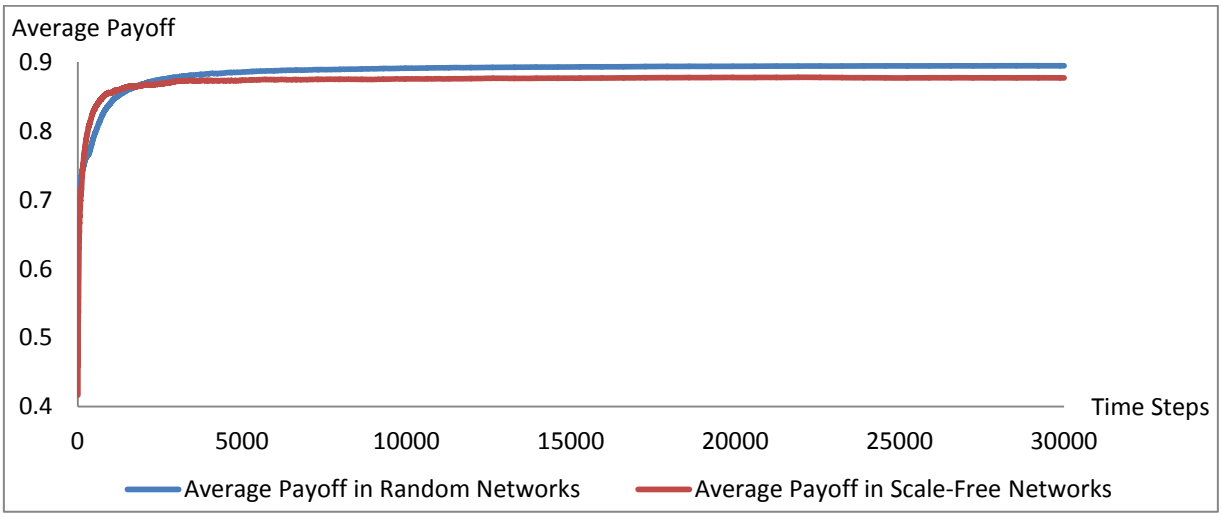

Figure 4.2: The Average Payoff of the original Q-learning algorithm when playing the Coordination game in random and scale-free networks

- **Coordinated Q-learning Algorithm**

The coordinated algorithm converges to a Nash Equilibrium (i.e. all learning agents execute the same action of their partners). The coordinated Q-learning algorithm converges faster in scale-free networks due to the same reason mentioned in the original Q-learning algorithm performance. However, there is no difference in the average payoff achieved in both network structures when

playing the coordination game using the coordinated Q-learning algorithm (see Figure 4.3). A reasonable justification of this is that the agents learning using the coordinated algorithm are not affected by the observability limitation because there is a communication during the execution time in which agents exchange their observations and choose their best action based on the joint observations through their delegate agent(s).

Also, it is observed that the coordinated Q-learning algorithm converges faster and performs slightly better than the original Q-learning algorithm. This is mainly because, besides the fact that there is no communication in the original Q-learning algorithm which prevents each agent from accessing some needed information of other agents except its connected neighbors, the coordinated Q-learning approach is distributed (since it distributes the learning of the global optimal policy among the groups of agents) resulting in a large computational savings of the policy space and making the algorithm scale to large domains and reducing the time required for the algorithm to converge.

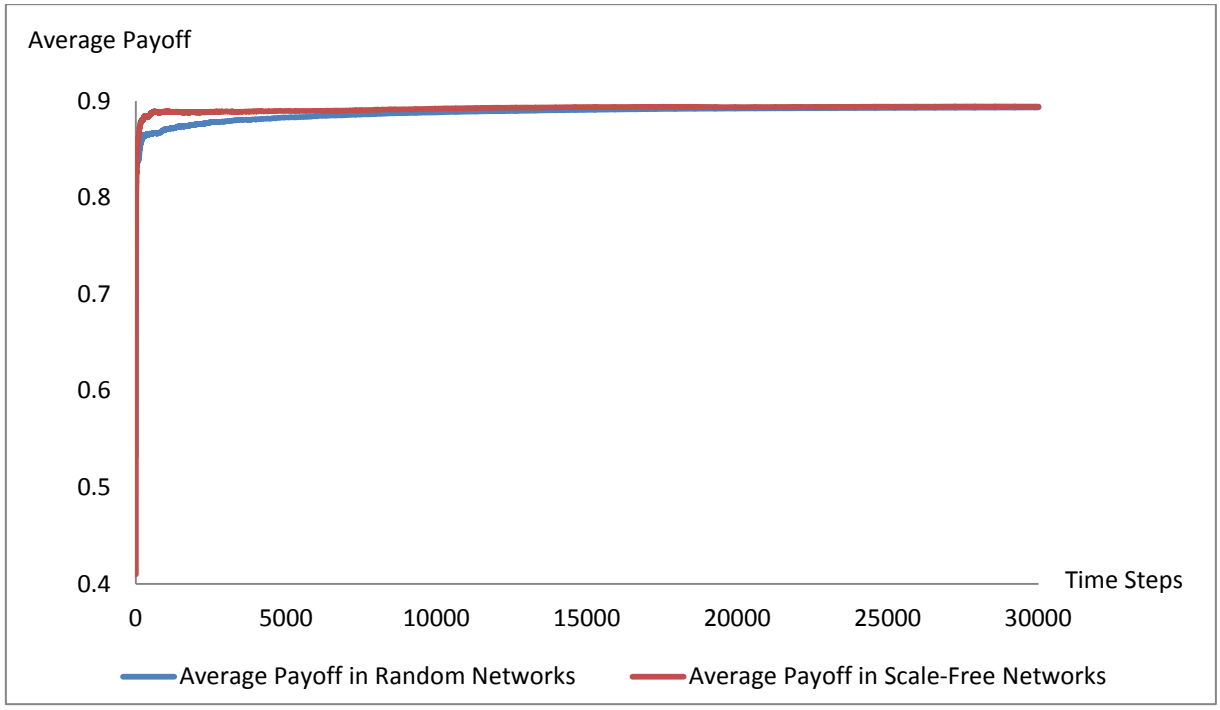

Figure 4.3: The Average Payoff of the coordinated Q-learning algorithm when playing the Coordination game in random and scale-free networks

