POLITECNICO DI TORINO

Master of Science
in Data Science and Engineering

Master of Science
Degree Thesis

Decentralized Value-Based Reinforcement Learning in Stochastic Potential Games

Supervisors

Prof. Fabio FAGNANI

Prof. Giacomo COMO

Candidate

Hafez GHAEMI
s289963

July 2022

Abstract

Multi-agent reinforcement learning (MARL) is a promising paradigm for learning problems that involve multiple decision makers. Contrary to centralized MARL with a central controller, decentralized (independent) MARL is a more practical paradigm in terms of scalibility, privacy, and computational cost, yet more challenging due to non-stationarity of the environment from an agent's perspective. The non-stationarity challenge arises as the evolution of the environment and each agent's payoffs will depend on the behavior of other agents. In value-based MARL, two-timescale learning is shown to address this issue. In such a learning dynamics, agents update their value function estimates at a timescale slower than their local Q-function estimates, and therefore, the game is rendered locally stationary with respect to the strategy of other agents. However, two-timescale dynamics in decentralized Q-learning has been studied only in two-player zero-sum games. In this thesis, we focus on a newly emerged and important class of stochastic games, stochastic potential games (SPG). We develop a many-player extension of the two-timescale decentralized Q-learning algorithm as the first game-agnostic value-based MARL algorithm in stochastic games, and analyze its asymptotic converges to Nash equilibria in SPGs. We evaluate the empirical performance of the algorithm on two SPG benchmarks, network routing games and distancing games.

Acknowledgements

I would like to thank my supervisors, Professor Fabio Fagnani and Professor Giacomo Como for their guidance during my thesis work. I would like to also thank Dr. Mark Jeeninga whose mathematical insight has been of extreme value in writing this thesis.

I dedicate my thesis to my parents, my brother, and my friends for their support during my studies.

Hafez Ghaemi, July 2022

Computational resources for running some of the experiments in this thesis has been provided by hpc@polito (`http://www.hpc.polito.it`).

Table of Contents

List of Tables

List of Figures

Acronyms

MARL
multi-agent reinforcement intelligence

RL
reinforcement intelligence

GT
game theory

SPG
stochastic potential game

MPG
Markov potential game

ML
machine learning

AI
artificial intelligence

MDP
Markov decision process

POMDP
partially observable Markov decision process

DP
dynamic programming

MCTS
Monte Carlo tree search

MC
Monte Carlo

RPE
reward prediction error

TD
temporal difference

PG
policy gradient

AC
actor critic

DDPG
deep deterministic policy gradient

SARSA
state–action–reward–state–action

DQN
deep Q-network

GLIE
greedy in the limit with infinite exploration

NE
Nash equilibrium

BR
best response

SG
stochastic game

IPG

independent policy gradient

INPG

independent natural policy gradient

ODE

ordinary differential equation

Chapter 1

Introduction

1.1 Motivations

1.1.1 MARL and Stochastic Potential Games

Multi-agent reinforcement learning (MARL) is a branch of reinforcement learning (RL) where multiple decision makers (agents or players) interact with a shared environment. As there are multiple players involved in MARL, it is closely related to game theory (GT), and especially repeated games. In a classical MARL setting, each agent tries to maximize its own notion of cumulative reward over repeated trials (games). The nature of the environment and the reward function of the agents can fit into one of three categories. In a purely cooperative setting, the agents share the same reward function (they have identical interests), and are incentivized to collaborate and maximize the shared reward. The second scenario is the fully competitive zero-sum environment with conflicting interests among agents. The pure competitive and collaborative settings have few practical and real-world use cases, and therefore, we may consider a third and middle-ground scenario, i.e., a non-zero sum mixed-motive environment where agents should balance collaboration and competition to achieve a stable equilibrium. An important class of mixed-motive environments is potential games that can be used to model a large number of real-world scenarios, in areas such as routing in computer and transportation networks that can be formulated as congestion games [1], resource allocation in settings like public good games [2], relaxation-labeling in image classification and segmentation [3, 4], and many more practical use cases [5]. Therefore, in this work, we focus on a newly emerged setup for MARL, known as stochastic potential game (SPG), also known as Markov potential game (MPG) [6, 7, 8, 9].

1.1.2 Decentralized Learning

MARL methods can be either centralized (coordinated), or decentralized (independent). The latter methods take advantage of a central coordinator who is in contact with all agents and can coordinate them to reach an equilibrium. On the other hand, in decentralized methods, the agents should make their decisions independently and only using local information. A degree of communication between neighboring agents may also be allowed in this setting.

In centralized MARL, the coordinator is fully aware of the game setup and can access the actions of all agents, and therefore, this setup is only feasible when we can simulate the game in a controlled environment [10, 11]. Furthermore, central algorithms may harm the privacy of agents and they may not scale as the number of agents grows. In decentralized MARL, many problems related to privacy and scalibility go away, as each agent only uses local information to make her decision. However, a great challenge that emerges in this setting is the non-stationarity of the environment [12, 13]. The changes in a MARL environment are a function of all agents' actions, however, each agent is not aware of the actions of other agents and has to act only based on her local observations and rewards. Therefore, since the opponents modify their strategies over time, the environment dynamics is non-stationary from the agent's prespective. Although information exchange between neighboring agents may be possible in some cases and may sometimes alleviate part of the non-stationarity problem, there are many cases where privacy, cost, or scale of the problem make the local communication between agents harmful, expensive, or futile. Therefore, a fully decentralized learning scheme that addresses the non-stationarity challenge and in which agents act with complete independence, could be beneficial in terms of privacy, cost, and scalibility in multi-agent reinforcement learning.

1.2 Contribution

Motivated by the previous section, in this work, we consider a two-timescale decentralized Q-learning algorithm that has been proposed for two-player zero-sum Markov games [14]. We extend this dynamics to multi-player settings and apply it to stochastic potential games. We observe that the learning dynamics asymptotically converge to a Nash equilibrium of the SPG. To our knowledge, this is the first value-based MARL work on SPGs. We run experiments on two SPG benchmarks, distancing games [8] and network routing games [15], to evaluate the empirical performance of the algorithm.

1.3 Outline

This thesis is organized in the following order. Chapter 2 provides the necessary background and preliminaries in reinforcement learning, game theory, and multi-agent reinforcement learning. Chapter 3 introduces the concept of stochastic potential game and its benchmarks, and discusses previous works on SPGs. Chapter 4 is dedicated to the learning dynamics and its convergence analysis. In Chapter 5, we present our experimental results. Finally, Chapter 6 concludes the thesis with a discussion on game-agnostic learning and possible directions for future works.

Chapter 2

Background and Preliminaries

In this chapter, we provide the introductory notions in reinforcement learning, game theory, and multi-agent reinforcement learning, and discuss the non-stationarity issue in decentralized MARL.

2.1 Single-Agent RL

Broadly speaking, reinforcement learning is a sub-field of machine learning (ML), and can be considered one of the three basic ML paradigms alongside supervised learning and unsupervised learning. In RL, the learner (agent) is not explicitly told what to do, but instead is left out in the wild to discover its environment and maps its states to actions that yield her the maximum cumulative reward overtime. She is highly dependent on trial and error during this process, and also may need to handle an important challenge named distant (delayed) reward, as the agent's actions may not only affect the immediate reward but also the reward values in the future [16].

RL lies between supervised and unsupervised learning. It is different from unsupervised learning which is mostly involved in finding patterns and structures in uncategorized data [17, 18]. It is also different from supervised learning where the model takes labelled data as its input and tries to generalize to recognize and label unlabelled samples [19]. In a complex and interactive environment, it is usually impractical to obtain and label sample actions that are representative enough for different situations that the agent may find herself in. Therefore, in RL, the agent uses the reward signal as its compass, and builds a model to maximize its long-term return, all through policy improvement by trial and error and without access to any labelled data [20]. Another distinct difference between RL and both supervised and

unsupervised learning is the trade-off between exploration and exploitation that is only present in RL. An agent should *exploit* the actions that she knows would yield the maximum cumulative reward with a high certainty, however, in order to find such actions in different situations, she has to *explore* untried actions in order to find them. How to balance exploration and exploitation is a dilemma that has been focused on by mathematicians and computer scientists through years [16]. The importance of RL in machine learning, and in general artificial intelligence (AI), is obvious when many renowned researchers believe that "Reward is Enough" to exhibit almost any ability related to natural and artificial intelligence [21].

After this general introduction, we now mathematically formulate a single-agent RL problem. A Markovian discrete environment, with which an agent interacts, can be modelled through a Markov decision process (MDP) defined below.

Definition 2.1.1 (Markov decision process)**.** A Markov decision process[1] may be described using a tuple of four key elements $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where,

- $\mathcal{S}$ is a set representing the different states in the environment.
- $\mathcal{A}$ is a set representing the possible actions an agent can take.
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a probability mapping[2] that at time step $t \in \mathbb{N}$ gives the transition probabilities that the agent taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ would go to the next state $s' \in \mathcal{S}$ at time step $t + 1$.
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function that returns a bounded scalar in the range $[-R_{max}, R_{max}]$ to the agent as a result of its action a in state s and the transition to s'.

Given an MDP, the goal of the agent interacting with it and following a policy π, is to maximize her expectation of discounted sum of rewards over time,

$$\mathbb{E}_{a_t \sim \pi}\left\{ \sum_{t=0}^{T} \gamma^t R_{a_t}(s_t, s_{t+1}) \right\}, \tag{2.1}$$

where T is the terminating time step and is finite in episodic tasks, or is equal to ∞ in continuing infinite-horizon tasks. $\gamma \in [0,1]$ is a discount factor determining the farsightedness of the agent (a lower value of gamma denotes a more myopic and greedy agent). In continuing tasks, γ should be strictly less than one to ensure

[1]There are some generalizations to MDP to model partially observable environments. For example, in a partially observable Markov decision process (POMDP), the environment dynamics are determined by an MDP, however, the agent can only observe the environment through a set of sensors and maintain a probability distribution of different observations given the current state.

[2]$\Delta(\mathcal{S})$ denotes a probability simplex over the set $\mathcal{S}$.

a finite value for the discounted reward. The policy $\pi : \mathcal{S} \to \Delta(A)$ denotes a probability mapping from each state to the set of possible actions. A policy is deterministic at a given state s, if a single action, a has a probability equal to one, and the other actions have zero probability of being selected. The ultimate goal of the agent is finding an optimal policy that maximizes the cumulative reward in (2.1).

In order to formulate a formal learning scenario, we can define some variables to denote the quality of different states and different actions in those states. Therefore, the state value function of state s under a policy π is defined as

$$V^{\pi}(s) := \mathbb{E}_{a_t \sim \pi}\left\{ \sum_{t=0}^{T} \gamma^t R_{a_t}(s_t, s_{t+1}) | s_0 = s \right\}. \tag{2.2}$$

We may also define the state-action value function, or Q-function under policy π as

$$Q^{\pi}(s, a) := \mathbb{E}_{a_t \sim \pi}\left\{ \sum_{t=0}^{T} \gamma^t R_{a_t}(s_t, s_{t+1}) | s_0 = s, a_0 = a \right\}. \tag{2.3}$$

The Q-function, where Q stands for quality, shows the value of taking a specific action in state s and following policy π afterwards. If we choose action a_0 also according to π, $V^{\pi} = \mathbb{E}^{\pi}\{Q^{\pi}(s, a)\}$. We may write value functions for every state in recursive form that are known as Bellman equations [22]

$$V^{\pi}(s) = \mathbb{E}_{a \sim \pi}\left\{ R_a(s, s') + \gamma \sum_{s' \in \mathcal{S}} V^{\pi}(s') \mathcal{P}(s'|s, a) \right\}, \quad \forall s \in \mathcal{S}. \tag{2.4}$$

The optimal values of these equations are represented using the Bellman optimality equations for V^* [22],

$$V^*(s) = \max_{a \in \mathcal{A}} (R_a(s, s') + \gamma \sum_{s' \in \mathcal{S}} V^*(s') \mathcal{P}(s'|s, a)), \quad \forall s \in \mathcal{S}. \tag{2.5}$$

We may also write the optimal values of Q^* for all state-action pairs using $V*$ as

$$Q^*(s, a) = R_a(s, s') + \gamma \sum_{s' \in \mathcal{S}} V^*(s') \mathcal{P}(s'|s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}. \tag{2.6}$$

The equations in (2.5) constitute a non-linear system whose solution is guaranteed to exist for all finite MDPs [23]. After solving this system of equations and finding $V*$, one may find the actions for which (2.5) yields its maximum. To calculate the optimal policy at each state, we may assign a non-zero probability to these actions, and a zero probability to others. Since there is no difference in value for the optimal actions if they are more than one, the optimal policy can also be a

purely deterministic policy favoring any of the optimal actions. Therefore, it can be said that any finite MDP has a deterministic optimal policy π^*.

As a side note, it should be mentioned that the MDP framework as a model of an RL environment is only valid if the environment possesses the Markov property

$$\mathcal{P}(s_{t+1} = s', r_{t+1} = r|s_t, a_t) = \mathcal{P}(s_{t+1} = s', r_{t+1} = r|s_t, a_t, r_t, ..., s_1, a_1, r_1 s_0, a_0). \quad (2.7)$$

Evidently, this assumption does not hold if one considers an arbitrary observation as the state in a complex environment. However, performing encoding tricks, such as augmenting several observations from consecutive time steps into a single state, may create state representations that include approximately all information needed for decision making. Furthermore, to theoretically prove convergence of many RL algorithms, the Markov assumption is necessary. Assuming that the Markov property holds and we perfectly know the dynamics of the environment, which is not always the case, solving the system of Bellman optimality equations (2.5) analytically for a complex environement with a large number of states is intractable. It requires an exhaustive search and looking ahead every possible trajectory in the MDP to find the optimal action in each state. A faster solution to find the optimal policy in MDPs is dynamic programming (DP) [22]. DP methods solve a complicated problem by breaking it down to smaller sub-problems recursively. The two DP algorithms in RL are policy iteration and value iteration. Although these algorithms find the optimal policy in polynomial time of the number of states and actions in the worst case, they are rarely used in practice because they require a complete model of the environment.

