Multiagent Social Learning in Large Repeated Games

Jean Oh

CMU-LTI-09-008

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:
Stephen F. Smith, Chair
Jaime Carbonell
Manuela M. Veloso
Sarit Kraus, Bar-Ilan University, Israel,
University of Maryland, College Park

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In Language and Information Technologies

Copyright © 2009 Jean Oh

This research was sponsored in part by the Department of Defense Advanced Research Projects Agency (DARPA) under contract #NBCHD030010.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Defense Advanced Research Projects Agency, or the U.S. Government.

Keywords: multi-agent learning, social learning, social norm, repeated games, congestion games, externality

This thesis is dedicated to my family for their love and support.

Abstract

This thesis studies a class of problems where rational agents can make suboptimal decisions by ignoring a side effect that each individual action brings to bear on the common good. It is generally believed that a mutually desirable strategy can be enforced as a stable outcome for rational agents if the imminent threat exists that any deviator from the strategy will be punished. This thesis expands this understanding, arguing that rationally bounded agents can learn to self-organize to stabilize on mutually beneficial outcomes *without* the explicit notion of threat. As an approach to demonstrate this capability, a double-layered multiagent learning algorithm, known here as IMPRES (implicit reciprocal strategy learning), has been developed.

In game theory, it is generally assumed that the players (agents) of a game are of equal ability. This thesis takes a contrasting view. The foundation of this work is inspired by the concept of “bounded rationality”, where some agents may have more privileges than others either because they are exposed to different parts of information in the environment, or because they simply have higher computational power. Based on this intuition, this thesis investigates how agents can boost their performance by utilizing the notion of social learning - learning from one another in an agent society.

Theoretical and empirical results show that the IMPRES agents learn to behave rationally as if they are in a virtually optimal Nash equilibrium of a repeated game. To my knowledge, IMPRES is the first algorithm that achieves this property in games involving more than two players under imperfect monitoring.

Acknowledgements

There are many people whom I must thank for their support in writing this thesis, although I claim that all errors are purely mine. Above all, I would like to express my special thanks to my thesis advisor Stephen F. Smith for letting me explore various research topics and for supporting me with thoughtful guidance throughout since I conceived this thesis. I am extremely fortunate to have had Steve as my advisor whom I truly respect.

I am also grateful to my thesis committee: Jaime Carbonell for his detailed feedback and for insightful suggestions for future directions of this research on decision making with multiple criteria; Manuela Veloso for her valuable guidance on this thesis and for general advice on persuasive written and oral presentations; and Sarit Kraus for the fruitful discussions on the efficiency of the algorithm and for the inspiration on multiagent systems research. I appreciate the members of the RADAR project that provided the rudiments for this thesis; and I extend my gratitude to the late Pragnesh Jay Modi who made significant contributions to the RADAR project. I also owe greatly to the authors of referenced work on which this thesis has been built.

Many thanks go to the members of Intelligent Coordination and Logistics Laboratory. I am especially grateful to Laura Barbulescu, Anthony Gallagher, Xiafang Wang, Pradeep Varakantham, Larry Kramer, Charlie Collins, David Crimm, Greg Barlow, Zack Rubinstein, Terry Zimmerman, Yangfang (Helen) Zhou, Matthew Danish, Susan Buchman, Vince Cicirello, David Hildum, and Marliese Bonk for detailed constructive comments and for their friendship.

In addition, I thank Geoff Gordon for helpful discussion on congestion games; Andrew Gilpin and Sam Ganzfried for their game theoretic insights; Reshef Meir and Archie Chapman for the productive discussion on correlated equilibria; Katia Sycara and the members of the Intelligent Software Agents group for the helpful feedback; the members of the CORAL research group for their honest comments; David Sarne and Sanmay Das for their kind encouragement when I was struggling with lack of confidence.

I am thankful to the faculty and staff members of Language Technologies Institute (LTI) for supporting this research; and I offer big thanks to my student colleagues of LTI: Satanjeev (Bano) Banerjee, Ming-yu (Robert) Chen, Brian Langner, Yik Cheung (Wilson) Tam, Antoine Raux, Jeongwoo Ko, Shinjae Yoo, Jaedong (Jaedy) Kim, Chanwoo Kim, Moonyoung Kang, Yanjun Qi, and specially to Betty Cheng for keeping me in the circle. Additionally, I appreciate Carol Frieze and all the women@SCS folks for making me feel proud to be a female computer scientist.

I am deeply indebted to the teachers and friends from my time prior to Carnegie Mellon: Craig Knoblock, Steven Minton, Jihie Kim, Pui Anusa, Cenk Gazen, Maria Muslea, Ion Muslea, Greg Barish, and Yigal Arens from Information Sciences Institute in Marina del Rey for the inspiration on intelligent systems research and for their friendship; T.C. Chiang from Bell Labs. for his encouragement as a mentor; and Newcomb Greenleaf and Salvatore Stolfo from Columbia University for introducing me to the fascinating field of Artificial Intelligence.

On a personal note, I am humbled by all the support from my dear friends: among all, I offer my sincere thanks to Eunji Chung for helping me out in numerous ways when I first came to Pittsburgh; Kathy Kim and her family for their generous support; and my soulmate friend Jiyeon Lee for her warm support. I am privileged to have my research collaborator and dear friend Jie-Eun Hwang; I am grateful to Jie-Eun for sharing academic insights on collective intelligence and for being there for me as a friend.

Lastly, I tribute my heartful thanks to my family: Mikyong Amy Park, Mooyeon Oh-Park, Chonghyo Thomas Park and the Park family, Seungmok Oh, Seungyong Oh, Father and Mother for their unconditional love and unyielding support. I extend my gratitude to Chulho Lee; without his trustworthy care for my parents I could not have focused on this research. My deepest appreciation goes to my Daughter Sylvie. These thesis chapters will always remind me of the smell of corn bread that she secretly baked in the early morning of my defense day, the sweetness of her caring letters, and the warmth of her hugs. I could not have done it without all the love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "Anything that gives us new knowledge gives us an opportunity to be more rational." - Herbert A. Simon.

1.1 Overview

This thesis studies a class of problems in which a large number of self-interested agents compete for common resources in an environment. In this context, perfectly rational agents can make suboptimal decisions by ignoring a side effect, referred to as an externality, that each individual action brings to bear on the common good. More broadly, short-sighted agents that only care about immediate rewards can limit themselves to suboptimal decisions by ignoring the forthcoming reward that they will receive later in time. Given the propensity to ignore future good, the main theme of this thesis is to explore multiagent learning algorithms with which agents learn to make mutually beneficial decisions in a repeated game setting.

In game theory, it is generally assumed that the players (agents) of a game are of equal ability. My research takes a constrasting view. The foundation of my work is inspired by the concept of "bounded rationality", where some agents may have more privileges than others either because they are exposed to different parts of information in the environment, or because they simply have higher computational power. Based on this intuition, my research investigates how agents can boost their learning performance by utilizing this asymmetry in abilities.

Specifically, this thesis proposes an algorithm for a large number of agents to learn mutually beneficial correlated[1] strategies, and characterizes the outcome of

[1] A technical definition of a correlated strategy is a probability distribution over a set of all possible joint strategy profiles; an English dictionary definition is sufficient in this context: to bear

the algorithm as behavior-equivalent to Nash equilibria of a repeated game. The algorithm is thoroughly evaluated in the context of congestion games; the algorithm is specifically focused on symmetric network congestion games where the complexity of the algorithm is polynomial to the number of agents and to network size. For the purpose of empirical analysis, a set of criteria for evaluating multiagent learning algorithms is also proposed.

Theoretical and empirical results show that the IMPRES agents learn to behave rationally as if they are in a mutually desirable Nash equilibrium of a repeated game. To my knowledge, IMPRES is the first algorithm that achieves this property in games involving more than two players under imperfect monitoring.

1.2 Problem domain

Congestion games refer to a class of games where the immediate payoff of a player depends only on the number of players that have also chosen the same action with the player [51]. A natural example of a congestion game is a traffic congestion problem where each driver independently chooses a path among alternative routes to reach her destination as quickly as possible, while actual travel time is determined by the traffic load of the chosen path. Congestion games concisely represent an important class of problems in transportation sciences, computer networks, and algorithmic game theory [44].

In this thesis, a congestion game is formulated as a multiagent decision making problem where a large number of agents select resources from a common set of resources; the objective of an individual agent is to minimize the cost of using the chosen resource. In particular, the main interest of this thesis is in a repeated game setting where the set of agents repeatedly participate in the same decision-making scenario; for instance, the agents travel from a certain origin to a certain destination on a regular basis. Figure 1.1 illustrates an example where a set of agents are trying to determine a path from a set of alternatives.

In congestion games, an agent can unintentionally cause a delay in the travel time of other agents on the same path. Generally, an indirect impact on the benefits of the others in an environment is called an *externality* [13]. In this sense, congestion games involve negative externalities[2].

reciprocal or mutual relations.

[2] Congestion externalities can be positive in certain problem domains such as peer-to-peer (P2P) sharing model where the quality of service can be increased as more sharers participate with the same server. In this thesis, we study negative externalities exclusively.

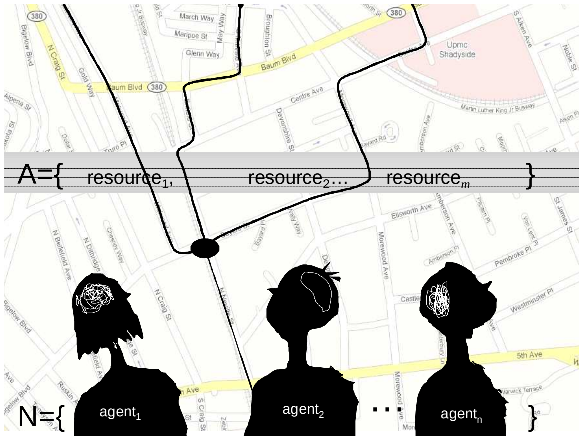

A={ resource$_1$, resource$_2$… resource$_m$ }
N={ agent$_1$ agent$_2$ … agent$_n$ }
Baum Blvd
Upmc Shadyside
Centre Ave
5th Ave
N Craig St
Ellsworth Ave

Figure 1.1: Multiagent modeling of a congestion game: each agent from set N chooses an action (path) from set A, where an agent's congestion cost depends not only on the agent's choice of action but also on the choices of others.

A similar sort of problem, known as *the tragedy of the commons*, emerges when rational agents fail to learn the best action based on their expected reward. This can occur due to agents' lack of awareness of the externalities to other agents [26]. As a result, completely rational agents can bring about substantially suboptimal outcomes in some problems; these problems are of keen interest to this research.

In particular, this thesis is focused on symmetric network congestion games where all agents travel from the same origin (source) to the same destination (sink). An in-depth survey on congestion games including recent discoveries in complexity analysis can be found in Chapter 2.

1.3 Background

In multiagent learning, two particularly important criteria for learning algorithms are rationality and convergence [4]. The former stresses that a learning algorithm must be adaptive to stationary opponents, while the latter sets the target of convergence to a Nash equilibrium in self-play - a setting in which opponents use the same learning algorithm as the learner. Consequently, the majority of existing multiagent learning algorithms aim to converge in self-play, particularly to Nash equilibria of a single-shot game [10, 37, 4, 11, 30].

A Nash equilibrium is the most celebrated solution concept in both game theory and the multiagent learning literature, where every agent plays its best response strategy against other agents [43]. In a single-shot game setting where a game is played only once, however, a Nash equilibrium can be arbitrarily inefficient in terms of the quality of system-wide solution [44].

When a game is repeated indefinitely, it is generally believed that any correlated strategy where every agent receives a better expected payoff than the *minimax* value can be enforced as a Nash equilibrium by use of threat, where the minimax value refers to the worst possible payoff that other agents can force against an agent; this general belief is dubbed the folk theorem [23]. The proof of the folk theorem is the existence of a meta-strategy known as the grim-trigger strategy that imposes an imminent threat: if an agent deviates from a correlated strategy, the other agents will, in perpetuity, use retaliatory strategies that will wipe out the deviator's gain.

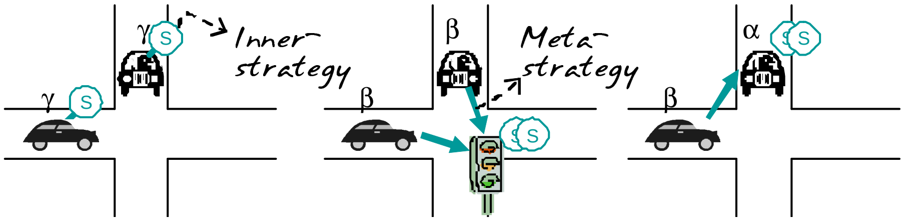
Although the folk theorem holds for multiple players, the majority of existing learning algorithms aiming at Nash equilibria of a repeated game are limited to 2-player games [58, 36, 54, 14], where direct reciprocity is feasible. In n-player games, rational learning (also known as Bayesian learning) can lead the agents to behave as if they are in a Nash equilibrium of a repeated game [32]; however, the algorithm relies strongly on the assumptions that the agents have perfect knowledge of the game, and that the agents a priori know a set of strategies that possess *a grain of truth* for the true strategies of other agents; that is, for every possible game playing history given the true strategies, there is a non-zero probability of the history given the agents' beliefs. Not only does this strong assumption limit rational learning, but games also exist where rational learning fails to converge [24, 21]; an example will be discussed in Section 1.5.2.

In terms of computational complexity, the problem of finding Nash equilibria in congestion games is intractable both in single-shot games [16] and in repeated games [3]. The complexity issues will be fully discussed in Chapter 2.

Given the difficulty of finding Nash equilibria, this thesis alternatively explores multiagent learning algorithms that can find stable solutions for rationally bounded agents, and studies how these alternative solutions relate to well-known solution concepts such as Nash equilibria.

1.4 Algorithm preview

The main hypothesis of this thesis echoes the folk theorem: in order to achieve mutually beneficial outcome in repeated games, a population needs a set of correlated strategies around a social norm. While the folk theorem states that such equilibria exist, this thesis seeks efficient learning algorithms for actually finding those stable strategies that can realize mutually beneficial outcomes in the context of large re-



An independent model: non-zero probability of collision

A centralized model: a traffic light

The **IMPRES** model: middle-ground solutions

Figure 1.2: A scenario of two driving agents at an intersection

peated games. As opposed to assuming that agents a priori know high-level strategies such as grim-trigger, the focus of this research is how the agents learn such reciprocal strategies, known here as **`implicit reciprocal strategy learning`** (**IMPRES**).

Generally, learning indirectly from the experiences of others (as opposed to one's own experiences) is referred to as *social learning* [57]. While the notion of social learning constitutes an important part of human learning, the idea has been underexplored in (artificial) agent learning. The IMPRES algorithm utilizes the notion of social learning in the context of multiagent learning; such that the agents act more rationally by adopting the strategies that are *given* by other agents.

The IMPRES algorithm consists of two learning layers such that: the agents learn a *correlated strategy* in the inner-layer, while they progressively self-organize to learn the *social norm of reciprocity* in the meta-layer. The algorithm requires that the agents know the cost functions, and that each agent can observe its own action selection and corresponding cost.

Consider two automobiles coming from orthogonal directions to an intersection depicted in Figure 1.2. When each agent independently makes decisions, there is a non-zero probability that there will be a collision. On the other hand, one can imagine a centralized model where there exists a traffic light that both agents can observe. If an agent observes a red light, it is best for the agent to stop given that the other agent obeys its corresponding (green) light.

The intuition behind the IMPRES algorithm can be illustrated in that situation without a traffic light or a stop sign. Without prior agreement, one of the drivers signals the other driver to cross first. The other driver then obeys the signal, and thus both drivers achieve mutually beneficial outcomes by avoiding a collision. Specifically, each driver in this example has three meta-strategies that it could adopt:

- α-agent (strategist) that computes an optimal correlated strategy to generate

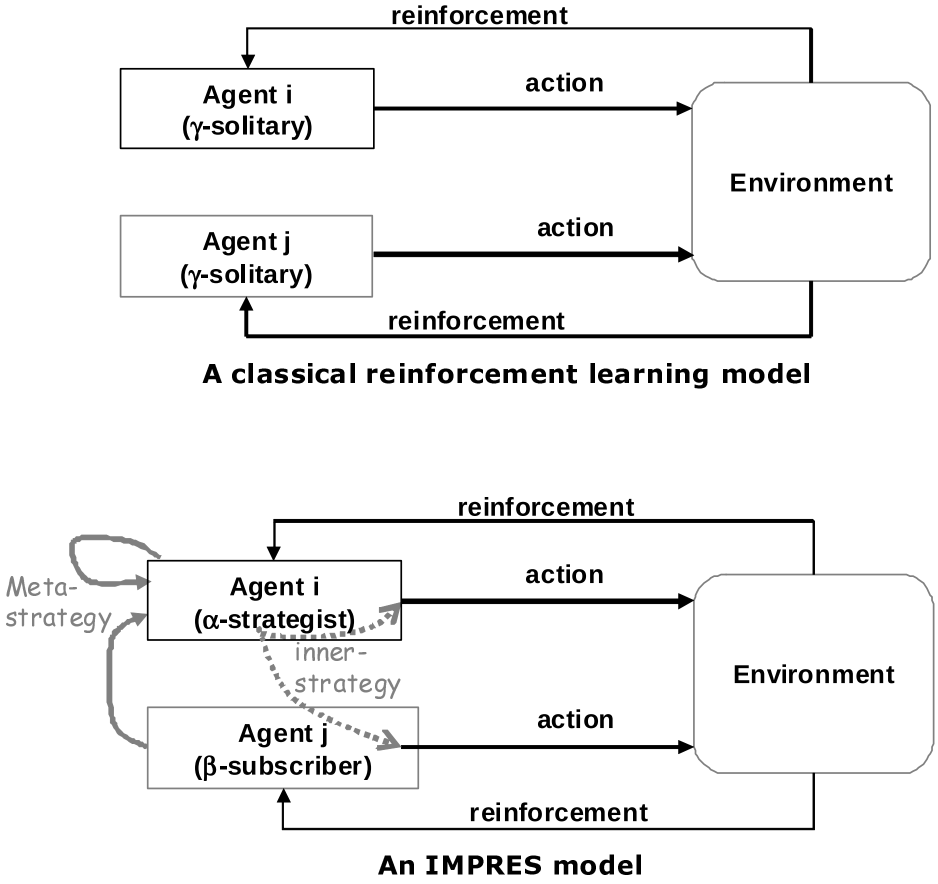


Figure 1.3: Extension to reinforcement learning: the IMPRES algorithm adds a high-level decision making to a classical reinforcement learning model such that an agent also learns to choose whose strategy will be used to select actual actions.

signals,

- β-agent (subscriber) that obeys the signals from the other driver, or
- γ-agent (solitary) that makes an independent decision.

In 2-player games such as this, an IMPRES α-strategist learns the system optimal strategy for itself and a subscriber (since that is the best-response strategy against an environment with no other players). In general, the IMPRES algorithm aims at middle-ground solutions between an independent solution and a centralized solution when there are more than two agents.

Specifically, the learning in the meta-layer is an extension to the classical reinforcement learning model. Figure 1.3 shows an abstract view of the learning in the two layers. The above figure represents a classical reinforcement learning model where each agent takes an action, receives reinforcement from an environment, and updates a policy to select better actions in subsequent rounds. Note that since the focus of this thesis is on stateless repeated games, state transition is omitted for simplicity. In this classical model, all agents can be viewed as γ-solitaries that independently make decisions.

The IMPRES model adds a meta-learning layer to this classical view. At each round of a game, an agent's current meta-strategy determines whose inner-strategy the agent should subscribe to, for instance whether to use its own strategy or to subscribe to the strategy of some other agent in the environment. According to the chosen inner-strategy, the agent selects an actual action, e.g. stop or enter the intersection. After taking the prescribed action, the agent receives reinforcement from its environment, and updates the strategies in both layers. The learning in the meta-layer induces emergent self-organization that leads an agent population towards more desirable outcomes. The IMPRES algorithm is described in detail in Chapter 4.

1.5 Motivating examples

In this section, three specific problems are presented. The first two are well-known examples of congestion games that illustrate the tragedy of the commons and the limitations of existing algorithms. The last example highlights the original insight behind this research.

1.5.1 Metro versus Driving

Let us examine a simple example of a symmetric network congestion game introduced in [49] that clearly exhibits the inefficiency of non-cooperative equilibria known as *selfish equilibria*. The following analysis is due to [53].

Suppose that there exist n self-interested agents that are deciding between two actions: taking a metro (denoted by M) or driving (denoted by D). Figure 1.4 illustrates such an example. The natural objective of this problem is to minimize the average travel time of all agents.

Let x_a denote the number of agents selecting action a, where $a \in \{M, D\}$, and let $t_a(x_a)$ be the travel time of taking action a. Note that travel time is a function of x_a, and the agents selecting the same action experience the same travel times.

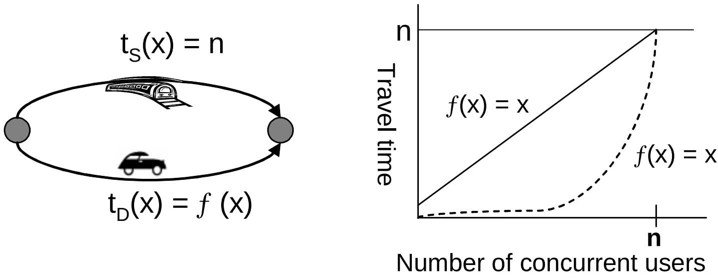


Figure 1.4: Metro versus Driving

Let us assume that $t_M(x_M) = n$ where n is a constant denoting the number of agents. On the other hand, let the travel time of driving be a linear function, $t_D(x_D) = x_D$ such that $t_D(n) = n$. That is, taking a metro takes a constant travel time that is always slower than driving except when traffic is fully congested.

In this context, driving is a dominant strategy of this game because driving is always faster than taking a metro no matter what other agents do. Therefore, self-interested agents converge to a dominant strategy equilibrium in which all agents will always choose to drive even when the road is fully congested, resulting in the average travel time of n. This is a Nash equilibrium since no one is motivated to deviate from their current choice of actions given that the choices of other agents are fixed. The average travel time of this solution is, however, suboptimal; to be precise, it is the worst possible solution.

Suppose now that there exists a centrally administered system that selects a small number of agents, ϵ, ($\epsilon < n$), and forces them to take the metro. In this case, the travel time of the selected agents is not any slower than that of their original decisions, i.e., n. This enforcement, however, enables the remaining drivers to travel faster, reducing the travel time of a driving agent to $n - \epsilon$.

Hence the average travel time of all agents is a convex function, $\frac{1}{n} \times \{n\epsilon + (n-\epsilon)^2\}$. By taking a partial derivative with respect to ϵ, an optimal value can be found when $\epsilon = \frac{n}{2}$. With this solution, the average travel time is reduced down to $\frac{3n}{4}$.

Alternatively, suppose the travel time function is non-linear (dotted line in Figure

1.4), e.g., an exponential function, $t_D(x_D) = x^P$, for some P. Then, in the limit, the travel time of driving can be ignored, i.e., $\lim_{P\to\infty}(n-\epsilon)^P = \lim_{P\to\infty}\{n^P(1-\frac{\epsilon}{n})^P\} = 0$. Thus, the average travel time of agents in this case is reduced to ϵ.

In summary, this example provides two interesting observations. First, selfish equilibrium solutions can be arbitrarily inefficient. Second, in some problems, a small subset of agents can significantly reduce inefficiency by taking altruistic actions, by which they are not worse off than under selfish equilibria, but other agents benefit from a strict decrease in cost.

In general, when a game possesses a dominant-strategy equilibrium, all stationary learning algorithms including fictitious play and no-regret algorithms converge to the dominant-strategy equilibrium that is generally suboptimal such as this. A set of experiments demonstrates that IMPRES agents learn close-to-optimal solutions in this example; and the detailed results can be found later in Section 5.5.1.

1.5.2 The El Farol bar problem

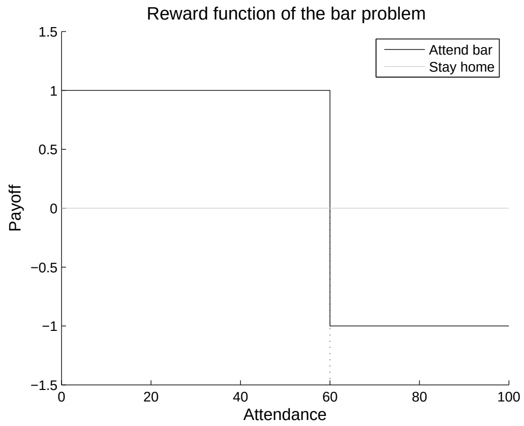


Figure 1.5: Reward function of EFBP

The El Farol bar problem (also known as the Santa Fe bar problem) introduced in [1] best illustrates the tragedy of the commons. The problem is defined as follows.

The El Farol bar presents nightly music entertainment in Santa Fe. A set of n agents make decisions about whether to attend the bar or not on certain nights. The only observations available to the agents are the past history of attendance at the bar. The reward function is discrete: attending the bar is fun only if the bar is not crowded; such that the number of attendees at the bar on the night does not exceed some threshold τ. On the other hand, an agent is better off staying home if the bar is overcrowded. An example of the reward function for $n = 100, \tau = 60$ is shown in Figure 1.5.

In this context, it can be difficult for rational agents to find optimal behaviors. For example, a rational agent decides to attend the bar if the agent predicts that the attendance at the bar will be lower than τ. Given that the other agents are also rational, everyone makes the same prediction based on the common information (history of attendance). Subsequently, the agents face contradictory outcomes based on rational decision making. That is, the agents attend the bar when the bar is full, or stay home when the bar is empty.

A formal analysis of this example explains the failure of rational learning [24]. Consider the rational learning algorithm where the agents always select the best response strategies based on their beliefs about the strategies of other agents [32]. In a nutshell, rational learning can converge to a Nash equilibrium if the agents have a "lucky" prior (although not necessarily accurate according to the truth). In general, the outcomes of rational learning in the El Farol bar problem are not even close to Nash equilibria as shown in Figure 1.6 (right).

Existing studies of the bar problem have generally focused on failure to converge, seeking algorithms that converge to selfish equilibria. For instance, agents adopting rationally bounded learning algorithms, such as an inductive reasoning algorithm [1] or a no-regret algorithm [24], converge to a symmetric mixed strategy Nash equilibrium shown in Figure 1.6 (left).

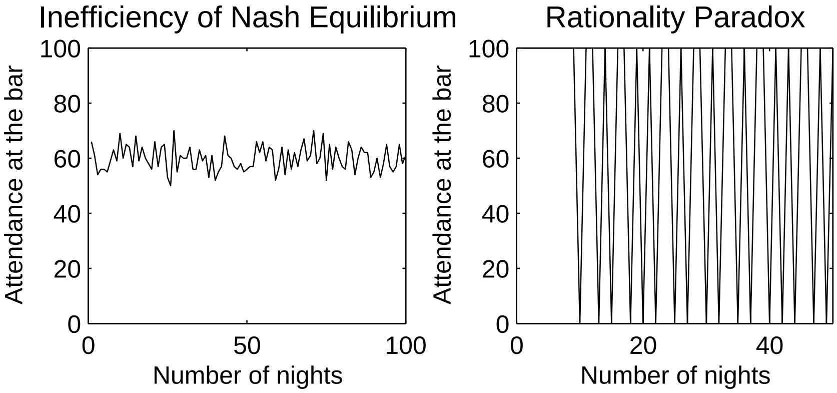


Figure 1.6: Selfish equilibria and tragedy of the commons in EFBP

As shown in Figure 1.6, however, the average payoff for the mixed-strategy Nash equilibrium (left) is as suboptimal as the oscillating solution (right), since positive payoffs on fun nights and negative payoffs on overcrowded nights offset each other.