4.2.2 Iterated Prisoner's Dilemma Results

The following presents the results of implementing the tested algorithms in the Coordination game under two different network structures. As mentioned in the sub-section 4.1, the average payoff is computed over 30000 simulation runs and the results are then averaged over 15 experiments per network structure for the original Q-learning algorithm and 45 experiments per network structure for the coordinated Q-learning algorithm.

- **Original Q-learning Algorithm**

The Q-learning algorithm converges to a Nash Equilibrium in which all agents choose "Defect" as their optimal joint action in both network structures. However, it is observed that the algorithm converges faster and performs better in scale-free networks than in random networks (see Figure 4.4). A reasonable justification is that, as mentioned in the coordination game results sub-section, each learning agent suffers from observability limitation in that it might not be able to access some needed information about agents other than its partners (such as their states and actions). Due to its structure, scale-free networks will converge faster than random networks but will suffer from the observability limitation more than them as this limitation increases when the number of agents' partners decreases (the case in scale-free networks).

Therefore, when the iterated prisoner's dilemma game is played in scale-free networks, learning agents that have a hub agent as their only possible partner will choose their action only based on the action of this hub agent even if this action is sub-optimal causing the average payoff to be higher than that of a random network since if some agents cooperated, instead of defecting which is the global optimal action, then their partners will get a higher payoff (i.e. 5 instead of 1) that increases the total average payoff more than if all agents were to defect (as in random networks).

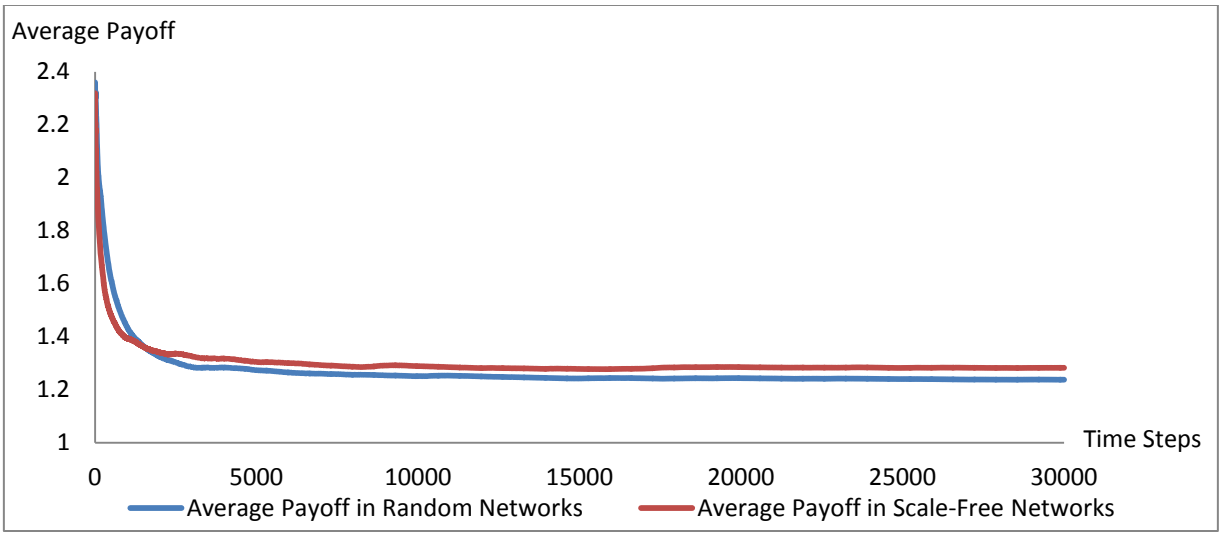

Figure 4.4: The Average Payoff of the original Q-learning algorithm when playing the Iterated Prisoner's Dilemma game in random and scale-free networks

- **Coordinated Q-learning Algorithm**

The coordinated algorithm converges to the optimal joint action in which, unlike the original Q-learning, all agents choose to "Cooperate" with converging faster in random networks and approximately no difference in the average payoff achieved in both network structures (see Figure 4.5). A reasonable justification of this is that the original Q-learning algorithm computes the global optimal joint action based on the local optimal action of each agent whereas the coordination approach computes the optimal action based on the global optimal policy of each group of learning agents and then it distributes the learning of the global optimal policy among groups and coordinates the distributed learning through using the Max-Sum algorithm as a message passing mechanism that enable agents to communicate and share their observations during the execution time. Therefore, and since choosing "Cooperate" by all agents will maximize the global Q-value function way better than choosing "Defect" (which is a local optimal action), the global optimal policy of the coordinated Q-learning is to make all agents "Cooperate". As a result, the coordinated Q-learning performs significantly better than the original Q-learning algorithm and due to its distributed nature it converges faster than the original algorithm. The following sub-section

provides an investigation of the effect of changing the value of some parameters in the coordinated learning algorithm.