The DP approach is the most basic algorithm in the realm of *model-based RL*. Any RL algorithm that uses a model, whether known (such as Monte Carlo tree search (MCTS)) or learned (such as World Models [24]), to solve the MDP and calculate an optimal value function or policy can be considered model-based [25]. On the other hand, *model-free RL* uses only experience in the environment without explicitly building a model of the MDP to adjust its policy and achieve maximum return. These model-free algorithms are either offline and wait until the end of the episode before updating their estimates (Monte Carlo (MC) methods), or they bootstrap in an online fashion, take advantage of the learned estimates up to now, and form the reward prediction error (RPE) to update the value or action-value estimates without the need to wait for the episode to terminate. This mode of learning, known as temporal difference (TD) learning is useful for tasks with long episodes and continuing tasks. It is worth mentioning that an algorithm may involve both model-free and model-based learning elements. An example of such agents is the renowned AlphaGo [10] that defeated the human world champion of

the ancient board game, Go, in 2016[3].

Model-free RL is usually divided into two main categories; *value-based* methods and *policy-based* methods.

In value-based RL, the agent optimizes its estimation of the action-value (Q) function to obtain the optimal action choice in each state. If the policy being optimized is the same as the one the agent is following, the method is called *on-policy*. State–action–reward–state–action (SARSA) is an example of on-policy value-based algorithms [26, 27]. On the other hand, if the followed policy differs from the policy being optimized, the algorithm is said to be *off-policy*. Tabular Q-learning [28] and deep Q-learning [29] that uses a deep neural network, known as deep Q-network (DQN) instead of a table as the action-value function approximator, are examples of off-policy value-based methods.

In policy-based RL, the objective is to directly optimize the policy (and not the action-value function) using sampled reward values from the environment. For instance, policy gradient (PG) methods optimize a parameterized policy with respect to the expected return by gradient descent [30]. There are also policy-based RL algorithms that do not rely on calculation of gradients, such as the ones using evolutionary strategies or cross-entropy method [31, 32].

It should be noted that many modern model-free RL algorithms do not fit exclusively under one of value-based or policy-based categories. Combining off-policy value-based techniques with policy-based gradient-based strategies, has given rise to a powerful family of model-free RL algorithms known as actor-critic (AC) methods. In AC, optimization of the action-value function guides the agent in improving her policy. After the success of DQN in solving RL tasks with continous state and discrete action spaces, AC methods inspired applying deep learning in tasks with continous action spaces, and in 2015, deep deterministic policy gradient (DDPG) [33] was proposed. Figure 2.1 shows a compact taxonomy of RL algorithms discussed up to now.

Since the decentralized MARL algorithm that we will be considering later on is built upon single-agent Q-learning, at the end of this section, we provide this algorithm and the conditions required for its convergence. Algorithm 1 represents the tabular Q-learning algorithm for discrete state-action spaces.

To state the convergence criteria for Q-learning in finite MDPs, we first introduce an important class of policies:

Definition 2.1.2 (GLIE Policy)**.** A learning policy is greedy in the limit with infinite exploration (GLIE) if it satisfies the following properties:

1. $\lim_{t\to\infty} N_t(s,a) = \infty$ if $\lim_{t\to\infty} N_t(s) = \infty$; all possible actions in a given state are selected infinitely often if the state is visited infinitely often.

[3] `https://www.alphagomovie.com/`

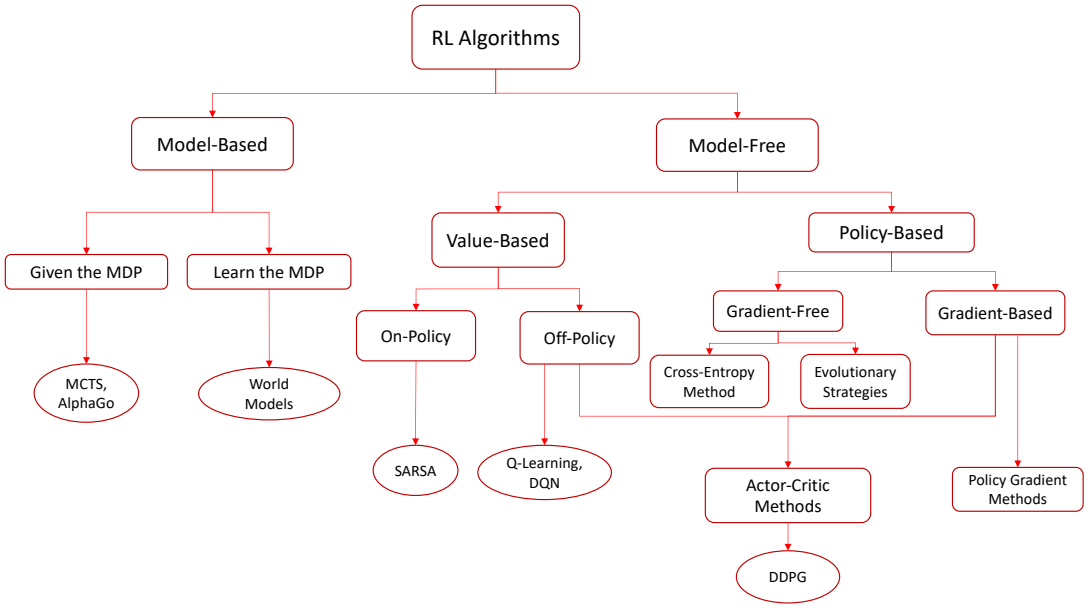


Figure 2.1: Categorization of RL algorithms; reproduced with slight modifications from [34].

Algorithm 1 Tabular Q-learning for Single-Agent RL

Parameters: set of states $\mathcal{S}$, set of actions $\mathcal{A}$, step size sequence $\{\alpha_t \in (0,1]\}$, an exploration probability $\epsilon \in [0,1)$.

Initialize $Q(s,a)$ for all $s \in \mathcal{S}$,$a \in \mathcal{A}$ arbitrarily except that $Q(s_{terminal},.) = 0$.

procedure LOOP FOR EACH EPISODE k:
  Randomly select an initial state s.
  Repeat for each state of the episode:
    Choose action a using a GLIE policy derived from Q, e.g., ϵ-greedy:
    With probability ϵ choose a random action a.
    Otherwise, $a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(s,a)$.
    Take action a, obtain reward a, and observe the next sate s'.

$$Q(s,a) \leftarrow Q(s,a) + \alpha_{t,(s,a)}[r + \gamma \max_{a' \in \mathcal{A}} Q(s',a') - Q(s,a)]. \tag{2.8}$$

    $s \leftarrow s'$.
  Until s is a terminal state.
end procedure

2. As the number of iterations goes to infinity, the policy converges to a greedy

policy with respect to the action-vale (Q) function; $\lim_{t\to\infty} \pi_t = \delta_{\underset{a\in A}{\text{argmax}}(Q_t(s,.))}$, where δ_j is a vector with all components equal to zero except the index corresponding to j which is equal to one.

Following a GLIE policy is an essential condition in the proof of convergence for RL algorithms with bounded estimates. An important policy that could easily satisfy GLIE conditions is the ϵ-greedy policy. In ϵ-greedy, the agent follows a greedy policy with respect to the Q-function with probability $1-\epsilon$, and selects a random action with probability ϵ.

Lemma 2.1.1. *The ϵ-greedy policy satisfies the GLIE conditions if at iteration t, $\epsilon_t = \frac{1}{t}$.*

Now, we can state the convergence criteria for single-agent Q-learning [35, 28]:

Theorem 2.1.1. *Given an MDP with finite state-action space, and a single agent following tabular Q-learning in Algorithm 1, as $t \to \infty$, Q_t converges to Q^*, and π_t converges to π^* given the following conditions,*

1. *The agent follows a GLIE policy.*
2. *The reward values are bounded; $R_{a_t}(s_t, s_{t+1}) \leq D$.*
3. *The learning rate satisfies the following properties as the number of episodes goes to infinity: $\sum_{t=1}^{\infty} \alpha_{t,(s,a)} = \infty$, $\sum_{t=1}^{\infty} \alpha_{t,(s,a)}^2 < \infty, \quad \forall s, a$.*

2.2 Game Theory

Before discussing the concept of multi-agent RL, as it is interrelated with game theory and repeated games, we first give an introductory overview of the main notions in game theory.

Game theory is a branch of mathematics that studies the interaction of multiple decision makers (players) with individual payoff functions whose decisions affect the payoff of other players. In a game, the players try to optimize their respective objective functions, i.e., they are rational, while taking into account the knowledge they have acquired and the expectation they have of the other players' behavior, i.e., they reason and act strategically [36]. Although a relatively new field, GT has been found applications in multiple fields, such as economic, military, and political strategic analysis, social sciences and social networks, robotics and multi-agent systems, and optimization of electrical grids and communication networks [37].

The most basic form of a game is an strategic game in which players choose their actions simultaneously once and for all after analyzing the reward function of

other players and forming an expectation of their behavior. We may define this type of game as below[4]:

Definition 2.2.1 (Strategic Game)**.** A strategic game $\mathcal{G}$ can be defined by a tuple of three sets $(\mathcal{N}, \mathcal{A} = \{\mathcal{A}_i | i \in N\}, U = \{u_i | i \in N\})$, where

- $\mathcal{N} = \{1,2,3,...,n\}$ is a set of n players with their own payoff function to maximize[5].
- $\mathcal{A}_i$ is a set of m_i possible actions that player i could play.
- $u_i : \mathcal{A} \to \mathcal{R}$ is the utility (payoff) function of player i and is dependent on the actions of all players. For the sake of simplicity, we may write $u_i(a_1, a_2, ..., a_n)$ as $u_i(a_i, a_{-i})$, where $a_{-i} = (a_j)_{j \in \mathcal{N}, j \neq i} = (a_1, ..., a_{i-1}, a_{i+1}, ..., a_n)$ is the action profile of all players except player i.

A common way to solve a strategic game when no communication or cooperation among agents is allowed, is to assume that all players are *rational* and try to maximize their payoff function. By acting rationally, the agents may reach an equilibrium point known as a Nash equilibrium (NE) of the game [38],

Definition 2.2.2 (Deterministic Nash equilibrium in strategic games)**.** An action profile $(a_1^*, a_2^*, ..., a_n^*)$ is a deterministic Nash equilibrium of the game if no player can increase her payoff by individually deviating from the profile, i.e.,

$$u_i(a_i^*, a_{-i}^*) \geq u_i(a_i, a_{-i}^*) \quad \forall a_i \in A_i, i \in N. \tag{2.9}$$

We may also define Nash equilibrium using the *best-response (BR)* set,

Definition 2.2.3 (Best-Response Set)**.** The best-response set for player i is the action set resulting from the maximization of u_i with respect to other players' strategies,

$$BR_i(a_{-i}) := \underset{a_i \in A_i}{\operatorname{argmax}}\, u_i(a_i, a_{-i}) \tag{2.10}$$

A Nash equilibrium in a strategic game happens when the action selected by all players falls into their respective best-response set,

$$(a_1^*, a_2^*, ..., a_n^*) \in NE(\mathcal{G}) \iff a_i^* \in BR_i(a_{-i}^*), \quad \forall i \in N \tag{2.11}$$

where $NE(\mathcal{G})$ is the set of deterministic Nash equilibria of the strategic game.

[4]All of the definitions from now on are given for games with discrete action space. Similar definitions can be provided for continous action spaces.

[5]Alternatively, we may also assume that players minimize their individual cost functions.

The best-response set provides us with a method to calculate deterministic Nash equilibria of a game. We may calculate BR for all players and then find the profiles where $a_i^* \in BR_i(a_{-i}^*)$ for all $i \in N$. This also implies that a game may not possess a *deterministic* Nash equilibrium because such action profile may not exist.

Although a game may not possess a deterministic Nash equilibrium, we may turn to another type of policies known as mixed strategies to extend the notion of Nash equilibrium. For player i, we define a mixed strategy over her possible actions as $\pi_i \in \Delta(A_i)$. In other words, π_i is a probability distribution over the possible actions of player i, and $\pi_i[a^j]$ is the probability of choosing action a^j in the game. Consequently, we may define the expected utility tensor of player i following a mixed strategy as,

$$U_i(\pi^i, \pi^{-i}) = (\pi^i)^T u^i \prod_{j \in N, j \neq i} \pi^j \tag{2.12}$$

where u^i is the tensor of the utilities corresponding to deterministic action profiles. Jon Nash proved the following theorem for *finite games* [39],

Theorem 2.2.1 (Existence of mixed NE in finite games)**.** *Every game with a finite set of players and finite strategy sets, admits at least one mixed Nash equilibrium profile* (π_*^i, π_*^{-i}) *where,*

$$U_i(\pi_*^i, \pi_*^{-i}) > U_i(\pi^i, \pi_*^{-i}), \quad \forall i, \pi^i \tag{2.13}$$

Comment. In this thesis, the terms player, utility/payoff, strategy, and learning dynamics are respectively equivalent to the terms agent, reward, policy, and learning algorithm, and are used interchangably. The first set of terms are more common in the game theory and learning in games literature, while the second set is usually used in ML/RL literature.

It should be noted that the concept of Nash equilibrium is defined in the context of non-cooperative games where agents do not cooperate with each other and are completely selfish. In such a setting, there may exist other action profiles where both agents can gain more than the Nash equilibrium of the game, yet they cannot reach that profile without cooperation. A famous example of such a game is the two-player prisoner's dilemma (Table 2.1) where the green payoff cell is not achievable when following the rationality principle because each agent has an individual motivation to change her strategy and gain the payoff equal to 4, and the only Nash equilibrium in this setting is the red cell.