The IMPRES algorithm, on the other hand, learns close-to-optimal solutions in this problem; corresponding experimental results are reported in Section 5.5.2.

1.5.3 CMRadar room-finding problem

The rudiments for this thesis hail from a multiagent scheduling system. Many recent studies show that *email overload* results in performance degradation at workplace [15]. The RADAR project started with an intention to develop a software system that can assist users to cope with email overload as efficiently as human assistants [22]. CMRadar [41], a scheduling component of RADAR, is a distributed calendar scheduling system wherein an individual CMRadar agent assumes responsibility for managing its user's calendar and for negotiating with other CMRadar agents 1) to schedule meetings and 2) to find rooms on its user's behalf. For clarity, we denote CMRadar performing the two tasks by a CMRadar meeting-scheduler and a CMRadar room-finder, respectively. On one hand, a CMRadar meeting-scheduler learns its user's scheduling preferences using passive machine learning algorithms by observing a series of meeting scheduling episodes [45]. On the other hand, a CMRadar room-finder agent utilizes learning to negotiate more efficiently in acquiring rooms that are already occupied by other users [20]. It is the latter that is most relevant to this thesis.

In general, each room has a set of features, providing various facilities, e.g., maximum capacity, types of table settings, size, number of projectors in the room, etc. Given a room request with a set of preferences for each room feature, quality is maximized when a room that best satisfies the preferences is found. Rooms are classified into two types: local and external. A CMRadar room-finder has complete access to room schedules of a set of local rooms within an institute, while it has only partial observation to the room schedule of external rooms, e.g., conference rooms in outside hotels. A CMRadar room-finder employs a specific learning technique according to the type of a room.

In a typical scenario, a room calendar is more than 90 percent filled in local rooms. Since a room can be reserved for only one user at a time, existing meetings are commonly rescheduled in favor of higher-priority events. When an existing reservation is preempted to accommodate a new meeting, the action is referred to as a *bumping*. In order to negotiate the possibility of bumping an existing meeting, a room-finder agent must send a bumping request to the organizer of the existing meeting (room owner). Each bumping request incurs some degree of penalty. The objective of a CMRadar room-finder in this context is to minimize the number of unsuccessful bumping requests, thus expediting the room-finding process. A CMRadar room-finder uses a Bayesian learning method to estimate the probability of a room owner approving a bumping request. More details about the learning of bumping probability can be found in Appendix A.

CMRadar's learning about external rooms can be formulated as a multiagent resource selection problem. Since CMRadar's access is limited for external rooms, bumping is not a possibility. Suppose that there exist a large number of hidden users in an environment that compete for the same set of external rooms with a CMRadar room-finder. Let us first assume that the other room users in an environment select rooms according to their stationary preferences about the set of room features. In this setting, a CMRadar room-finder can adaptively learn the collective preferences of other room users by approximating the ordinal ranks of preferred rooms. More specifically, a room-finder agent updates the conditional probability of a room being available as the agent observes more information about the availability of external rooms. For example, the agent may avoid vain attempts to acquire most commonly preferred rooms that are more likely to be taken by other users, given evidence that a less preferred room has already been taken.

Suppose now that the other users are learning concurrently in a manner similar to CMRadar; in this case, an adaptive learning method may fail to learn the best strategy, a result similar to that seen in earlier examples. Figure 1.7 depicts the tragedy of the commons for CMRadar when there exist multiple agents competing for the same set of rooms.

1.6 Thesis statement

When a large number of self-interested agents make independent decisions in a shared environment, the resulting outcome may be suboptimal. It is generally believed that a mutually desirable strategy can be enforced as a stable outcome for rational agents if the imminent threat exists that any deviator from the strategy will be punished. This thesis expands this understanding, arguing that rationally bounded agents can learn to self-organize to stabilize on mutually beneficial outcomes *without* the explicit notion of threat. Specifically, a structure of mutually dependent information sources and information recipients can emerge when two specific conditions are satisfied. First, an agent is willing to share its knowledge with other agents if it expects a better payoff by sharing than by not doing so. Second, an agent willingly takes prescribed actions from a certain information source if the agent expects a better payoff by acting according to the decisions from the source than from any other sources including itself. Under these two conditions, any correlated strategy that

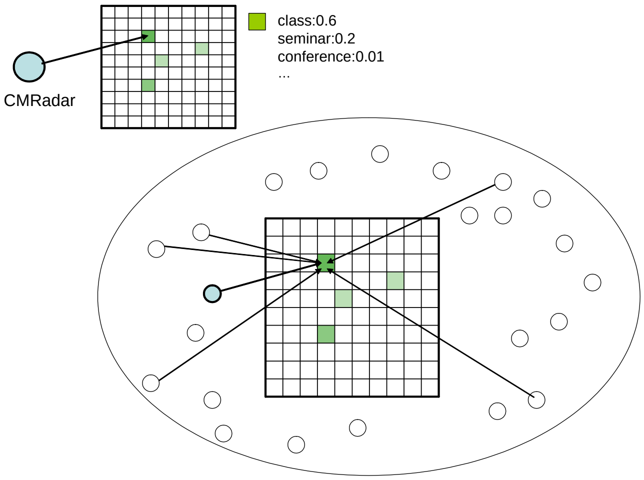


Figure 1.7: The tragedy of the commons in the CMRadar room-finding problem: given that all agents have learned about room availability from common history, consider a case where a set of CMRadar-like agents are concurrently trying to find a room. In contrast to CMRadar's belief, the rooms with a high probability of being available are more likely to be occupied because these rooms appeal to other agents at the same time.

ensures a better payoff than the agents' subjective valuation of independent strategies can be stabilized as a social norm.

1.7 Main contributions of the thesis

This thesis states that rationally bounded agents can learn to self-organize to stabilize mutually beneficial outcomes without an explicit notion of threat. As an approach to demonstrate this capability, a double-layered multiagent learning algorithm, known here as IMPRES, has been developed. In brief, IMPRES agents act as if they are perfectly rational, although in truth they choose the best actions only according to their (possibly imperfect) subjective beliefs. The IMPRES algorithm is theoretically justified: the outcome of IMPRES in self-play is *behavior-equivalent* to a Nash equilibrium of a repeated game.

Empirically, the algorithm is evaluated in the context of congestion games, where the performance of the algorithm is measured with respect to both individual and social rationality criteria. In conjunction with empirical evaluation, a set of desired properties of a multiagent social learning algorithm is proposed. The basic premise of social learning is that by acquiring new knowledge from others, the agents must be better off than by not doing so; formal definitions can be found in Chapter 5. The main set of experiments are conducted on symmetric network congestion games with linear, polynomial, exponential, and discrete cost functions.

The main results can be summarized as follows:

- The outcome of IMPRES in self-play is behavior-equivalent to a Nash equilibrium of a repeated game; that is, the outcome is individually rational in terms of expected cost.

- With respect to social welfare, the performance of IMPRES is generally close to optimal; IMPRES is the first multiagent learning algorithm that achieves this property in games involving more than 2 players.

- The algorithm is scalable to large problems involving up to 1,000 agents and networks containing 15 alternative paths.

- The algorithm is robust against moderate population changes (such as when a small number of agents are replaced with new ones over time).

- In addition to symmetric congestion games, the algorithm has also been evaluated for some well-known 2-player matrix games, namely the iterative prisoner's dilemma, the (asymmetric) coordination game, and the game of chicken. In all three games, agents adopting IMPRES learned fair and optimal solutions; the algorithm is comparable with state-of-the-art algorithms in terms of learning rate.

To sum up, this thesis proves that using social learning, rationally bounded agents can learn to behave rationally, achieving mutually desirable outcomes with respect to long-term average rewards.

1.8 Roadmap

The rest of the chapters are organized as follows.

Chapter 2 gives a broad survey on congestion games - the main target problem domain of this thesis, concentrating on the complexity analysis of finding both individually rational solutions and socially optimal ones. After discussing the complexity results, a motivation of this research is given: to find an efficient multiagent learning algorithm that can realize rational behaviors based on long-term expected rewards.

Chapter 3 through 5 constitute the core parts of this thesis. First, Chapter 3 describes the IMPRES algorithm in detail; the focus of this chapter is to explain how the algorithm works mechanically. Next, the IMPRES algorithm is theoretically justified in Chapter 4; the structure of the algorithm is meticulously analyzed within the boundaries of established theories. Finally, Chapter 5 provides a comprehensive set of empirical evidence to support the main point of this thesis: rationally bounded agents can learn to enforce mutually beneficial outcomes through self-organization.

Chapter 6 summarizes the main contributions of this thesis and discusses future work.

Chapter 2

Congestion Games

> "For that which is common to the greatest number has the least care bestowed upon it. Every one thinks chiefly of his own, hardly at all of the common interest; and only when he is himself concerned as an individual.", Aristotle, Politics, Book II(3), 350 B.C.

2.1 Introduction

Congestion games refer to a class of games where a player's immediate payoff depends only on the number of players that have also chosen the same action with the player [51]. A natural example of a congestion game is a traffic congestion problem where each driver independently chooses a path among alternative routes to reach her destination as quickly as possible, while actual travel time is determined by the traffic load of the chosen path. Congestion games constitute an important subject of research in transportation sciences, computer networks, and algorithmic game theory [44].

It is well known that every congestion game possesses at least one pure-strategy[1] Nash equilibrium. Given that, this chapter opens with a discussion of Nash equilibrium solutions in the context of congestion games in both single-shot and repeated play environments. The focal points of the discussion are: 1) how difficult, computationally, it is to find a Nash equilibrium, and 2) how efficient a Nash equilibrium solution is with respect to the quality of solution. After that, the discussion closes with a more fundamental topic of the goal of multiagent learning in repeated games.

In general, Nash equilibria have been the most celebrated solution concept both in game theory and the multiagent learning literature. In fact, congestion games were

[1] A pure-strategy refers to a deterministic strategy; see Section 2.2.

introduced as a class of games that possesses a pure-strategy Nash equilibrium [51]. This celebrated solution concept, however, carries two dark sides: computational intractability and inefficiency of the solution.

The notion of individual rationality centered around Nash equilibria depends on how agents value the future. For instance, a single-shot game can be described as a setting where agents only appreciate their immediate payoffs (as if there is no tomorrow). In a single-shot game setting, finding a Nash equilibrium in a general normal-form game is computationally intractable (PPAD-complete [48]) even in the case of 2-player games [8, 16]. More specifically, the problem of finding a single-shot Nash equilibrium of a general congestion game belongs to the most difficult class of local search problems, referred to as the complexity class PLS-complete [18].

A more serious disadvantage of a single-shot Nash equilibrium is its inefficiency. Congestion games naturally yield negative externalities; for example, an agent can unintentionally cause a delay in the travel time of other agents on the same path. Because externalities are not taken into account when agents attempt to minimize their costs, individually rational solutions, known as selfish equilibria, can be suboptimal in this domain.

One of the main contributions of algorithmic game theory is to discover interesting classes of games where the inefficiency of selfish equilibria can be tightly bounded. In nonatomic games where the impact of an individual agent is insignificant, promising results are found that selfish equilibria are not as bad as had been speculated. On the contrary, however, in atomic games where the impact of an individual agent is non-negligible, the inefficiency of selfish equilibria can be arbitrarily high even in symmetric network congestion games with linear cost functions [44]. More radically, a system-optimal solution can only be achieved through an explicit coordination among agents unless the cost function is logarithmic [40].

The next part of discussion is set on repeated games where agents take their future payoffs into consideration. According to the folk theorem, when a game is played repeatedly better-quality solutions can turn up as individually rational outcomes. Specifically, for any payoff profile where every agent receives a better payoff than the *minimax* value on average, there exists a Nash equilibrium strategy that can realize the payoff profile; where the minimax value of an agent denotes the best payoff of the agent when all other agents change their objectives to turn against the agent. For example, any correlated strategy that satisfies the premise can be sustained if all agents adopt the *grim-trigger* strategy such that a deviator from the correlated strategy will be punished by merely receiving the minimax values ever after.

In contrast to the general belief that finding Nash equilibria of a repeated game is easier (than finding single-shot Nash equilibria), computing the minimax values of a repeated congestion game turns out to be NP-complete [3]. The NP-completeness still holds even in the case of network congestion games with linear cost functions.

In addition to the complexity of finding the minimax values (punishing strategy), the folk theorem also assumes that there exists some correlated strategy (mutually desirable strategy) that Pareto dominates the minimax values; this entails additional complication. For instance, computing a system-optimal solution in general congestion games is NP-hard, and a polynomial-time approximation scheme does not exist except for limited classes of games [38]. Especially when the cost function is player-specific, it is NP-hard to achieve socially optimal solutions even in the balls-in-bins model (the simplest form of symmetric network congestion games) [7].

Given the disadvantages of Nash solutions, we can ask the following research question:

> What is a desirable target solution of multiagent learning in repeated congestion games?

The rest of this chapter is organized as follows. First, formal definitions are given for general solution concepts and congestion games. Next, discussions on the complexity and the inefficiency of selfish equilibria follow. Finally, the main point of this chapter - the fundamental question of what should be the goal of multiagent learning in repeated congestion games – is discussed.

2.2 Preliminaries

Remark. The two terms "agents" and "players" are used exchangeably. Since we treat the players of a game as (artificially intelligent) agents, a single player will be referred by a pronoun "it". After taking some action, agents receive reinforcement from an environment, which is generally referred to as a reward (payoff). In congestion games, agents select resources and receive the actual "cost" of using the selected resources. Thus, the objective of agents in a congestion game is to *minimize* the expected cost (as opposed to maximizing the expected reward).

Notations. Let $N = \{1, 2, ..., n\}$ denote a set of n players (agents) of a game. Let A_i denote a set of available actions for player i. A strategy of an agent denotes a probability distribution over the set of actions; thus, a strategy by default refers to a (stochastic) mixed-strategy. A pure-strategy is a special case when the whole probability mass is on one particular action. Let S_i denote a set of strategies of agent i over A_i. A joint strategy profile s refers to a strategy vector such that $s \in \prod_{i \in N} S_i$.

The expected cost of agent i is denoted by $c_i(s_i, s_{-i})$ where agent i chooses actions according to s_i and the other agents follow strategies s_{-i}.

Definition 1 (correlated strategy). *A correlated strategy refers to a probability distribution over a set of all possible joint strategy profiles.*

Informally, the strategies of agents are in a Nash equilibrium if and only if every agent plays its best response strategy against other agents [43]. The notion of best response depends on the setting where the agents play a game; for instance, whether the game is played only once or repeatedly[2]. Formal definitions follow, addressing each game setting.

2.2.1 Nash equilibria of a single-shot game ($\mathbf{NE}^1$)

A Nash equilibrium of a single-shot game is a solution concept where no one benefits in the immediate payoff by changing its strategy given the strategies of other agents are fixed. Formally, joint strategy profile s is in a single-shot Nash equilibrium (NE^1) if and only if for every agent $i \in N$, a unilateral deviation from strategy profile s does not reduce its cost; such that $c_i(s_i, s_{-i}) \leq c_i(s'_i, s_{-i})$ for every strategy $s'_i \in S_i$ of agent i.

Theorem 1. *Every finite game possesses an equilibrium point in which players cannot reduce their costs by a unilateral deviation [43].*

2.2.2 Nash equilibria of a repeated game ($\mathbf{NE}^\infty$)

Let G^∞ be a repeated game in which game G is repeated indefinitely. The constituent game G is also called a stage game. A payoff vector $V = [v_1, ..., v_n]$ is an n-tuple of real numbers where v_i denotes the payoff (cost) of player $i \in N$. A vector of the worst possible payoff values that the other agents can force against an agent is referred to as the vector of *minimax values* (also known as the threat point).

A convex combination is a linear combination of values where the linear coefficients are non-negative and sum to 1. The set of points that can be formed by convex combinations of a set of points X is called a convex hull of X. A payoff vector V is *feasible* if and only if V is in the convex hull of payoff vectors that can be achieved when the agents play pure strategies; that is, a feasible payoff vector can only be realized when the agents play according to some correlated strategies.

A payoff vector is *enforceable* (or *individually rational*) if every agent $i \in N$ is better off than its minimax value.

[2] This thesis studies infinitely repeated games as opposed to finitely repeated games.

Theorem 2 (The folk theorem)**.** *For any payoff vector V that is both feasible and individually rational, there exists a Nash equilibrium strategy s such that the payoff vector V represents the average payoffs of strategy profile s [23].*

The folk theorem dictates that any outcome that Pareto dominates[3] the minimax payoff can be enforced by an equilibrium strategy.

A set Ψ^{G^∞} of Nash equilibria of a repeated game G^∞ subsumes the set Ψ^G of Nash equilibria of a constituting game G, such that $\Psi^{G^\infty} \supseteq \Psi^G$.

2.3 Definitions

A congestion game, which was first introduced in [51], is an n-player game in which players share a common set of resources. Thus, a set of available resource selection strategies for each agent can be represented as a subset of the common resources. A congestion cost function is defined in terms of the number of agents that have chosen the same resource. For instance, a linear cost function indicates that the cost of using a resource increases linearly with respect to the number of agents using the same resource. It is well known that every game in this class possesses a pure-strategy Nash equilibrium.

This section includes concepts in a broad context of congestion games; and Definition 3, 4, and 8 are most relevant to the rest of discussions.

Definition 2 (General congestion games)**.** *Congestion game Γ is defined as a quadruple $(N, E, S^{i\in N}, F^{e\in E})$ where N is a set of agents; E is a set of resources; $S = S_1 \times ... \times S_n$ is a set of joint strategies of N where agent i's strategy S_i is a subset of resources such that $S_i \subseteq E$; and $F^{e\in E}$ denotes a set of cost functions. Given a strategy profile $s \in S$, let the number of agents using resource e be denoted by $\sigma_e(s) = \sum_{i=1}^{n} \sum_{e' \in s_i; e'=e} 1$, where s_i denotes the strategy of agent i prescribed by s. Each $f_e(l) \in F$ corresponds to a cost function of resource e defined in terms of load $l \in [0, n]$. Given a strategy profile s, the cost $c_i(s)$ of player $i \in N$ is the sum of all resource costs in its chosen subset of resources s_i, such that $\sum_{e\in s_i} f_e(\sigma_e(s))$.*

Every agent makes decisions at the same time, and finds out the congestion cost of each resource in the chosen set only after using the resource.

Definition 3 (Network congestion games)**.** *Network congestion games refer to a special class of congestion games that can be represented more succinctly as a directed*

[3] A payoff vector x Pareto dominates another payoff vector y, if in payoff vector x at least one agent is better off and no one else is worse off than in payoff vector y.

graph $G = (V, E)$ in which a set E of edges represents a set of common resources. In this model, each player $i \in N$ is given a pair of vertices (s, t) for its origin (source) and destination (sink), thus a set s_i of strategies available to player i is a set of simple (acyclic) paths between vertices s and t.

Definition 4 (Symmetric congestion games)**.** *A congestion game is* symmetric *when all agents have the same set of strategies (resources) in addition to the common cost function.*

Definition 5 (Player-specific cost congestion games)**.** *Cost function F can be customized for each agent as $F^{e \in E, i \in N} : N \times E \times [0, 1, ..., n] \rightarrow \Re$, such that $f_{e,i}(l)$ defines agent i's cost of using edge e as a function of load l.*

Milchtaich's work [39] was the first to address player-specific cost functions in congestion games, and to prove that a congestion game possesses at least one pure-strategy Nash equilibrium even when the cost functions are player specific. Unless otherwise specified, a common cost function is assumed in this thesis.

Definition 6 (Multi-commodity flow model)**.** *In networks research, as opposed to having a distinct set of agents in the model, a multi-commodity flow model is more commonly used. A commodity is defined in terms of a source and sink pair, and a demand for such a pair. For instance, a single-commodity flow model is equivalent to a symmetric congestion game. A general k-commodity flow model can also be viewed as a congestion game that has k classes of agents where the members of each class share a common source and sink pair, thus the size of each class determines the demand for the corresponding commodity.*

In general, a multi-commodity flow problem is a (centralized) optimization problem to find an assignment of flows to a set of paths to minimize the cost of transmitting the flows over the edges in the path. A distributed agent-based model is ideal for source-routing networks in which each end-user makes independent decisions in choosing its own route [44, pg. 461–463].

Definition 7 (Atomic/nonatomic congestion games)**.** *When the impact of each individual agent on congestion cost is nonnegligible, it is called an atomic game. On the other hand, when individual action has little or no significance to the cost function, it is called a nonatomic game. A multi-commodity model suits a nonatomic congestion game more naturally since a flow can be continuous.*

This thesis focuses on atomic congestion games.

Definition 8 (Minimum-cost-flow algorithm)**.** *A minimum-cost-flow algorithm iteratively chooses a path for one agent at a time that has the lowest travel cost according to the current load. An everyday example of this algorithm can be found at a cashers' line where customers sequentially choose the shortest line among several cashers [12].*

The algorithm needs to evaluate the cost of all alternative paths for each agent, thus the complexity is polynomial to the number of agents and to the number of alternative paths. This algorithm will be used to establish proofs in the later sections; and it will also be used as a subroutine of the proposed algorithm.

2.4 Existence of pure-strategy Nash equilibria

This section gives a historical background of congestion games. For those readers who are not interested in proofs, this section can be summarized in Theorem 3 that every congestion game possesses at least one pure-strategy Nash equilibrium.

Rosenthal introduced congestion games as a class of games that admits at least one pure-strategy Nash equilibrium [51]. Rosenthal's method of proof is directly related to the potential approach later studied by Monderer and Shapley [42]. Specifically, Rosenthal's proof involves an exact potential function. This section describes a proof that consists of two lemmas using the potential approach.

Consider any two pure-strategy profiles (x_i, s_{-i}) and (z_i, s_{-i}) that differ only in the strategy of some agent i. Let "deviator" refer to the agent that has changed the strategy. A *potential* is a value assignment to each strategy profile that gives the same ordering of the two strategy profiles as when they are sorted with respect to a deviator's payoff. Specifically, for all agent $i \in N$, a potential Φ satisfies the following.

$$c_i(x, s_{-i}) > c_i(z, s_{-i}) \iff \Phi(x, s_{-i}) > \Phi(z, s_{-i}), \forall x, z \in A_i$$

Generally, a class of games that admits a (ordinal) potential is referred to as *potential games.*

Definition 9 (Nash dynamics graph)**.** *A finite game can be represented as a directed graph $G = \{S, D\}$ where each vertex $s \in S$ represents a joint strategy profile, and each arc $(u, v) \in D$, where $u, v \in S$, represents a unilateral deviation such that strategy vertices u and v differ only in exactly one player's strategy. Each arc is directed towards a more favorable strategy profile for the deviator. Such a graph is called a Nash dynamics graph.*

Lemma 1. *Every finite game that admits an ordinal potential possesses a pure-strategy Nash equilibrium [42].*

Proof. Let G denote a Nash dynamics graph of some game. If there exists a node in a Nash dynamics graph G that has only inwards arcs (i.e., the node is a sink) the strategy profile that is represented by the node is a pure-strategy Nash equilibrium by its definition - no one benefits by a unilateral deviation. Suppose there exists a valuation Φ to every node such that the same graph can be generated by using the values in Φ in replace of a deviator's payoff. For instance, the direction of an arc is consistent with the original graph if it points to a node that has a lower value (cost) of Φ. Such a valuation is called an *ordinal potential* since it preserves the ordering of the nodes. Let G^{Φ} denote this new graph. Straightforwardly, a sink node in the new graph G^{Φ} coincides with a pure-strategy Nash equilibrium of the original graph G.

The proof follows the fact that a sink node exists in the new graph. Since there is a finite number of strategy profiles, a potential also has a finite set of values. Thus, a node with the minimum potential value exists in a finite space. Any nodes that are associated with the minimum potential value can only have inwards edges since the node has the best value (lowest cost), thus are Nash equilibria of the game (although they may not be the only ones). Therefore, there must be at least one pure-strategy Nash equilibrium in every game that has a finite potential.

More generally, if the Nash dynamics graph satisfies the finite improvement property (FIP) - the length of any path towards an improvement is finite, then a Nash equilibrium exists. Every potential game possesses the finite improvement property since a valuation guarantees that the graph is acyclic. □

Let us show that a congestion game is a potential game by using the exact potential function that was used by Rosenthal.

Lemma 2. *Every congestion game is a potential game.*

Proof. Given congestion game $\Gamma = (N, E, S^{i \in N}, F^{e \in E})$, Rosenthal defined the following valuation function $\Phi(s)$ for each joint strategy $s \in S$.

$$\Phi(s) = \sum_{e \in s} \left(\sum_{l=1}^{\sigma_e(s)} f_e(l) \right)$$

Let us prove that function Φ is an exact potential. That is, for any two nodes that are connected by a unilateral deviation arc in the Nash dynamics graph, the difference in the function values using Φ matches exactly with the difference in the payoffs of the deviator of the arc.

The proof is straightforward by regrouping terms. Let d denote a deviator, and let s and s' denote strategy profile before and after the deviation, respectively. Let $\Delta = \Delta^+ \cup \Delta^-$ be a set of edges (resources) involved in deviation, such that Δ^+ denotes a set of new edges that are added to the deviator's path after the deviation, and Δ^- denotes a set of edges that have been removed from the path.

Let us write the potential value of the strategy after a deviation in terms of Δ.

$$\begin{aligned}
\Phi(s') &= \sum_{e\in s; e\notin\Delta}\left(\sum_{l=1}^{\sigma_e(s)} f_e(l)\right)\\
&+ \sum_{e\in\Delta^+}\left(\sum_{l=1}^{\sigma_e(s)} f_e(l) + f_e\{\sigma_e(s)+1\}\right)\\
&+ \sum_{e\in\Delta^-}\left(\sum_{l=1}^{\sigma_e(s)} f_e(l) - f_e\{\sigma_e(s)\}\right)\\
&= \left[\sum_{e\in s; e\notin\Delta}\left(\sum_{l=1}^{\sigma_e(s)} f_e(l)\right) + \sum_{e\in\Delta^+}\left(\sum_{l=1}^{\sigma_e(s)} f_e(l)\right) + \sum_{e\in\Delta^-}\left(\sum_{l=1}^{\sigma_e(s)} f_e(l)\right)\right]\\
&+ \sum_{e\in\Delta^+} f_e\{\sigma_e(s)+1\} - \sum_{e\in\Delta^-} f_e\{\sigma_e(s)\}\\
&= \Phi(s) + \sum_{e\in\Delta^+} f_e\{\sigma_e(s)+1\} - \sum_{e\in\Delta^-} f_e\{\sigma_e(s)\}
\end{aligned}$$

Then, the difference in potential values can be written as

$$\Phi(s') - \Phi(s) = \sum_{e\in\Delta^+} f_e\{\sigma_e(s)+1\} - \sum_{e\in\Delta^-} f_e\{\sigma_e(s)\},$$

which is precisely the difference in the costs of deviator d after the deviation.

$$\Phi(s') - \Phi(s) = c_d(s') - c_d(s)$$

Therefore, function Φ is an exact potential of congestion game Γ. Since congestion game Γ admits an exact potential, congestion game Γ is a potential game. □

Theorem 3. *Every congestion game possesses at least one pure-strategy Nash equilibrium [51].*

The proof directly follows Lemma 1 and Lemma 2.

Remark. A potential is strictly based on the difference in the payoffs of each individual player. Therefore, optimization of a potential function is irrelevant to social welfare.

2.5 Complexity of solving congestion games

This section discusses how difficult it is to find a solution for a congestion game. The first two sections of discussion involves individually-rational solutions in the single-shot game and repeated game setting respectively, and the last section is on socially-rational (centralized optimization) approaches. This section surveys existing complexity results including proofs. For those readers who are not interested in proofs, this section can be summarized as follows: generally in congestion games, finding a Nash equilibrium is computationally intractable both in single-shot and repeated play environments; and computing a socially optimal solution or a fair solution is also intractable.

2.5.1 Finding Nash equilibria of a single-shot game

The complexity results in this section are from [18]. Let us first define the complexity class PLS-complete [31].