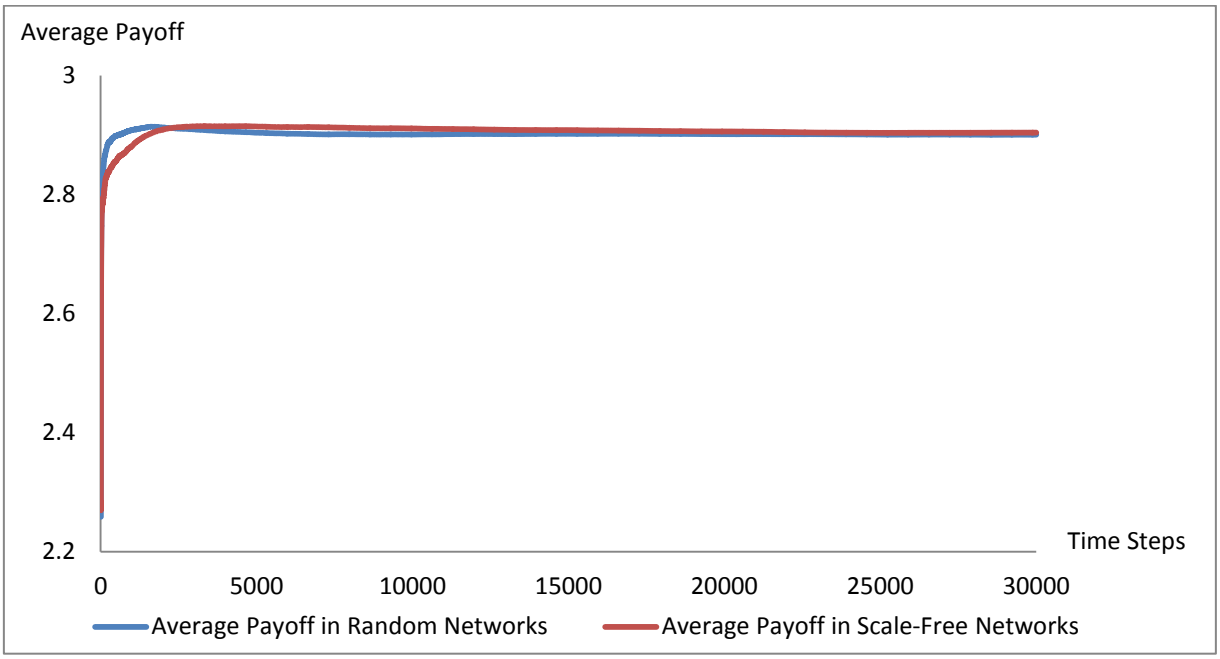

Figure 4.5: The Average Payoff of the coordinated Q-learning algorithm when playing the Iterated Prisoner's Dilemma game in random and scale-free networks

4.2.3 Further Investigations on the Coordinated Q-learning algorithm

Two essential parameters are investigated to evaluate the effect of changing their values on the performance of the coordinated. These parameters are: The Horizon H and The number of learning groups of agents (since each learning group has only one delegate agent that learns on behalf of its group as mentioned in the 1st section of chapter 3, the number of learning groups is equal to the number of delegate agents). The value of the Horizon ranges from 1 to 3 and the number of learning delegates ranges from 1 to 5 delegates. The average payoff is computed over 30000 simulation time steps and the results are then averaged over 45 simulation runs per network structure (i.e. conduct 3 experiments per learning group value for each horizon value (3*(3*5))).

Figure 4.6 illustrates the average payoff of each learning group per horizon value when playing the iterated prisoner's dilemma using the Coordinated Q-learning algorithm. It is obviously

observed that as the value of the horizon increases the average payoff increases (i.e. the algorithm performs better in both network structures and tested games). This is mainly because, as the horizon increase, each agent will have access to more information (i.e. previous actions and rewards) about other agents which will greatly help in the learning process leading to better performance. Also, it is observed that as the number of learning groups increases the coordinated Q-learning algorithm converges faster in both network structures and in both tested games. A reasonable justification is that the distribution property leads to a large saving of the computation complexity of the policy space which results in decreasing the time required for both agents to learn and algorithm to converge. The following sub-section demonstrates an illustrative comparison between the performance of both Original and Coordinated Q-learning algorithms. Also, the standard deviation is computed for each case and its value is so small (under 0.03) to be illustrated.