We now introduce two important classes of games that have also been studied in the context of repeated games and multi-agent RL; zero-sum games, and potential games [40].

Definition 2.2.4 (Two-player zero-sum game)**.** A two-player game is zero-sum if

Table 2.1: Two-player prisoner's dilemma

P2 P1	Confess	Don't confess
Confess	**3, 3**	0, 4
Don't confess	4, 0	**1, 1**

the sum of two players' payoffs for each possible action profile is equal to zero[6],

$$u_1(a_1, a_2) + u_2(a_1, a_2) = 0, \quad \forall (a_1, a_2) \in A_1 \times A_2 \tag{2.14}$$

Definition 2.2.5 (Potential game)**.** A game $(N, A = \{\mathcal{A}_i\}, U = \{u_i\})$ is (exact) potential, if there exists a potential function, $\Phi : \mathcal{A} \to \mathbb{R}$, s.t.,

$$u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i}) = \Phi(a_i, a_{-i}) - \Phi(a'_i, a_{-i}), \quad \forall i \in N \tag{2.15}$$

In a potential game, the change in the utility of player i, when changing her action unilaterally is proportional to the variation in the potential function. As discussed in chapter 1, potential games are capable of modeling several practical and real-world scenarios. Because the incentive of all players are mapped into a single function, the pure Nash equilibria of the game correspond to the local minima of the potential function. This is an important property that is used in studying convergence of learning dynamics to Nash equilibria in repeated games. Indeed, in the real world, games are not normally played once, and players have to play a game with *partial observability* repeatedly and learn an optimal strategy. Partial observability implies a player may not be fully-rational, she may not be aware of her reward function and even may not be able to observe other players' actions and rewards during the game. To analyze the game dynamics and the equilibria that the game may converge to in the long run in such settings, we can turn to algorithms designed for learning strategies in repeated games [41]. When players adopt a learning strategy, they may not necessarily converge to a Nash equilibrium, or they may converge to an approximate Nash equilibrium when the number of iterations of the game is limited. Therefore, we may define an $\epsilon-$Nash equilibrium as

Theorem 2.2.2 ($\epsilon-$Nash equilibrium)**.** *A mixed strategy profile (π^i_*, π^{-i}_*) is an ϵ-Nash equilibrium if*

$$U_i(\pi^i_*, \pi^{-i}_*) \geq U_i(\pi^i, \pi^{-i}_*) - \epsilon, \quad \forall i, \pi^i, \tag{2.16}$$

[6]This class of games can be considered the purest form of non-cooperative games as there is absolutely no incentive for the players to cooperate with each other [37].

and a Nash equilibrium happens when ϵ is zero.

An important algorithm for learning in repeated games is fictitious play [42]. In fictitious play, each player has full observability over her own reward function. However, she can only observe the actions of other players at each iteration, and not the reward they get. Each player assumes that other players follow a stationary mixed strategy, and at each round, she plays her best response to the empirical frequency of the previous actions played by her opponents. In other words, at each iteration, she gives a weight equal to the frequency of actions played by other players. Therefore, the the probability player i assigns to player $-i$ for playing a_j^{-i} will be,

$$\gamma_t^i(a_j^{-i}) = \frac{\eta_t^i(a_j^{-i})}{\sum_{a_k^{-i} \in A^{-i}} \eta_t^i(a_k^{-i})} \tag{2.17}$$

where $\eta_t^i(a_k^{-i})$ is the number of times that player $-i$ has played a^{-i} up to iteration t. η^i can be considered a mixed strategy profile for player $-i$. At each round, player i plays her best response with respect to the mixed utility calculated over η^i,

$$BR_i(\eta^i) := \underset{a_j \in A_i}{\operatorname{argmax}}\, U_i(\delta_j, \eta^i) \tag{2.18}$$

where δ_j is a vector with all components equal to zero except the index corresponding to j which is equal to one.

Fictitious play has been proven to converge to a pure Nash equilibrium in many classes of games, including any two-player game with a generic payoff matrix [43] and potential games with arbitrary number of actions and players [40].

Since deterministic best response in fictitious play may change abruptly (discretely), in case of a mixed Nash equilibrium, the actual behavior overtime may not converge to the equilibrium, even though the beliefs converge to one [44]. To avoid such a problem, stochastic fictitious play was introduced by Fudenberg and Kreps [45], in which the deterministic BR is substituted by perturbed BR dynamics. In perturbed BR, the player payoffs are perturbed with a random noise at each iteration. Perturbed BR is proven to converge to a pure Nash equilibrium in zero-sum, potential, and supermodular games as the noise vector corresponding to each player's payoff becomes sufficiently small [44].

2.3 Multi-agent RL

Unlike single-agent RL where the dynamics of the environement was only determined by the actions of a single player, in MARL, the evolution of the environment and the reward that each agent receives is dependent not only on her action, but on the actions of all players. Apart from this difference, each agent is again assumed

to solve a decision-making problem sequentially, and through trial and error. Due to the involvement of multiple decision makers, it is reasonable to model a MARL problem using a game-theoretic framework, known as a stochastic (Markov) game [46, 47]:

Definition 2.3.1 (Stochastic (Markov) game)**.** A stochastic (Markov) game[7] is defined by extending the definition of MDP to a multi-player setting, and therfore can be described using a tuple of five key elements $(N, \mathcal{S}, \mathcal{A} = \{\mathcal{A}_i | i \in N\}, \mathcal{P}, R = \{R_i | i \in \mathcal{N}\})$, where,

- $\mathcal{N} = \{1,2,3, ..., n\}$ is a set of n players.
- $\mathcal{S}$ is a set representing the different states of the game, and is shared by all players.
- $\mathcal{A}_i$ is a set of m_i possible actions that player i could play at each state.
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a probability mapping that at time step $t \in \mathbb{N}$ gives the transition probabilities of the agents going from state $s \in \mathcal{S}$ to the next state $s' \in \mathcal{S}$ at time step $t + 1$ given the action profile $(a_1, ..., a_n) \in \mathcal{A}$.
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to R$ is the reward function that returns a bounded scalar in the range $[-R_{max}, R_{max}]$ to each agent as a result of the action profile $(a_1, ..., a_n)$ taken in state s and the transition to s'.

Based on this definition, every state of a stochastic game (SG) can be viewed as a normal-form game with the important caveat that after taking actions, the agents may transit to a different state based on the transition probability matrix.

There are generally two different paradigms when approaching an MARL problem. The first one is coordinated (centralized) MARL in which a central coordinator has full observability over the joint actions, rewards, and observations of all agents, and it coordinates their behavior in order to optimize their individual policy and achieve an equilibrium profile. This setting requires either a simulation of the game or an express two-way connection between the agents and the controller, and considering the task of the controller, it usually needs to have a high computational capacity that grows with the number of agents. Up to now, the centralized paradigm has been the focus of many seminal works in MARL [11, 10, 48, 49].

Centralized MARL deprives agents of their autonomy and competitive drive that are important factors in non-cooperative games. Indeed, in many practical

[7]It is important to distinguish between stochastic games where agents make their decisions simultaneously, and extensive-form or tree games where agents take their actions sequentially after observing the actions of previous players.

multi-agent scenarios, such as decision making in economics [50], the nature of the game and the competition present, dictates that the agents take their actions independently and without coordination. Furthermore, the scale and complexity of a game that increases with the number of players, e.g., in large-scale robot systems [51], motivates the use of distributed algorithms. The second paradigm in MARL is decentralized and based on independent decision making of individual agents. The main benefits of this paradigm are better scalability of the algorithm, and a reduction in computational cost, both of which are the results of eliminating the central controller. Furthermore, users may not be inclined to share their personal data with a coordinator server, and therefore, decentralized techniques may also be beneficial in terms of privacy. Before discussing the existing algorithms and methods in decentralized MARL, we discuss an important challenge that any MARL algorithm needs to address.

Unlike single-agent RL where the evolution of the environment is only a function of the single agent's behavior, in MARL, the joint policy of all participating agents shape the environment dynamics. Coupling this joint dependence with the fact that agents do not have access to other agents' policies gives rise to the *non-stationarity* challenge in MARL. From a single agent's prespective, the environment dynamics is dependent on hidden and observable variables which makes independent policy optimization and decision making difficult. Adopting the centralized paradigm is an evident way to deal with non-stationarity [52, 53, 49]. Therefore, in the next section, we examine the methods that deal with this challenge in the decentralized paradigm.

2.4 Dealing with Non-stationarity in Decentralized MARL

Hernandez-Leal et al. [12] provides an interesting categorization for the techniques that decentralized MARL algorithms use to deal with non-stationarity. The first technique is ignoring the non-stationarity of the environment, and acting as if the opponents are playing a stationary strategy. This approach may easily fail to capture an optimal strategy if the opponents alter their policies. In the second category, called *forget*, the agents continuously update their value estimates or policies based on recent observations, and therefore, forget about the information in the past. The third and fourth categories rely on communication channels between agents, or a higher degree of observability, to respond to the respond to target opponents [54, 55], or learn models of the opponents [56, 57]. The last category is based on theory of mind and recursive reasoning where the agent assumes that the opponent is strategically modeling her and creates her best response set based on recursive reasoning [58, 59].

Our area of interest in this thesis is the second category, *forget*. These algorithms are mostly model-free as they are usually based on iterative updates of value and Q functions, and policies. One approach that these algorithms use to deal with non-stationarity is asymmetric and variable step sizes for different agents to continuously adjust to opponents' behavior [60, 61, 62]. Another approach is two-timescale learning [63, 14] to create periodic local stationary environments for agents. We particularly focus on this approach and the one proposed by Sayin et al. [14] in decentralized Q-learning dynamics. In this dynamics, in the faster timescale, the environement can be assumed to be *locally* stationary from an agent's prespective, while it changes in the background with the slow timescale updates.

Chapter 3

Stochastic Potential Games

A large portion of works that approach MARL from a game theoretical prespective, have focused on either zero-sum [64, 14, 65, 62], or fully-cooperative games [66, 67, 68, 69, 70], and the study of other types of games have been limited. However, as discussed in Section 1.1, an important family of games that connects cooperation and competition, and appears in numerous real-world scenarios, is potential games. Furthermore, every general-sum game can be decomposed into three games, a non-strategic game, a potential game, and a harmonic game [71]. Therefore, studying MARL algorithms for stochastic potential games can be a step towards finding algorithms for arbitrary general-sum games.

3.1 Related Works and Definitions

Stochastic (also known as dynamic or Markov) potential games were first studied outside the MARL framework and assuming that the dynamics of the environement are known, either with a centralized controller [72, 73, 74], or in a decentralized fashion [75]. Recently, the study of SPGs from an RL prespective has become more prevalent. Extending the policy gradient method [76] to a multi-player setting, Leonardos et al. [8] and Zhang et al. [9] proved the last iterate convergence of independent policy gradient (IPG) in SPGs. Fox et al. [77] showed the convergence of independent natural policy gradient (INPG), and experimentally showed its faster convergence compared to IPG. These three works define the SPG, and its corresponding potential function, based on the state value function of the agents. On the other hand, Mguni et al. [7] provides an alternative definition based on reward functions in each state of the SG, and extend the game to continuous state and action spaces. They propose two algorithms for finding the Nash equilibrium in SPGs. The first one, SPotQ, is based on Q-learning and requires the potential function to be known. The second one, SPot-AC, is an actor-critic algorithm that

needs to learn the potential function using minibatch data samples. Therefore, in both algorithms, the agents should be aware that the game is potential and either have to know or estimate the potential function. To best of our knowledge, up to know there has been no work adopting a pure value-based MARL approach in SPGs. In this thesis we consider the SPG formulation of Mguni et al. [7] for discrete action spaces, and study such a value-based dynamics to bridge this gap. An SPG is defined as:

Definition 3.1.1 (Stochastic potential game)**.** A stochastic game is potential, if there exists a potential function $\Phi : \mathcal{S} \times \mathcal{A} \to R$ such that for every player $i \in \mathcal{N}$, every state $s \in \mathcal{S}$, and action profiles $(a_i, a_{-i}), (a'_i, a_{-i}) \in \mathcal{A}$, the following holds,

$$r_i(s, a_i, a_{-i}) - r_i(s, a'_i, a_{-i}) = \Phi(s, a_i, a_{-i}) - \Phi(s, a'_i, a_{-i}). \tag{3.1}$$

An intuitive condition that holds in many SPGs is state transitivity:

Definition 3.1.2 (State transitivity)**.** An SPG is said to be state transitive if for every player $i \in \mathcal{N}$, every action profile $(a_i, a_{-i}) \in \mathcal{A}$, and every state transition from s to s', the following holds,

$$r_i(s, a_i, a_{-i}) - r_i(s', a_i, a_{-i}) = \Phi(s, a_i, a_{-i}) - \Phi(s', a_i, a_{-i}). \tag{3.2}$$

In other words, when transiting to another state, the difference in rewards is the same for all agents. This assumption holds in many real-world applications of SPGs, such as network routing congestion games [15] and distancing games [8]. Similar to Mguni et al. [7], in our analysis, we also assume that state transitivity condition is satisfied.

3.2 Benchmarks

We consider two SPG benchmarks for experimental evaluation; network routing games, and distancing games. It is worth stating that the SPG literature is relatively young, and therefore, there are only few reported experimental results on these benchmarks.