Definition 10 (Complexity class PLS (Polynomial Local Search))**.** *A problem in PLS defines an optimization problem, the goal of which is to find a local optimum as opposed to a global optimum. The problem instance is represented in a language L of a binary string, and so are its solutions. Each solution is associated with a cost and a set of neighboring solutions.*

- *Given an instance x and a set S^x of its solutions, the length of each solution $s \in S^x$ is bounded by the length of input x according to some polynomial function p, such that $s \in \{0,1\}^{p(|x|)}$.*
- *A polynomial function λ_1 exists, such that given an input x it can determine whether $x \in L$, and if so, an initial solution $s_0 \in S^x$ is returned.*
- *A polynomial function λ_2 exists, such that given $x \in L$ and $s \in \{0,1\}^{p(|x|)}$, it can determine whether $s \in S^x$, and if so, the cost of solution s denoted by $c(s)$ is returned.*

- *A polynomial function λ_3 exists, such that given $x \in L$ and $s \in S^x$ it can determine whether solution s is a local optimum in terms of cost c. If not, a strictly better solution in the neighborhood $s' \in N(s)$, where $N(s)$ denotes a set of neighbors of s, is returned.*

Lemma 3. *A problem L is in PLS-complete if L is in PLS, and L is reducible from some $L' \in$ PLS-Complete.*

Theorem 4. *It is PLS-complete to find a pure-strategy Nash equilibrium in congestion games of the following classes [18]:*

(i) Symmetric congestion games

(ii) Asymmetric network congestion games

(iii) General congestion games

The proofs for the asymmetric congestion games and general network congestion games are somewhat involved, which I refer to the work in [18]. This thesis instead focuses on symmetric cases.

Proof. The symmetric congestion game case is proved by constructing a symmetric congestion game from an asymmetric game. Consider an asymmetric congestion game $\Gamma = (N, E, S, F)$ where S_i denotes a distinct set of strategies of each player $i \in N$. For each set S_i, a bogus edge $e_i^{0,\infty}$ is created, such that its cost is zero for a single user, or infinitely large for two or more users. Let S'_i denote a new set of strategies such that $S'_i = \{s \cup \{e_i^{0,\infty}\}, \forall s \in S_i\}$. Consider now a symmetric congestion game $\Gamma' = (N, E \cup_{i\in N} \{e_i^{0,\infty}\}, \bigcup_{i\in N} S'_i, F)$, such that every player shares the union of all strategies. In any equilibrium of this symmetric game Γ', it is implicitly forced that exactly one from each original set of strategies can be taken due to the infinite cost of bogus edges. Therefore, the solutions of symmetric game Γ' after excluding the bogus edges coincide with those of the original asymmetric game Γ. Since the problem of finding a Nash equilibrium of a symmetric congestion game is reducible from that of an asymmetric congestion game, finding a pure-strategy Nash equilibrium in symmetric congestion games is PLS-complete. □

Theorem 5. *A pure-strategy Nash equilibrium of a symmetric network congestion game can be found in polynomial time [18].*

Proof. Given a network congestion game $G = (V, E)$ and n agents, consider a new game $G' = (V, E')$ where each edge $e \in E$ in the original game G is substituted by n unit-capacity edges $(e_1, ..., e_n)$, such that the cost of each new edge e_k is $f_e(k)$. Let

$\sigma_s(e)$ denote the load on edge e. Given strategy profile s, a potential function Φ of game G is the sum of edge cost of new game G' as follows:

$$\Phi(s) = \sum_{e \in E} \sum_{j=1}^{\sigma_s(e)} f_e(j) = \sum_{e' \in E'} f_{e'}(1)$$

That is, minimizing the sum of edge-cost of game G' minimizes the potential of the original game G. Therefore, the minimum-cost flow of new game G' is a pure-strategy Nash equilibrium of the original game G. □

2.5.2 Finding Nash equilibria of a repeated game

According to the folk theorem, there may exist infinitely many Nash equilibria in a repeated game. Ironically, abundance does not mean that it is easy to find. The folk theorem relies on the premise of a tangible threat; that is, an equilibrium strategy is enforceable due to a high risk of retaliation. Thus, it is safe to say that the hardness of finding an equilibrium depends on how hard it is to find a concrete threat such as a vector of minimax values. Unfortunately, finding a vector of minimax values for a repeated congestion game is an intractable problem. In what follows, complexity analysis on general congestion games is discussed.

Minimax values

In 2-player zero-sum games, the minimax value can be efficiently computed using linear programming. In any 2-player normal-form games, a pair of minimax values for a row player and a column player can be computed efficiently by substituting the game with two zero-sum games; such that in one game, a row player forgets its own payoff matrix, and instead uses the complement of the column player's payoffs, and vice versa.

A similar approach can be applied to n-player games, which provides an informal insight in understanding the complexity of computing the minimax values. For simplicity, assume that every agent has the same number of actions, say k. An n-player game can be substituted with a set of n 2-player zero-sum games. That is, for each player $i \in N = \{1, ..., n\}$, construct a zero-sum game of player i against a team of the other players $N - \{i\}$. Then, a set of actions for the team becomes a set of all possible joint actions of $n - 1$ agents. In terms of the number of actions k in the original game, therefore, the size of payoff matrix is $k \times k^{n-1}$ for each of the n zero-sum games.

Complexity of computing the minimax values

This section describes a formal complexity result for finding the minimax values. The following analysis is due to [3], and I elaborated the proof by constructing an example.

Theorem 6. *Computing the threat point (a vector of minimax values) in a network congestion game on a Directed Acyclic Graph (DAG) is NP-complete [3].*

Proof. The proof follows a reduction from a SAT problem that belongs to the complexity class NP-complete. Consider a SAT problem that have 2-variable clauses or 3-variable clauses only. Let $V = \{x_1, ..., x_n\}$ denote a set of variables in an expression. Each variable $v_i \in V$ appears three times: once positively and once negatively in any 2-variable clause, and once in a 3-variable clause either positively or negatively; for example, $(x_1 \vee \bar{x}_2) \wedge (x_2 \vee \bar{x}_3) \wedge (x_3 \vee \bar{x}_1) \wedge (x_1 \vee x_2 \vee x_3)$.

From this SAT construction, a network congestion game can be constructed by representing each variable $x_i \in V$ as a player i with source s_i and sink t_i. Each clause is represented as clause-edge (u, v), the cost $c(u, v)$ of which is defined in terms of the number of users of the edge denoted by $\sigma(u, v)$, such that $c(u, v) = 0$ if $\sigma(u, v) \leq 1$, $c(u, v) = 1$ otherwise ($\sigma(u, v) > 1$). Those edges that are not clause-edges have the cost of zero.

Let 2-edge and 3-edge denote clause-edges for 2-variable clause and 3-variable clause, respectively. For every 2-edge (u, v) representing a clause $(x_i \vee x_j)$, node u has two incoming edges from the sources of its variables, s_i and s_j, and two outgoing edges to the sinks of its variables, t_i and t_j. In addition, node v may have an additional outgoing edge to a 3-edge that also has either variable x_i or x_j, such that every 3-edge is connected from three 2-edges that also have one of its three variables in them. Each 3-edge has three outgoing edges to the sinks of its three variables.

Now suppose an additional player z for whom n players are computing a minimax value. The source and sink of additional player is linked to all clause edges (both 2-edges and 3-edges), such that the player has access to all alternative paths in the graph; that is, two new edges (s_z, u) and (v, t_z) are added for each clause-edge (u, v). Figure 2.1 shows a network congestion game reduced from the simple example of an expression above. Every clause-edge is labeled with a clause accordingly, and a solid line is used for a positive appearance of a variable, and a dotted line for a negative one. For visibility, the node and edges of the additional player z are omitted from the graph.

If the expression is satisfiable, exactly one variable should be positive in each 2-edge, which implies that each 2-edge is already filled with one user. Thus, whichever path player z chooses, its cost is 1 at best. In other words, the worst cost (the

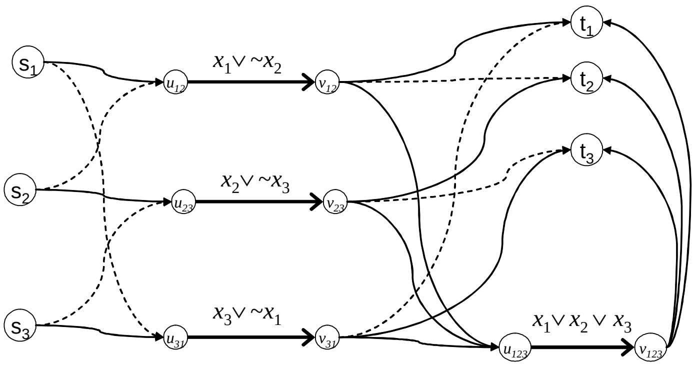


Figure 2.1: Network congestion game reduced from $(x_1 \vee \bar{x}_2) \wedge (x_2 \vee \bar{x}_3) \wedge (x_3 \vee \bar{x}_1) \wedge (x_1 \vee x_2 \vee x_3)$

minimax value) that the other n players can force against player z is 1. If the expression is unsatisfiable, it means that at least one clause-edge is not used by any other players. In the worst case, there can be only one open path and $3n-1$ paths of cost 1. If a path is chosen at random, therefore, the worst possible cost of player z is $\frac{3n-1}{3n}$. □

The NP-completeness still holds even in the case of network congestion games with linear cost functions. Furthermore, it is still an open question whether there exists a class of non-trivial games for which Nash equilibria of a repeated game can be found in a reasonable time.

2.5.3 Computing social welfare and fairness

This section summarizes the work from [38] and [7]. If a central administrator can control the agents' decision making, a congestion game can be optimized with respect to more global objectives than individual rationality, such as social welfare or fairness. Social welfare is typically measured by the sum of expected payoffs of all agents. On the other hand, fairness is generally measured by the payoff of the worst performing agent in the population.

In terms of complexity, optimizing for social welfare in congestion games is generally NP-hard even in the case of symmetric network congestion games with non-decreasing cost functions. In particular, if the cost function is player-specific, it is NP-hard to compute optimal social welfare or fairness even in the simplest form of symmetric network congestion games, known as the “balls-in-bins” model [7]. A comprehensive analysis can be found in [38], from which excerpts on network congestion games are copied in Table 2.1.

cost \ type	symmetric	asymmetric
nondecreasing	NP-hard	NP-hard; inapprox.
convex nondecreasing	P	NP-hard; inapprox.
nonincreasing	P	NP-hard; inapprox.
concave nonincreasing	P	NP-hard
nonmonotomic	NP-hard;inapprox.	NP-hard; inapprox.

Table 2.1: Complexity of finding system-optimal solutions of network congestion games [38] (notion inapprox. indicates that no polynomial time approximation scheme exists)

In the special case of symmetric network congestion game with convex nondecreasing cost functions, there exists a polynomial-time algorithm to compute a system-optimal solution.

Theorem 7. *A system-optimal solution of a symmetric network congestion game with a convex nondecreasing cost function can be found in polynomial time [38].*

Proof. Let $c_e(x)$ denote a cost function of a resource when x agents are using resource e. A cost function is *nondecreasing* if adding an additional agent on a resource does not decease the cost of the resource; that is, $c_e(x+1) \geq c_e(x)$. A cost function is *convex* if the rate of cost increase in relation to the load is also nondecreasing; that is, $c_e(x+1) - c_e(x) \geq c_e(x) - c_e(x-1)$.

Given a network congestion game $G = (V, E)$ and n agents, consider a new game $G' = (V, E')$ where each edge $e \in E$ in the original game G is substituted by a set of n unit-capacity[4] edges ($E'_e = e_1, ..., e_n$), such that the cost of each new edge e_k is $kc_e(k) - (k-1)c_e(k-1)$.

Since the cost functions are nondecreasing, it can be said $c_e(k-1) - c_e(k-2) \geq 0$. Also by definition, convex functions satisfy $\{c_e(k) - c_e(k-1)\} - \{c_e(k-1) - c_e(k-2)\} \geq 0$. Therefore, if the cost functions are convex nondecreasing, the costs of the new edges are in an increasing order as follows:

$$\begin{aligned} c_{e_k}(1) &\geq c_{e_{k-1}}(1) \\ \{kc_e(k) - (k-1)c_e(k-1)\} &\geq (k-1)c_e(k-1) - (k-2)c_e(k-2) \\ k[\underline{\{c_e(k) - c_e(k-1)\} - \{c_e(k-1) - c_e(k-2)\}}] &+ 2\{\underline{c_e(k-1) - c_e(k-2)}\} \geq 0 \end{aligned}$$

[4] A unit-capacity edge can accommodate at most one agent.

Since the costs of the new edges are in an increasing order, the new edges will be occupied in that order under the minimum-cost-flow algorithm. Subsequently, the sum of the new edge costs becomes the sum of costs of the agents that have chosen the original edge e, such that when k edges have been chosen from the new set E'_e:

$$\sum_{e'=e_1}^{e_k} c_{e'}(1) = c_e(1) + (2c_e(2) - c_e(1)) + ... + (k(c_e(k) - (k-1)c_e(k-1)) = kc_e(k)$$

Therefore, a minimum-cost flow of new game G' minimizes the total cost of all agents in the original game G in the case of nondecreasing convex functions. □

This theorem provides a kernel for a subroutine of the proposed approach in Chapter 4.

2.6 Inefficiency of Nash equilibria

This section highlights the main point of this chapter. As seen in the example of metro versus driving in Section 1.5.1, Nash equilibrium solutions can be substantially suboptimal. While the inefficiency of single-shot Nash equilibria has been a core subject of algorithmic game theory, the issues remained underexplored on what it takes to actually implement the folk theorem to reach Nash equilibria of a repeated game. In the next subsections, the inefficiency of Nash solutions and existing remedies for coping with the inefficiency are discussed in detail.

2.6.1 Price of Nash equilibria in single-shot games

The price of anarchy was introduced in [35] as a criterion for measuring the inefficiency of selfish equilibria. In this context, the objective function value of a solution refers to social welfare (sum of all agents' cost). Let φ_s denote the objective function value of a target problem under strategy profile s; S a set of selfish equilibrium strategy profiles; and o the optimal strategy profile. The price of anarchy, denoted here by $\$^A$, is defined as the worst ratio of the objective function value of a selfish equilibrium to that of the optimum. Formally,

$$\$^A = \max_{s \in S}(\frac{\varphi_s}{\varphi_o}).$$

In nonatomic network congestion games with affine cost functions (in the form of $ax + b$), the price of anarchy is bounded by $\frac{4}{3}$, meaning that a single-shot Nash equilibrium is virtually optimal. The price of anarchy can be, however, higher in atomic games even in the case of affine cost functions. When the cost functions are nonlinear, the price of anarchy of a congestion game can be arbitrarily high [53].

Existing approaches to reducing the price of anarchy seek solutions from two different sources: one from the environment and the other from the users of the environment. The former addresses making adjustments directly to the environment to make it more efficient. Specific examples are to increase resource capacity or redesign of the network routing structure [53]; or to design an efficient mechanism [19, 6]. The interest of this thesis resides in the latter, assuming that the environment is not under our control.

Generally in congestion games, optimal solutions cannot be achieved without an external intervention or an explicit coordination among agents [40], such that some subset of agents must conduct altruistic acts at times.

Learning of a periodic policy was introduced in [60] in which agents alternate a set of unfair Nash equilibria. In their problem domains, it was assumed that agents have access to the performance of other agents. Subsequently, agents act under "homo-egualis" principle, thus fairness is embedded inside the agents' objective function. For instance, an agent is evaluated not only by its individual performance, but also by the score of the poorest performing agent in the population.

Another common method is to install a centralized control to force a set of agents to take certain actions that are dictated by the administration as opposed to their choices of actions. Although a completely centralized approach is avoided due to practical reasons, a mixed model of selfish agents and centrally managed agents is commonly used in practice. Virtual Private Network (VPN) is such an example in which intermediate nodes are centrally managed while private users still make independent decisions [34].

The Stackelberg strategy is another partial centralization approach [52], wherein a set of (market) leaders make moves first, inducing desirable responses from the followers. This approach requires leaders to always sacrifice their own benefits because followers will still choose selfish actions regardless of what moves leaders make. For instance, a Stackelberg strategy performs poorly if a leader adopts a proportional strategy such that it shares the burden only proportionally in the hope that the followers will also share the remaining burden. Thus, a Stackelberg strategy always exploits the centrally controlled set of agents since the followers are not obligated nor motivated for altruistic acts.

2.6.2 Price of Nash equilibria in repeated games

Hitherto, the quality of solution has been measured based strictly on congestion costs. In that sense, the concept of Nash equilibria of a repeated game is established on the premise of low-cost solutions. In this section, let us discuss what must be traded in for the (congestion) cost reduction to realize Nash equilibria of a repeated game.

Recall that the folk theorem is based on two policies: a mutually beneficial correlated strategy and a punishing strategy. Subsequently, implementing the folk theorem requires two prerequisite computations: a good-quality correlated strategy and a vector of threat (minimax) values. As discussed earlier, these are computationally elaborate tasks. For the moment, assume that these two steps have been undertaken. Let us now probe further hidden costs.

In order to implement a correlated strategy, agents must have a common source of information to observe. For instance, an external mediator may be needed for providing signals to all agents, which entails a notion of centralization. When agents are physically distributed, an agent may be able to receive a signal from a mediator only through explicit communications. In such cases, agents must reason about tradeoffs between congestion-cost benefits and communication-cost expenses.

In order to punish a deviator, agents first must be able to detect a deviator. It is commonly assumed that every agent has a complete view of game to detect a deviator and to simultaneously enact retaliation against the deviator. In the context of congestion games, however, it is unrealistic to assume that an agent can observe the other agents' strategies even when they are not on the same path. Given that, the punishing strategy can only be implemented in a centralized way similar to the case of a correlated strategy.

In summary, a Nash equilibrium of a repeated game can be viewed as an agreement among the constituents of a society for following a certain rule given that any violator will be penalized by the society. The implementation of such rules generally involve centralized methods, which may incur extra expenditure such as communication costs.

2.7 Summary

A Nash equilibrium is a beautiful solution concept that best reflects the notion of individual rationality, and has been the limelight of the multiagent learning literature. In this chapter, we discussed various aspects of Nash equilibria in two different game settings: single-shot and repeated play environments. The problem of finding

Nash equilibria is computationally intractable in both settings. In terms of solution quality, Nash solutions can be arbitrarily suboptimal especially in a single-shot game setting. In a repeated game setting where better quality solutions can be sustained, an actual implementation of Nash solutions may entail centralized methods involving coordination costs.

In fact, the set of individually rational payoffs (as defined by the folk theorem) are desirable in any repeated games. Nonetheless, actual strategy structures that support those payoffs and how the strategies will put into operation are underexplored. In particular, it has received little attention that realizing a correlated strategy among a large number of agents may involve significant overhead.

Having said that, the focus of this thesis should not be confused with convergence to Nash equilibria. Instead, the goal of multiagent learning in this thesis is to efficiently find stable strategy structures that can bring about individually rational payoffs. Under imperfect monitoring, agents can be rational only according to their subjective information. Thus, the solutions found by subjectively rational agents may or may not coincide with objectively[5] rational solutions such as Nash equilibria. This chapter mainly discussed objectively rational solution concepts, which will provide a basis for later discussions in Chapter 4, where subjectively rational solutions are analyzed within the boundaries of well-known objectively rational solution concepts including Nash equilibria and correlated equilibria.

[5] The notion of Nash equilibria is generally accepted as an objectively rational solution concept. I discuss a different view in Chapter 4.

Chapter 3

Multiagent social learning

> “There are too many ideas, things, and people. And, too many directions to go. I was starting to believe that the reason it matters to care passionately about something, is that it whittles the world down to a more manageable size.” –Susan Orlean, The orchid thief, 1998

3.1 Introduction

This chapter introduces the multiagent learning approach that constitutes the heart of this thesis. Multiagent learning refers to an agent’s learning of optimal behaviors with respect to the long-term expected reward, when the reward depends not only on the agent’s strategy but also on the strategies of other (possibly also learning) agents in an environment. In essence, this thesis proposes the idea of “learning from others” [57]. Generally, learning indirectly from the experiences of others (as opposed to one’s own experiences) is referred to as *social learning*. This thesis initiates an effort to establish the notion of social learning in the context of multiagent learning, and proposes a specific social learning algorithm, known here as IMPRES (implicit reciprocal strategy learning), where agents learn to act more rationally by using the strategies *given* by others. An earlier work was published in [46].

The multiagent social learning model is a break from earlier thinking in two general assumptions about other agents. First, when an agent is learning in the presence of other agents, those other agents in the environment are generally considered only as additional sources of uncertainty that obscures the agent’s decision-making process. Under this assumption, the agents may be limited to suboptimal decisions by neglecting a possibility of finding mutually beneficial solutions. In contrast, my approach views other agents as potential “sources of information” that may be able to facilitate the learner's decision making even in a competitive setting.

Second, the majority of work in game theory is based on the assumption that the agents in an environment are of equal ability. My research takes a contrasting view. The foundation of my work is inspired by the concept of "bounded rationality", where some agents may have more privileges than others either because they are exposed to different parts of information in the environment, or because they simply have higher computational power.

Based on these two intuitions, this thesis investigates how agents can improve their performances by utilizing the presence of other agents in an environment. Note that the notion of social learning is different from coalitional scenarios where self-interested agents form a prior agreement based on a full evaluation of potential benefit; this view point will be discussed further in Chapter 6.

From the game theoretic perspective, it is generally believed that rational agents can abide by a mutually beneficial strategy if the imminent threat exists that any deviator from the strategy will be punished. This thesis expands this understanding, arguing that rationally bounded agents can learn to self-organize to stabilize mutually beneficial outcomes *without* the explicit notion of threat.

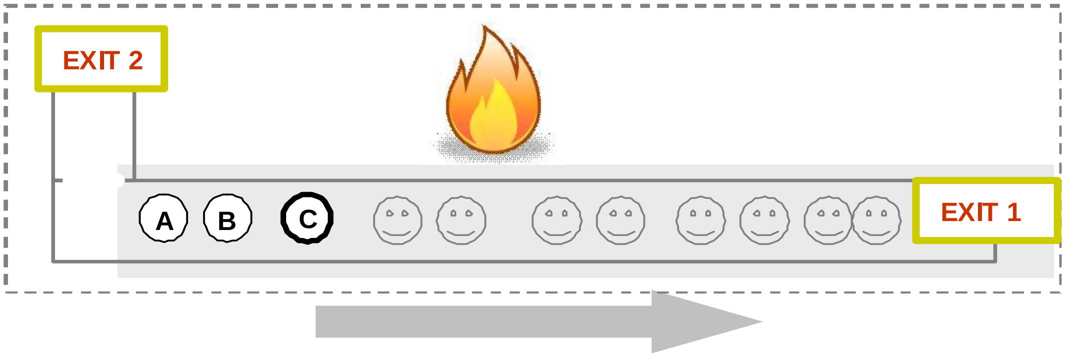


Figure 3.1: Example: necessary conditions for a correlated strategy

For example, consider a scenario where a large number of agents are trying to evacuate a building in an emergency as depicted in Figure 3.1. In the figure, only agent C has a complete view that includes both exit 1 and exit 2. The observations of the rest of the agents are limited to the shaded area that includes only exit 1. In terms of the amount of information, agent C is more advantageous than the rest. Given the shared objective that every agent wants to evacuate the building as quickly as possible, the strategies of agents A, B, and C can be coordinated as follows. Agent C signals agents A and B to move to the left, and both agents A and B follow the direction from agent C to move towards exit 2. Subsequently, all three agents evacuate quicker than would have been if they had to use exit 1.

Let us examine the rationale behind these decisions. This analysis gives an insight to answer two fundamental questions of this research:

- Under what conditions, are more privileged agents motivated to share their strategies?

- Under what conditions, are agents motivated to follow the strategies of other agents?

Given the subjective view, an optimal strategy for every agent but agent C is to move to the right towards exit 1. For agent C, moving towards exit 2 is a better strategy. The rationale of agent C for sharing its strategy with agents A and B is self-interest: in order for it to move to the left, the agents on its left also have to move to the left. The rationale of agents A and B for following the prescribed actions is also self-interest: given that agent C's objective is to evacuate the building, if there is an exit on the left that is closer to agent C, then the exit must be even closer to agents A and B.

More generally, a structure of mutually dependent information sources and information recipients can emerge when two specific conditions are satisfied. First, an agent is willing to share its knowledge with other agents if it expects a better payoff by sharing than by not doing so. Second, an agent willingly takes prescribed actions from a certain information source if the agent expects a better payoff by acting according to the decisions from the source than from any other sources including itself. Under these two conditions, any correlated strategy that ensures a better payoff than the agents' subjective valuation of independent strategies can be stabilized as a social norm.

In the following sections, a brief literature review is given on the topic of multiagent learning; followed by detailed descriptions of the algorithmic procedures.

3.2 Related work

This section reviews existing multiagent learning algorithms under two categories. The algorithms are classified according to the type of learned strategies.

3.2.1 Stationary policy learners

Definition 11 (Stationary strategy)**.** *A strategy (policy) is stationary if it does not depend on the past play.*

For instance, a single-shot Nash equilibrium strategy is stationary since the strategy depends only on the immediate payoff that does not change according to the actual play. Note that a stationary policy does not mean that the policy is deterministic, and can be stochastic.

Bowling and Veloso proposed the use of two criteria for evaluating multiagent learning algorithms [4]:

(i) (The rationality property) An agent must learn to play the best-response strategy against a stationary opponent; and

(ii) (The convergence property) The learning of an agent must converge.

The second criterion assures that a play of rational learners necessarily converges to a Nash equilibrium (of a single-shot game). Thus, this set of criteria suggests that a learned policy be stationary.

A gradient ascent learning algorithm using a variable learning rate converges to a single-shot Nash equilibrium in 2-player 2-action games [4]. In this algorithm, the learner dynamically adjusts its learning rate according to the "Win or Learn Fast" (WoLF) principle - when winning, persist a winning strategy so that the opponent would better adapt to it, and vice versa.

AWESOME is a more general algorithm that guarantees convergence to a single-shot Nash equilibrium in n-player games in self-play [11]. This algorithm does not learn a Nash equilibrium strategy. Instead, the algorithm is given a pre-computed single-shot Nash equilibrium strategy. When the agent detects that the opponents are using stationary strategies, the agent plays a best response strategy accordingly. On the other hand, when the agent detects that the opponents are using pre-computed equilibrium strategies, then the agent plays its corresponding part of the equilibrium strategy profile. Other algorithms aiming at convergence to a single-shot Nash equilbrium include [10, 37, 30].

Regret-based learning algorithms take a distinct view in the learning process. Assuming that a game play follows some stationary probability distribution, the goal of this class of algorithms is to learn its respective part to realize the distribution. This class of learning algorithms exhibit the following property:

Definition 12 (Hannan-consistency property [25])**.** *A strategy of a player is called* Hannan-consistent *if, in a long run, the average cost of the player is as low as it can drop when played against the empirical distribution of the strategies of other players.*

Fictitious play is an iterative method to compute an equilibrium that always selects the best response against the average play of the past [5]. Although the original fictitious play is not Hannan-consistent, a *smoothed* fictitious play that probabilistically chooses a *better* action, as opposed to the best action, is Hannan-consistent [28].

A play of agents adopting a Hannan-consistent learning algorithm converges to a set of correlated equilibria in the limit [27]. In accordance with the learning assumption that a game play follows some stationary probability distribution, the solution concept describing the outcome is also stationary. The stationary characteristic of correlated equilibria will be further discussed in Section 4.3.1.

Let us revisit the two multiagent learning criteria. While the rationality property is unmistakably clear, the convergence property is less clear, especially if the purpose of learning is to optimize with respect to long-term rewards.