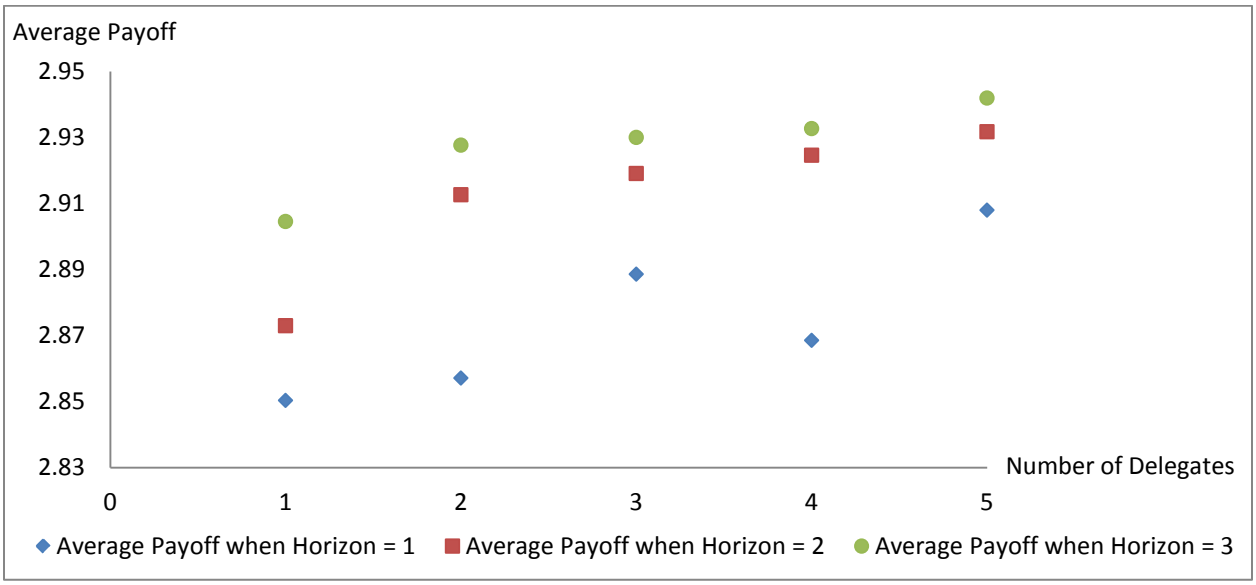

Figure 4.6: The Average Payoff against the number of delegate agents in the Coordinated Q-learning algorithm when playing the iterated prisoner's dilemma

4.2.4 Illustrative comparison between learning algorithms

Figure 4.7 illustrates a comparison of the global performance of both learning algorithms when playing the coordination game. It is clearly observed that the coordinated Q-learning algorithm outperforms the original Q-learning algorithm slightly in the coordination game. In order to

investigate if this result is affected by the exploration rate value Ɛ, smaller values of the exploration rate are used. Figure 4.8 and 4.9 represent the global performance of both algorithms when the exploration rate Ɛ is set to 0.01 and 0.001 respectively. It is clear that as the exploration rate decreases the difference in the performance between both learning algorithm increases in which the coordinated Q-learning outperforms the original Q-learning algorithm. Figure 4.10 illustrates a comparison of the global performance between both algorithms when playing the iterated prisoner's dilemma. It is clearly shown that the coordinated Q-learning algorithm outperforms the original Q-learning algorithm significantly even when the exploration rate is set to 0.1 and the distance between the values of the performance of both algorithms will keep on increasing as the exploration rate decreases.

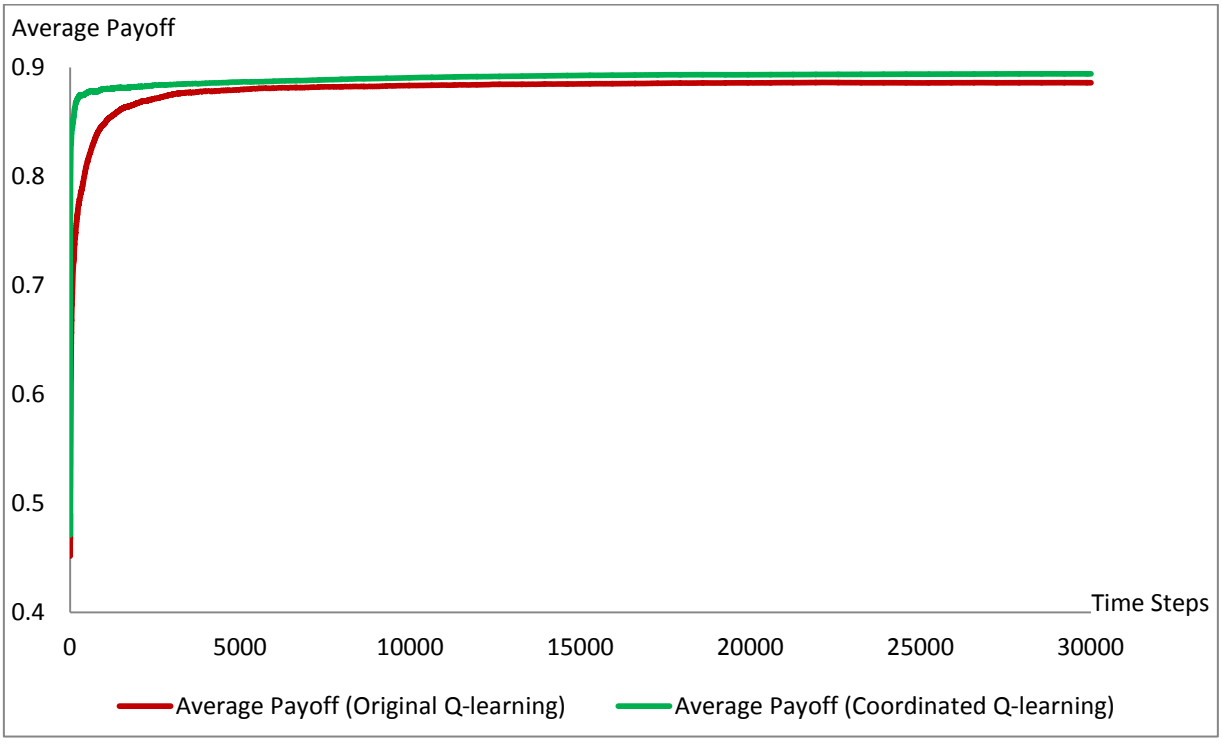

Figure 4.7: A comparison of the performance of the original and coordinated Q-learning algorithms when playing the coordination game with exploration rate Ɛ = 0.1