3.2.1 Network Routing Games

Network Routing Games A network routing game is defined over an acyclic directed network of nodes, through which self-interested agents transfer their commodities. Passing through every edge has a cost that is proportional to the number of agents that are using the edge simultaneously. We consider an atomic routing game [15] that involves n agents who are transporting their commodities from a source node

to a destination node. The atomic term implies that the commodity of each agent cannot be divided into smaller parts and should be transferred along a single edge. Figure 3.1 shows two consecutive states of an atomic routing game with four agents. If we do not impose any restriction on the simultaneity of the agents' movement through the network, we will have a total of $|V|^n$ states in the MDP that constitutes the underlying SPG, where $|V|$ is the number of nodes in the network. In other words any configuration of the agents on different nodes is a state in the SPG. Every edge has a cost function $c_e(.)$, and therfore, each state of this SPG is a congestion game, and the potential function function is given by,

$$\Phi(s, a^i, a^{-i}) = \sum_e \sum_{k=1}^{n_e} c(k) \tag{3.3}$$

where e is any edge that can be chosen as action by players in state s, and n_e is the number of players who choose edge e.

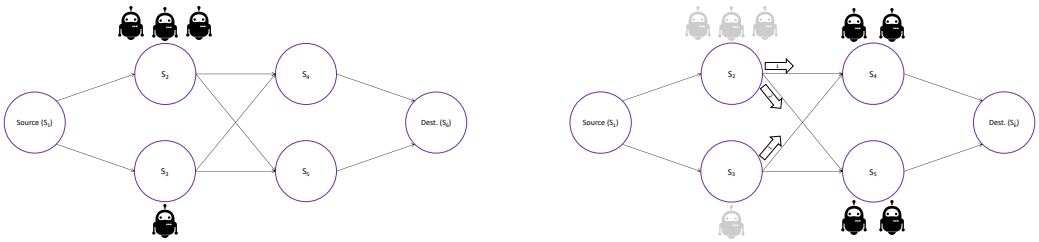


Figure 3.1: Two consecutive states in an atomic network routing game

3.2.2 Distancing Game

Distancing game [77] is an SPG environement that was introduced by Leonardos et al. [8], and consists of two states, both of which are congestion games. Each state has $m = 4$ *facilities* from which agents have to choose to stay when they are in that state. The first state is a safe state, and agents will receive a positive reward proportional to the value (weight) of their chosen facility and the number of agents in that facility. The weight of the facilities $\{A, B, C, D\}$ have the property $w_D > w_C > w_B > w_A$. Whenever there are more than $N/2$ number of agents in a single facility in the safe state, the agents will transit to the second state, *spread* state. In this state, the agents receive a reward consisting of the same positive reward as the safe state, but with a large negative constant added to it. In order to transit back to the safe state, agents have to spread throughout the facilities as much as possible; if there are more than two $N/4$ players in the same facility, the

agents stay in this harmful spread state until they spread and move back to the safe state. The potential function this game for each state is:

$$\Phi(s_{safe}, a^i, a^{-i}) = \sum_f \sum_{k=1}^{n_f} k w_f, \tag{3.4a}$$

$$\Phi(s_{spread}, a^i, a^{-i}) = \sum_f \sum_{k=1}^{n_f} k(w_f - c), \tag{3.4b}$$

where f is one of the facilities, and n_f is the number of agents choosing facility f, and c is the constant penalty term. Figure 3.2 shows a configuration of $n = 8$ agents in the safe state.

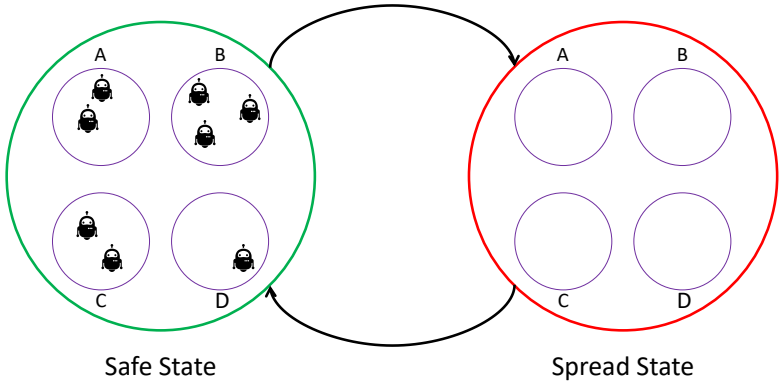


Figure 3.2: Distancing game

Chapter 4

Learning Dynamics and Convergence Analysis

In this chapter, we outline the preliminary definitions and notations, and afterwards, present the decentralized Q-learning two-timestep algorithm for n-player stochastic games. Decentralized Q-learning was first introduced for two-player zero-sum stochastic games by Sayin et al. [64, 78]. Afterwards, we analyze the asymptotic convergence of this algorithm in stochastic potential games.

4.1 Definitions and Notations

In a given SG (Definition 2.3.1), where player i follows a stochastic policy $\pi^i \in \Delta(A^i)$, we denote the joint policy profile as $\pi = (\pi^1, \pi^2, ..., \pi^n) \in \Delta(A)$ and define the expected utility of player i when playing the SG over an infinite horizon as

$$U^i(\pi^i, \pi^{-i}) = \mathbb{E}_{(a_t^i, a_t^{-i}) \sim \pi} \left\{ \sum_{t=0}^{\infty} \gamma^t r_{s_t}^i(a^i, a^{-i}) \right\}. \tag{4.1}$$

where s_0 is drawn from a probability distribution $p_{s_0} \in \Delta(s)$, and the consecutive states are determined by the transition mapping $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$. An ϵ-Nash equilibrium can be defined for an SG similar to a normal-form game as a strategy profile $\pi_* = (\pi_*^1, \pi_*^2, ..., \pi_*^n)$ if the following holds,

$$U_i(\pi_*^i, \pi_*^{-i}) \geq U_i(\pi^i, \pi_*^{-i}) - \epsilon, \quad \forall i, \pi^i \in \Delta(A^i), \tag{4.2}$$

where $\epsilon = 0$ results in a Nash equilibrium. It is proven that every finite SG possesses at least one Nash equilibrium [79]. To attribute a value to a given state from each agent's prespective, we define the value function of player i in state s as

$$v_\pi^i(s) = \mathbb{E}_{(a^i,a^{-i})\sim\pi}\left\{ r_s^i(a^i,a^{-i}) + \gamma \sum_{s'\in S} v_\pi^i(s')p(s'|s,a^i,a^{-i}) \right\}. \tag{4.3}$$

It follows that $U_i(\pi^i,\pi^{-i}) = \mathbb{E}_{s_0\sim p_{s_0}}\{v(s_0)\}$. Furthermore, the Q-function for state s and a given action profile (a^i,a^{-i}) is defined as

$$Q_\pi^i(s,a^i,a^{-i}) = r_s^i(a^i,a^{-i}) + \gamma \sum_{s'\in S} v_\pi^i(s')p(s'|s,a^i,a^{-i}). \tag{4.4}$$

This function can be called the *global* Q-function because it is a function of all players' actions and can be obtained only by full-observability over the game. The global Q-function at state s can be interpreted as the utility function of player i corresponding to that state over the iterations of the stochastic game. However, there are two obstacles in the way of using algorithms designed for repeated games, such as fictitious play, over this global Q-function; first, the Q-function is not constant and evolves alongside the value function and the policy of other players, and second, even if we could consider Q constant, in decentralized learning, the agents would not have access to the actions of other players, and therefore, cannot calculate the Q-function. Both of these challenges can be addressed by employing the two-timestep decentralized Q-learning algorithm. In this algorithm, the agents employ their state value estimations, that have become *locally stationary* in a slow timescale, to approximate the expected quality of a given action with respect to the other players' strategy. This is done by introducing the *local* Q-function for player i as

$$q_\pi^i(s,a^i) = \mathbb{E}_{a^{-i}\sim\pi^{-i}}\left\{ Q_\pi^i(s,a^i,a^{-i}) \right\} \tag{4.5}$$

4.2 Decentralized Q-Learning

The decentralized Q-learning is based on a two-timescale update rule. In the faster timescale, agents assume that the environement is stationary, which is equivalent to assuming that the value function is constant and as if they are playing a single-state normal-form game. Consequently, they update their estimates of the local Q-function accordingly (Q-**update**). In the slower timescale, the agents apply the knowledge that they have gained from their observations since the previous slow timescale update and update their estimates of the value function to make the stage ready for another sequence of fast timescale updates with assumed stationarity. The algorithm has been designed in a way that each agent may be oblivious to the nature of the game or even the presence of other players.

Before presenting the algorithm, we define the type of policy that the agents follow in the algorithm. Perturbed best-response (also known as smoothed or noisy

best-response), that was mentioned in Section 2.2, can be defined for player i in general terms as follows

$$Br(q^i) = \underset{\mu \in \Delta(A)}{\operatorname{argmax}} \left\{ \mu q^i - \nu(\mu) \right\} \tag{4.6}$$

where μ is a probability distribution (stochastic policy) over A, and ν is a perturbation (noise) function, and $\mu \cdot q^i$ denotes the inner product of vectors μ and q^i. If the perturbation function ν is smooth, strictly concave, and with an unbounded gradient at the boundary of $\Delta(A)$, (4.6) is shown to have a unique maximizer. Choosing ν as the negative entropy function with a noise level (temperature) τ.

$$\nu(\mu, \tau) = \tau \sum_{a_j \in A} \mu_j \log(\mu_j) \tag{4.7}$$

will result in a logit choice (softmax) BR function,

$$Br(q^i, \tau) = \frac{\exp(q^i/\tau)}{\sum_{a_j \in A} \exp(q^i_j/\tau)} \tag{4.8}$$

Algorithm 2 presents the two-timescale decentralized Q-learning dynamics for a SG with n players who follow a noisy Br policy. As can be seen in the algorithm, the step sizes are normalized at every time step [14, 80, 41]. As will be seen in the convergence proof, the normalization addresses the asynchronous updates of the local Q-function entries. With this procedure, although different actions in a given state may not be chosen with the same frequency, they are updates at the same rate in the expectation.

The following assumptions on step sizes and the temperature parameter are needed for the dynamics to converge.

Assumption 1. *The step size sequences* $\{\alpha_t \in (0,1) | t \in \mathbb{Z}\}$ *and* $\{\beta_t \in (0,1) | t \in \mathbb{Z}\}$ *are non-increasing and satisfy*

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \quad \sum_{t=1}^{\infty} \beta_t = \infty, \quad \lim_{t \to \infty} \alpha_t = \lim_{t \to \infty} \beta_t = 0 \tag{4.12}$$

This assumption, similar to the one used in tabular Q-learning (Theorem 2.1.1) is common in proving convergence of iterative algorithms using stochastic approximation theory [81, 82].

Assumption 2. *Given* $J \in (0,1)$*, a polynomial function* $P(.)$*, that may depend on* J*, exists such that for any factor* $\lambda \in (0,1)$*, if the set* $\{k \in \mathbb{Z}_+ | k \leq c, \frac{\beta_k}{\alpha_c} > \lambda\}$ *is not empty, we will have*

$$\max \left\{ l \in \mathbb{Z}_+ | l \leq c, \quad and \quad \frac{\beta_l}{\alpha_c} > \lambda \right\} \leq Jc, \quad \forall c \geq P(\lambda^{-1}) \tag{4.13}$$

Algorithm 2 Decentralized Q-learning Dynamics

Preliminary comments:
$q^i_{s,t}$ denotes the local Q-values of player i in state s and time step t, and is a vector with $|A_i|$ elements. $q^i_{s,t}[j]$ denotes the j^{th} entry this vector.
$v^i_{s,t}$ is a scalar denoting the value of state s from the prespective of player i at time step t.
$\#s$ is the number of visits to state s up to now. This value is used to update the step sizes and temperature parameter overtime.
$\alpha_{\#s}$ is the smaller step size (corresponding to the faster timescale) for updating local Q-functions.
$\beta_{\#s}$ is the larger step size (corresponding to the slower timescale) for updating state value functions.
$\tau_{\#s}$ is the temperature parameter of the noisy best response policy.
α, β and τ are calculated at every visit to a state and all of them are inversely proportional to the number of visits to that state ($\#s$), and therefore decay overtime.

Require: Keep track of $\{q^i_{s,t}, v^i_{s,t}, \#s\}$.

1: **procedure** SEPARATELY FOR EACH PLAYER i, RUN THE FOLLOWING DYNAMICS FOR A PRE-DETERMINED NUMBER OF ITERATIONS OR UNTIL A CONVERGENCE CRITERION IS MET.

2: Observe the current state s_t and receive the reward r^i_{t-1} for the previous action a^i_{t-1} performed at state s_{t-1}.

3: Calculate the step sizes $\alpha_{\#s}$, $\beta_{\#s}$ and the temperature parameter $\tau_{\#s}$ based on the number of visits to s_t up to now.

4: Normalize the step sizes $\bar{\alpha}^i_{t-1} = \min(1, \alpha_{\#s})$, $\bar{\beta}^i_{t-1} = \min(1, \beta_{\#s})$ to have expected synchronous updates of the Q-function entries.

5: Update the entry of the local Q-function corresponding to the chosen action a^i_{t-1} (other entries and other states' estimates will not change):

$$q^i_{s_{t-1},t}[a^i_{t-1}] = q^i_{s_{t-1},t}[a^i_{t-1}] + \alpha_{\#s_{t-1}}(r^i_{t-1} + \gamma(v^i_{s_{t-1},t-1}) - q^i_{s_{t-1},t}[a^i_{t-1}]) \tag{4.9}$$

6: Construct the noisy best response policy:

$$\pi^i_t = Br(q^i_{s_t}, \tau_{\#s_t}) \tag{4.10}$$

7: Update the value function estimate of state s (the value of other states will not change):

$$v^i_{s_t,t+1} = v^i_{s_t,t} + \beta_{\#s_t}(\pi^i_t \cdot q^i_{s_t,t} - v^i_{s_t,t}) \tag{4.11}$$

8: **end procedure**

The above assumption ensures two-timescale learning but it is stronger than $\lim_{t\to\infty} \frac{\beta_t}{\alpha_t}$ to ensure that when states are not visited with the same rate, the fast timescale updates of states with less frequent visits do not lag behind and interfere with two-timescale learning.