For instance, consider a 2-player game. Suppose the learner detects a sign of stationary policy in the other agent's play (e.g., by observing certain actions more frequently than others). If the learner is *adaptive*, it should play its respective best response more frequently; therefore, the game will eventually converge. This scenario is queer that the learner is indifferent to winning or losing as long as its policy converges to some stationary one. Suppose that the other player is in a winning situation. If the learner is rational, its objective should be to overturn the momentum of the game, instead of playing adaptively to lose fast. If the learner cannot win the game, the second best option should be a draw instead of losing. Therefore, having the convergence property makes the learning algorithms *adaptive*, but not necessarily more rational.

3.2.2 Non-stationary policy learners

A set of criteria for multiagent learning more recently proposed by [50] stresses performance guarantees against various types of opponents. In particular, the expected payoff of a learner must be at least as good as the *minimax* value if played against another rational player.

When all agents are rational, the folk theorem dictates that mutually beneficial solutions can be sustained by use of threat. In order to enact a threat, agents must be able to choose their strategies conditioned on the previous plays of other agents. That is, agents must employ *non-stationary* strategies to realize Nash equilibria of a repeated game.

Littman and Stone proposed a polynomial-time algorithm to learn an optimal strategy for any repeated 2-player games [36]. The idea is like the folk theorem, but this algorithm focuses on how the strategies can be constructed as a pair of automata. This algorithm assumes that agents have complete information, and is restricted to 2-player games.

Crandall proposed two properties for multiagent learning in repeated games: 1) the average payoff of a learning agent must be at least as good as the minimax value, and 2) an agent should learn to cooperate or compromise when beneficial (C/C property) [14]. The M-Qubed (M^3 for Max or MiniMax) learning algorithm exhibits these two properties in several well-known 2-player normal-form games. In M-Qubed, agents learn to play a mixed strategy in competitive games, or to play a pure-strategy otherwise to intentionally expose its strategy so that the other agent can also play the corresponding pair.

Sen et al. introduced another algorithm similar to M-Qubed where agents intentionally expose their own strategies to the other agent using signals [54]. This algorithm is also limited to 2-player games.

In both M-Qubed and the signaling algorithms, agents can be exploited by revealing their strategies. Other work in repeated games includes [58], the focus of which was exclusively on prisoner's dilemma.

Given that existing non-stationary learning algorithms are limited to 2-player games, this thesis proposes a non-stationary multiagent learning algorithm that is scalable to n-player games for a large value of n ($n \geq 2$).

3.3 Implicit reciprocal strategy learning (IMPRES): the Algorithm

The IMPRES algorithm presented here is specifically focused on symmetric network congestion games. Throughout the section, a strategy may also be referred to as a path.

3.3.1 Overview of double-layered learning

The crux of the algorithm lies in its double-layered learning structure: the agents learn a *correlated strategy* in the inner-layer, while they progressively self-organize to learn the *social norm of reciprocity* in the meta-layer.

The intersection example from Chapter 1 is repeated. Consider two automobiles coming from orthogonal directions to an intersection. When each agent independently makes decisions, there is a non-zero probability that there will be a collision. On the other hand, one can imagine a centralized model where there exists a traffic light that both agents can observe. If an agent observes a red light, it is best for the agent to stop given that the other agent obeys its corresponding (green) light.

The intuition behind the IMPRES algorithm can be illustrated in the situation without a traffic light or a stop sign. Without prior agreement, one of the drivers signals the other driver to cross first. The other driver then obeys the signal, and thus both drivers achieve mutually beneficial outcomes by avoiding a collision. Specifically, each driver in this example has three meta-strategies that it could adopt:

- α-agent (strategist) that computes an optimal correlated strategy to generate signals,
- β-agent (subscriber) that obeys the signals from the other driver, or
- γ-agent (solitary) that makes an independent decision.

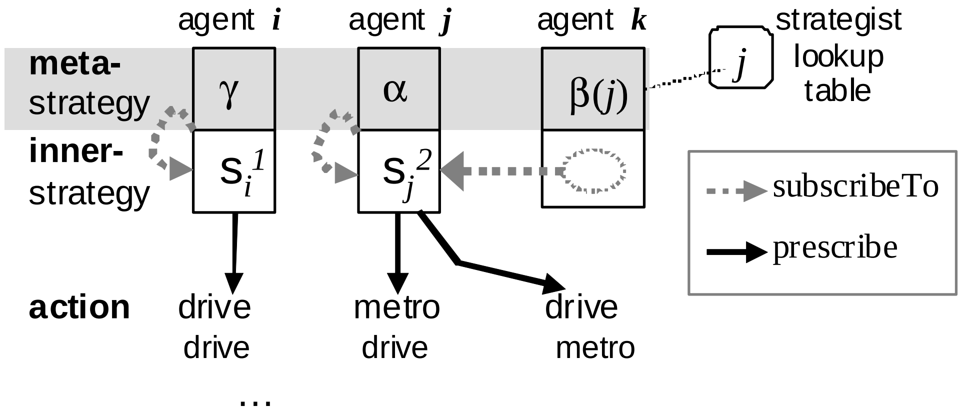


Figure 3.2: Metro vs. Driving: an example of 2-layered strategies for 3 agents. In the meta-layer, each agent makes a high-level decision of whose strategy to use in choosing the actual action.

Figure 3.2 illustrates an example of the 2-layered strategies for three agents i, j, and k. Agent i is a γ-agent, meaning that the agent has its own inner-policy π_i^1 that prescribes which action to take. Agent j is an α-agent, denoting that the agent is a registered strategist that is learning inner-policy π_j^2 for 2 agents (itself and a subscriber), and that it is acting as a signal generator. Lastly, agent k is a β-agent that subscribes to the strategies of agent j, meaning that agent k selects a path that strategist j prescribes for agent k. The required amount of communication in this model is minimal since a subscriber observes only its respective part of a strategy signal.

The algorithm requires that the agents know the cost functions, and that each agent can observe its own path selection and the corresponding cost. The algorithm is also limited to a symmetric case where agents have the same set of strategies. It is also assumed that there exists a strategist lookup table that lists current registered strategists. Note that the lookup table does not provide any further information about the strategists except their names.

3.3.2 Main procedure

Algorithm 1 shows the main procedure of the IMPRES algorithm. Initially, all agents are γ-solitaries. The agents have an alternative meta-strategy α to become a strategist, but meta-strategy β (subscriber) is not yet available to the agents (since there are no strategists in the beginning).

In the IMPRES algorithm, an agent takes an optimal action according to some knowledge; and it is the meta-strategy that specifies the source of knowledge. At each round, an agent subscribes to a path prescription (knowledge) according to its current meta-strategy m (line 2–7). When an agent is a solitary (γ) or a strategist (α), a prescription comes from itself. When an agent is a subscriber ($\beta(l)$), the agent sends a prescription-request to strategist l, and receives a path from agent l (Algorithm 3 line 9). A subscription may fail if the requested strategist has changed its meta-strategy; if a subscriber-agent fails to receive a prescribed strategy, the agent turns into a solitary (and selects an action according to its own strategy).

Each agent keeps the number of current subscribers f. When the agent is the sole subscriber, the value of f is 1. Each agent also keeps a strategy stack of size f. The general purpose of subscribing and prescribing algorithms is to establish communication between agents only; and the actual strategy stack is updated in the inner-learning layer one time step earlier. When the stack is empty, as in the initial time step t_0, a best response strategy is computed on an on-demand basis.

After taking the recommended path, the agent observes the load of the chosen path, and receives the congestion cost.

So far, only the mechanics at a high-level has been described. The learning algorithms for how an agent selects an actual path (inner-learning) and a meta-strategy (outer-learning) will be described in the next sections in detail.

```
Init: a set of meta-strategies $M = \{\alpha, \gamma\}$
Init: Q-value $Q(m) = 0$, policy $\pi(m) = \frac{1}{|M|}, \forall m \in M$
Init: meta-strategy $m = \gamma$                                /* solitary */
Init: a set of available paths $P$ in game $G(V, E)$
Init: estimated load $l_e = 0, \forall$ edges $e \in E$
Init: inner-strategy stack $S = []$
Init: the number of subscribers $f = 0$
repeat
    path $s \leftarrow$ subscribeTo($m$)
    if $s$ is nil then                              /* obsolete strategist */
        $M \leftarrow M - \{m\}$
        $m \leftarrow \gamma$                                 /* back to solitary */
        $s \leftarrow$ subscribeTo($m$)
    end
    Take the recommended path $s$
    Receive congestion cost $c$
    $S \leftarrow$ innerLearn($s, f$)
    $m \leftarrow$ outerLearn($c$)
forever
```

Algorithm 1: The main procedures of IMPRES

```
send(request to meta-strategy $m$)          /* $m$:  self or a strategist */
listen
    receive(path)                                        /* reply from $m$ */
    return path
for one round
```

Algorithm 2: β subscribeTo

3.3.3 Inner-learning of a correlated strategy

Algorithm 4 shows the inner-learning algorithm that learns a correlated strategy. Note that β-agents (subscribers) do nothing in the inner-layer (line 2). Otherwise, at each round, the learner first updates the expected number of other agents in all paths based on the observed loads on the edges from the path that it took in the last round (line 4–7). After the loads are updated as a weighted sum of the old value and the newly observed value, the loads are normalized so that the sum does not exceed the number of agents n.

Next, the learner computes the best-response joint strategy for the number of

```
reset: the number of subscribers $f = 0$
listen
    receive(request from subscriber)
    if not a strategist ($\alpha$) nor self-subscription then
        return nil                                        /* obsolete */
    end
    if inner-strategy stack $S$ is empty then
        $s \leftarrow$ bestResponse(1)
    else
        $s \leftarrow$ pop(inner-strategy stack $S$)
    end
    $f \leftarrow f + 1$
    send($s$ to subscriber)
for one round
```

Algorithm 3: α Prescribe (reply to subscribeTo)

```
input: Path s, the number of subscribers $f$
switch meta-strategy $m$ do
    case subscriber $\beta$: return                       /* do nothing */
    case solitary $\gamma$ or strategist $\alpha$:
        foreach edge $e$ in path $s$ do
            Observe $load(e)$          /* the number of agents on edge $e$ */
            $l_e \leftarrow (1 - \eta^i) l_e + \eta^i load(e)$
        end
        inner-strategy stack $S \leftarrow$ BestResponse($f$)
    end
end
```

Algorithm 4: InnerLearn

subscribers f from the previous round (Algorithm 5). In the case of solitary γ-agents, the number of subscribers f is 1, thus the algorithm simply selects a path for the agent alone.

The idea behind the best-response algorithm is due to the following algorithm for computing a system optimal solution [38] that has been discussed earlier in Chapter 2. For easier reading, the algorithm is repeated briefly. Let $c_e(x)$ denote a cost function of resource e when x agents are using resource e. Given a network congestion game $G = (V, E)$ and n agents, consider a new game $G' = (V, E')$ where each edge $e \in E$

```
   input: the number of subscribers $f$
   reset: inner-strategy Stack $S = []$
   reset: $a_e = 0, \forall e \in E$
1  for $v = 1$ to $f$ do
2      if random() $< p$ then            /* explore with probability $p$ */
3          $s \leftarrow$ randomPick($P$)
4      else                              /* exploit with probability $1 - p$ */
5          $s \leftarrow \arg\min_{k \in P} \sum_{j \in k}$ {Equation 3.1}
6      end
7      push($s$ to $S$)
8      foreach edge $e$ in path $s$ do
9          $a_e \leftarrow a_e + 1$
10     end
11 end
12 Shuffle S
```

Algorithm 5: Select BestResponse

in the original game G is substituted by n unit-capacity edges $(e_1, ..., e_n)$, such that the cost of each new edge e_k is $kc_e(k) - (k-1)c_e(k-1)$.

Since the costs of the new edges in G' are in an increasing order, the new edges will be occupied in that order under the minimum-cost-flow algorithm. Subsequently, when k new edges are chosen, the sum of the new edge costs becomes the sum of costs of the agents that have chosen the original edge e; such that $\sum_{e'=e_1}^{e_k} c_{e'}(1) = c_e(1) + (2c_e(2) - c_e(1)) + ... + (k(c_e(k) - (k-1)c_e(k-1)) = kc_e(k)$. Therefore, a minimum-cost-flow of new game G' minimizes the total cost of all agents in the original game G in the case of nondecreasing convex functions.

In IMPRES, this idea is modified to find the best-response strategies for one or more subscribers. Recall that the minimum-cost-flow algorithm sequentially selects a path at a time. Let l_j be the estimated load of edge j, and let a_j be the number of subscribers that the strategist has already decided to send to edge j. Let $c_j(x)$ and $c'_j(x)$ denote the flow cost of edge j in the original graph G and that in the new graph G', respectively. The algorithm computes the cost of an edge in the new graph G' when one more agent is added to the existing load, as a criterion to determine the minimum-cost-flow path using the following equation (line 5):

$$c'_j(l_j + a_j + 1) = (a_j + 1)c_j(l_j + a_j + 1) - a_j c_j(l_j + a_j) \tag{3.1}$$

Ties are broken randomly. With a decaying probability, the agent also explores randomly selected paths instead of myopic best responses (line 3).

Once a best-response joint strategy is computed, the learner shuffles the strategy stack (line 12), so that all subscribers (including itself) fairly experience the average cost of the strategy profile.

3.3.4 Outer-learning of a meta-strategy

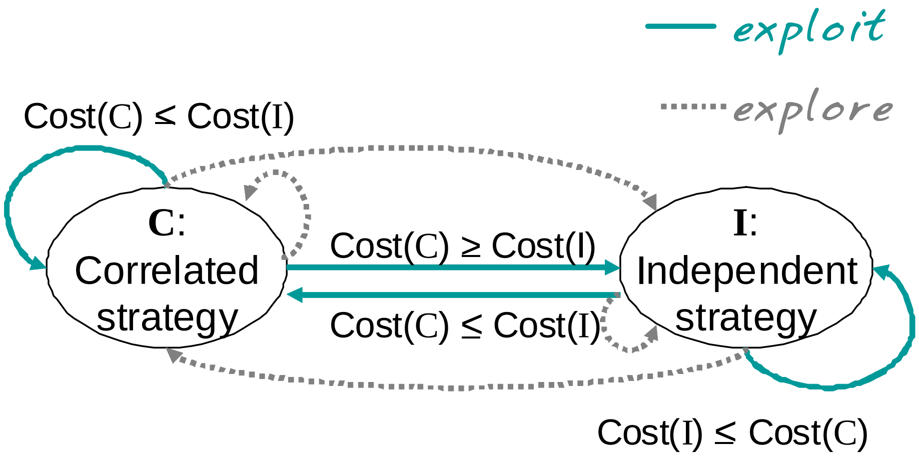


Figure 3.3: An example of an IMPRES strategy

It is important to note that the learning in the meta-layer is a significant extension to classical reinforcement learning. In traditional reinforcement learning, a set of actions available to a learner does not change dynamically although the reward from an environment may not be stationary. For example, in a k-armed bandit problem, the reward from each arm may change over time, but the learner will always have the same set of k arms. On the contrary, one of the main challenges of IMPRES is rooted in the fact that the set of meta-strategies changes dynamically, as different strategists emerge and submerge over time.

Figure 3.3 illustrates a simple example of the learning in the meta-layer. In this example, two meta-strategies are available to the learner: a correlated strategy (denoted by C) and an independent strategy (denoted by I). A solid line represents the optimal choice based on the expected costs, referred to as exploitation, and a dotted line represents a randomized choice that may be suboptimal according to the learner's current belief but perhaps a better choice in a long run, referred to as exploration. This reinforcement learning structure constitutes a non-stationary strategy, which will be analyzed in the next chapter.

Algorithm 6 describes the outer-learning procedure. At each round, the learner updates the Q-value of its current meta-strategy m (line 1). Next, if the number of meta-strategies is smaller than parameter κ, then a new strategist is randomly selected from the lookup table as a new meta-strategy; and the initial value of the new strategist is set to ι (line 2–6). The two constants κ and ι will be further discussed later in Section 5.4.2.

Finally, the agent updates the meta-policy using the Boltzmann update rule (line 7); the rule assigns more probability mass to the meta-strategies that have performed better than other alternatives. As temperature T in the equation (line 7) drops, the rule becomes more sensitive to the value differences; therefore, when the temperature is very low the algorithm mostly exploits the current best choice rather than exploring alternatives.

```
   **Global**: LookupTable L, maximum cost MaxCost
   **input**  : current cost $c$
 1 $Q(m) \leftarrow (1 - \eta^m)Q(m) + \eta^m$(MaxCost - c)
 2 if $|M| < \kappa$ then                                   /* max size $\kappa$ */
 3     $l \leftarrow randomPick(L)$
 4     $Q(l) = \iota$
 5     $M \leftarrow M \cup \beta(l)$
 6 end
 7 $\pi(m') = \frac{\exp \frac{Q(m')}{T}}{\sum_{m'' \in M} \exp(\frac{Q(m'')}{T})}, \forall m' \in M$
 8 $T \leftarrow \delta T$                                    /* $0 < \delta < 1$ */
                                   /* meta-strategy for the next round */
 9 if probability $\frac{1}{\max(f,1)^\omega}$ then
10     $m \leftarrow$ selectMetaStrategy($\pi$)
11     if $m = \alpha$ then                               /* new strategist */
12         $L \leftarrow L \cup \{id\}$
13     end
14 end
```

Algorithm 6: OuterLearn

After updating the policy, a new meta-strategy is selected for the next round (line 9–14). Although a strategist does not directly receive additional feedback from its subscribers, the number of subscribers is a reasonable indicator for the performance of a strategist. Based on this intuition, a boosting parameter ω is added such that the probability of staying as a strategist is proportionally increased with respect to the number of its subscribers f. This parameter affects only when the agent is a strategist since the number of subscribers f is at most 1 otherwise.

The time complexity of the algorithm is polynomial in the number of agents and in the number of alternative paths, and the space complexity is linear in the number of resources (edges).

3.4 Summary

In this chapter, the general idea of "learning from others" is introduced into the multiagent learning context. Specifically, I proposed IMPRES (implicit reciprocal strategy learning) - a double-layered learning framework where agents learn to act more rationally by correlating their strategies. I have concentrated on technical details in this chapter; and a more general discussion of the theoretical model on which the IMPRES algorithm is based on will follow in the next chapter.

Chapter 4

Theoretical Analysis

> "The tyranny of a prince in an oligarchy is not so dangerous to the public welfare as the apathy of a citizen in a democracy." – Charles de Secondat, Baron de Montesquieu, The Spirit of the Laws, 1748.

4.1 Introduction

This chapter describes a theoretical model that frames `implicit reciprocal strategy learning` (IMPRES). The model is closely related to the notion of Nash equilibria of a repeated game. Clearly, a set of cost vectors defined by the folk theorem *includes* a set of cost vectors that is mutually desirable for all players of a game. There can be, however, infinitely many strategies that satisfy the folk theorem including a set of single-shot Nash equilibria. Due to abundance, the predictive power of Nash equilibria is significantly reduced in repeated games. The main contribution in studying repeated games, both in general [47] and in this thesis, is to discover meaningful social norms (strategies) that can support mutually beneficial outcomes. In this context, one can view multiagent learning as a "search" for a better choice in the vast space of enforceable strategies; and one of the main goals of this research is to be able to predict more likely outcomes with which *reasonably* rational agents will settle.

This chapter is organized as follows: first, the conceptual model behind the IMPRES algorithm is described. Next, my interpretations for well-known solution concepts are given. Finally, the outcome of IMPRESS is generally characterized as behavior-equivalent to approximate Nash equilibria, and an additional set of criteria to further specify interesting properties of the solutions is discussed.

4.2 The models of social norms

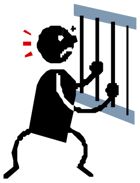
The main hypothesis of this thesis echoes the folk theorem: in order to achieve mutually beneficial outcomes in repeated games, a population needs a set of correlated strategies around a social norm. While the folk theorem states that such equilibria exist, this thesis seeks efficient learning algorithms for actually finding such stable strategies in the context of games that involve a large number of agents.

4.2.1 The folk theorem

In this section, an example of prisoner's dilemma is used to illustrate the folk theorem that is relevant to the discussion of the IMPRES model.

Prisoner's dilemma

A prisoner's dilemma is a 2-player matrix game where two prisoners under a criminal charge are making decisions between two alternative actions: to cooperate with the other prisoner by keeping silent (denoted by C), or to defect by confessing the crime (denoted by D). If one prisoner cooperates when the other prisoner defects, the cooperative prisoner faces the severest punishment while the defective one is set free. An example of a complete penalty matrix is shown in Table 4.1. In this example, when both prisoners defect the strategies are in a Nash equilibrium where each prisoner faces 6 years in prison. The minimax value is also 6 years in prison for both players.



row,column	Cooperate	Defect
Cooperate	1,1	10,0
Defect	0,10	6,6

Table 4.1: Prisoner's dilemma: a penalty for each corresponding strategy profile is defined in terms of the years in prison. For instance, when both prisoners cooperate, each serves 1 year in prison.

Grim-trigger

A grim-trigger strategy can be formulated for this game of prisoner's dilemma as follows: given an *enforceable* correlated strategy, obey the correlated strategy as long

row,column	Cooperate	Defect
Cooperate	0	0.5
Defect	0.5	0

A suboptimal strategy

row,column	Cooperate	Defect
Cooperate	1	0
Defect	0	0

The optimal strategy

Table 4.2: Enforceable correlated strategies of prisoner's dilemma: a number in each cell represents the probability of choosing the corresponding strategy profile. The left table represents a correlated strategy of alternating (C, D) and (D, C) with an equal probability, and the right table represents a strategy where both prisoners cooperate with probability 1.

as the other prisoner also obeys the strategy, otherwise defect forever. This strategy can be succinctly represented in a state transition diagram as shown in Figure 4.1.

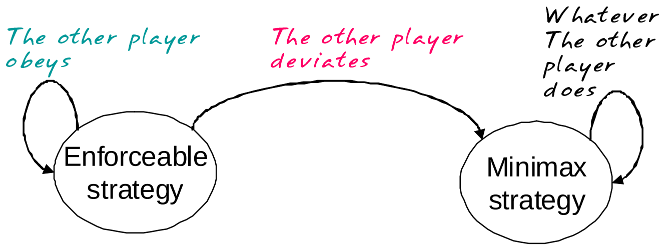


Figure 4.1: A grim-trigger strategy

Recall that any correlated strategy that is better in average payoff than the minimax values is *enforceable*. Table 4.2 shows two examples of enforceable correlated strategies of the prisoner's dilemma game. Consider a correlated strategy of alternating (C, D) and (D, C) (left). It is easy to see that this correlated strategy is suboptimal since the optimal correlated strategy is for both prisoners to always cooperate (right). The alternating strategy is, nonetheless, enforceable since its average penalty of 5 years in prison is better when compared to the minimax penalty of 6 years.

Consider a suboptimal grim-trigger strategy using the alternating correlated strategy. If both prisoners adopt this suboptimal grim-trigger strategy, the play is in a Nash equilibrium of a repeated game; this equilibrium solution provides two interesting observations.

First, it is assumed that both agents are aware of a punishment that will follow a deviation. Suppose not. Since a single deviation results in the minimax (punishment) state, the most likely outcome is a single-shot Nash equilibrium where both prisoners defect. Since all single-shot Nash equilibria are also Nash equilibria of a repeated game, theoretically the statement still holds, but the outcome is less desirable.

Second, at least one agent is rationally bounded; for instance, it only knows about a limited set of available strategies. Specifically in this example, the only enforceable strategy available to the agents is the alternating strategy. Suppose not. Let x and y denote the two agents. Consider an optimal grim-trigger strategy where the prisoners cooperate as long as the other prisoner continues to cooperate, and defect forever once the other prisoner defects. Let s and s' denote the optimal and the suboptimal grim-trigger strategies, respectively. When agent x uses suboptimal strategy s', agent y's best response is to use a counter-threat to enforce optimal-strategy s. Given the counter-threat, if agent x is objectively rational, it must change its strategy to s since the expected payoff of s is strictly better than that of s'. Therefore, if both agents are using strategy s', at least one agent is rationally bounded. When both agents adopt the optimal grim-trigger strategy, the pair of strategies represents an objectively rational choice given a complete set of available correlated strategies.

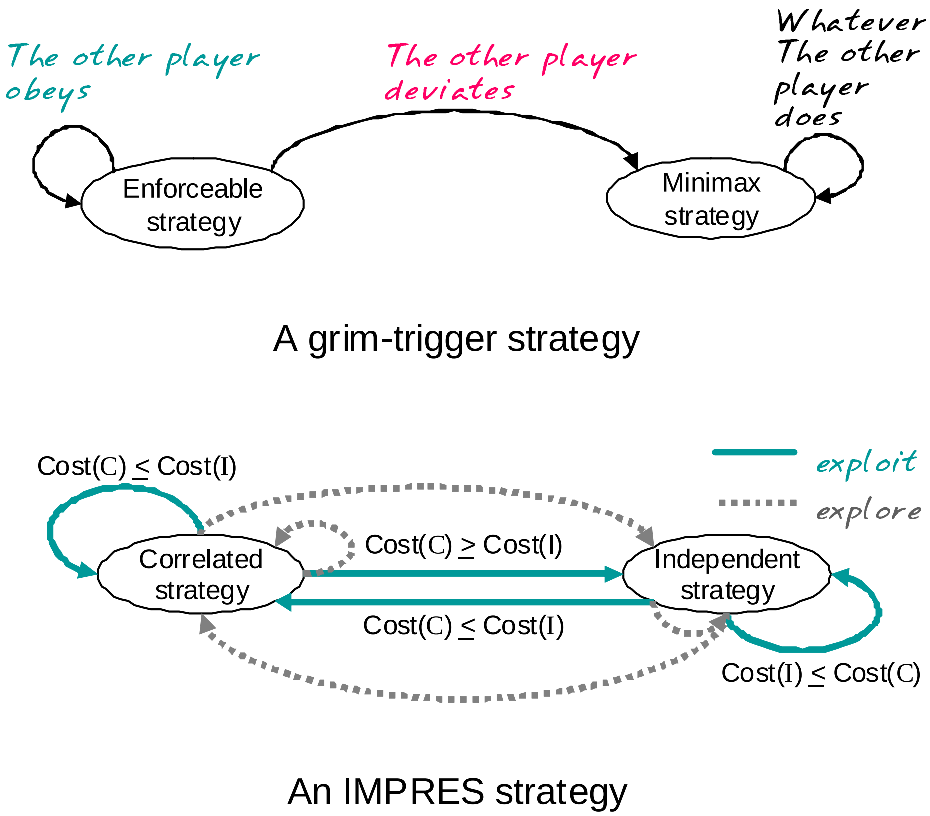
Remark. The folk theorem states that any enforceable strategy profile s can be stabilized under two assumptions: 1) all agents are aware of the enforceable strategy profile s and the penalty of a deviation from it, and 2) at least one agent believes that there is no better strategy profile than strategy profile s.

4.2.2 The IMPRES model

The IMPRES approach starts from the assumption that an enforceable strategy and an absolute criterion such as the minimax values are *not* readily available. As opposed to assuming that agents a priori know meta-strategies such as grim-trigger, the focus of this research is on how the agents can *learn* such reciprocal strategies.