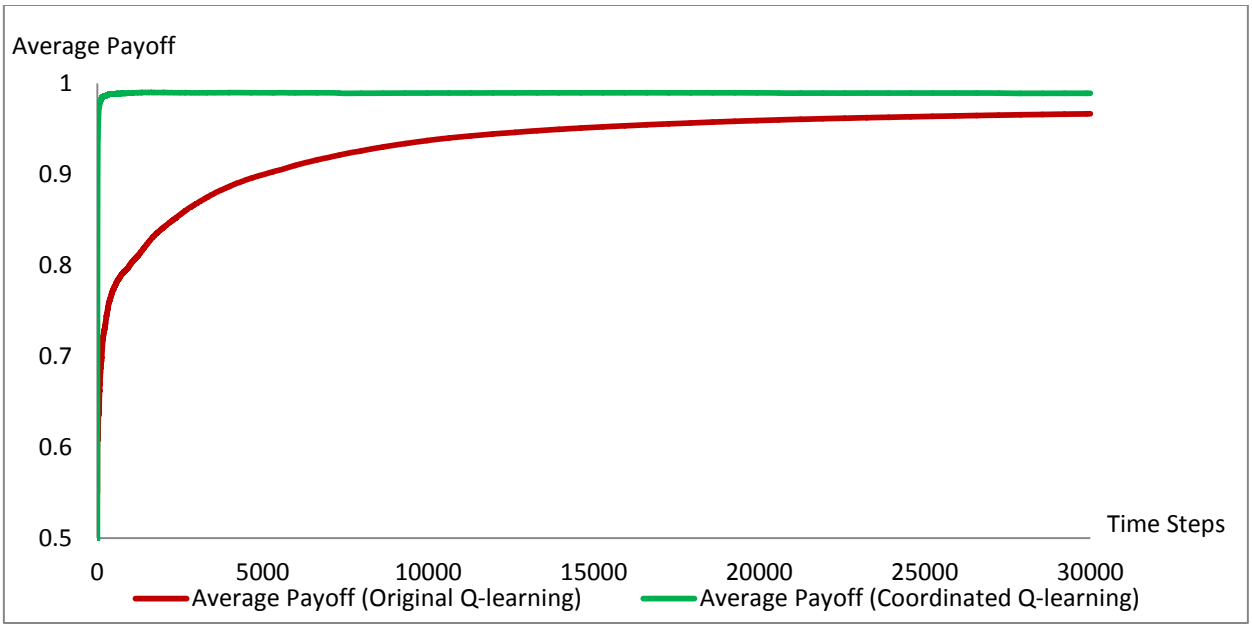

Figure 4.8: A comparison of the performance of the original and coordinated Q-learning algorithms when playing the coordination game with exploration rate $\varepsilon = 0.01$

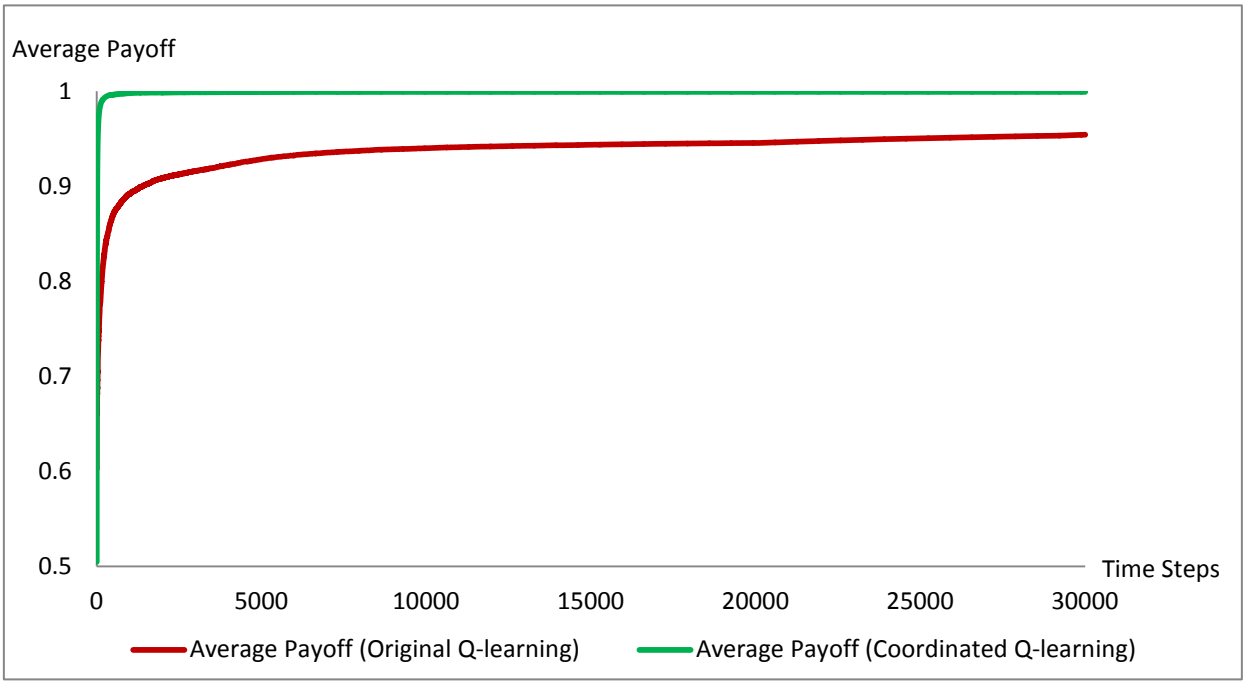

Figure 4.9: A comparison of the performance of the original and coordinated Q-learning algorithms when playing the coordination game with exploration rate $\varepsilon = 0.001$