Assumption 3. *For any infinite sequence of actions, the environement dynamics are such that we may reach from any state s to any other state s' with a positive probability and within a finite number of iterations n.*

Assumption 4. *The temperature sequence $\tau_{t>0}$ is non-increasing and satisfies*

$$\lim_{t\to\infty} \frac{(\tau_{t+1} - \tau_t)}{\alpha_t} = 0, \quad \textit{and} \quad \lim_{t\to\infty} \tau_t = 0 \tag{4.14}$$

Furthermore, the step size sequence $\{\alpha_t\}_{t>0}$ satisfies $\sum_{t=1}^{\infty} \alpha_t^{2-\rho} < \infty$, for some $\rho \in (0,1)$, and there exists $T, T' \in (0,\infty)$ such that $\alpha_t^{\rho} \exp(4D/\tau_t) \leq T'$ for all $t \geq T$.

These two assumptions ensure that states are visited infinitely often, so that the random noise in the perturbed BR vanishes and the policy becomes deterministic as the number of iterations approach infinity.

Now, we state the main theorem for convergence of decentralized Q-learning in SPGs.

Conjecture 4.2.1. Given a SPG with n players, if all agents follow the dynamics in Algorithm 2, and Assumptions 1 to 4 hold, the agents asymptotically converge to a pure Nash equilibrium policy of the SPG.

An idea for the potential proof of this conjecture is given in Appendix A. The first part of the given analysis is similar to Sayin et al. [14] as the dynamics of a single state are decoupled from other states, yet it is reformulated for a game with n players instead of two. The following limiting ODE is extracted for the decoupled dynamics

$$\frac{dq_s^i(t)}{dt} = Q_s^i Br^{-i}(q_s^{-i}(t), \tau(t)) - q^i(t) \quad \forall (i,s) \in N \times \mathcal{S}, \tag{4.15a}$$

$$\frac{dv_s^i(t)}{dt} = 0, \quad \forall (i,s) \in N \times \mathcal{S}, \tag{4.15b}$$

$$\frac{d\tau(t)}{dt} = 0, \quad \forall s \in \mathcal{S}, \tag{4.15c}$$

with the following simplification of notation

$$\begin{aligned} &Q_s^i Br^{-i}(q_s^{-i}(t), \tau(t)) \\ &:= Q_s^i \overline{Br}^1(q_s^1(t), \tau(t)) ... \overline{Br}^{i-1}(q_s^{i-1}(t), \tau(t)) \overline{Br}^{i+1}(q_s^{i+1}(t), \tau(t)) ... \overline{Br}^n(q_s^n(t), \tau(t)). \end{aligned} \tag{4.16}$$

Afterwards, we associate this ODE to the stochastic fictitious play dynamics [44] by quantifying the difference error between them in order establish that the learning dynamics converges to the same point as the fictitious play dynamics, which is a Nash equilibrium, as this error and the temperature sequence converges to zero. At the end of Appendix A, we also prove the convergence to a Nash equilibrium under an extra assumption.

Chapter 5

Experimental Results

In this chapter, we first consider two preliminary SPGs with a single state and analyze the behavior of the learning dynamics and the limiting ODE (4.15a). After that, we provide the experimental results obtained on the two SPG benchmarks, network routing games, and distancing games. All codes related to experiments in this chapter are publicly available online[1].

5.1 ODE Simulation

In order to intuitively grasp the evolution of the dynamics, we simulate the limiting ODE (4.15a) in two single-state potential games. Consider the two-player normal-form game in Table 5.1. Considering, we simulated the ODE (4.15a) using $dt = 10^{-6}$, $\tau = 0.01$, and $\gamma = 0.5$. As the state value functions are constant in (4.15b), they are set to zero, so that the Q function would be equal to the reward function. evolution of the local Q-functions for both players with two different initial conditions are plotted in Figure 5.1.

Table 5.1: A single-state potential game with two players and two actions alongside the potential function

P2 P1	a1	a2
a1	0.2, -0.3	**0.4, 0.1**
a2	**0.5, 0.2**	0.2, 0.1

Φ

0.1	**0.5**
0.4	0.3

[1] `https://github.com/hafezgh/PoliTo-MSc-Thesis`

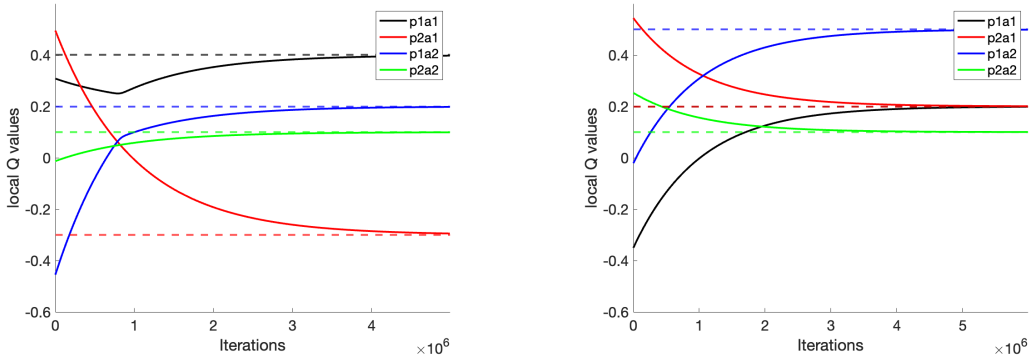


Figure 5.1: The ODE simulation for the two-player normal-form potential game in Table 5.1 with two different random initializations.

It can be seen that the local Q-functions converge to the global Q-values corresponding to a Nash equilibrium. As supported by theory, this equilibrium could be any of the possible Nash equilibria of the game, and not necessarily the one corresponding to a strict Nash equilibrium or the maximum of the potential function.

Figure 5.2 presents similar simulation results for the game in Table 5.2.

Table 5.2: A single-state potential game with two players and four actions alongside the potential function

P2 P1	a1	a2	a3	a4
a1	0.1,-0.4	0.3,-0.2	**0.4,0.3**	-0.1,-0.5
a2	0.5,0.2	**0.7, 0.4**	0.2,0.3	0.1,-0.1
a3	0.2,-0.2	0.3,-0.1	0.0,0.0	**0.4,0.1**
a4	**0.6, 0.9**	0.4,0.7	0.1,0.8	0.0, 0.4

Φ

0.2	0.4	**0.9**	0.1
0.6	**0.8**	0.7	0.3
0.3	0.4	0.5	**0.6**
0.7	0.5	0.6	0.2

Again, the local Q-functions converge to the global Q-values corresponding to a Nash equilibrium on the game depending on the initialization.

5.2 Single-state Potential Games

In this section, we run the dynamics for the two games in Table 5.1 and Table 5.2. For this experiment and all the folloeing experiments, we use step sizes α_t and β_t, and the temperature parameter τ_t with the following expressions and conditions

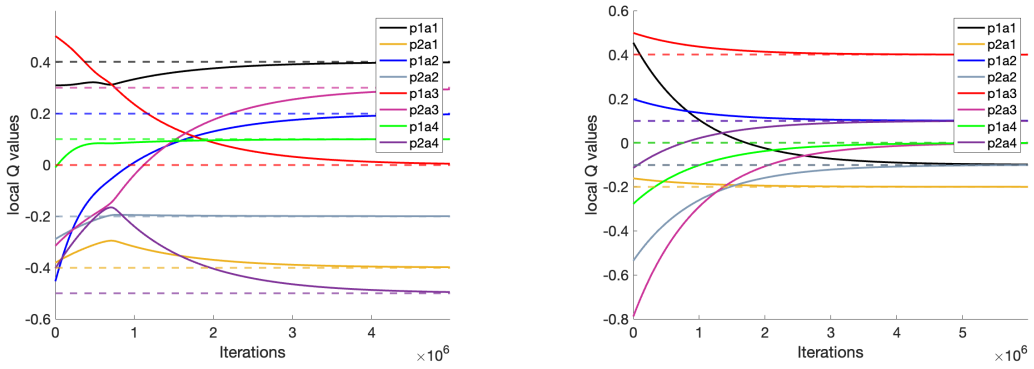


Figure 5.2: The ODE simulation for the four-player normal-form potential game in Table 5.2 with two different random initializations.

that have been shown to satisfy Assumptions 1, 2, and 4 [14]:

$$\alpha_t = t^{-\rho_\alpha}, \tag{5.1a}$$

$$\beta_t = t^{-\rho_\beta}, \tag{5.1b}$$

$$\tau'_t = \bar{\tau}\Big(1 + \bar{\tau}\frac{\rho_\alpha \rho}{4D}\log(c)\Big)^{-1}, \tag{5.1c}$$

where $0.5 < \rho_\alpha < \rho_\beta \leq 1$, $\rho \in (0, 2 - 1/\rho_\alpha)$, and $\bar{\tau} > 0$.

Figures 5.1 and 5.1 show the results of two runs corresponding to the games in Table 5.1 and Table 5.2, respectively. The actions corresponding to the maximum of local Q-values in both cases correspond to a Nash equilibrium of the game ($(a1, a2)$, and $(a3, a4)$ respectively).

5.3 Distancing Games

In this section, we report our results on an example of the distancing game benchmark introduced in Section 3.2.2. The game is played with $n = 8$ players, and $m = 4$ facilities in each state. Therefore, if more than 4 players choose the same facility in the safe state, the agents transit to the spread state. In order to go back to the safe state, exactly 2 agents should choose each facility in the spread state. The optimum strategies in these two states are visualized in Figure 5.5.

In order to run the decentralized Q-learning dynamics on the distancing game, we consider as the convergence criterion the L1 accuracy metric, which is the average distance between the current policy and final perturbed BR policy of the agents:

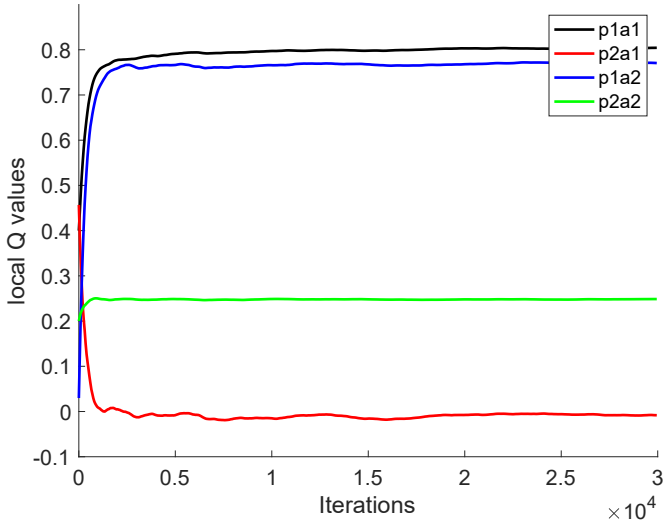


Figure 5.3: An individual run of the decentralized Q-learning dynamics for the potential game in Table 5.1

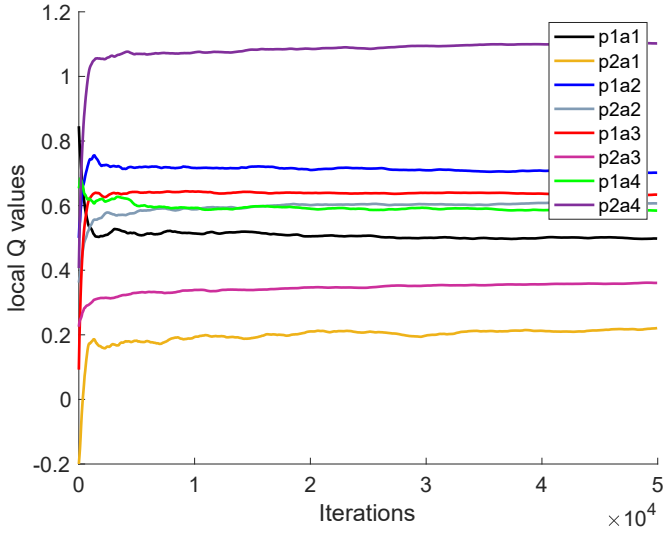
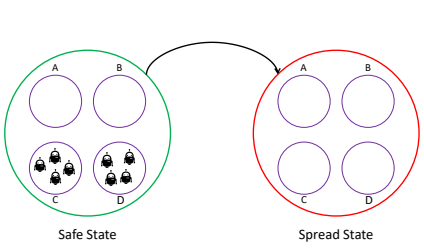
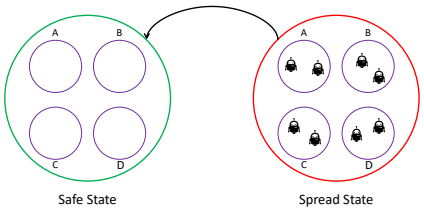


Figure 5.4: An individual run of the decentralized Q-learning dynamics for the potential game in Table 5.2

$$L1 - accuracy = \frac{1}{N}\sum_{i\in N}|BR^i - BR^i_{final}| = \frac{1}{N}\sum_{i\in N}\sum_{s\in\mathcal{S}}\sum_{a^i_j\in\mathcal{A}_i}|BR^i(s,a^i_j) - BR^i_{final}(s,a^i_j)|. \tag{5.2}$$



(a) The optimum strategy in the safe state



(b) The optimum strategy in the spread state

Figure 5.5: Optimum strategies in a distancing game

We stop the iterations when the L1-accuracy goes below a small threshold ϵ. Note that the convergence of $L1$ to zero indicates that the temperature parameter in BR have become small enough and the the perturbed BR has converged to a deterministic BR that chooses the optimum action corresponding to a Nash equilibrium in each state.