The IMPRES model represents a strategy as a double-layered structure that resembles a trigger strategy. To enable a direct comparison, the examples used earlier when describing the grim-trigger and the IMPRES models are combined in Figure 4.2. Both models reflect non-stationary strategies that depend on previous plays. Specifically, they both impose some sort of threat to enforce a mutually beneficial strategy profile. The IMPRES model is distinguishable from the grim-trigger model in several important respects:

- (Incomplete monitoring) The grim-trigger strategy requires complete monitoring; that is, the minimax strategy is triggered when an agent observes a deviator from the enforceable strategy. On the other hand, the IMPRES strategy does



A grim-trigger strategy

An IMPRES strategy

Figure 4.2: Grim-trigger versus IMPRES

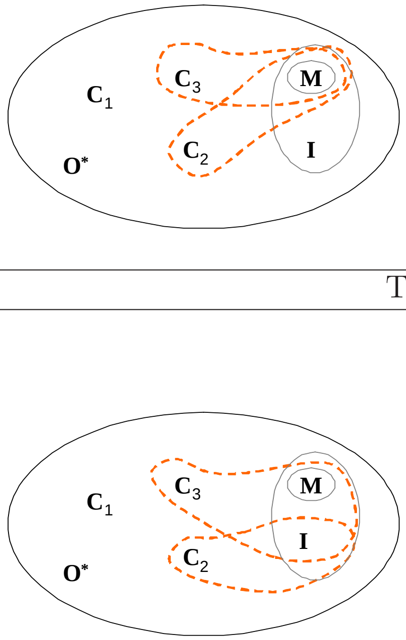
not require complete monitoring because a transition between meta-strategies is purely based on the agent's expectations (expected costs).

- (Stochastic transition) While the strategic transition (between the enforceable strategy and the minimax strategy) is deterministic in the grim-trigger model, IMPRES uses a stochastic transition; that is, IMPRES allows the agents to explore with a small probability.

- (Less coordination overhead) Whereas implementing the minimax strategy may require a complete coordination among all agents, the punishment strategy of IMPRES is simply a break from a correlated strategy.

- (Learned correlated strategy) The grim-trigger model can only be implemented when both the enforceable and the minimax strategies are available; as discussed in Chapter 2, both the problem of finding an optimal strategy and the problem of finding the minimax values are generally intractable. In contrast, IMPRES is a learning algorithm for efficiently finding those strategies that can be stabilized.

Perhaps the most important difference is where a mutually beneficial strategy and a punishing strategy come from. The learning in this context can be viewed as a search for a self-enforceable strategy, as opposed to a minimax-enforceable strategy, in the space of correlated strategies. Figure 4.3 compares the folk theorem and the IMPRES models in the search space of correlated strategies. While the folk theorem defines an enforceable strategy based on an absolute criterion of the minimax strategy (denoted by M in the figure), the IMPRES model uses the agent's independent strategy (denoted by I) as its relative criterion for determining enforceable strategies.



The strategy profile realizing the minimax payoff is denoted by M. A set of C_i denotes a set of correlated strategies. The figures are simplified, but there can be infinitely many correlated strategies in the space. O^* denotes optimal strategies.

The folk theorem

The strategy profile when the agent's strategy is independent from the strategies of other agents is denoted by I. Note that these figures represent a view of a single agent, thus the strategies of other agents may still be correlated in profile I as in the minimax strategy profile M.

The IMPRES model

Figure 4.3: The search space of correlated strategies

Remark. The IMPRES model generalizes the folk theorem such that any correlated strategy can be stabilized if, for every agent, using this strategy produces a better outcome than using the agent's independent strategy. When the expected payoff of an agent's independent strategy is at least as good as its minimax value for every agent, the set of payoff vectors that are enforceable in the IMPRES model corresponds to the set supported by the folk theorem.

4.3 Analysis of solution concepts

This section analyzes the outcome of IMPRES within the boundaries of well-known solution concepts: specifically the notions of correlated equilibria and of Nash equilibria.

4.3.1 Correlated equilibria

In particular, the outcome of IMPRES may appear as a correlated equilibrium since the learned strategy is correlated. The purpose of this section is to draw a clear line that the notion of correlated equilibria does not reflect the non-stationary nature of the IMPRES algorithm.

A correlated strategy is a probability distribution over the space $S = S_1 \times ... \times S_n$ of strategy profiles. Suppose that a strategy profile is drawn from set S according to some correlated strategy f, and each agent is given its respective part of strategy f_i. If, for every agent i in N, the expected cost of following the received strategy f_i is lower than not doing so, given other agents obey their respective strategies f_{-i}, the correlated strategy f is in a correlated equilibrium [2].

An intuitive example of a correlated equilibrium is a traffic light. Let us revisit the example of two drivers at an intersection from Section 1.4. The agents (drivers) have two action choices: stop or enter an intersection. Given that both agents know how a traffic light works - when one side is a green light the other side is a red light – the strategies of the two agents can be in a correlated equilibrium around a traffic light. For instance, suppose that an agent observes a red light. Given that the other agent obeys its corresponding green light (thus enters the intersection), obeying the red light (thus stopping) is the optimal strategy. A similar reasoning can be applied to the other agent that observes a green light. Since none of the agents are motivated to deviate from the current strategy profile, the strategies are in a correlated equilibrium.

Definition 13. *A correlated strategy f is in a correlated equilibrium if and only if $E[c_i(f_i, f_{-i})] \leq E[c_i(g_i, f_{-i})]$, for all $g_i \in S_i, i \in N$ where E denotes the expected* *value*[1] *and* $c_i(s)$ *denotes the cost of agent* i *under strategy profile* s*.*

i) (Comparison with Nash equilibria of a single-shot game NE^1) Since a correlated strategy profile is a probability distribution over all possible joint strategies, NE^1 strategy profiles are special cases of correlated equilibria where the strategies of individual agents are independent. Therefore, a set of correlated equilibria (CE) subsumes a set of single-shot Nash equilibria (NE^1); such that CE $\supseteq$ NE^1.

ii) (Comparison with Nash equilibria of a repeated game NE$^\infty$) In principle, the concept of correlated equilibria assumes that the agents are stationary, thus *not* reactive. Specifically, every agent i compares the expected cost of following strategy f_i recommended by the signal to that of taking alternative strategy g_i, assuming that the other agents will perseveringly follow their respective strategies f_{-i} recommended by the signal in subsequent rounds even after agent i' deviation causes an increase in their costs. Therefore, a set of Nash equilibria of a repeated game generally do not coincide with a set of correlated equilibria.

To see the properties of correlated equilibria more clearly, consider an unusual prisoner's dilemma[2] example used in Aumann's seminal article that introduced the notion of correlated equilibria [2] (copied[3] in Table 4.3). The table (right) shows an optimum-correlated strategy distribution that is *not* a correlated equilibrium; for instance, when a prisoner is recommended to cooperate by the signal, the expected penalty of defecting is lower, given the other prisoner still obeys its respective signal.

row,column	C	D
C	4,4	6,0
D	0,6	5,5

The penalty matrix (years in prison)

row,column	C	D
C	0	0.5
D	0.5	0

The optimum-correlated strategy

Table 4.3: An unusual prisoner's dilemma (C: cooperate, D: defect) [2]

The folk theorem suggests that this distribution can, in fact, represent an equilibrium strategy of a repeated game, since this strategy is both *feasible* - as are any

[1] For simplicity, $c_i(s)$ elsewhere denotes the expected cost without being prefixed with E.

[2] We call this game "unusual prisoner's dilemma" since its penalty matrix does not comply with a general definition of the prisoner's dilemma game where the penalty of mutual cooperation is generally lower than the average of the lightest and the harshest penalties.

[3] In Aumann's original article, the objective of the prisoners is to maximize the expected payoff. For consistency, the payoff matrix has been modified such that the objective is to minimize the expected penalty, but the preference order of the strategy profiles is preserved.

correlated strategies – and *enforceable* as the expected penalty vector $(3, 3)$ [4] Pareto dominates the minimax values $(5, 5)$.

If both agents choose their strategies to minimize the expected penalty, a threat naturally exists in this example, which also illustrates an intuition for the IMPRES algorithm that a mutually beneficial outcome can be stabilized without an explicit notion of threat. If the row player deviates from following the signal then the column player will also be motivated to disobey the signal, since the expected penalty of following the recommendation is no longer the best choice for the column player. Thus, the expected penalties of both agents after a deviation become higher than would have been if the agents continued following the correlated strategy.

More generally, a set of correlated equilibria does not include all equilibrium points of a repeated game that are suggested by the folk theorem. Particularly, as just seen in the prisoner's dilemma example, it excludes those correlated strategies that can only be sustained by non-stationary strategies such that the agents have contingent strategies when other agents suddenly change their strategies.

Remark. A correlated equilibrium is a *stationary* solution concept. Whereas a set of correlated equilibria includes a set of single-shot Nash equilibria, there exists a set of Nash equilibria of a repeated game that does not belong to a set of correlated equilibria.

Remark. Although IMPRES learns correlated strategies, the learned strategies are non-stationary in nature; therefore, the outcome of IMPRES cannot generally be characterized as a set of correlated equilibria.

4.3.2 Nash equilibria

In conjunction with the notion of subjective equilibria that will be described in the next section, the notion of Nash equilibria embodies the targeted solution concept of IMPRES.

It is generally stated that the notion of Nash equilibria is objectively rational. I take a different view, and argue that the notion of Nash equilibria is rationally bounded, particularly in repeated games. I briefly described this view in an earlier section when discussing the folk theorem. The next example shows the irrational decision-making process using a Nash dynamics graph. Note that a Nash dynamics graph does not represent an actual play of a game. It is used here to examine the conditions behind the definition of a unilateral deviation.

The high-low game shown in Table 4.4 is a two-player matrix game where both

[4] For each prisoner, the expected penalty when playing according to the optimum-correlated strategy is 3 years in prison, such that $0.5 \times 0 + 0.5 \times 6 = 3$.

agents are better off when they play the same action. In this example, there are two Nash-equilibrium profiles: (L,L) and (H,H). The strategy profile (L,L) is a Pareto optimal solution that is more desirable for both players.

row (x),column (y)	Low (L)	High (H)
Low (L)	1,1	10,10
High (H)	10,10	2,2

Table 4.4: The high-low game (cost matrix)

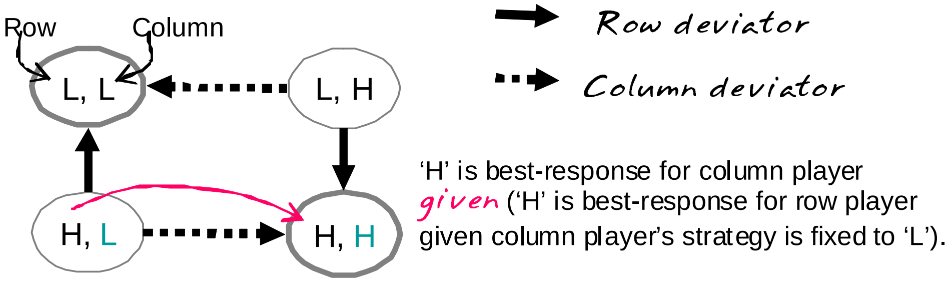

Row
Column
L, L
L, H
H, L
H, H
Row deviator
Column deviator
'H' is best-response for column player
given ('H' is best-response for row player
given column player's strategy is fixed to 'L').

Figure 4.4: Nash dynamics graph of the high-low game that possesses two Nash equilibria (L,L) and (H,H).

I will show that at least one player is rationally bounded in the suboptimal Nash-equilibrium profile (H,H). Consider the deviation arc from profile (H,L) to (H,H) in the Nash dynamics graph in Figure 4.4. Let x and y denote the row player and the column player, respectively. For player y, strategy H is the best response given a condition that player x's strategy is fixed to H; that is, player x in profile (H,L) will not change its strategy to the best-response strategy L. This condition, in turn, may mean that player x is rationally bounded, for instance, due to one or more of the following reasons:

- (incomplete set) player x's set of strategies is incomplete and does not include the best-response strategy L.

- (imperfect precision) player x is indifferent between the expected costs of the two strategies ($10 \leq 1 + \epsilon$ for some $\epsilon > 9$).

- (subjective belief) player x's belief is inaccurate such that given its belief (e.g., the strategy of player y is fixed to H instead) H is the best response.

Otherwise, agent y's belief that agent x is not willing to change its strategy even when a better alternative is given must be incorrect. The reasoning process from profile (L,H) to (H,H) is analogous. To sum up, the two inwards arcs of profile (H,H) exist only when either or both agents are rationally bounded.
Remark. If the strategy profile of a set of agents converges to a suboptimal Nash equilibrium of a repeated game, one or more agents in the set are rationally bounded.

Definitions of rationality

Let us formally define the rationality of agents. The terms and their definitions are compiled from [33, 56]. The notion of objective rationality follows a stricter definition from [56].

Definition 14. *An agent is objectively rational if it chooses the best option from a complete set of available choices, based on the objective truth.*

Definition 15. *An agent is unconsciously (objectively) rational if its behavior is objectively rational, but it is unaware of the fact. When the behavior of a rationally bounded agent is equivalent to the behavior of an objectively rational agent, the rationally bounded agent is unconsciously rational.*

Definition 16. *An agent is rationally bounded if one or more of the following conditions are satisfied:*

- *(incomplete set) the agent chooses the best option from an incomplete set of choices available to them. Let ι denote the level of incompleteness.*
- *(imperfect precision) the agent's criterion for determining the best option is approximate; such that agent i's best strategy s_i satisfies $c(s_i, s_{-i}) \leq c(s_i', s_{-i}) + \epsilon$, for all $s_i' \in S_i$ for arbitrarily small ϵ.*
- *(subjective belief) the agent's belief about the strategies of other agents is inexact with respect to the objective truth. Let η denote the difference between the subjective belief and the objective truth.*

In other words, an agent is objectively rational if and only if the agent can accurately valuate every choice from a complete set of options, and always chooses the best option from the set.

Nash equilibria and bounded rationality

I have made a remark earlier that if the strategies of a set of agents converges to a suboptimal Nash equilibrium at least one agent is rationally bounded. Let us rephrase the remark more formally using the terms from the section above. The discussion henceforth will be specifically focused on repeated games under the following assumption.

Assumption 8 (incomplete set)**.** *Agents have a complete set of independent strategies; however, agents may not have a complete set of correlated strategies. In other words, agents can explore all possible independent strategies to choose the best independent strategy, but can choose the best correlated strategy only among those available. In this sense, an agent's optimal strategy (best-response) refers to the best option from a set of options that are available to the agent.*

Note that a correlated strategy is available to an agent when there exists a common signal that prescribes the agent its respective action. Unless otherwise specified, the above assumption holds throughout the section.

Let N^ι denote a set of agents that are rationally bounded due to incomplete set (but possess perfect precision and accurate belief). Let $\hat{S} = \hat{S}_1 \times ... \times \hat{S}_n$ denote a set of strategy profiles available to agents in N^ι.

Proposition 9. *For all agent $i \in N^\iota$, agent i plays an optimal strategy if other agents' strategies are stationary.*

Proposition 10. *For all agent $i \in N^\iota$, the expected cost of agent i in the worst case is its minimax value.*

In the minimax strategy profile, a deviator's strategy is independent from the strategies of the rest, although the strategies of a team of punishers may be correlated. Therefore, if the deviator has a complete set of independent strategies (Assumption 8), and can correctly choose the best-response given the strategies of other agents are fixed to the minimax profile (Proposition 9), the highest penalty of a deviator is at most its minimax cost.

Proposition 11. *If strategy profile $\pi \in \hat{S}$ is stabilized by the agents in N^ι, then strategy profile π is in a Nash equilibrium of a repeated game (NE^∞).*

In strategy profile π, every agent is playing its best response (by Proposition 9), and no one changes its strategy (stabilized). Therefore, profile π is in a Nash equilibrium.

Corollary 12. *Let $N^{\iota,\epsilon}$ denote a set of agents that are rationally bounded due to incomplete set and imperfect precision for some ϵ (but possessing accurate belief). If strategy profile π is stabilized by the agents in $N^{\iota,\epsilon}$, then strategy profile π is in an approximate Nash equilibrium of a repeated game (ϵ-NE$^{\infty}$).*

Theorem 13. *Let N denote a set of agents that are objectively rational; that is, the set $\hat{S}_i$ of strategies that is actually available to agent i coincides with the complete set S_i for all agent i in N. If some strategy profile π is stabilized by the agents in N, then strategy profile π is in a Pareto-optimal Nash equilibrium of a repeated game.*

Proof sketch. By Proposition 11, profile π is a Nash equilibrium of a repeated game. Suppose that there exists strategy profile π' that Pareto dominates profile π. The profile π' can be enforced by use of threat, if all agents are rational. If the agents settle with profile π instead of a better option π', at least one agent is acting irrationally according to π_i, forcing the other agents to play their respective best response strategies π_{-i}. This contradicts the assumption that all agents are objectively rational. □

Corollary 14. *In a repeated game, if some strategy profile π is in a Nash equilibrium, either strategy profile π is Pareto optimal, or at least one agent is rationally bounded.*

4.3.3 Subjective equilibria

The notion of subjective equilibria introduced in [33] generalizes the notion of Nash equilibria. Whereas the notion of a rational choice under a Nash equilibrium is conditioned on the true strategies of other agents, a subjectively rational agent selects the best option based on its subjective (thus maybe incorrect) belief. If for all agents the realized actions of other agents matches the agent's belief, the belief is enforced and so is the corresponding best-response strategy; thus a strategy profile can eventually converge to an equilibrium point.

Let $N^{\iota,\eta}$ be a set of agents that are rationally bounded due to incomplete set and subjective belief (but possess perfect precision). Let H denote a set of all possible play histories of N, and let μ_g be a probability distribution over H that is induced by strategy profile $g = [g_1, ..., g_n]$. For some agent $i \in N^{\iota,\eta}$, let $g^i = [g^i_1, ..., g^i_n]$ denote agent i's belief vector about the strategies of agents in $N^{\iota,\eta}$. Formally,

Definition 17 (Subjective equilibria (η-SE))**.** *A strategy profile $g = [g_1, ..., g_n]$ is in a subjective equilibrium if and only if for all agent $i \in N$:*

(i) agent i knows its strategy, such that $g^i_i = g_i$;

(ii) agent i plays best response based on its belief, such that $c(g_i, g^i_{-i}) \leq c(x, g^i_{-i})$ *for all* $x \in \hat{S}_i$*; and*

(iii) the realized play is consistent with agent i's belief, such that $\mu_g = \mu_{g^i}$*.*

Under two specific assumptions: 1) complete information about the payoff matrix, and 2) perfect monitoring of players' actions, the notion of subjective equilibria is equivalent to that of Nash equilibria as follows:

Proposition 15. *If the agents' subjective beliefs perfectly match the objective truth (i.e.,* $\eta = 0$*), a set of subjective equilibria coincides with a set of Nash equilibria [33].*

More generally, a subjective equilibrium is "behavior-equivalent" to a Nash equilibrium. A formal result from [32, 33] follows.

Definition 18 (ϵ-closeness)**.** *Let* μ *and* $\bar{\mu}$ *denote two probability measures defined in the same space. We say that* μ *is* ϵ*-close to* $\bar{\mu}$ *for some* $\epsilon > 0$*, if there exists a measurable set* Q *of events that satisfies:*

(i) $\mu(Q)$ *and* $\bar{\mu}(Q)$ *are greater than* $1 - \epsilon$*; and*

(ii) for every measurable set $A \subseteq Q$*,* $(1-\epsilon)\bar{\mu}(A) \leq \mu(A) \leq (1+\epsilon)\bar{\mu}(A)$*.*

Definition 19 (ϵ-like)**.** *Let* $\epsilon \geq 0$*. Given two strategy profiles* f *and* g*, we say that* f *plays* ϵ*-like* g *if* μ_f *is* ϵ*-close to* μ_g*; also referred to here as "behavior-equivalent".*

Theorem 16. *For every* $\epsilon > 0$*, there exists* $\eta > 0$ *such that if* g *is a subjective equilibrium (*η*-SE) then there exists* $\bar{f}$ *such that: 1)* g *plays* ϵ*-like* $\bar{f}$*, and 2)* $\bar{f}$ *is an approximate Nash equilibrium (*ϵ*-NE) [33].*

That is, in a subjective equilibrium (η-SE), the agents act as if they were in an approximate Nash equilibrium (ϵ-NE).

4.4 Characterizing the outcome of IMPRES

The above notions can be related to characterize the behavior of IMPRES. Briefly, in n-player symmetric games, the outcome of IMPRES in self-play is behavior-equivalent to an approximate Nash equilibrium of a repeated game. A formal proof is given in Section 4.4.2.

The discussion from this section is tightly coupled with the empirical analysis in the next chapter. In contrast to the notion of single-shot Nash equilibria where the number of matching solutions is relatively small, the notion of Nash equilibria of a repeated game comprises infinitely many solutions including a set of suboptimal single-shot Nash equilibria. In that regard, I propose the use of a new set of criteria to further classify the solutions in the space of Nash equilibria of a repeated game; these criteria will also be used to demonstrate that the agents adopting IMPRES exhibit highly rational group behaviors.

4.4.1 Subclassifying Nash equilibria of a repeated game

In order to give formal proofs, a set of new concepts and their definitions are introduced. These concepts will be used both for analyzing the behavior of IMPRES, and for dissecting the large space of Nash equilibria of a repeated game.

k-correlated strategy

The notion of a correlated strategy generally assumes that every agent shares common prior knowledge, which implies a centralized signal. To be precise, the concept is more general and includes the profiles where only the strategies of some subset of agents are correlated. The concept of k-correlated strategy subclasses the notion of a correlated strategy. The definitions for the k-correlated strategy and a set of auxiliary concepts follow.

Definition 20 (configuration / demographic)**.** *A configuration (demographic) is a generic term to describe a composition of meta-strategies for a set of agents.*

Definition 21. *k-correlated strategy profile denotes a configuration where the agents are partitioned into k subgroups such that the strategies are correlated only within the same subgroup.*

Definition 22 (singleton)**.** *Given a k-correlated strategy profile, a singleton denotes a subgroup that consists of a single agent.*

Let n denote the number of agents. When a configuration is composed of n singleton-subgroups, the profile represents an independent strategy profile, referred henceforth as an *anarchy.* On the other hand, if the strategies of all agents are correlated according to a single signal, then the configuration contains one subgroup, referred henceforth as a *monarchy* (autocracy).

Price of anarchy

Figure 4.5 characterizes the subspaces defined by various solution concepts including correlated equilibria (CE), Nash equilibria of a single-shot game (NE^1), Nash equilibria of a repeated game (NE$^\infty$), and optimal solutions (Opt). Note that there can be special classes of games where these spaces further overlap with one another.

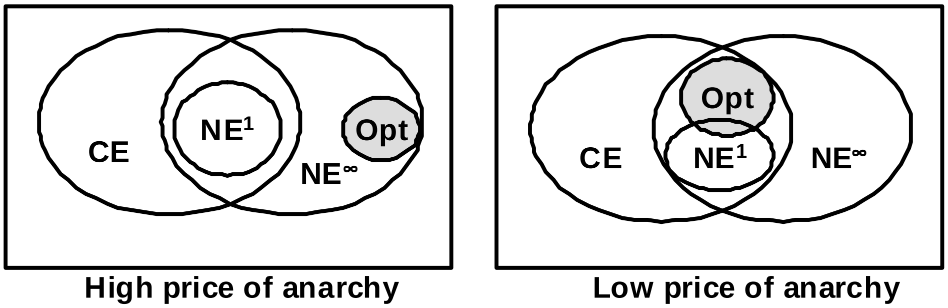


Figure 4.5: Various solution concepts in the space of correlated strategies

Another criterion that is relevant to our discussion is the price of anarchy that measures a distance between optimal solutions and selfish equilibrium solutions strictly from the system-wide quality perspective [35]. The left figure represents the problems with high price of anarchy where the solutions defined by stationary solution concepts are farther away from the optimal solutions. On the other hand, the right figure represents the problems with low price of anarchy where stationary solution concepts generally reflect system-wide optimality.

Price of monarchy

With respect to system-wide cost minimization, a monarchy configuration is perhaps the best option; however, implementing a centralized configuration is not only difficult, but it also incurs high communication overhead since all agents need to communicate with a centralized administrator. Analogous to the price of anarchy, a new metric is introduced to measure the coordination cost, referred to here as the price of monarchy. Whereas the price of anarchy measures potential quality loss due to selfish decisions, the price of monarchy estimates the practical cost of installing cooperative strategies in multiagent systems.

Both price analyses will be further discussed in Chapter 5 when evaluating the IMPRES algorithm empirically.

4.4.2 Proof of behavior-equivalency to Nash equilibria

The analysis henceforth is strictly focused on symmetric games under the following assumptions. The names of properties are coined by [4, 55].

Assumption 17. *In the decision making of a given agent, uncertainty exists only in the strategies of other agents; that is, their environment is stationary when the strategies of other agents have converged to stationary ones.*

Assumption 18. *The inner-learning algorithm satisfies the following two properties:*

(i) (the rationality property) The algorithm learns an optimal strategy in a stationary environment.

(ii) (the ϵ-safety property) When agent i is solitary, the inner-learning algorithm guarantees that the agent's expected cost does not exceed its minimax value v_i^m; such that $c_i \leq v_i^m + \epsilon$ for arbitrarily small ϵ.

Note that the safety property does not mean that an agent can efficiently learn its minimax value, but rather means that an agent can learn to choose a best response strategy even in the worst possible scenario.

This assumption may sound strange to some readers since it states that agents in an anarchy configuration (thus without the meta-learning layer) can already accomplish an average cost vector that satisfies the folk theorem. But, as discussed in Chapter 2, independent solutions are generally suboptimal; and an anarchy configuration represents at best a set of single-shot Nash equilibria (See Figure 4.5). The purpose of having the meta-learning layer is to improve the quality from that of selfish equilibria by exploring non-stationary strategies.

In this chapter, I first show that the outcome of IMPRES generally belongs to a class that satisfies the folk theorem where every agent in the worst case receives its minimax value. I then evaluate the algorithm more thoroughly according to the quality metric in the next chapter.

Lemma 4. *Suppose that the inner-learning exhibits the* 0*-safety property. Then, in n-player symmetric games, the expected cost of an agent adopting IMPRES in self-play is at most its minimax value.*

Proof. The proof is by exhaustion. At any point of time, the agents are under one of the three following configurations: an anarchy ($k = n$), a monarchy ($k = 1$), and a middle-ground ($1 < k < n$) configurations. Let $v^m = [v_1^m, ..., v_n^m]$ denote the minimax cost vector.

The analyses on an anarchy and a monarchy configurations directly follow the basic assumptions. By Assumption 18, an anarchy-cost vector Pareto dominates the minimax vector v^m. In a monarchy configuration, since there are no other agents in the environment, the environment is stationary (Assumption 17). By the rationality property from Assumption 18, α-strategist learns a fair and optimal strategy for all agents; thus, the monarchy-cost vector Pareto-dominates the minimax cost vector.