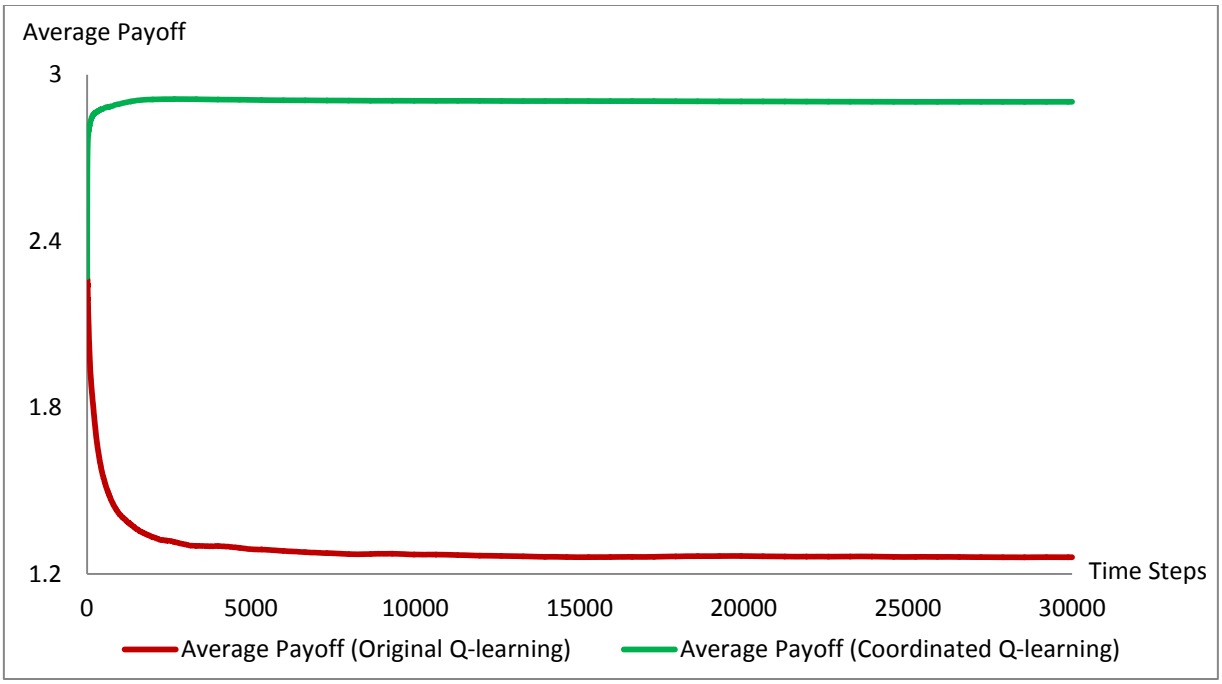

Figure 4.10: A comparison of the performance of the original and coordinated Q-learning algorithms when playing the iterated prisoner's dilemma with exploration rate $\varepsilon = 0.1$

4.3 Experiments Summary

Several critical characteristics of the coordinated Q-learning algorithm have been discovered. Firstly, we found that, unlike the original Q-learning algorithm, the performance of the coordinated algorithm is not affected by both random and scale-free network structures (i.e. the coordination approach is robust against the tested network structures). Secondly, the coordinated Q-learning algorithm outperforms the original Q-learning algorithm slightly in the coordination game and significantly in the iterated prisoner's dilemma game and, as the exploration rate value decreases, the difference of the performance between both algorithms increases in which the coordinated Q-learning algorithm converges faster and outperforms the original algorithm. This is mainly due to the distribution and communication property of the coordinated Q-learning algorithm. Thirdly, the number of delegate agents and the horizon are found to be very important parameters that can affect the speed of convergence and the global performance of the coordinated algorithm respectively. When the value of the horizon increases, the coordinated algorithm performs better and when the number of delegate agents increases, the coordinated algorithm converges faster. Finally, we can answer the thesis research questions now that we achieved the previous results:

- Does the coordinated Q-learning approach help in improving the performance of multi-agent learning algorithms in networks when applied in two-player two-action cooperative/semi-cooperative games?

Based on the conducted experiments, the coordinated Q-learning approach demonstrates a superior performance in cooperative and semi-cooperative two-player two-action games in networks.

- Is the performance of the coordinated Q-learning approach affected by different network structures such as random and scale-free networks?

Based on the experimental results, the performance of the coordination approach is not affected by random and scale-free networks.

- Is there a simple grouping methodology to cluster agents in a network automatically? Can such methodology ensure cycle-free clustering?

Yes, the proposed grouping methodology in this thesis automatically clusters agents in both random and scale-free networks and ensures cycle-free grouping.

- Is the performance of the coordinated Q-learning approach affected by some of its parameters?

Based on the conducted experiments, the performance of the coordination approach is affected by both horizon value and the number of learning groups of agents.

4.4 Generalizing the Coordination Approach

We thought of generalizing the coordinated approach by adding one more level of a new delegate agent, called super delegate agent, to be placed on top of other delegate agents and learn on behalf of them in the same manner delegate agents used to learn on behalf of agents in their group (see Figure 4.11). However, in this generalization there are two ways to use the super delegate agent: it will either learn on behalf of delegate agents who learn on behalf of agents in their groups only when it is needed, or it will learn on behalf of delegate agents and give them instructions about what to send to their agents. In the second way, the super delegate agent is the only one who has a Q-table since both agents and delegate agents will not learn.

We realized that in either ways the convergence will be slower than in the current coordination approach since the distribution property is reduced and both of the time needed to converge and the computational complexity of the policy space are increased. Therefore, it is better not to generalize the coordination approach by adding a super delegate agent who will learn on behalf of both delegate agents and normal agents. We are currently investigating other possible ways to further enhance the performance of the coordination approach.