Figure 5.6 plots the L1-accuracy of 8 independent runs of decentralized Q-learning on the distancing game with $\gamma = 0.5$, and Figure 5.7 plots the corresponding means and shaded standard deviation of these runs. All independent runs converged to the optimal strategies in 5.5. The configuration of specific agents in facilities in each state may differ in different runs depending on the random seed.

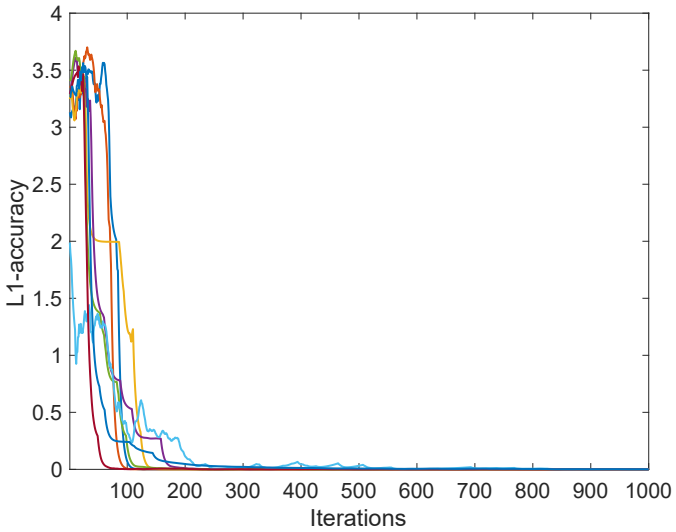


Figure 5.6: L1-accuracy trajectories for 8 independent runs of decentralized Q-learning dynamics on the distancing game benchmark

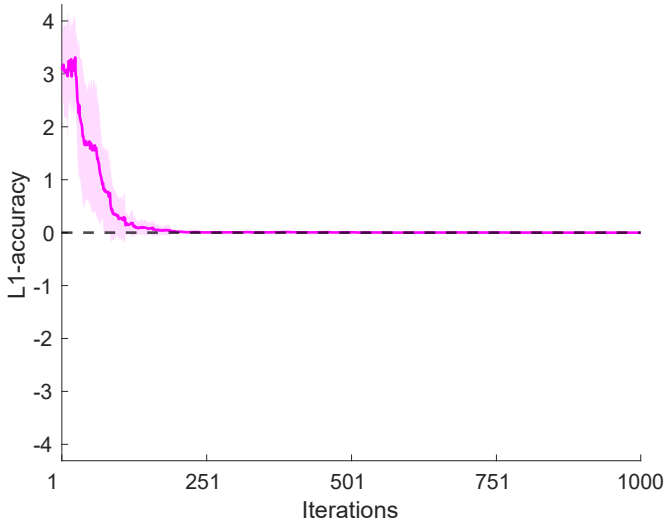


Figure 5.7: The mean and standard deviation of L1-accuracy trajectories for 8 independent runs of decentralized Q-learning dynamics on the distancing game benchmark

5.4 Congestion Games

As the second benchmark, we report our results on an example of atomic network routing game introduced in Section 3.2.1. We consider the game in Figure 5.8 where agents all start in the source node, and have to take the left or right action in each intermediate stage until they reach the destination node. To make the SG infinite-horizon, we move the agents back to the source node after they reach the destination node. Based on this condition, the number of states in the game will be $(l-1) \times 2^{|N|} + 1$, where l is the number of internal layers, and N is the number of agents. Therefore, the optimal policy in each state is for the agents to spread throughout the possible routes in each state and minimize congestion in the network. For example, the optimal number of agents in each route when they are all going from source to destination is written on arrows in Figure 5.8. We run the decentralized Q-learning dynamics in Figure 5.8 for $N = 4$ and $N = 4$ agents corresponding to $|S| = 17$ and $|S| = 17$ states with a constant cost for each route and $\gamma = 0.5$.

Figures 5.9 and 5.11 plot the L1-accuracy of 8 independent runs for $N = 4$ and $N = 8$ players, respectively, playing the game in Figure 5.8. Figures 5.10 and 5.12 plot the corresponding means and shaded standard deviation of these runs for $N = 4$ and $N = 8$ players, respectively. All independent runs converged to the optimal strategies for all states, i.e., minimum congestion at each route

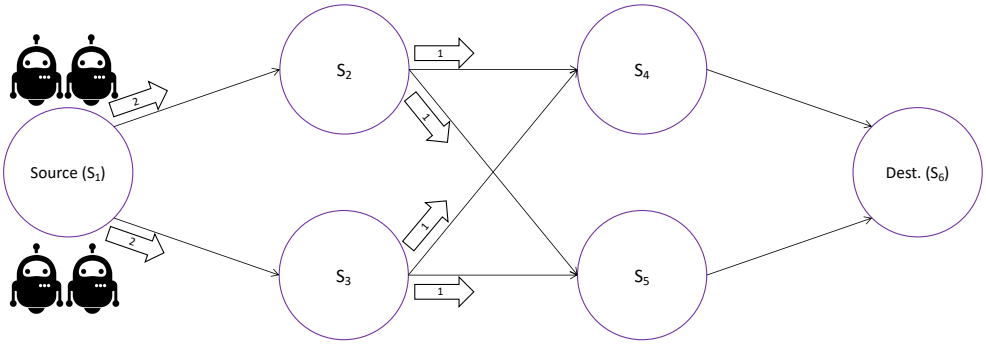


Figure 5.8: The atomic network routing game and its optimal strategy for reaching the destination node

was achieved by agents splitting throughout the routes in each state. Again, the configuration of specific agents in the network, i.e., their choice of routes in each state differs in different runs depending on the random seed.

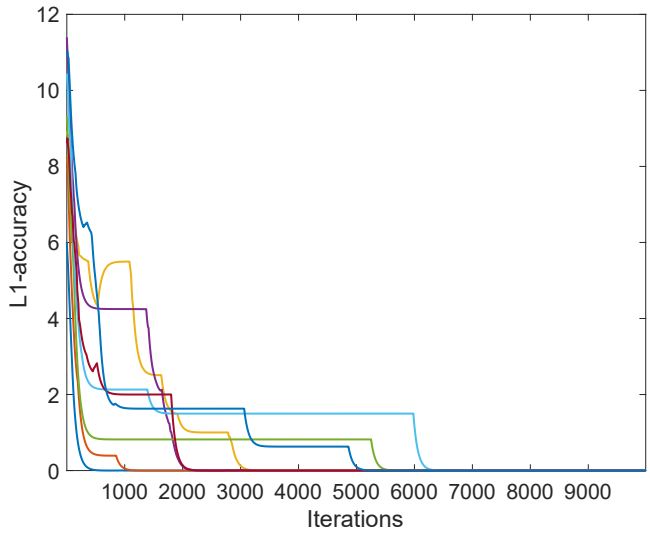


Figure 5.9: L1-accuracy trajectories for 8 independent runs of decentralized Q-learning dynamics on the network routing game benchmark for $N = 4$ agents

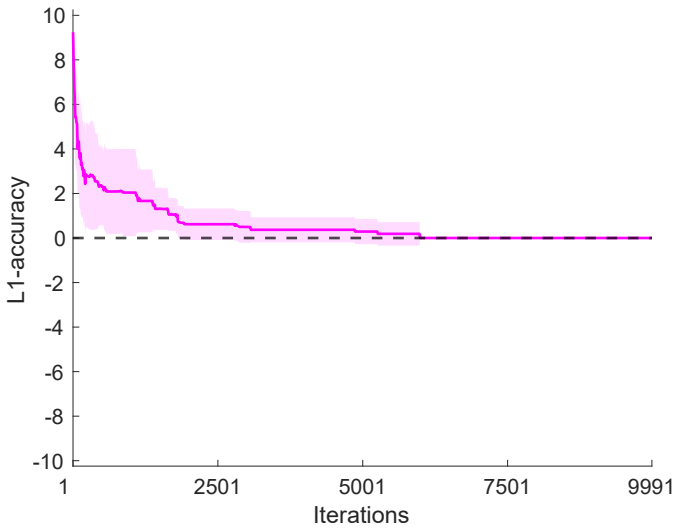


Figure 5.10: The mean and standard deviation of L1-accuracy trajectories for 8 independent runs of decentralized Q-learning dynamics on the network routing game benchmark for $N = 4$ agents

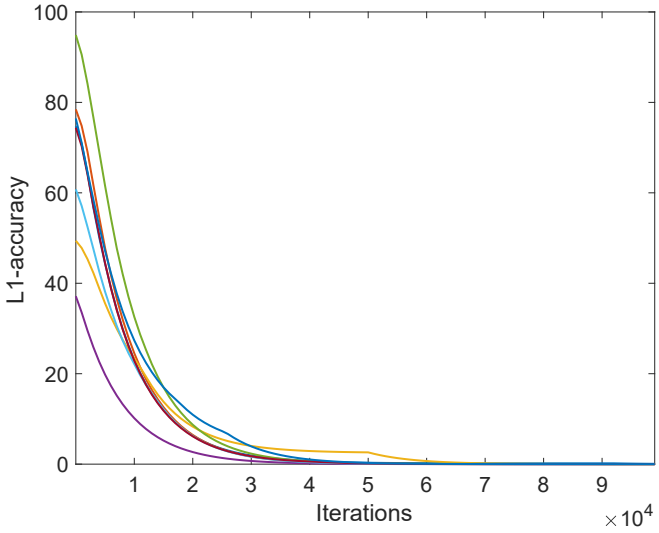


Figure 5.11: L1-accuracy trajectories for 8 independent runs of decentralized Q-learning dynamics on the network routing game benchmark for $N = 8$ agents

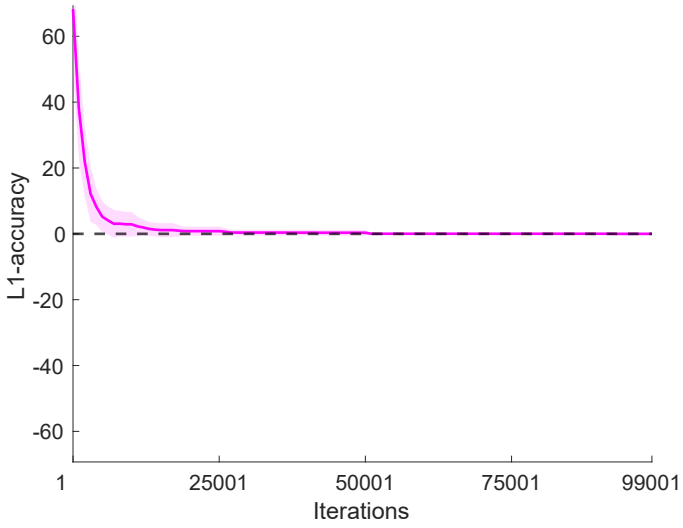


Figure 5.12: The mean and standard deviation of L1-accuracy trajectories for 8 independent runs of decentralized Q-learning dynamics on the network routing game benchmark for $N = 8$ agents

Chapter 6

Discussion and Conclusion

In the final chapter of this thesis, we discuss an important property of the decentralized Q-learning dynamics, i.e., game-agnostic convergence. We conclude the thesis by summarizing our contributions, and outlining the potential directions for future research.

6.1 Game-agnostic Learning Dynamics in MARL

In a very recent paper, Ding et al. [83] discuss a desirable property that a learning dynamics may possess, named *game-agnostic* convergence. This property indicates that the learning dynamics converges regardless of the type of the game being played or the players' awareness of this type. Although convergence has been established for both zero-sum and potential games when players adopt fictitious play [40, 44] or Q-learning [84, 85] in non-stochastic matrix games, this property has not been explored in the realm of MARL and stochastic games. Ding et al. [83], for the first time, show that a gradient-based algorithm, named Optimistic Gradient Descent/Ascent [65] converges to Nash equilibrium in both Markov zero-sum and potential two-player Markov cooperative games (an identical-interest case of Markov potential games). Therefore, our results indicate that decentralized Q-learning is potentially the first game-agnostic value-based MARL algorithm, and the first one to converge in multi-player SPGs.

6.2 Conclusion and Future Work

In this thesis, for the first time, we applied a purely value-based MARL approach, named decentralized Q-learning, to stochastic potential games, a class of Markov games with the ability to model many real-world scenarios. Following the previous established convergence in zero-sum games, our results indicate that decentralized

Q-learning is a game-agnostic MARL algorithm. Possible directions for future research include: analyzing the finite-time convergence properties of decentralized Q-learning, extending the algorithm to continuous state spaces using function approximation, and delving deeper into studying game-agnostic convergence of the algorithm by considering other families of games.