Now considering the case of middle-ground configurations for $1 < k < n$, the expected cost is at most the minimax value for all agents. Consider a non-singleton case first, followed by a singleton case. Without the loss of generality, consider a set of agents that belong to some (non-singleton) subgroup G^k. The expected cost is the same for all members of subgroup G^k since the strategist's inner-learning algorithm is fair (Algorithm 4). Let $v^\gamma = [v_1^\gamma, ..., v_n^\gamma]$ denote the agents' solitary-cost vector. Similarly, let v^α and v^β denote the agents' strategist-cost vector and subscriber-cost vector, respectively. According to the reasoning in the meta-layer, for all β-agents in subgroup G^k, the expected cost must be better than their solitary-cost values; such that for all $i \in G^k$, $v_i^\beta \leq v_i^\gamma$. By the safety property from Assumption 18, for all $i \in N$, $v_i^\gamma \leq v_i^m$. By joining the safety property with the reasoning in the meta-layer, the expected cost of β-agents are at most their minimax values; such that for all β-agent $i \in G^k, m(i) = \beta$, $v_i^\beta \leq v_i^m$, where $m(i)$ denotes the current meta-strategy of agent i. The same logic applies to an α-agent. Therefore, all agents in subgroup G^k are better off than their minimax values. In the case of a singleton, the safety property suffices to complete the proof. □

Theorem 19. *Suppose that the inner-learning exhibits the* 0*-safety property. Then, the outcome of IMPRES algorithm in self-play in n-player symmetric games is behavior-equivalent to a Nash equilibrium of a repeated game (NE$^\infty$), for any $n \geq 2$.*

Proof. Combined with the folk theorem, Lemma 4 suffices the proof. Let s denote the correlated strategy that represents the group behavior of the agents. Since the average cost vector associated with strategy s Pareto dominates the minimax cost vector, there exists a Nash equilibrium (trigger) strategy that can support correlated strategy s. Therefore, the agents act as if their strategies are in a Nash equilibrium of a repeated game. □

Corollary 20. *Suppose the inner-learning algorithm exhibits the ϵ-safety property for some $\epsilon > 0$. Then, the outcome of IMPRES algorithm in self-play in n-player symmetric games is behavior-equivalent to an approximate Nash equilibrium of a repeated game (ϵ-NE$^\infty$), for any $n \geq 2$.*

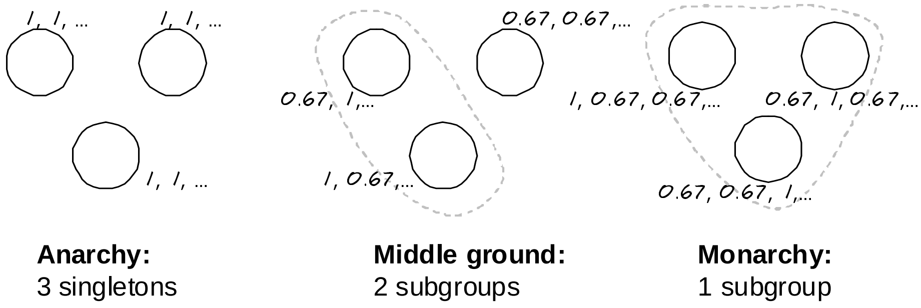


Figure 4.6: Metro vs. Driving: various demographics for 3 agents

Example

The proof is elaborated using an example that illustrates a 3-player case. Let us formulate the metro versus driving example from Section 1.5.1 for three agents: i, j and k. Let the cost of taking metro always be 1; and let the cost function of driving be $\frac{d}{n}$, where d is the number of driving agents and n is the total number of agents ($n = 3$ in this example).

Figure 4.6 illustrates three configurations of this example. The minimax cost in this example is 1.0 for every agent; this is also the value of a Nash equilibrium for all agents (left). The system-optimal solution is achieved in a monarchy configuration (right) where exactly two agents drive (while one agent takes a metro) at each round. The average cost for all agents in the monarchy configuration is approximately $\frac{0.67+0.67+1}{3} = 0.78$.

The most interesting case is the middle-ground configuration where two agents form a correlated strategy while the third agent remains as a singleton. Without the loss of generality, let i be a singleton; and let j and k belong to a subgroup. The double-layered strategy structure for this example was shown earlier in Figure 3.2. In this configuration, agent i's optimal strategy is to always drive. In the correlated strategy, the two agents j and k take turns to drive. Since there will always be two drivers, the driving cost becomes 0.67. Thus, the average cost of agents j and k is $\frac{0.67+1}{2} = 0.84$. On the other hand, agent i's average cost is 0.67 since it is always driving.

Given the strategies of others are fixed, agent i's best response is to stay in a singleton (since moving on to the monarchy only increases its cost). Given the strategy of i is fixed, agents j and k are better off in staying correlated; that is, if either one breaks from a correlated strategy to become a singleton, the configuration turns into an anarchy (the minimax configuration). One may argue that a truly optimal reaction would be to use the anarchy configuration as a threat to move to a monarchy configuration. Nonetheless, using the same reasoning as seen earlier in suboptimal Nash equilibrium examples, this middle-ground configuration is an enforceable Nash equilibrium of a repeated game; more importantly, the quality of the middle-ground solution is significantly better than a single-shot Nash equilibrium in terms of agents' average cost.

4.5 Summary

In this chapter, I described the conceptual model behind the IMPRES algorithm that supports implicit reciprocal strategies, and discussed how the IMPRES model relates to game-theoretic solution concepts.

In general, the interpretations of solution concepts may lead to a philosophical debate. My goal was to bridge the notion of rationality from the theories of decision making and the notion of rational outcome from game theory, in particular when a solution concept describes an output of some learning process over repeated interactions. I argued that if the learning of a set of agents converges to a suboptimal Nash equilibrium, at least one agent is rationally bounded. I stress that this remark is not meant to scrutinize the notion of Nash equilibria. On the contrary, I am strongly inclined to believe that the reason why the notion of Nash equilibria highly appeals is perhaps because it also represents "reasonable" rationality in addition to an ideal one.

To sum up, I formally proved that in symmetric games the agents adopting IMPRES in self-play make reasonably rational decisions as if they are in an approximate Nash equilibrium of a repeated game. Since the boundary of Nash equilibria of a repeated game is large, I proposed the use of a new set of criteria to dissect the space; this analysis will be more meaningful when combined with an empirical study that will be discussed in the next chapter.

Chapter 5

Experiments

"The remedy is to reinforce each of these moods from the other. Conversation will not corrupt us if we come to the assembly in our own garb and speech and with the energy of health to select what is ours and reject what is not." – Ralph Waldo Emerson, Society and solitude

5.1 Introduction

The premise of social learning is that with the knowledge learned from others, agents should always perform at least as good as acting individually. In the last chapter, I presented theoretical results that IMPRES in self-play learns a mutually beneficial solution within the range that is supported by the folk theorem. This chapter provides the empirical counterpart of that argument. The empirical study was carried out on various sets of problems; each problem set is composed of some well-known examples and a comprehensive set of complex problems randomly generated according to the specification of each category.

The evaluation results are organized in three parts. First, a set of desired properties of social learning is defined that will be used to evaluate the experimental results. Next, the main set of experiments performed on symmetric network congestion games are presented; this problem set is further divided into a set of controlled experiments to discuss interesting cases in relation to the tradeoffs of evaluation criteria. Finally, preliminary results on well-known 2-player matrix games are presented to demonstrate potential usefulness of social learning beyond symmetric network congestion games.

5.2 Evaluation criteria

Before defining the desired properties of social learning, this section discusses a set of more general criteria for evaluating a system-wide performance. Some of the criteria have been briefly mentioned in the last chapter; and formal definitions are given here.

5.2.1 Price of anarchy

The price of anarchy introduced in [35] is a relative criterion for measuring the inefficiency of a selfish equilibrium (of a problem). In this context, the efficiency of solution is measured by the objective function value of a target problem. For instance, the objective used in the experiments here is to minimize the sum of all agents' costs, also known as social welfare.

Let Q denote a set of selfish equilibria; and let φ_s denote the objective function value of some solution s. The price of anarchy, denoted here by $\$^A$, is defined as the worst ratio of the objective function value of a selfish equilibrium $q \in Q$ to that of an optimum o^* as follows:

$$\$^A = \max_{q \in \mathrm{Q}} (\frac{\varphi_q}{\varphi_{o^*}}).$$

As discussed earlier, the price of anarchy can be arbitrarily high. On the other hand, there exists a class of games where the price of anarchy is bounded low.

Definition 23 (thin middle-ground)**.** *When the price of anarchy (of problem) is low, selfish equilibria are virtually optimal; that is, the quality gap between selfish solutions and optimal ones is narrow. Specifically in this thesis, the problems with the price of anarchy lower than $\frac{4}{3}$ are referred to as "thin middle-ground" problems.*

I generalize the definition such that the price of anarchy (of a learning algorithm) measures how well a given algorithm can cope with the inefficiency. Given learning algorithm l, let φ_l denote the objective function value of a solution when all agents use algorithm l. The price of anarchy of a learning algorithm, denoted here by $\$^A_l$, is defined as:

$$\$^A_l = \frac{\varphi_l}{\varphi_{o^*}}. \tag{5.1}$$

Depending on how the objective function is determined, the price of anarchy can be used for different purposes. For instance in repeated symmetric games, one can define the objective function φ as the average cost of the worst-performing agent; such that the price of anarchy can be used to verify whether a certain solution is enforceable as a Nash equilibrium.

As discussed in the last chapter, the price of anarchy will be used to further evaluate the solutions concentrating on their quality. In the experiments, the price of anarchy of a pure-strategy Nash equilibrium $\$^{A}_{\mathrm{PNE}}$ is used as a baseline; it is straightforward to see that this sets a higher standard than the general price of anarchy that counts the worst case ratio.

5.2.2 Price of monarchy

The original definition of the price of anarchy suggested that the quality loss of a selfish solution is a direct exchange for a coordination cost [9]. In fact, in the price of anarchy literature, the price of anarchy is commonly referred to as a coordination cost (or coordination ratio) as well. In order to disambiguate the difference between the quality loss and the actual coordination cost, I define a new criterion referred to here as the price of monarchy. Whereas the price of anarchy measures potential quality loss due to selfish decisions, the price of monarchy estimates the practical cost of installing a certain coordination scheme on a multiagent system. For instance in this thesis, the coordination cost will be defined in terms of communication cost.

When the agents act independently, no coordination cost incurs; that is, the coordination cost is optimal when the agents do not communicate at all. On the other hand, the upper bound cost is open ended. In general, the coordination cost depends on how a coordination mechanism is implemented; for instance, supporting a complex negotiation mechanism can be very expensive. In the problems where coordination is in their nature, the agents may be willing to trade in a high communication cost for other merits such as privacy; for instance, while a lengthy negotiation process typifies meeting-scheduling problems, distributed solutions may still be preferred due to privacy reasons. In this experiment, however, the upper bound of the price of monarchy is set to that of a centrally administered system, and thus the algorithms are disregarded when their communication costs exceed this upper bound.

Let ς_l and ς_A denote the coordination cost function of learning algorithm l, and the optimal coordination cost at an anarchy configuration, respectively. The price of monarchy, denoted here by $\$^{M}_{l}$, is:

$$\$^{\mathrm{M}}_{\mathrm{l}} = \frac{\varsigma_l}{\varsigma_A}. \tag{5.2}$$

In the experiments, the coordination cost function ς is defined as an exponential function of communication bandwidth δ; such that $\varsigma = e^{\delta}$. This function avoids a division by zero, and possesses a characteristic that the value grows only gradually (almost linearly) when the value of δ is less than 1 (the bandwidth of a centrally administered system), but rises quickly as the value of δ exceeds 1.

5.3 Experimental hypotheses

This section describes preliminary conditions of the experiments and a set of desired properties of social learning.

5.3.1 Learning algorithms

For the purpose of experiments, two subordinate variations on the IMPRES algorithm are formulated.

IMPRES (I)

First of all, IMPRES is the algorithm that is being evaluated in the experiments.

Without meta-learning (-m)

Without the meta-learning layer, IMPRES operates only on the inner-learning algorithm. Since an anarchy is the only possible configuration in this case, the learned strategies are independent. The inner-learning algorithm described in Chapter 3 learns a best-response strategy, but it does not guarantee convergence. This variation serves two main purposes: 1) to evaluate the inner-learning algorithm alone (when all agents act individually); and 2) to observe the impacts after adding the meta-learning layer.

With meta-learning and with a pre-computed Nash solution (+m+n)

As stressed earlier, IMPRES does not require an absolute threshold such as the minimax values. In this variation, a pure-strategy Nash equilibrium is pre-computed; and the average cost of the pre-computed Nash equilibrium profile is given to all agents in the beginning. Also, the agents have an option of following a centralized signal that *fairly* distributes the Nash equilibrium strategies. Fairness is an important feature here because a pure-strategy Nash equilibrium can generally be unfair. Note that fairness is guaranteed only in the case of symmetric games; and an asymmetric case is discussed in Section 5.6.4. The purpose of having this variation is to verify whether IMPRES (using only a subjective criterion) performs comparably with an algorithm that has an absolute criterion.

5.3.2 Properties of social learning

A set of experiments was carried out to verify whether IMPRES in self-play achieves the following set of desired properties in a finite time.

Given a set of agents $N = \{1, ..., n\}$, let N^a denote a self-play configuration where all agents in N adopt some learning algorithm a; such that N^I denotes IMPRES in self-play. Let $v^m = [v_1^m ... v_n^m]$ denote the minimax cost vector. There exists a finite time T such that at any point of time $t > T$, the agents adopting IMPRES (N^I) achieve the following:

I. (minimax-safety) For all agent $i \in N^I$, its average cost does not exceed its approximate minimax cost; such that $c_i \leq v_i^m + \epsilon$ for some arbitrarily small ϵ.

II. (collusion-safety) The sum of average costs of N^I is at least as low as that of N^{-m}; such that $\sum_{i \in N^I} c_i \leq \sum_{i \in N^{-m}} c_i + \epsilon$ for some arbitrarily small ϵ.

III. (comparability) The sum of average costs of N^I is comparable to that of N^{+m+n}; such that $\sum_{i \in N^I} c_i - \sum_{i \in N^{+m+n}} c_i \leq \epsilon$ for some arbitrarily small ϵ.

First of all, if a learning algorithm satisfies the minimax-safety property, its outcome is behavior-equivalent to an approximate Nash equilibrium. Since the games of interest are symmetric the minimax value is common for all agents; hence, an evaluation on the worst-performing agent will suffice in the results. Next, the collusion-safety property ensures the basic premise of social learning that the agents' performance should be improved when extra knowledge is available from other agents; this can be verified by comparing the results with the $-m$ setting (without social learning). Finally, the comparability property further verifies that rationally bounded agents can learn a mutually beneficial outcome *without* an explicit threat. The last two properties are defined in terms of social welfare, thus will be discussed in terms of the price of anarchy in the results.

5.3.3 How to read results

For the sake of clear comprehension of results, a brief instruction is given on how to read the plots and the tables in the results section. As a reminder, the payoff of an agent is defined in terms of congestion cost; therefore, *the lower, the better.*

Legend

Generally, the results compare the performances of the three variations of the IMPRES algorithm and two absolute baselines as listed in the table below. Note that absolute baselines are used here since IMPRES is the first algorithm that learns non-stationary strategies for more than two players.

Label	Description
IMPRES	the general IMPRES algorithm
-m	IMPRES without meta-learning, inner-learning algorithm only (an anarchy configuration)
+m+n	IMPRES with meta-learning where a centralized signal is available to all agents that prescribes a pre-computed PNE strategy profile (when an absolute criterion exists)
NE[1]	A pure-strategy Nash equilibrium of a single-shot game (PNE) that is pre-computed by using the centralized algorithm described in Section 2.5.1. In some cases, mixed-NE[1] is also discussed.
Optimum (O^*)	An optimal solution that is pre-computed using the centralized algorithm described in Section 2.5.3 (a monarchy configuration)

The minimax-safety result tables

The experimental results verifying the minimax-safety property for a given experiment is presented in a table as shown in the example below. For each problem, the minimax value is pre-computed according to Theorem 21. The property is satisfied if the cost of the worst performing agent (denoted by worst in the table) is lower than the minimax value; the value of ϵ in such cases is 0. The value of ϵ measures the maximum offset from the minimax value when the property is satisfied approximately; such that the outcome of IMPRES is behavior-equivalent to an ϵ-Nash equilibrium of a repeated game.

ID	minimax	worst	ϵ
exp-0	1.000	0.571	0.000

Table 5.1: Example: the minimax-safety property

Figures on the price of anarchy

The price of anarchy figure is introduced to concisely visualize the performance of learning algorithms in terms of the quality of solution.

An example of the price of anarchy figure is shown in Figure 5.1 (left). Let PNE denote a pure-strategy Nash equilibrium of a single-shot game that is pre-computed by using the centralized algorithm described in Section 2.5.1. The x-axis

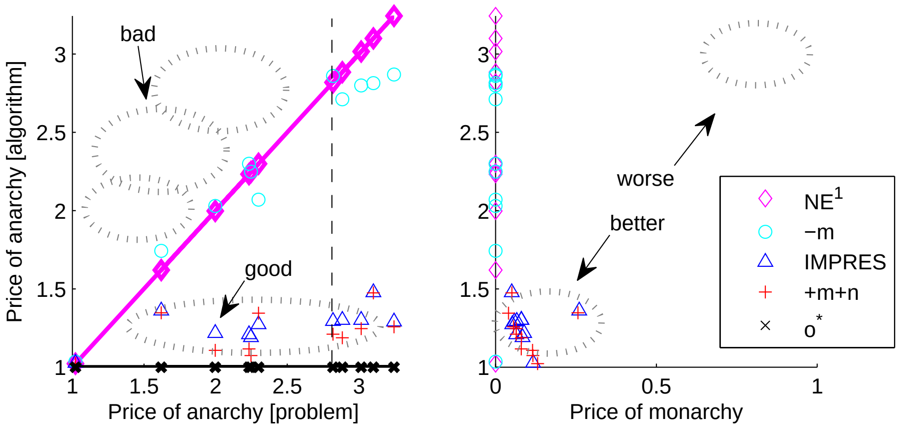


Figure 5.1: Sample figures on price analysis

exhibits the price of anarchy of a PNE, and the y-axis represents the prices of anarchy of the considered algorithms and the baselines. Therefore, the x-axis represents the optimum-baseline ($y = 1$), whereas the diagonal line ($y = x$) represents the PNE-baseline. For each problem g, the averaged social welfare from the last 100 rounds when using each algorithm a represents the objective function value $\varphi_a(g)$ of algorithm a. The resulting prices are plotted on $x = \$^{A}_{\mathrm{PNE}}(g)$.

Hence, a price of anarchy figure visualizes two things: 1) whether the IMPRES algorithm satisfies the collusion-safety and the comparability properties; and 2) how close the quality of solution is to the optima. In brief, the closer to the x-axis, the better the quality of solution is.

Figures on price curve

The price curve of social learning is introduced to eloquently capture tradeoffs between the quality loss and the communication cost. An example of a price curve is shown in Figure 5.1 (right). The x-axis represents the price of monarchy (communication cost), while the y-axis represents the price of anarchy (quality loss). Since the objective on both axes is to minimize the costs, the intersection[1] holds the holy grail of an ideal solution.

[1] Since the price starts from 1, the intersection in the figures is not the origin $(0, 0)$.

The price of anarchy (monarchy) tables

For each set of experiments, a complete result on the price of anarchy (monarchy) is also presented in a table in the appendix section. The columns include the problem ID, the number of alternative paths, followed by a set of prices for the algorithms and the PNE baseline. Since the price of anarchy is relative to the optimum, the optimal price is 1. The price of monarchy is specified inside a parenthesis only when the meta-learning was used. For easier reading, the price of monarchy is expressed in percentage with respect to the cost of a centralized solution. Note that the last column represents the performance of the general IMPRES algorithm.

Problem		Price of anarchy (monarchy)			
ID	$\|S\|$	PNE	-m	+m+n	IMPRES
linear-0	9	1.30	1.30	1.10 (0.22)	1.09 (0.27)

Table 5.2: An example of a complete result

For instance, an example is shown in Table 5.2 indicates that problem *linear-0* contains 9 alternative paths, and the price of anarchy in the case of PNE is 1.30; that is, the average cost of agents in the PNE solution can be 30% higher than in the optimal solution. The performance of the IMPRES algorithm without meta-learning (-m) is comparable with PNE in this problem. The IMPRES algorithm when a PNE is given in the outset achieves lower price of anarchy (1.10) by using 22% of communication cost of what would have been used in a centralized approach. Finally, the general IMPRES algorithm achieves lower price of anarchy (1.09) at 27% of communication cost when compared to a centrally administered system.

5.4 Experimental settings

The results presented in the rest of the chapter are averaged over 5 – 30 trials. During each trial, the current costs are logged for all agents at a discrete interval (e.g., at every 10^{th} iteration). The price of anarchy values on social welfare are in general based on the results over 50,000 iterations in each trial; the number of iterations vary for different problems.

The parameter values that are used in the experiments are empirically chosen; and the rationale for the selected values is discussed in the following subsections. The default parameter values are shown in the table below.

Parameter	Value	Description
T_0	10.0	initial temperature at time 0 for the Boltzmann update rule
δ	0.95	temperature drop rate ($T_{i+1} = \delta T_i$) where i denotes time step
T_l	0.01	the lowest temperature
η^m	$\max(0.01, \frac{1}{10+\text{trials}})$	learning step size; the weight of a new value in value update functions, where *trials* denotes the number of visits to the chosen meta-strategy
ι	0	an initial value of a new meta-strategy
κ	10	the maximum number of meta-strategies
ω	2	inertia parameter for the current meta-strategy (algorithm chooses a new meta-strategy with probability $\frac{1}{\max(f,1)^\omega}$ where f is the number of subscribers)

Table 5.3: Learning parameters

5.4.1 Exploration versus exploitation in meta-learning

In the meta-layer, a policy is represented as a probability distribution over a set of available choices (where each choice of action corresponds to an actual strategy of how to select a path). After trying out a choice of action m, the algorithm updates the value of the choice as follows:

$$Q(m) \leftarrow (1 - \eta^m)Q(m) + \eta^m cost(m). \tag{5.3}$$

Subsequently, the algorithm adjusts the policy (probability distribution) according to the Boltzmann equation below, such that a better-performing choice is selected more frequently.

$$\pi(m') = \frac{\exp \frac{Q(m')}{T}}{\sum_{m'' \in M} \exp(\frac{Q(m'')}{T})}, \forall m' \in M \tag{5.4}$$

Given that, the learning performance depends on the values of step-size η^m and temperature T from the equations above.

First, the learning step size η^m in Equation 5.3 denotes the weight of a new value (cost) in the update equation above; and the value of η^m is determined for each choice m by taking the number of trials (denoting how many times a certain choice has been tried) into account as follows:

$$\eta = \max(z, \frac{1}{x + \frac{\text{trials}}{y}}).$$

While parameters x and y control how quickly the weight of the new value is diminished as the number of trials grows, parameter z limits the minimum weight. If the value of parameter z is 0, then the value of η^m will approach 0 in the limit. On the other hand, if a positive value of z is used, the value of step-size eventually becomes constant z; this makes the algorithm more adaptive to dynamically changing values. At the same time, if the true value of a choice becomes stationary, the algorithm using a constant step-size can still learn the correct value [59].

Second, the Boltzmann equation (Equation 5.4) becomes more sensitive as temperature T drops; that is, when the temperature is very low, the algorithm greedily exploits the current best choice as opposed to exploring for alternatives. In general, this type of exploration scheme is referred to as *decaying exploration* and the algorithm becomes less adaptive to dynamically changing environment. Let T_t denote the temperature at time t; δ some discount factor such that $0 < \delta < 1$; and T_l the lowest positive temperature. The IMPRES algorithm gradually cools down the temperature as follows:

$$T_t = \max(T_l,\, \delta^t T_0) \tag{5.5}$$

Although the value of T_l cannot be 0 (since temperature T is used as a denominator in Equation 5.4), a reasonably small value makes the algorithm greedy. Figure 5.2 displays the algorithm's sensitivity to temperature T for decision making between two alternatives. For example, consider two choices the costs of which differ by 0.01. When the temperature is high ($T = 10$), the algorithm chooses the two options with an equal probability; but when the temperature is low, the better option is exploited, e.g. with probability .99 when $T = 0.02$. The default value of the lowest temperature $T_l = 0.01$ has been chosen to allow persistent exploration; similarly with the step-size, the rationale is to keep the algorithm more adaptive to dynamic settings.

Figures 5.3 – 5.6 show how the performance changes depending on the values of parameters z and T_l, using the metro versus driving example (Section 1.5.1) with linear and polynomial cost functions. This example will be further discussed in the later section; thus the discussion in this section will focus on how the algorithm performs under various parameter settings. For example, the algorithm with the first setting ($z = 0, T_l = 0.002$) employs decaying exploration, and the last setting ($z = 0.01, T_l = 0.01$) is for the default persistent exploration.

In all four settings, a resource usage pattern (e.g., the number of drivers) emerges early in the learning phase (within 2,000 iterations); but it takes a longer time to

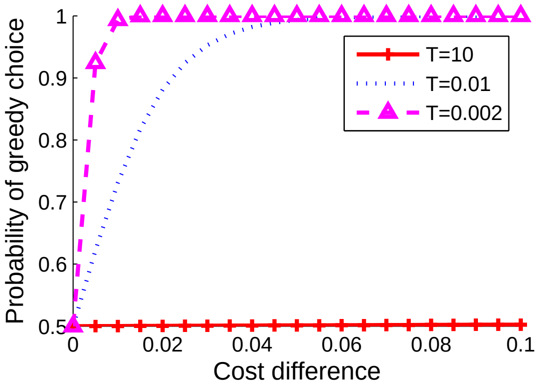


Figure 5.2: Sensitivity to temperature: given 2 alternatives x-axis represents the value difference between the two choices, while y-axis represents the probability of choosing a better option.

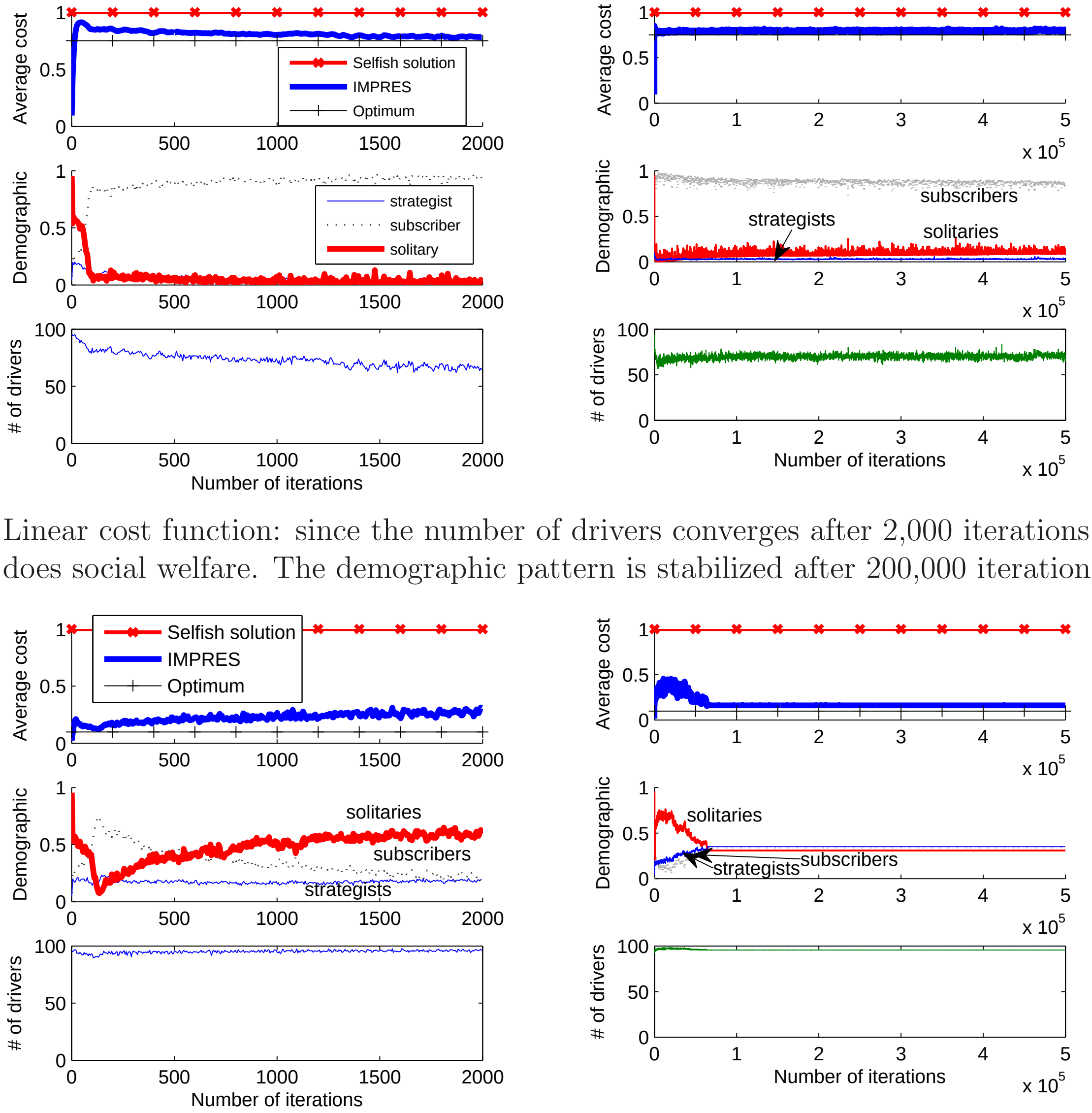
observe convergence[2] in the demographic pattern. In order to illustrate this issue, learning curves of the first 2,000 iterations and of the long-term performances are displayed for each setting.