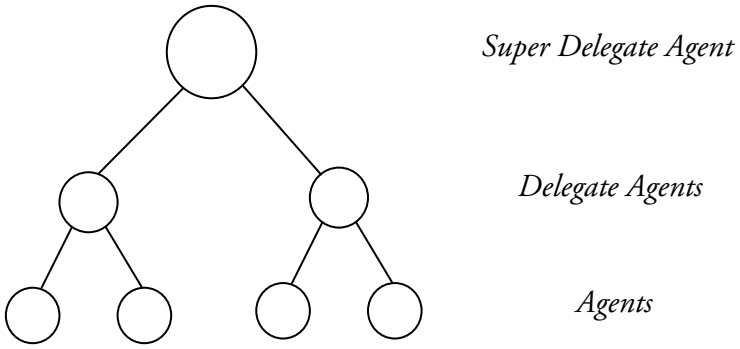

Figure 4.11: A Generalized form of the Coordination Approach

Chapter 5

Conclusion and Future Work

This chapter concludes the work done in this thesis and discusses a set of possible works that can be carried out in the future.

5.1 Conclusion

This thesis attempts to study and analyze the performance of one of the recent coordinated multi-agent reinforcement learning approaches, the coordinated Q-learning approach, in cooperative and semi-cooperative two-player two-action games under different network structures (i.e. random and scale-free networks) to better understand its characteristics, strength and weakness points. Since the adopted coordination approach is based on distributing learning agents among a number of groups in which there is a delegate agent for each group that will learn in behalf of the group members and then using the Max-Sum algorithm, a DCOP technique, agents are allowed to communicate with each other during the execution time and the optimal joint action is computed, we have proposed a novel grouping mechanism to perform the grouping process in a way that will ensure cyclic-free grouping which will in turn ensure the optimality of the solution computed using the Max-Sum algorithm. In addition, a simulator of the tested learning algorithm, tested games and networks has been built using NetLogo to carry out the experiments which will evaluate the performance of the coordinated Q-learning algorithm and compare it with that of the original Q-learning algorithm.

After conducting several experiments, the research questions of this thesis have been answered. For the first question, **the coordination approach is proved to significantly enhance the performance of multi-agent learning algorithms when applied in two-player two-action cooperative/semi-cooperative games in networks**. Experimental results show that the coordinated Q-learning algorithm significantly outperforms the original Q-learning algorithm in all tested games and networks in which the original Q-learning algorithm converges to Nash-

Equilibrium, while the coordinated Q-learning algorithm converges to Pareto Optimal. This is mainly due to the fact that the coordinated Q-learning algorithm allows the communication between agents during the execution time and, unlike the original Q-learning algorithms which computes the local optimal joint action, computes the global optimal joint action. For the second research question, **results show that the performance of the coordinated Q-learning algorithm, unlike the original Q-learning algorithm, is unaffected by the difference presented in random and scale-free network structures** (i.e. the coordinated Q-learning algorithm is robust against random and scale-free networks).

As for the third research question, **a simple, yet effective, grouping methodology has been proposed in this thesis in which agents are automatically distributed among groups and a cycle-free grouping is ensured**. This grouping technique ensures the optimality of the policy computed using the Max-Sum algorithm. Furthermore, a set of experiments have been conducted to check if there is any parameter that affects the performance of the coordinated Q-learning algorithm to answer the last question, and results show that **there are two parameters which affect the performance of the coordinated Q-learning algorithm, the number of delegate (i.e. the number of groups) and the horizon value**. While the number of groups affects the speed of convergence (in which coordinated Q-learning algorithm converges faster as the number of groups increases), the horizon value affects the global performance of the coordinated Q-learning algorithm in which the coordinated Q-learning performs better as the value of the horizon increases.

Finally, an attempt to generalize the coordinated Q-learning approach by adding one more level of a new agent (called the super delegate agent) which learns on behalf of delegate agents is shown to be unnecessary. In contrast, adding such a level can be considered as an inefficient step since it removes the distributive feature in the current coordinated Q-learning and therefore, slows down the convergence of the coordination algorithm due to increasing the state space and computational complexity.

5.2 Future Work

After achieving the previously mentioned results, several ideas can be rendered as possible works to be carried out in the future as an extension of this thesis. This thesis has tested the coordinated Q-learning algorithm in two-player two-action games; therefore, a worthwhile work to be conducted in the future is to test the same coordination approach in n-player n-action games. Another possible work to be done in the future is to apply the adopted coordination approach using different multi-agent learning algorithm other than the Q-learning algorithm such as one of the gradient ascent algorithms (e.g. IGA, GIGA-WoLF and WPL algorithms) in the same domain and network structures applied in this thesis and check how the coordination approach affects the performance of such algorithms.

Furthermore, since the adopted coordination approach assumes that there is only one delegate agent per a group of agents, it will be interesting to carry out a work which studies and analyzes the effect of increasing the number of delegates per group of agents on the performance of the coordination algorithm. Another possible work to be carried out in the future is to study the effect of modifying the learning parameter values (i.e. the learning rate and the discount factor values) during the learning process on the performance of the coordinated Q-learning algorithm. Finally, a good work to be done in the future is to test the coordination approach under different network structures other than the ones tested in this thesis to ensure the robustness of the coordinated learning algorithm.

References

Abdallah, S. & Lesser, V. (2006). Learning the Task Allocation Game. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 850-857.

Abdallah, S. & Lesser, V. (2007). Multiagent Reinforcement Learning and Self-Organization in a Network of Agents. In *Proceedings of the 6th International Joint conference on Autonomous Agents and Multiagent Systems (AAMAS'07)*.

Abdallah, S. & Lesser, V. (2008). A Multiagent Reinforcement Learning Algorithm with Non-linear Dynamics. *Journal of Artificial Intelligence Research*, vol. 33, pp. 521-549.

Aji, S.M. & McEliece, R.J. (2000). The Generalized Distributive Law. *IEEE Transactions on Information Theory*.

Babes, M., Wunder, M. & Littman, M.(2009). Q-Learning in Two-Player Two-Action Games. In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '09)*.

Barabási, A. L. & E. Bonabeau. (2003). Scale-free networks. *Scientific American*, 288(5), pp. 60–69.