Appendix A

Analysis of Conjecture 4.2.1

The following theorem that is needed for decoupling the dynamics of each state, is based on Proposition 4.1 and Corollary 6.6 of Benaim [82] is given as Theorem 2 in Sayin et al. [14]:

Theorem A.0.1. *Consider the following discrete-time update rule,*

$$x_{t+1} = x_t + \lambda_t(F(x_t) + \epsilon_t + \omega_t), \tag{A.1}$$

and its limiting ordinary differential equation (ODE),

$$\frac{dx(t)}{dt} = F(x_t). \tag{A.2}$$

Assuming that the following are true,

1. *There exists a Lyapunov function $V : \mathbb{R}^m \to [0, \infty)$ for the ODE in* (A.2).
2. *The sequence of learning rates are bounded, their sum is divergent, and they are square-summable,*

$$\lambda_t \in [0,1] \quad \forall t, \quad \sum_{t=0}^{\infty} \lambda_t = \infty, \quad \sum_{t=0}^{\infty} \lambda_t^2 < \infty \tag{A.3}$$

3. *All iterates $x_t \in \mathbb{R}^m$ are bounded, $\sup_t ||x_t||_\infty < \infty$.*
4. *The vector field $F : \mathbb{R}^m \to \mathbb{R}^m$ is globally Lipschitz continuous.*
5. *The stochastic approximation term $\omega_t \in \mathbb{R}^m$ satisfies the following condition for all $T > 0$,*

$$\lim_{t\to\infty} \sup_{n>t:\sum_{l=t}^{n-1}\lambda_l \leq T} \left\{ \left|\left| \sum_{l=t}^{n-1} \lambda_l \omega_l \right|\right| \right\} = 0. \tag{A.4}$$

6. *The error term* $\epsilon_t \in \mathbb{R}^m$ *is asymptotically negligible, i.e.,* $\lim_{t\to\infty} ||\epsilon_t|| = 0$ *with probability 1.*

the limit set of (A.1) *will be contained in the following set with probability 1,*

$$\{x \in \mathbb{R}^m : V(x) = 0\}. \tag{A.5}$$

We now show that the normalization of the learning rates at each timestep, would make the evolution of every entry of the local Q- function, synchronous in the expectation. For player i, the stochastic approximation term in A.0.1 for all $a^i \in A^i_s$ and iteration t is

$$\omega^i_{s_t,t}[a^i_t] := \mathbf{1}_{a^i_t=a^i} \frac{r^i_s(a^i, a^{-i}_t) + \gamma v^i_{s_{t+1},t} - q^i_{s_t,t}[a^i]}{\overline{\pi}^i_t[a^i]} - \mathbb{E}\left\{ \mathbf{1}_{a^i_t=a^i} \frac{r^i_s(a^i, a^{-i}_t) + \gamma v^i_{s_{t+1},t} - q^i_{s_t,t}[a^i]}{\overline{\pi}^i_t[a^i]} \Big| h_t \right\}, \tag{A.6}$$

where $h_t = \{q^j_{s,t}, v^j_{s,t} | (j,s) \in \{1,...,n\} \times \mathcal{S}\}$ is the set of all current value and local Q-function estimates of players. To simplify the notation, we denote $\overline{\pi}^1_t[\tilde{a}^1]...\overline{\pi}^{i-1}_t[\tilde{a}^{i-1}]\overline{\pi}^{i+1}_t[\tilde{a}^{i+1}]...\overline{\pi}^n_t[\tilde{a}^n]$ as $\overline{\pi}^{-i}_t[\tilde{a}^{-i}]$, and $\sum_{\tilde{a}^1} ... \sum_{\tilde{a}^{i-1}} \sum_{\tilde{a}^{i+1}} ... \sum_{\tilde{a}^n}$ as $\sum_{\tilde{a}^{-i}}$. Now, we can expand the expectation above,

$$\mathbb{E}\left\{ \mathbf{1}_{a^i_t=a^i} \frac{r^i_s(a^i, a^{-i}_t) + \gamma v^i_{s_{t+1},t} - q^i_{s_t,t}[a^i]}{\overline{\pi}^i_t[a^i]} \Big| h_t \right\} = \overline{\pi}^i_t[a^i] \sum_{\tilde{a}^{-i}} \overline{\pi}^{-i}_t[\tilde{a}^{-i}] \frac{Q^i_{s_t,t}[a^i, \tilde{a}^{-i}] - q^i_{s_t,t}[a^i]}{\overline{\pi}^i_t[a^i]} \tag{A.7}$$

Therefore, after eliminating $\overline{\pi}^i_t[a^i]$, (A.6) can be written as,

$$\omega^i_{s_t,t}[a^i_t] := \mathbf{1}_{a^i_t=a^i} \frac{r^i_s(a^i, a^{-i}_t) + \gamma v^i_{s_{t+1},t} - q^i_{s_t,t}[a^i]}{\overline{\pi}^i_t[a^i]} - \left(\sum_{\tilde{a}^{-i}} Q^i_{s_t,t}(a^i, \tilde{a}^{-i}) \overline{\pi}^{-i}_t[\tilde{a}^{-i}] - q^i_{s_t,t}[a^i] \right), \tag{A.8}$$

Denoting the multi-dimensional tensor multiplication $Q^i_{s^t,t}\overline{\pi}^1_t...\overline{\pi}^{i-1}_t\overline{\pi}^{i+1}_t...\overline{\pi}^n_t$ as $Q^i_{s_t,t}\overline{\pi}^{-i}_t$, based on Proposition 2 in Sayin et al. [14], we can write the update rule in (4.9) as

$$q^i_{s_t,t+1} = q^i_{s_t,t} + \alpha_{\#s_t}\left(Q^i_{s_t,t}\overline{\pi}^{-i}_t - q^i_{s_t,t} + \omega^i_{s^t,t} \right) \tag{A.9}$$

We now consider the learning dynamics focused on a single state between two consecutive visits of that state. We may write the system of equation between these two visits of state s, as

$$q^i_{s,t'} = q^i_{s,t} + \alpha_{\#s}\Big(Q^i_{s,t}Br^{-i}(q^{-i}_{s,t}, \tau_{\#s}) - q^i_{s,t} + \mathbf{0} + \omega^i_{s,t}\Big) \quad \forall i \in N, \tag{A.10a}$$

$$v^i_{\hat{s},t'} = v^i_{\hat{s},t} + \alpha_{\#s}(0 + \epsilon^i_{\hat{s},t'} + 0), \quad \forall (i, \hat{s}) \in N \times \mathcal{S}, \tag{A.10b}$$

$$\tau_{\#s+1} = \tau_{\#s} + \alpha_{\#s}(0 + \frac{\tau_{\#s+1} - \tau_{\#s}}{\alpha_{\#s}} + 0). \tag{A.10c}$$

where

$$\epsilon^i_{\hat{s},t'} := \frac{v^i_{\hat{s},t'} - v^i_{\hat{s},t}}{\alpha_{\#s}}, \quad \forall (i, \hat{s}) \in N \times \mathcal{S}. \tag{A.11}$$

In (A.10), the first term inside the parentheses correspond to $F(x_t)$, the second term corresponds to ϵ_t, and the last term corresponds to ω_t in (A.1).

Based on Lemma 2 in Sayin et al. [14], the error terms in (A.11) are asymptotically zero. Now, based on Proposition 3 in Saying et al. [14] that ensures that $s \to \infty$ as $t \to \infty$ for every s, following Theorem A.0.1, we can write the limiting ODE of (A.10) as

$$\frac{dq^i_s(t)}{dt} = Q^i_s Br^{-i}(q^{-i}_s(t), \tau(t)) - q^i(t) \quad \forall (i, s) \in N \times \mathcal{S}, \tag{A.12a}$$

$$\frac{dv^i_s(t)}{dt} = 0, \quad \forall (i, s) \in N \times \mathcal{S}, \tag{A.12b}$$

$$\frac{d\tau(t)}{dt} = 0, \quad \forall s \in \textstyle\int. \tag{A.12c}$$

With the following notation

$$\begin{aligned} &Q^i_s Br^{-i}(q^{-i}_s(t), \tau(t)) \\ &:= Q^i_s \overline{Br}^1(q^1_s(t), \tau(t)) ... \overline{Br}^{i-1}(q^{i-1}_s(t), \tau(t)) \overline{Br}^{i+1}(q^{i+1}_s(t), \tau(t)) ... \overline{Br}^n(q^n_s(t), \tau(t)) \end{aligned} \tag{A.13}$$

Before approaching to the next step of the proof, we state the following lemma about SPGs:

Lemma A.0.1. *Assuming that the agents are playing an SPG (3.1.1) with state transitivity (3.1.2), based on Proposition 4 in Mguni et al. [7], the global Q-function of the game admits a potential function G:*

$$Q^i(s,(a^i)',a^{-i}) - Q^i(s,a^i,a^{-i}) = G(s,(a^i)',a^{-i}) - G(s,a^i,a^{-i}), \quad \forall (i,s) \in N \times \mathcal{S}. \tag{A.14}$$

We may add n auxiliary ODEs for π^i that trivially hold when we follow a noisy BR policy to the system of ODEs as

$$\frac{d\pi^i_s(t)}{dt} = Br^i(q^i_s(t),\tau(t)) - \pi^i_s(t), \quad \forall (i,s) \in N \times \mathcal{S} \tag{A.15}$$

Merging (A.12) and (A.15) and removing the state subscript, we will have the following system of ODEs for all i:

$$\frac{dq^i(t)}{dt} = Q^i Br^{-i}(q^{-i}(t),\tau) - q^i(t), \tag{A.16a}$$

$$\frac{d\pi^i(t)}{dt} = Br^i(q^i(t),\tau) - \pi^i(t), \tag{A.16b}$$

Our ultimate goal is to prove that the policies π^i will converge to a Nash equilubrium profile as $t \to \infty$.

To move towards this goal, we need to define a suitable Lyapunov function on the system of ODE at hand. We first define the following property for q^i,

Definition A.0.1 (Belief-based Q-function)**.** We call the local Q-function, q^i for player i, *belief-based*, if there exists a strategy $\pi^{-i} \in \Delta(A)$ (following the same notation as (A.13)), where we could establish the following equality:

$$q^i(t) = Q^i \pi^{-i}(t) \tag{A.17}$$

If we could establish that q^i are belief-based for every time step, then, the ODEs in (A.0.1) would follow the same trajectory as the stochastic fictitious play [44] with the following system of ODE:

$$\frac{d\pi^i(t)}{dt} = Br^i(Q^i \pi^{-i}(t),\tau) - \pi^i(t), \tag{A.18}$$

In Proposition 4.1, [44] proposes the following strict Lyapunov function for stochastic fictitious play in potential games,

$$L(\pi^i,\pi^{-i}) = (\pi^i)^T Q^i \pi^{-i} - \sum_{i \in N} \nu(\pi^i) \tag{A.19}$$

where ν is the perturbation function in (4.6) that can be chosen as (4.7).

Proposition 4.2 of Hofbauer [44] proves that the function in (A.19) is strictly increasing and converge to rest points as long as the perturbation function ν is sufficiently smooth, and in Proposition 3.1, it states that as the perturbations become small enough, these rest points approximate a Nash equilibrium of the underlying game.

The aforementioned proof is only valid if the belief-based property in Definition A.0.1 holds for every time step. If that is true, based on Lemma A.0.1, the trajectories of A.18 and A.16a would follow the same path towards a Nash equilibrium. However, the belief-based property does not necessary hold for the whole trajectory. It is possible to prove that q^i becomes belief-based in the limit using the following Lyapunov function,

$$H(t) = \sum_{i \in N} ||q^i - Q^i \pi^{-i}||^2 \tag{A.20}$$

By definition, (A.20) is always greater or equal to zero. Using (4.15) and (A.15), we can take the derivative of H

$$\frac{dH(t)}{dt} = -nH(t) \tag{A.21}$$

Therefore, $H(t)$ is a Lyapunov function for the flow, and,

$$\lim_{t \to \infty} ||q_t^i - Q^i \pi_t^{-i}||^2 = 0 \tag{A.22}$$

Which proves that q^i is belief-based asymptotically for every i.

We can form the belief-based error along the trajectory of A.22 as

$$\varepsilon(t) := q^i(t) - Q^i \pi^{-i}(t). \tag{A.23}$$

A possible direction to complete the proof would be quantifying this error term and establishing its rate of convergence towards zero.

Convergence Proof with an Extra Assumption: If we are able to establish the belief-based property (A.0.1) at the beginning of the trajectory with a communication-based initialization among the agents, then based on Corollary 8.17 in Kelly and Peterson [86], the system of linear ODEs formed by merging (A.18) and (A.16a) become coupled together and the system will have a unique solution that based on Hofbauer [44] is a Nash equilibrium.

Bibliography

[1] Robert W Rosenthal. «A class of games possessing pure-strategy Nash equilibria». In: *International Journal of Game Theory* 2.1 (1973), pp. 65–67 (cit. on p. 1).

[2] Tiina Heikkinen. «A potential game approach to distributed power control and scheduling». In: *Computer Networks* 50.13 (2006), pp. 2295–2311 (cit. on p. 1).

[3] Shan Yu and Marc Berthod. «A game strategy approach for image labeling». In: *Computer Vision and Image Understanding* 61.1 (1995), pp. 32–37 (cit. on p. 1).

[4] Marc Berthod, Zoltan Kato, Shan Yu, and Josiane Zerubia. «Bayesian image classification using Markov random fields». In: *Image and vision computing* 14.4 (1996), pp. 285–295 (cit. on p. 1).

[5] Yakov Babichenko and Omer Tamuz. «Graphical potential games». In: *Journal of Economic Theory* 163 (2016), pp. 889–899 (cit. on p. 1).

[6] David Mguni. «Stochastic potential games». In: *arXiv preprint arXiv:2005.13527* (2020) (cit. on p. 1).

[7] David H Mguni, Yutong Wu, Yali Du, Yaodong Yang, Ziyi Wang, Minne Li, Ying Wen, Joel Jennings, and Jun Wang. «Learning in nonzero-sum stochastic games with potentials». In: *International Conference on Machine Learning*. PMLR. 2021, pp. 7688–7699 (cit. on pp. 1, 18, 19, 43).

[8] Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. «Global convergence of multi-agent policy gradient in markov potential games». In: *arXiv preprint arXiv:2106.01969* (2021) (cit. on pp. 1, 2, 18–20).

[9] Runyu Zhang, Zhaolin Ren, and Na Li. «Gradient play in multi-agent markov stochastic games: Stationary points and convergence». In: *arXiv preprint arXiv:2106.00198* (2021) (cit. on pp. 1, 18).

[10] David Silver et al. «Mastering chess and shogi by self-play with a general reinforcement learning algorithm». In: *arXiv preprint arXiv:1712.01815* (2017) (cit. on pp. 2, 7, 15).

[11] Oriol Vinyals et al. «Grandmaster level in StarCraft II using multi-agent reinforcement learning». In: *Nature* 575.7782 (2019), pp. 350–354 (cit. on pp. 2, 15).