Generally, if there exist some number of agents already coordinating their actions through social learning, being a solitary is generally a more advantageous option (by taking advantage of the agents exercising social learning); due to this reason, the number of solitaries gradually grows in early iterations. If the number of solitaries becomes high, however, utilizing social learning may appear as a better option. Over a long period of time, the demographic pattern is eventually stabilized.

Fairness : The fairness of an algorithm is commonly measured by the cost of the worst-performing agent. In addition, a standard deviation of the costs incurred by individual agents can be used to measure general fairness of the algorithm. Table 5.4 compares the algorithm's fairness in the metro versus driving example under the four parameter settings. The results from this example show that persistent exploration generally results in fairer solutions. In a later section, the fairness measure will be revisited in 2-player games, where the algorithm under decaying exploration converges to an unfair single-shot Nash equilibrium more frequently.

To sum up, persistent exploration parameters were selected for fairness and adaptivity.

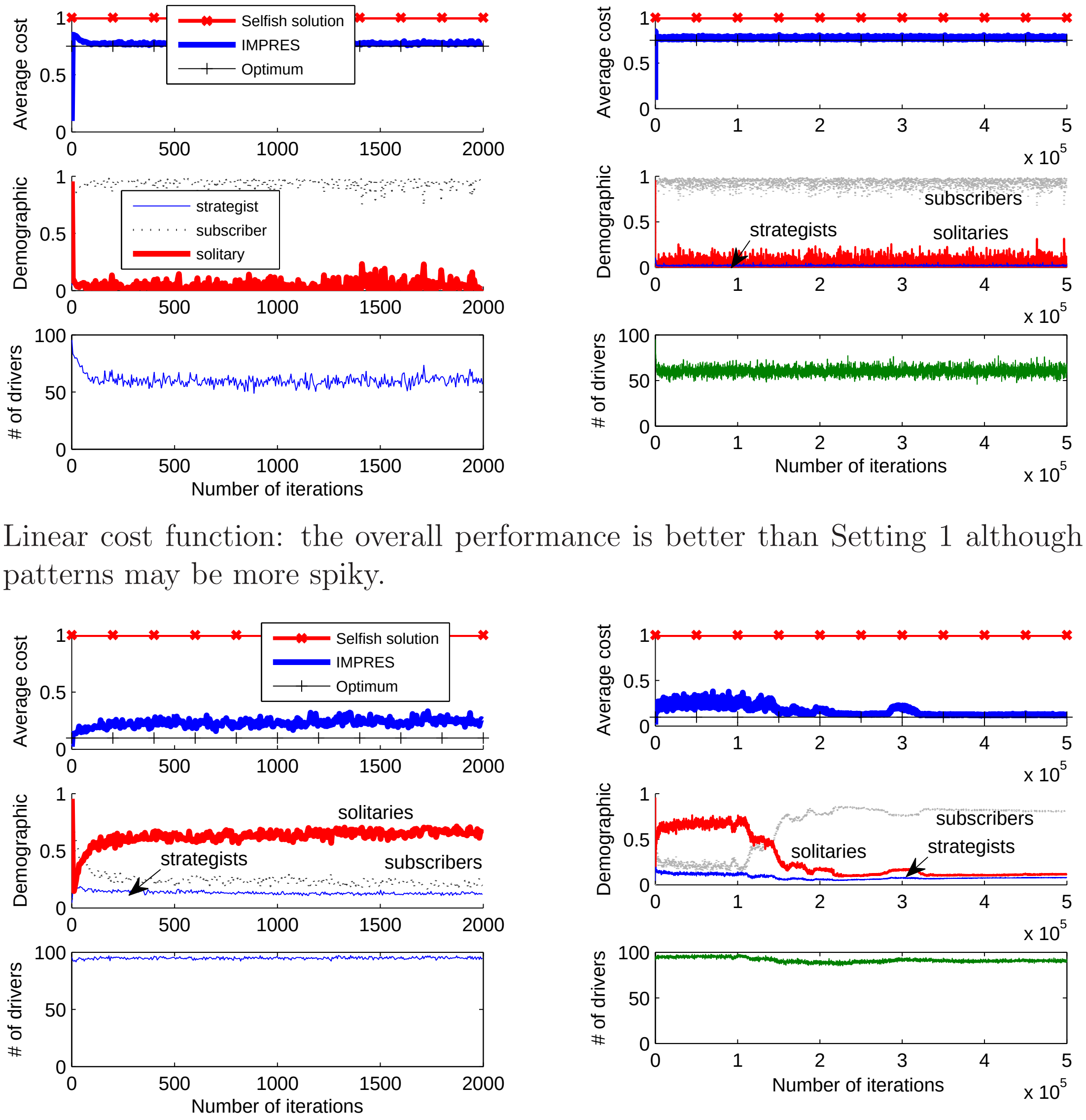
[2] In this section, an algorithm is said to be converged when the averaged value is no longer increased nor decreased over some arbitrarily small variance.



Linear cost function: since the number of drivers converges after 2,000 iterations, so does social welfare. The demographic pattern is stabilized after 200,000 iterations.

Polynomial cost function: the number of solitaries grows in the early learning phase, but converges after 50,000 iterations.

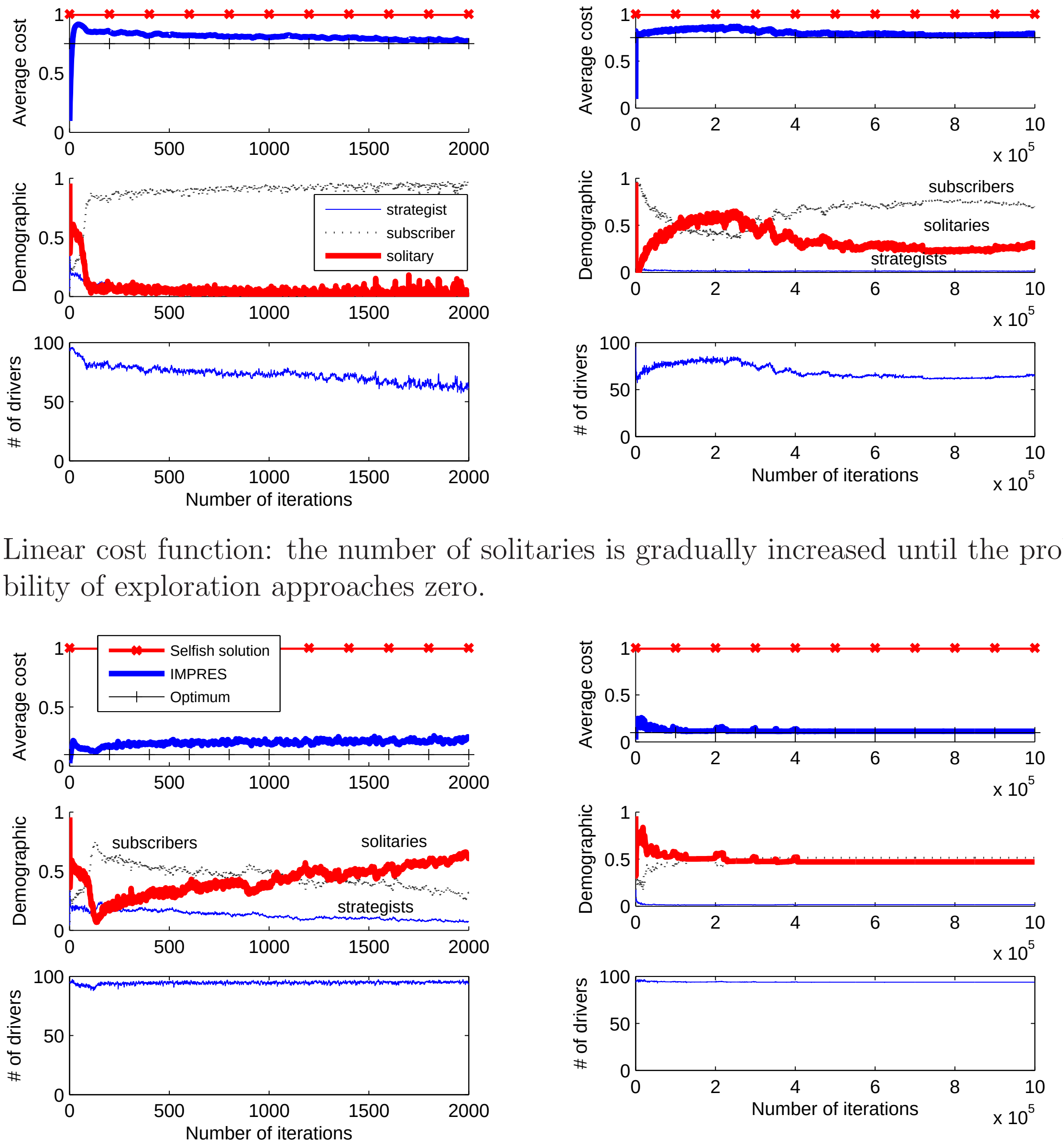
Figure 5.3: Metro vs. Driving (Setting 1. $z = 0, T_l = 0.002$): this setting employs decaying exploration.



Linear cost function: the overall performance is better than Setting 1 although the patterns may be more spiky.

Polynomial cost function: the demographic pattern converges after 200,000 iterations; but the number of solitaries are smaller than Setting 1.

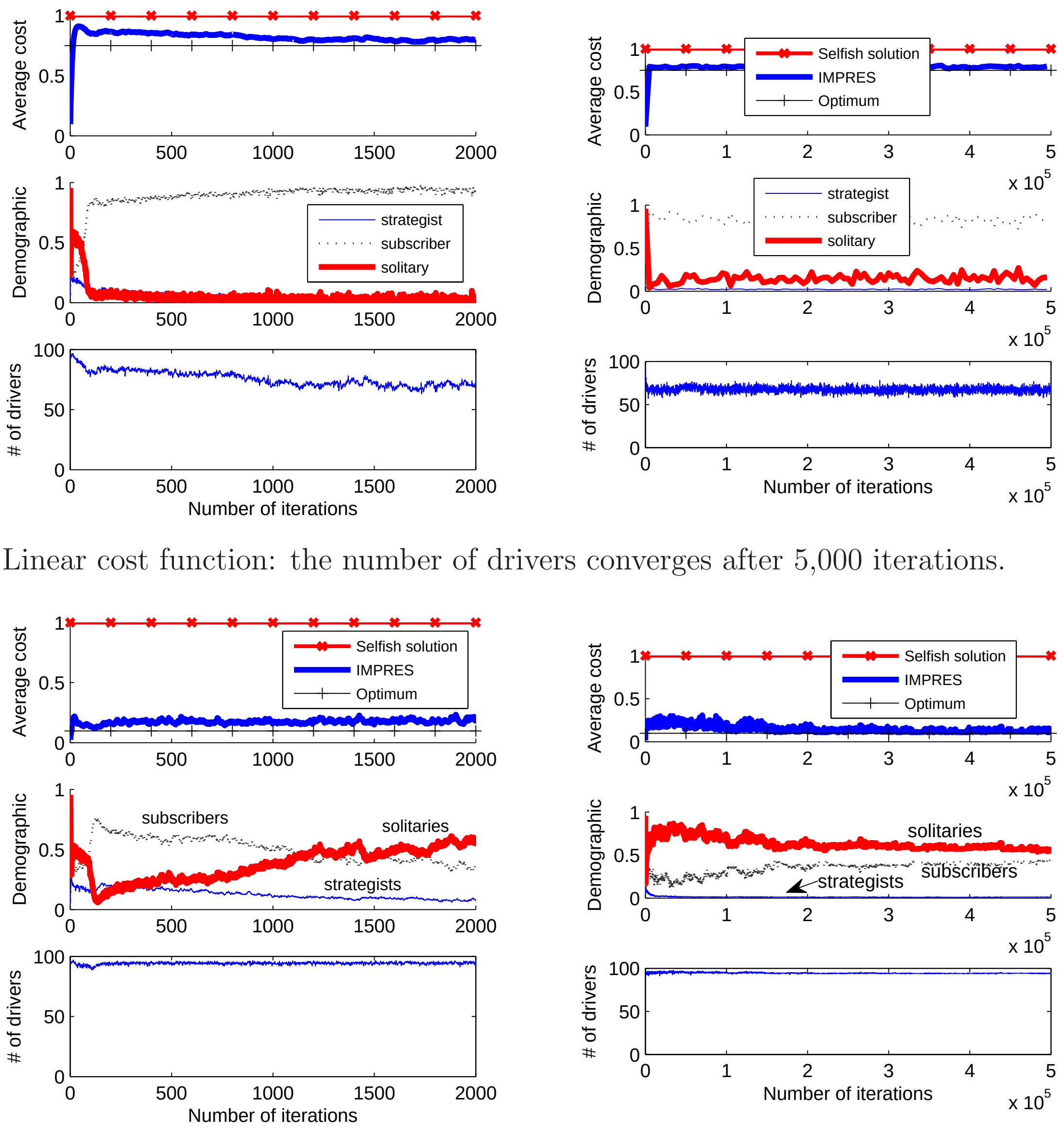
Figure 5.4: Metro vs. Driving (Setting 2. $z = 0, T_l = 0.01$): this setting allows persistent exploration, but the algorithm becomes less adaptive after some time since the weight of new value becomes ignorably light.



Linear cost function: the number of solitaries is gradually increased until the probability of exploration approaches zero.

Polynomial cost function: the average cost is immediately decreased at early learning phase, although demographic pattern converges at around 200,000 iterations.

Figure 5.5: Metro vs. Driving (Setting 3. $z = 0.01, T_l = 0.002$): since the algorithm becomes greedy in this setting, more number of solitaries (free riders) are expected than other settings.



Linear cost function: the number of drivers converges after 5,000 iterations.

Polynomial cost function: similarly with other settings, the average cost drops early, although the demographic pattern converges after 200,000 iterations.

Figure 5.6: Metro vs. Driving (Setting 4. $z = 0.01, T_l = 0.01$): this is the default setting for the main experiments that employs persistent exploration.

Linear cost function

Setting	1	2	3	4
z, T_l	0,0.02	0,0.01	0.01,0.002	0.01,0.01
mean	0.795	0.775	0.8	0.787
max (worst)	0.813	0.778	0.831	0.793
standard deviation	0.03	0.002	0.035	0.002

Polynomial cost function

Setting	1	2	3	4
z, T_l	0,0.02	0,0.01	0.01,0.002	0.01,0.01
mean	0.172	0.15	0.119	0.132
max (worst)	0.290	0.202	0.182	0.203
standard deviation	0.064	0.03	0.049	0.042

Table 5.4: Fairness of the IMPRES algorithm in Metro versus Driving under four parameter settings. The values represent the mean, the max and the standard deviation of the costs incurred by individual agents.

5.4.2 Social learning parameters

As mentioned in Chapter 3, IMPRES introduces a new set of boosting parameters that may have significant impacts on the learning process. The focus of this section is to empirically analyze each parameter's impact on the learning performance of IMPRES.

Initial value of a new meta-strategy (ι) : Recall that the IMPRES algorithm maintains a set of strategists (α-agents) as alternative sources of a strategy; such that each meta-strategy corresponds to an agent (either itself or another agent). A meta-strategy can become unavailable if the corresponding agent is no longer a strategist; at the same time, an agent may add a new strategist from the lookup table. When an agent adds a new meta-strategy, how the agent initially values the newly added meta-strategy plays a significant role in meta-level exploration. For instance, if the initial valuation (expected cost) of a new strategist is set too high, then the new strategy will seldom be explored. On the contrary, if the agent is optimistic about the new strategist (thus expects a low cost), the new strategy will be frequently tried out and thus be fully vetted.

The empirical results shown in Figure 5.7 reflects this speculation that optimistic agents in the long-run achieves better performance in terms of average cost. At the same time, the communication cost is slightly increased as the agents explore more on the other agents' strategies.

Maximum number of meta-strategies (κ) : Continuing on the discussion on the meta-strategies from above, each agent has a set of meta-strategies that dynamically change over time. Parameter κ sets the maximum number of meta-strategies that an agent can keep concurrently. For instance, $\kappa = 7$ means that each agent has choices from being: a solitary, a strategist, or a subscriber to the strategy of one of 5 or fewer other agents. One may consider parameter κ as a search span in the space of correlated strategies as discussed earlier in Figure 4.3.

A set of values of $\{5, 7, 10, 15\}$ were tested for parameter κ, and $\kappa = 10$ was the best-performing value from the set; the results are shown in Figure 5.8. The price of anarchy is mitigated as the size of meta-strategies grows, but the mitigation is statistically insignificant after the size reaches 10. On the other hand, the price of monarchy increases when the size of meta-strategies is larger than 10.

Subscribers' weight (ω) : Although a strategist provides correlated strategies to its subscribers, the strategist does not receive explicit rewards from the subscribers. The parameter ω is introduced to enable a strategist to take implicit rewards from its subscribers by increasing the probability of staying as a strategist with respect to the number of subscribers. Let f_t be the number of subscribers at time t, and let p_t be the probability of choosing a meta-strategy at time t; such that an agent keeps its current meta-strategy with probability of $(1 - p)$.

$$p_t = \frac{1}{f_t^\omega} \tag{5.6}$$

Recall that this parameter affects only when the agent is a strategist, since the number of subscribers f_t is at most 1 otherwise.

When $\omega = 0$, the meta-level decision making is deterministic; that is, an agent selects a meta-strategy at every round. As shown in Figure 5.9, the coordination cost is more sensitive to the values of parameter ω. Specifically, when $\omega = 2$ the price curve indicates that IMPRES learns virtually optimal solutions by forming small-size subgroups.

5.5 Symmetric network congestion games

The main set of experiments are performed on symmetric network congestion games with convex cost functions. As described in Chapter 2, polynomial-time algorithms

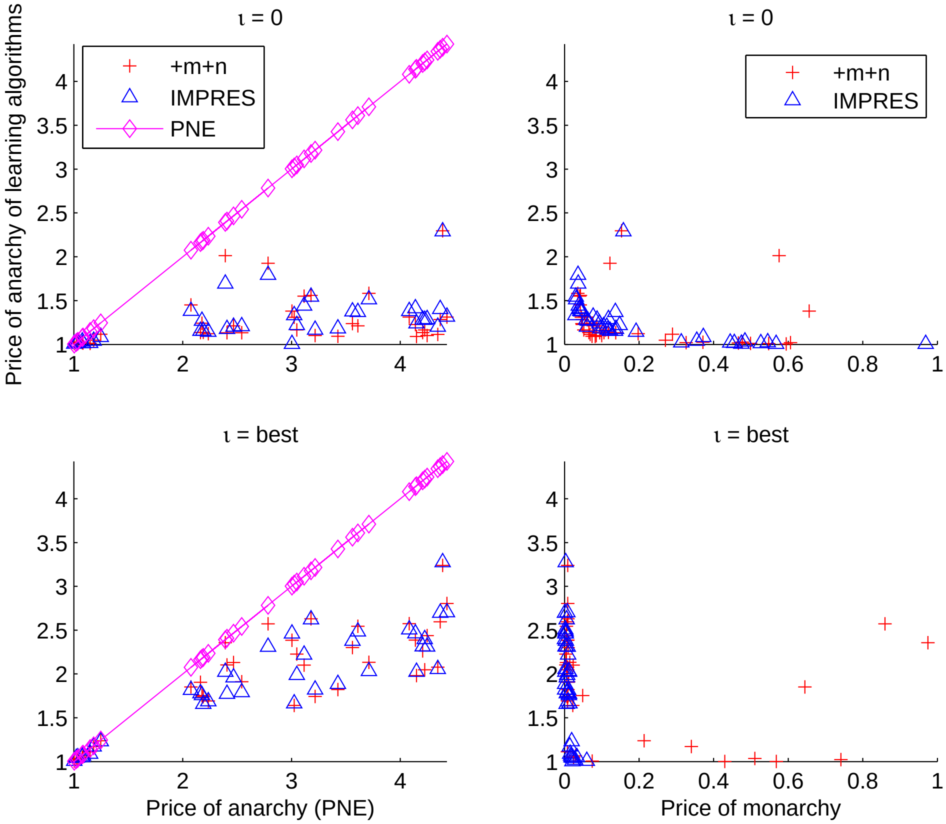


Figure 5.7: On varying ι (value of a new leader): The ι value of 0 fully encourages the agents to explore a new leader. When compared to a case where the current best value of al meta-strategies was used, using constant 0 outperformed significantly.

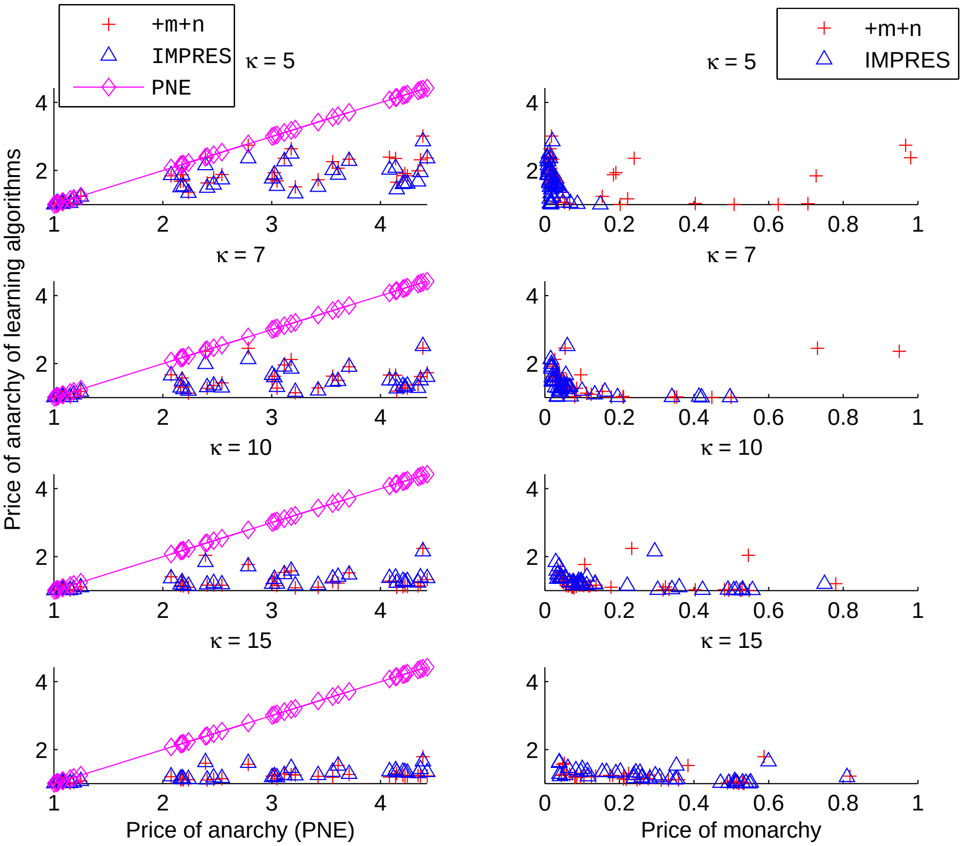


Figure 5.8: On varying κ (maximum number of meta-strategies): The price of anarchy is significantly lower when $\kappa \geq 10$ and the price of monarchy is lower when $\kappa \leq 10$. Thus, the performance is optimized when $\kappa = 10$ in this experiment.

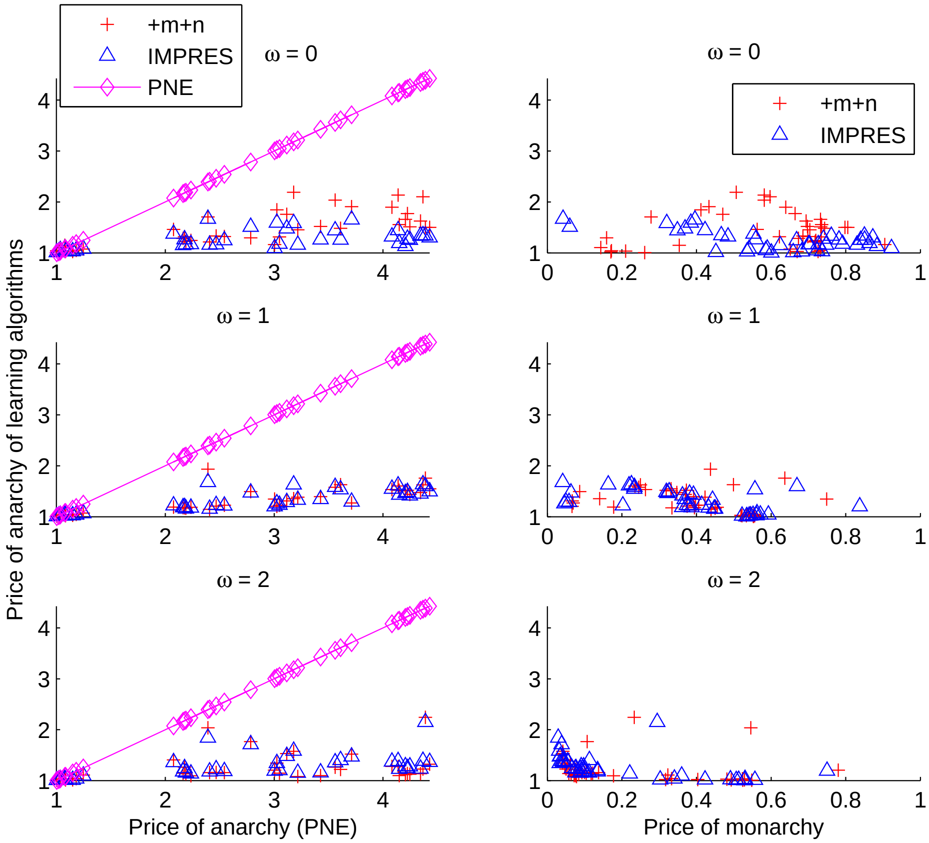


Figure 5.9: On varying ω (voters' weight): The performance of social learning is empirically optimized when $\omega = 2$ both in terms of the price of anarchy and the price of monarchy.

exist for computing both a single-shot Nash equilibrium and a system optimal solution in this class of games, which is convenient for a performance evaluation. Recall that finding a system optimal solution of a symmetric network congestion game with general nondecreasing cost functions is strictly NP-hard [38]. In addition to convex cost functions, some examples of discrete cost functions are also included in the problem set, the optimal solutions of which were pre-computed.

First, I show that the worst punishment for an agent in repeated symmetric congestion game is realized at a single-shot Nash equilibrium. The rest of the section reports on experimental results as follows. The first two sets are grouped according to the type of cost functions: 1) convex cost functions and 2) discrete cost functions.

The last sets of experiments comprise controlled studies designed to analyze the performance of the algorithm on various conditions.

Minimax values of symmetric network congestion games

This section describes how the minimax values are computed in the experiments on symmetric congestion games. Recall that the congestion cost depends only on the number of agents that have chosen the same resource; this requires a full coordination among all agents (excluding a deviator) to realize the minimax value for a deviator.