Bernstein, D.S., Zilberstein, S. & Immerman, N. (2000). In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI '00)*, pp. 32-37.

Boutilier, C. (1996). Planning, Learning and Coordination in Multiagent Decision Processes. In *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge (TARK'96)*, pp. 195-210

Boutilier, C. (1999). Sequential Optimality and Coordination in Multiagent Systems. In *International Joint Conference on Artificial Intelligence*, pp. 478-485.

Bowling, M., & Veloso, M. (2002). Multiagent Learning Using a Variable Learning Rate. *Artificial Intelligence*, 136(2), pp. 215–250.

Bowling, M. (2005). Convergence and No-Regret in Multiagent Learning. In *Proceedings of the Annual Conference on Advances in Neural Information Processing Systems*, pp. 209–216.

Claus, C. & Boutilier, C. (1998). The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *Proceedings of National Conference on Artificial Intelligence (AAAI'98)*, pp. 746-752.

Conitzer, V. & Sandholm, T. (2007). AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2), pp. 23–43.

de Groot,B. (2008). Wolf-*Gradient, a Multi-Agent Learning Algorithm for Normal Form Games*. M.Sc. Thesis. University of Amsterdam.

Erdős, P. & Rényi, A. (1959). On Random Graphs I. *Publicationes Mathematicae*, vol. 6, pp. 290-297.

Guestrin, C., Lagoudakis, M. & Parr, R. (2002). Coordinated Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML'02)*, pp. 227-234.

Hansen, E.A. (2004). Dynamic Programming for Partially Observable Stochastic Games. In *Proceedings of The Nineteenth National Conference on Artificial Intelligence*, pp. 709-715

Howard, R.A. (1960). *Dynamic Programming and Markov Processes*. The MIT Press.

Hoen,P.J., Tuyls,K., Panait,L., Luke, S. & La Poutré, J.A. (2006). An Overview of Cooperative and Competitive Multiagent Learning. In *Learning and Adaption in Multi-Agent Systems*, vol. 3898, pp. 1- 46.

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

Kaelbling, L.P., Littman, M.L. & Cassandra, A.R. (1998). Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*, vol.101, pp. 99-134.

Leyton-Brown, K. & Shoham, Y. (2008). *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*, San Rafael, CA: Morgan & Claypool Publishers.

Littman, M.L. (1994). Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 157-163.

Modi, P. J., Shen,W., Tambe, M., & Yokoo,M. (2003). An Asynchronous Complete Method For Distributed Constraint Optimization. In AAMAS, pp. 161-168.

Mostafa, H. (2011). *Exploiting Structure in Coordinating Multiple Decision Makers*. Ph.D. Thesis. University of Massachusetts Amherst.

Nair, R., Varakantham, P., Tambe, M. & Yokoo, M. (2005). Networked Distributed POMDPs: A Synthesis of Distributed Constraint Optimization and POMDPs. In *Proceedings of the 20th conference on Association for the Advancement of Artificial Intelligence (AAAI '05)*, pp. 133-139.

Nash, J. (1950). Equilibrium Points in n-Person Games. In *Proceedings of the National Academy of Sciences*, 36(1), pp. 48-49.

Osborne, M.J. & Rubinstein, A. (1994). *A Course in Game Theory*. The MIT Press.

Panait,L. & Luke, S. (2005). Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multiagent Systems,* 11(3), pp. 387- 434.

Sandholm, T.W. & Crites, R.H. (1995). Multiagent Reinforcement Learning in the Iterated Prisoner's Dilemma. *Biosystems*, vol. 37, pp. 147-166.

Sandholm, T.W. & Crites, R.H. (1996). On Multiagent Q-Learning in a Semi-Competitive Domain. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence Workshop on Adaptation and Learning in Multi-agent Systems (IJCAI '95),* Springer Verlag, pp. 191-205.

Singh, S., Kearns, M. & Mansour, Y. (2000). Nash Convergence of Gradient Dynamics in General-Sum Games. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 541–548.

Shoham, Y. & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge, U.K.: Cambridge University Press.

Shoham, Y. & Powers, R. (2010). Multi-Agent Learning I: Problem Definition. *Encyclopedia of Machine Learning*, pp. 694-696.

Stranders, R., Farinelli, A., Rogers, A. & Jennings, N. R. (2009). Decentralised Coordination of Mobile Sensors Using the Max-Sum Algorithm. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pp. 299-304.

Watkins, C.J.C.H. & Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3), pp. 279-292.

Weib, G. (1993). Learning to Coordinate Actions in Multi-Agent Systems. In *Proceedings of the 13th International Conference on Artificial Intelligence*, pp. 311-316.

Wilensky, U. (1999). *NetLogo* [online]. [Accessed 20 March 2012]. Available at: http://ccl.northwestern.edu/netlogo/.

Yagan, D. & Tham, C.-K. (2007). Coordinated Reinforcement Learning for Decentralized Optimal Control. In *Proceedings of the IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL '07)*, pp. 296-302.

Zhang, C., Abdallah, S. & Lesser, V. (2010). Self-Organization for Coordinating Decentralized Reinforcement Learning. In *Proceedings of the 9th international joint conference on Autonomous Agents and Multi-Agent Systems (AAMAS'10)*, pp. 739-746.

Zhang, C. & Lesser, V. (2011). Coordinated Multi-Agent Reinforcement Learning in Networked Distributed POMDPs. In *Proceedings of the 25th conference on Association for the Advancement of Artificial Intelligence (AAAI'11)*.

Zinkevich, M. (2003). Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of the International Conference on Machine Learning*, pp. 928–936.

Declaration

I declare that this thesis was done and carried out by myself, and that all of the work contained here is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Mayada Mohamed Oudah