[12] Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. «A survey of learning in multiagent environments: Dealing with non-stationarity». In: *arXiv preprint arXiv:1707.09183* (2017) (cit. on pp. 2, 16).

[13] Lucian Busoniu, Robert Babuska, and Bart De Schutter. «A comprehensive survey of multiagent reinforcement learning». In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.2 (2008), pp. 156–172 (cit. on p. 2).

[14] Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. «Decentralized Q-learning in zero-sum Markov games». In: *Advances in Neural Information Processing Systems* 34 (2021) (cit. on pp. 2, 17, 18, 24, 26, 30, 40–42).

[15] Tim Roughgarden. «Routing games». In: *Algorithmic game theory* 18 (2007), pp. 459–484 (cit. on pp. 2, 19).

[16] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cit. on pp. 4, 5).

[17] Peter Dayan, Maneesh Sahani, and Grégoire Deback. «Unsupervised learning». In: *The MIT encyclopedia of the cognitive sciences* (1999), pp. 857–859 (cit. on p. 4).

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. «Unsupervised learning». In: *The elements of statistical learning*. Springer, 2009, pp. 485–585 (cit. on p. 4).

[19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. «Overview of supervised learning». In: *The elements of statistical learning*. Springer, 2009, pp. 9–41 (cit. on p. 4).

[20] Yaodong Yang and Jun Wang. «An overview of multi-agent reinforcement learning from game theoretical perspective». In: *arXiv preprint arXiv:2011.00583* (2020) (cit. on p. 4).

[21] David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. «Reward is enough». In: *Artificial Intelligence* 299 (2021), p. 103535 (cit. on p. 5).

[22] Richard Bellman. «Dynamic programming». In: *Science* 153.3731 (1966), pp. 34–37 (cit. on pp. 6, 7).

[23] Csaba Szepesvári. «Algorithms for reinforcement learning». In: *Synthesis lectures on artificial intelligence and machine learning* 4.1 (2010), pp. 1–103 (cit. on p. 6).

[24] David Ha and Jürgen Schmidhuber. «World models». In: *arXiv preprint arXiv:1803.10122* (2018) (cit. on p. 7).

[25] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. «Model-based reinforcement learning: A survey». In: *arXiv preprint arXiv:2006.16712* (2020) (cit. on p. 7).

[26] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*. Vol. 37. Citeseer, 1994 (cit. on p. 8).

[27] Richard S Sutton. «Generalization in reinforcement learning: Successful examples using sparse coarse coding». In: *Advances in neural information processing systems* 8 (1995) (cit. on p. 8).

[28] Christopher JCH Watkins and Peter Dayan. «Q-learning». In: *Machine learning* 8.3 (1992), pp. 279–292 (cit. on pp. 8, 10).

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. «Playing atari with deep reinforcement learning». In: *arXiv preprint arXiv:1312.5602* (2013) (cit. on p. 8).

[30] J. Peters. «Policy gradient methods». In: *Scholarpedia* 5.11 (2010). revision #137199, p. 3698. DOI: 10.4249/scholarpedia.3698 (cit. on p. 8).

[31] Dmitry Kalashnikov et al. «Scalable deep reinforcement learning for vision-based robotic manipulation». In: *Conference on Robot Learning*. PMLR. 2018, pp. 651–673 (cit. on p. 8).

[32] Yao Lu et al. «AW-Opt: Learning Robotic Skills with Imitation andReinforcement at Scale». In: *Conference on Robot Learning*. PMLR. 2022, pp. 1078–1088 (cit. on p. 8).

[33] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. «Continuous control with deep reinforcement learning». In: *arXiv preprint arXiv:1509.02971* (2015) (cit. on p. 8).

[34] Hao Dong, Hao Dong, Zihan Ding, Shanghang Zhang, and Chang. *Deep Reinforcement Learning*. Springer, 2020 (cit. on p. 9).

[35] Csaba Szepesvári. «The asymptotic convergence-rate of Q-learning». In: *Advances in neural information processing systems* 10 (1997) (cit. on p. 10).

[36] Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994 (cit. on p. 10).

[37] Dario Bauso. *Game theory with engineering applications*. SIAM, 2016 (cit. on pp. 10, 13).

[38] John Nash. «Non-cooperative games». In: *Annals of mathematics* (1951), pp. 286–295 (cit. on p. 11).

[39] John F Nash Jr. «Equilibrium points in n-person games». In: *Proceedings of the national academy of sciences* 36.1 (1950), pp. 48–49 (cit. on p. 12).

[40] Dov Monderer and Lloyd S Shapley. «Potential games». In: *Games and economic behavior* 14.1 (1996), pp. 124–143 (cit. on pp. 12, 14, 38).

[41] Drew Fudenberg, Fudenberg Drew, David K Levine, and David K Levine. *The theory of learning in games*. Vol. 2. MIT press, 1998 (cit. on pp. 13, 24).

[42] George W Brown. «Iterative solution of games by fictitious play». In: *Act. Anal. Prod Allocation* 13.1 (1951), p. 374 (cit. on p. 14).

[43] Ulrich Berger. «Fictitious play in 2× n games». In: *Journal of Economic Theory* 120.2 (2005), pp. 139–154 (cit. on p. 14).

[44] Josef Hofbauer and William H Sandholm. «On the global convergence of stochastic fictitious play». In: *Econometrica* 70.6 (2002), pp. 2265–2294 (cit. on pp. 14, 27, 38, 43, 44).

[45] Drew Fudenberg and David M Kreps. «Learning mixed equilibria». In: *Games and economic behavior* 5.3 (1993), pp. 320–367 (cit. on p. 14).

[46] Lloyd S Shapley. «Stochastic games». In: *Proceedings of the national academy of sciences* 39.10 (1953), pp. 1095–1100 (cit. on p. 15).

[47] Michael L Littman. «Markov games as a framework for multi-agent reinforcement learning». In: *Machine learning proceedings 1994*. Elsevier, 1994, pp. 157–163 (cit. on p. 15).

[48] Matteo Hessel et al. «Rainbow: Combining improvements in deep reinforcement learning». In: *Thirty-second AAAI conference on artificial intelligence*. 2018 (cit. on p. 15).

[49] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. «Multi-agent actor-critic for mixed cooperative-competitive environments». In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 15, 16).

[50] Daniel Friedman. «On economic applications of evolutionary game theory». In: *Journal of evolutionary economics* 8.1 (1998), pp. 15–43 (cit. on p. 16).

[51] Julia Robinson. «An iterative method of solving a game». In: *Annals of mathematics* (1951), pp. 296–301 (cit. on p. 16).

[52] Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. «Dealing with non-stationarity in multi-agent deep reinforcement learning». In: *arXiv preprint arXiv:1906.04737* (2019) (cit. on p. 16).

[53] Shariq Iqbal and Fei Sha. «Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning». In: *arXiv preprint arXiv:1905.12127* (2019) (cit. on p. 16).

[54] Amy Greenwald, Keith Hall, Roberto Serrano, et al. «Correlated Q-learning». In: *ICML*. Vol. 3. 2003, pp. 242–249 (cit. on p. 16).

[55] Pablo Hernandez-Leal and Michael Kaisers. «Learning against sequential opponents in repeated stochastic games». In: *The 3rd Multi-disciplinary Conference on Reinforcement Learning and Decision Making, Ann Arbor*. Vol. 25. 2017 (cit. on p. 16).

[56] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. «Learning to communicate with deep multi-agent reinforcement learning». In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 16).

[57] Dylan P Losey, Mengxi Li, Jeannette Bohg, and Dorsa Sadigh. «Learning from my partner's actions: Roles in decentralized robot teams». In: *Conference on robot learning*. PMLR. 2020, pp. 752–765 (cit. on p. 16).

[58] Piotr J Gmytrasiewicz and Prashant Doshi. «A framework for sequential planning in multi-agent settings». In: *Journal of Artificial Intelligence Research* 24 (2005), pp. 49–79 (cit. on p. 16).

[59] Michael Wunder, John Robert Yaros, Michael Kaisers, and Michael L Littman. «A framework for modeling population strategies by depth of reasoning.» In: *AAMAS*. Citeseer. 2012, pp. 947–954 (cit. on p. 16).

[60] Michael Bowling and Manuela Veloso. «Multiagent learning using a variable learning rate». In: *Artificial Intelligence* 136.2 (2002), pp. 215–250 (cit. on p. 17).

[61] Hongyi Guo, Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. «Decentralized Single-Timescale Actor-Critic on Zero-Sum Two-Player Stochastic Games». In: *International Conference on Machine Learning*. PMLR. 2021, pp. 3899–3909 (cit. on p. 17).

[62] Yulai Zhao, Yuandong Tian, Jason D Lee, and Simon S Du. «Provably efficient policy gradient methods for two-player zero-sum Markov games». In: *arXiv preprint arXiv:2102.08903* (2021) (cit. on pp. 17, 18).

[63] David S Leslie and EJ Collins. «Convergent multiple-timescales reinforcement learning algorithms in normal form games». In: *The Annals of Applied Probability* 13.4 (2003), pp. 1231–1251 (cit. on p. 17).

[64] David S Leslie, Steven Perkins, and Zibo Xu. «Best-response dynamics in zero-sum stochastic games». In: *Journal of Economic Theory* 189 (2020), p. 105095 (cit. on pp. 18, 22).

[65] Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. «Last-iterate convergence of decentralized optimistic gradient descent/ascent in infinite-horizon competitive markov games». In: *Conference on learning theory*. PMLR. 2021, pp. 4259–4299 (cit. on pp. 18, 38).

[66] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. «Fully decentralized multi-agent reinforcement learning with networked agents». In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5872–5881 (cit. on p. 18).

[67] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. «Finite-sample analysis for decentralized batch multiagent reinforcement learning with networked agents». In: *IEEE Transactions on Automatic Control* 66.12 (2021), pp. 5925–5940 (cit. on p. 18).

[68] Qichao Zhang, Dongbin Zhao, and Frank L Lewis. «Model-free reinforcement learning for fully cooperative multi-agent graphical games». In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–6 (cit. on p. 18).

[69] Yan Zhang and Michael M Zavlanos. «Distributed off-policy actor-critic reinforcement learning with policy consensus». In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 4674–4679 (cit. on p. 18).

[70] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. «Networked multi-agent reinforcement learning in continuous spaces». In: *2018 IEEE conference on decision and control (CDC)*. IEEE. 2018, pp. 2771–2776 (cit. on p. 18).

[71] Ozan Candogan, Ishai Menache, Asuman Ozdaglar, and Pablo A Parrilo. «Flows and decompositions of games: Harmonic and potential games». In: *Mathematics of Operations Research* 36.3 (2011), pp. 474–503 (cit. on p. 18).

[72] David González-Sánchez and Onésimo Hernández-Lerma. *Discrete–time stochastic control and dynamic potential games: the Euler–Equation approach*. Springer Science & Business Media, 2013 (cit. on p. 18).

[73] Sergio Valcarcel Macua, Javier Zazo, and Santiago Zazo. «Learning parametric closed-loop policies for markov potential games». In: *arXiv preprint arXiv:1802.00899* (2018) (cit. on p. 18).

[74] Santiago Zazo, Sergio Valcarcel Macua, Matilde Sánchez-Fernández, and Javier Zazo. «Dynamic potential games with constraints: Fundamentals and applications in communications». In: *IEEE Transactions on Signal Processing* 64.14 (2016), pp. 3806–3821 (cit. on p. 18).

[75] David Mguni, Joel Jennings, and Enrique Munoz de Cote. «Decentralised learning in systems with many, many strategic agents». In: *Thirty-second AAAI conference on artificial intelligence*. 2018 (cit. on p. 18).

[76] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. «On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift.» In: *J. Mach. Learn. Res.* 22.98 (2021), pp. 1–76 (cit. on p. 18).

[77] Roy Fox, Stephen M Mcaleer, Will Overman, and Ioannis Panageas. «Independent natural policy gradient always converges in Markov potential games». In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 4414–4425 (cit. on pp. 18, 20).

[78] Muhammed O Sayin, Francesca Parise, and Asuman Ozdaglar. «Fictitious play in zero-sum stochastic games». In: *arXiv preprint arXiv:2010.04223* (2020) (cit. on p. 22).

[79] Arlington M Fink. «Equilibrium in a stochastic n-person game». In: *Journal of science of the hiroshima university, series ai (mathematics)* 28.1 (1964), pp. 89–93 (cit. on p. 22).

[80] David S Leslie and Edmund J Collins. «Individual Q-learning in normal form games». In: *SIAM Journal on Control and Optimization* 44.2 (2005), pp. 495–514 (cit. on p. 24).

[81] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Vol. 48. Springer, 2009 (cit. on p. 24).

[82] Michel Benaım. «Dynamics of stochastic approximation algorithms». In: *Seminaire de probabilites XXXIII*. Springer, 1999, pp. 1–68 (cit. on pp. 24, 40).

[83] Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo Jovanovic. «Independent policy gradient for large-scale Markov potential games: Sharper rates, function approximation, and game-agnostic convergence». In: *International Conference on Machine Learning*. PMLR. 2022, pp. 5166–5220 (cit. on p. 38).

[84] Stefanos Leonardos, Georgios Piliouras, and Kelly Spendlove. «Exploration-exploitation in multi-agent competition: convergence with bounded rationality». In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 26318–26331 (cit. on p. 38).

[85] Stefanos Leonardos and Georgios Piliouras. «Exploration-exploitation in multi-agent learning: Catastrophe theory meets game theory». In: *Artificial Intelligence* 304 (2022), p. 103653 (cit. on p. 38).

[86] Walter G Kelley and Allan C Peterson. *The theory of differential equations: classical and qualitative*. Springer Science & Business Media, 2010 (cit. on p. 44).