Theorem 21. *In symmetric network congestion games, the minimax value of an agent is the worst payoff from a pure-strategy single-shot Nash equilibrium profile.*

Proof. Consider a strategy profile where the agent in jeopardy of punishment, referred henceforth as a deviator, chooses the best option when other agents congest all available paths as best as they can. Given a set A of paths, let a denote the best path for the deviator against the minimax strategy of the others. Let $-a$ denote other paths $-a \in A - \{a\}$. Since a is the best option for the deviator given the punishing strategies of others, the cost of adding one more agent to the existing load l_a of path a must be lower than that of the rest; such that $f_a(l_a + 1) \leq f_{-a}(l_{-a} + 1)$ for all $-a \in A$. At the same time, the cost of path a after the deviator is added must be higher than the cost of other paths; such that $f_a(l_a + 1) \geq f_{-a}(l_{-a})$. Suppose not. Then, for each path $-a$ that violates the inequality, there exists at least one extra agent that could have been moved to path a to increase the minimax value for the deviator; this contradicts that the current punishment is the worst possible cost for the deviator. Now, suppose the other agents reason with their original objectives (i.e., to minimize their own costs). Given the strategies of other agents are fixed, each agent stays with its current choice a' since it is the best option; such that $f_{a'}(l_{a'}) \leq f_a(l_a + 1) \leq f_{-a}(l_{-a} + 1)$ for all $-a \in A - \{a\}$. Therefore, it is a pure-strategy Nash equilibrium where the deviator receives the worst possible payoff in the profile. □

Since a pure-strategy Nash equilibrium can be computed in polynomial-time in symmetric network congestion games, the minimax value can also be found in polynomial time.

5.5.1 Games with convex cost functions

In this problem set, the type of cost function is homogeneous for all edges in the same problem. Specifically, three types of convex cost functions are examined: linear, polynomial, and exponential functions. The cost functions elsewhere are a mixture of these three types. As a reminder, the cost function is defined in terms of the number of agents on the same path. For instance, a linear cost function indicates that the cost of a path increases linearly as more agents decide to use the path.

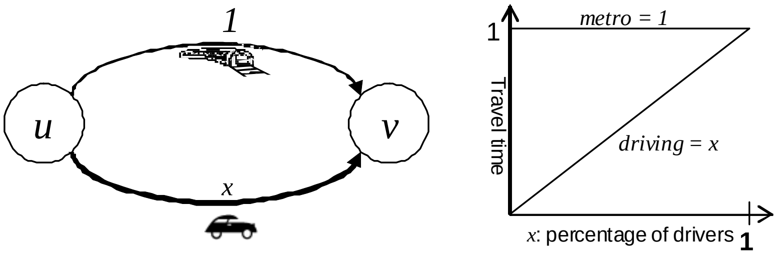


Figure 5.10: Metro vs. Driving: linear cost function ($\$^A = 1.33$)

Simple example: Metro versus Driving

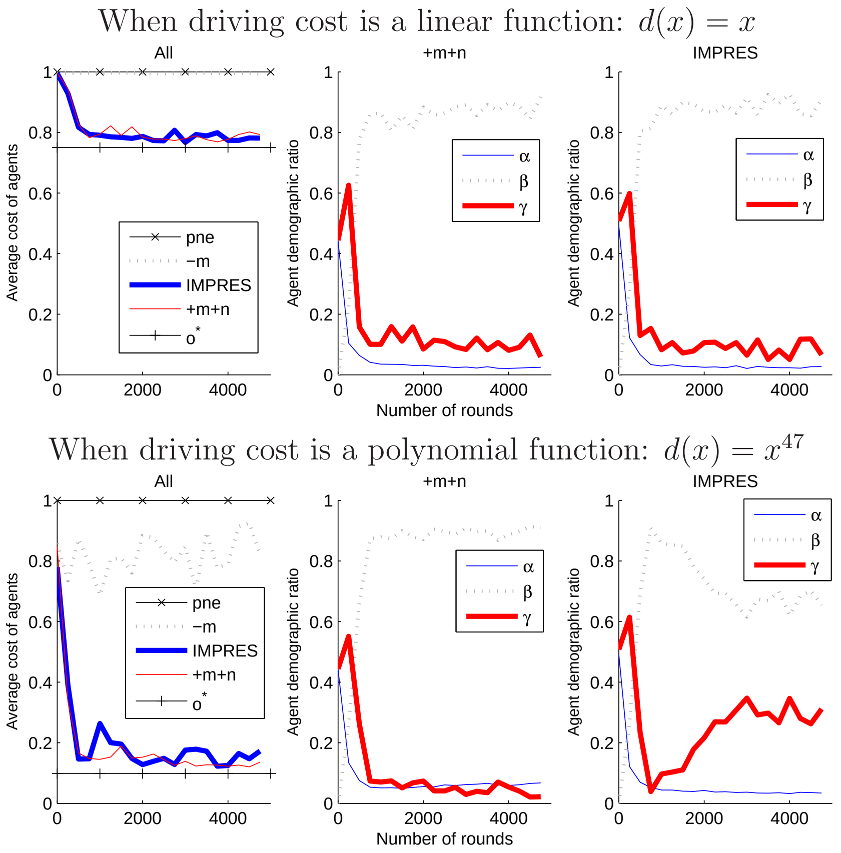
Let us revisit the metro versus driving example from Section 1.5.1 to illustrate symmetric network congestion games. Recall that the cost (travel time) of taking a metro is a constant, say 1; and the cost of driving is a function of the percentage of driving agents x.

This game possesses a dominant-strategy Nash equilibrium where all agents drive. In general, if a game possesses a dominant-strategy equilibrium all stationary learning algorithms including fictitious play and no-regret algorithms necessarily converge to the dominant strategy equilibrium.

Let $d(x)$ denote the driving cost function for load x. Figure 5.11 shows the results for the metro versus driving example with two different driving cost functions: 1) $d(x) = x$; and 2) $d(x) = x^{47}$. The first column shows the travel cost averaged over the set of agents, and the second and third columns show how the agent demographics changed in the meta-learning layer for the $+m+n$ and the IMPRES algorithms, respectively.

It is undoubtable that the minimax-safety property is satisfied, since the minimax value is the worst possible value in this example; also the collusion-safety property is safely achieved since the -m algorithm converged to the dominant strategy equilibrium[3]. For the comparability property, a statistical significance test failed to reject

[3] For exactly one agent, driving is not a dominant strategy; thus there exists one more pure-strategy equilibrium where exactly one agent takes a metro. Due to this reason, the average cost of -m appear better when compared to the worst possible equilibrium solution.


When driving cost is a linear function: $d(x) = x$

When driving cost is a polynomial function: $d(x) = x^{47}$

Figure 5.11: Metro vs. Driving with convex cost functions: Each row corresponds a linear and a polynomial cost functions. The system optimal solutions are 0.75 and 0.098 for each case, respectively. In both cases, the average cost of a dominant-strategy Nash equilibrium is 1.

the hypothesis that the IMPRES algorithm and the $+m+n$ variation are comparable in the linear cost function case. Since the confidence interval includes 0, the evidence supports that the two algorithms perform comparably. In the polynomial function case, the hypothesis was rejected, but the confidence interval for the cost difference falls below 0.047 with 95% confidence level. I conclude that the comparability property is achieved for $\epsilon < 0.05$ in this example. Throughout the experiments, a similar trend is observed.

An interesting difference is observed in the agent demographics in the two results. As discussed earlier in Section 1.5.1, in the case of nonlinear cost functions, just a few agents can make a huge improvement not only to their own benefits but also to social welfare. When a more concrete threat existed ($+m+n$), a largely correlated strategy was enforced. On the other hand, without an explicit threat IMPRES learned correlated subgroups that are smaller in size, leaving nearly 40% of the population as singletons; yet in terms of social welfare, the two algorithms were indifferent.

This set of results concurs with my claim from the last chapter that: while the optimal solutions may be difficult to stabilize without an explicit threat, close-to-optimal middle-ground solutions can be enforced as a stable outcome if the agents can learn to incorporate other agents' strategies only when doing so improves their performances.

Complex network congestion games with convex cost functions

This section presents a set of results on complex games with linear, polynomial, and exponential cost functions. All network congestion games in this set is composed of 5 - 10 vertices and 10 - 20 edges; and the number of agents is 100.

First, the minimax-safety property was verified by comparing the cost of the worst-performing agent to the minimax value; this result can be found in Table 5.5. The system-wide performance results are compactly represented in Figure 5.12; and the complete set of results for each type of cost function can be found in Table B.1 – B.3 in Appendix.

In summary, the IMPRES algorithm achieved the following on symmetric network congestion games with linear, polynomial, and exponential cost functions:

- In problems with high price of anarchy ($\$^A \geq \frac{4}{3}$),
  - the minimax-safety property is satisfied for $\epsilon = 0$; and
  - the collusion-safety property is satisfied for $\epsilon = 0$
- In thin middle-ground problems with low price of anarchy ($\$^A < \frac{4}{3}$),
  - the minimax-safety property is satisfied for $\epsilon = 0.03$; and
  - the collusion-safety property is satisfied for $\epsilon = 0$.
- The comparability property was generally satisfied. Statistically, IMPRES significantly outperformed the $+m+n$ variation in this set, but I conclude that the performance is comparable since the confidence interval is near to 0 (-0.0271 -0.0017).

Games with linear cost functions

ID	minimax	worst	ϵ
linear-0	0.992	0.903	0.000
linear-1	0.997	0.844	0.000
linear-2	0.991	0.847	0.000
linear-3	0.982	0.846	0.000
linear-4	0.992	0.871	0.000
linear-5	0.787	0.792	0.005
linear-6	0.937	0.873	0.000
linear-7	1.000	1.055	0.055
linear-8	2.628	2.626	0.000
linear-9	0.891	0.924	0.033

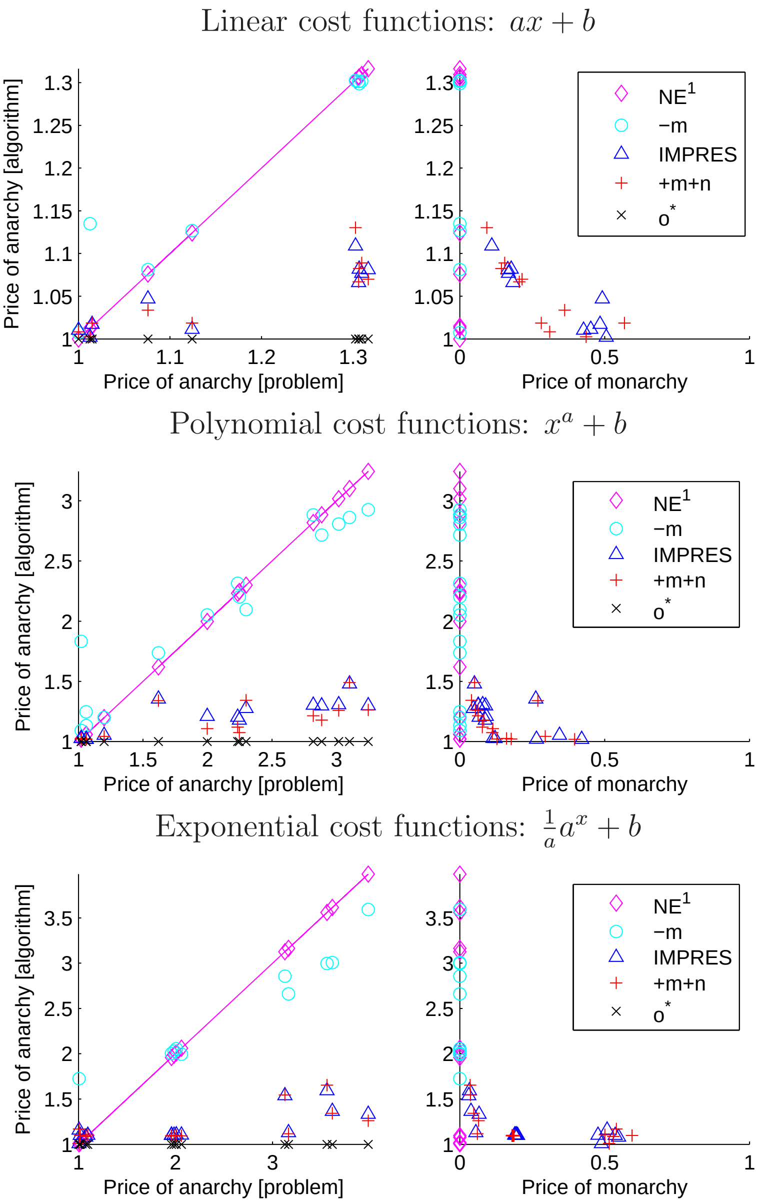
Games with polynomial cost functions

ID	minimax	worst	ϵ
poly-0	0.760	0.413	0.000
poly-1	1.051	0.605	0.000
poly-2	0.969	0.475	0.000
poly-3	0.886	0.483	0.000
poly-4	1.128	0.600	0.000
poly-5	2.005	1.749	0.000
poly-6	0.758	0.448	0.000
poly-7	0.923	0.548	0.000
poly-8	0.791	0.533	0.000
poly-9	1.331	0.840	0.000
poly-10	0.463	0.484	0.021
poly-11	0.602	0.604	0.002
poly-12	0.382	0.357	0.000
poly-13	1.216	1.189	0.000
poly-14	0.595	0.623	0.028

Games with exponential cost functions

ID	minimax	worst	ϵ
exp-0	1.000	0.571	0.000
exp-1	1.000	0.575	0.000
exp-2	1.000	0.567	0.000
exp-3	1.000	0.577	0.000
exp-4	1.000	0.579	0.000
exp-5	0.149	0.173	0.024
exp-6	0.341	0.367	0.026
exp-7	0.275	0.275	0.000
exp-8	0.219	0.223	0.004
exp-9	1.119	1.123	0.004
exp-10	1.256	0.532	0.000
exp-11	1.107	0.574	0.000
exp-12	1.081	0.523	0.000
exp-13	1.012	0.433	0.000
exp-14	1.163	0.657	0.000

Table 5.5: The minimax-safety property on the games with convex cost functions



Linear cost functions: $ax + b$

Polynomial cost functions: $x^a + b$

Exponential cost functions: $\frac{1}{a}a^x + b$

Figure 5.12: Games with convex cost functions

- Most importantly, the quality of solution found by IMPRES is close to optimal. Specifically, when the price of anarchy of a problem is greater than 2, the average cost was reduced by more than 40%.
- The use of communication among agents adopting IMPRES is effective; the average communication cost of IMPRES is approximately 30% of that in a centralized approach.

5.5.2 Games with discrete cost functions

The problems with discrete cost functions are the most challenging class in general. Since polynomial-time algorithms do not exist for computing the baseline solutions in this class, the experiments is conducted on two known examples.

The El Farol bar problem

The El Farol bar problem described in Section 1.5.2 is a well-known congestion game with a discrete cost function. Any combination of exactly τ agents attending the bar while the rest stay home is a pure-strategy Nash equilibrium of this problem. For instance, if an agent knows that other τ agents will definitely attend the bar, its best response is staying home, and vice versa. Although such a pure-strategy NE is optimal, it is an unlikely outcome not only because it is unfair, but also because there are so many of them[4] such that it is difficult for the agents to agree on one of them. On the other hand, a mixed-strategy NE where approximately τ agents attend the bar is a more natural outcome, but it is suboptimal.

The results are displayed in Figure 5.13. The -m variation slowly approaches the mixed-strategy equilibrium after some period of oscillation. When a fair NE signal is available (+m+n), the agents formed a centralized correlated strategy around the NE signal. In this example, therefore, the +m+n configuration can be viewed as a centrally administered system. On the other hand, the IMPRES agents self-organized into a 10-correlated strategy on average (including 6 singletons on average); and approximately 59 agents (below threshold $\tau = 60$) alternately attended the bar each night.

In this example, IMPRES exhibits all three desired properties of social learning. In particular, IMPRES learns a solution the quality of which is comparable to a cen-

[4] Given n agents and threshold τ, the number of pure-strategy Nash equilibria of the bar problem is the number of combinations of n elements, taken τ at a time. For instance, for $n = 100, \tau = 60$, $\begin{pmatrix} 100 \\ 60 \end{pmatrix} = 1.3746e + 028$.

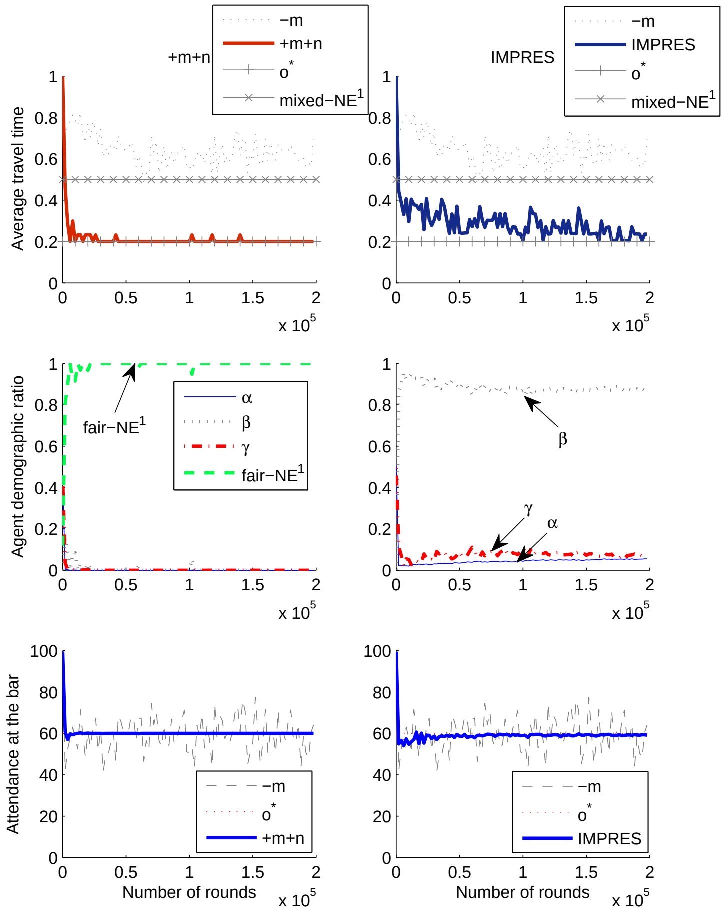


Figure 5.13: The El Farol bar problem: The two columns compare the performances of IMPRES with the $+m+n$ variation. The $-m$ variation (appeared in both columns) approaches a mixed-strategy NE^1; and with meta-learning, the average costs of IMPRES and the $+m+n$ variation approach the optimal value. While the $+m+n$ variation forms a monarchy around the pre-computed NE strategy, IMPRES forms on average 10 subgroups (10-correlated strategy), among which 6 subgroups were singletons.

trally administered system, yet its coordination is far more effective (approximately 20% of a centrally administered system) since the number of agents in each correlated strategy was small.

A game with a high price of collusion

In contrast to the price of anarchy that measures the inefficiency of selfish equilibria, the price of collusion is a criterion for measuring the amount of damage that a collusion can cause to social welfare [29]. Suppose that agents are partitioned into a set of coalitional subgroups such that the agents in a group share a common objective of minimizing the "group average cost" as opposed to minimizing each individual cost. A set of subgroups is in a coalitional equilibrium if, for all subgroups, changing one agent's strategy does not decrease the average cost of the group. Let C denote a set of coalitional equilibria. Analogously to the price of anarchy, the price of collusion $\$^C$ is defined as the worst ratio of the objective function value of a coalition equilibrium to that of a system optimum as follows:

$$\$^{\mathrm{C}} = \max_{c \in C}\left(\frac{\varphi_c}{\varphi_{o^*}}\right).$$

In symmetric nonatomic congestion games, the price of collusion is always 1, meaning that cooperation always contributes positively to social welfare. Unfortunately, this result does not hold in the atomic congestion games. Table 5.6 shows an example borrowed[5] from [29].

$$l_1(x) = \begin{cases} 0 & x \leq k-1 \\ 0.2 & x = k \\ 1 & x \geq k+1 \end{cases} \qquad l_2(x) = \begin{cases} 0 & x \leq k \\ 0.1 & x = k+1 \\ 1 & x \geq k+2 \end{cases}$$

Table 5.6: Discrete cost network congestion game (Hayrapetyan's example)

This problem is similar to the bar problem from the previous section except that it is far more difficult to realize the optimal solution since the load has to be divided into exactly $k-1$ and $k+1$, and all the other solutions are extremely undesirable including any approximate ones.

Suppose that there are $2k$ agents. A pure-strategy Nash equilibrium $(k-1, k+1)$ exists that sends $k-1$ agents to path 1 and $k+1$ agents to path 2. This Nash equilibrium is also the system optimal solution, and the average cost is $\frac{1}{2k}(0 \times (k -$

[5] I re-scaled the costs to $[0, 1]$ range.

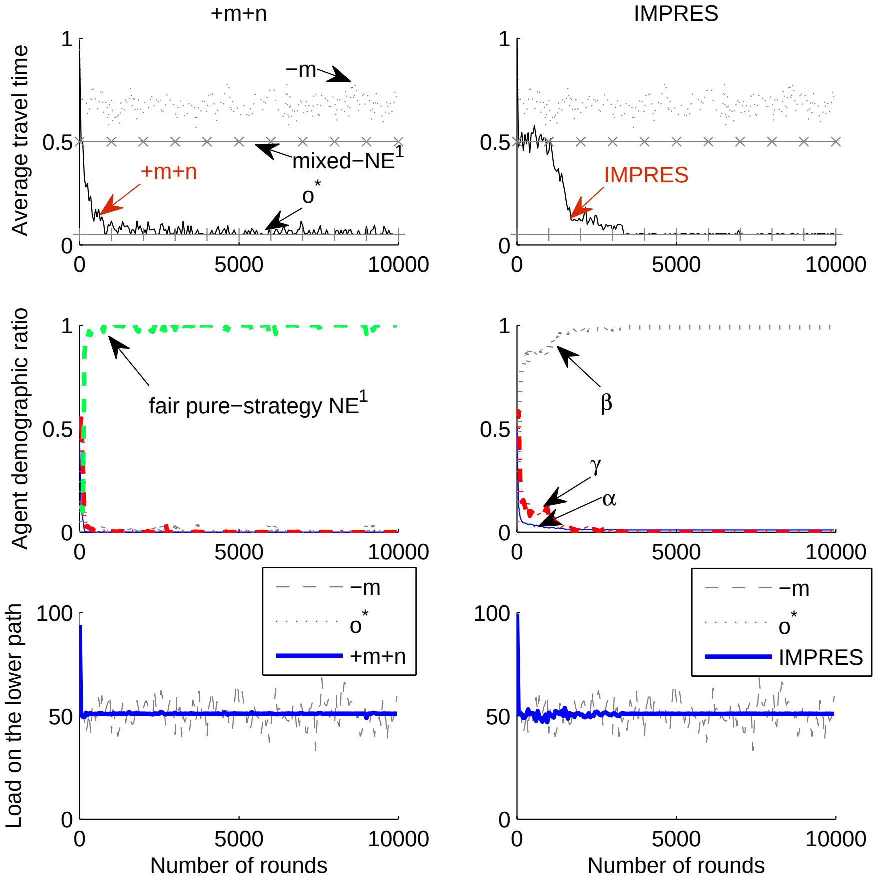


Figure 5.14: Hayrapetyan's example: The -m variation failed to converge within 30,000 iterations. With meta-learning, both IMPRES and +m+n achieved optimal outcomes by forming a monarchy configuration. Whereas agents adopting +m+n formed a monarchy around the precomputed NE signal, IMPRES agents do so through self-organization.

$1) + 0.1 \times (k+1)) = \frac{0.1(k+1)}{2k}$. Similarly with the bar problem, there are $\begin{pmatrix} 2k \\ k-1 \end{pmatrix}$ different optimal but unfair equilibria in this case.

If each pair of agents forms a coalition, then a strategy profile that splits each pair into two paths is a coalition equilibrium (k, k), and the average cost is $\frac{1}{2k}(0.2k + 0k) = 0.1$. Thus, the price of collusion is $\frac{2k}{k+1}$; this is the worst possible case in symmetric atomic network congestion games with convex cost functions.

This example is used to demonstrate that agents adopting social learning algorithms learn a correlated strategy only when the correlated strategy is an improvement from a selfish solution. By Theorem 21, the minimax value of this example is 0.1. Note that the coalition equilibrium described above is *not* a feasible outcome of the IMPRES algorithm, since the cost when participating in a coalitional subgroup is higher than the minimax value (0.1) for the k agents whose cost is 0.2.

Figure 5.14 compares the performance of IMPRES with the $+m+n$ variation. Similarly with the bar problem, the resulting strategy of $+m+n$ represents a centralized correlated strategy that randomly divide the agents into $k-1$ and $k+1$ each time. In this example, the outcome of IMPRES also stabilized in a monarchy configuration. Given the nature of the discrete cost function that does not allow any approximate solution, a monarchy is a rational choice for the population.

To sum up, in the two known problems with discrete cost functions where a large set of "optimal but unfair" pure-strategy Nash equilibria exists, IMPRES satisfies all three properties of social learning; and achieves close-to-optimal solutions.

5.5.3 On varying number of players

This set of experiments evaluates the scalability of IMPRES with respect to the number of agents. For the same set of network congestion games, the performance of the algorithm was measured for: 100, 500, and 1000 agents. Note that an increased population also slightly increases the price of anarchy in some problems; for instance in problem pop-0, the price of anarchy is increased from 3.03 to 3.27 when population was increased from 100 to 500.

Figure 5.15 presents the results averaged over 7 trials; the values from each trial are based on the average cost of all agents from the last 3000 iterations[6]. The complete result can be found in Table B.8 in Appendix. According to the desired properties, IMPRES is scalable with respect to the number of agents. The performance slightly degrades as the number of agents increases. In the experiments, all

[6] During the experiments, the costs were logged at every 100^{th} iteration, thus the values are averaged over 30 data points to be precise.

three sets are conducted for 10,000 iterations. A more probing will be necessary to determine whether the performance on larger population set can be further improved by adjusting other parameters such as running a longer experiment.

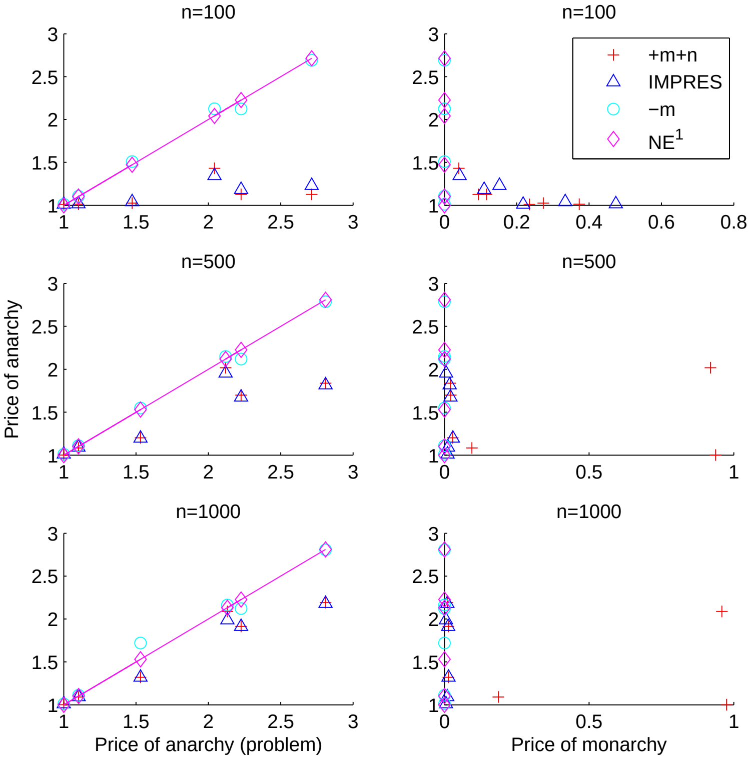


Figure 5.15: Scalability with respect to population size

5.5.4 On varying problem size

This set of experiments evaluates the scalability of IMPRES in games of various sizes. The size of a network congestion game is determined based on the number of alternative paths (denoted by $|S|$) in the game. Specifically, the problem set consists of games with 3, 6, 10, and 15 paths.

The results on the minimax-safety property can be found in Table 5.7; and the results on social welfare are plotted in Figure 5.16 – 5.17, and the complete set of results for each size problems can also be found in Table B.4 – B.7 in Appendix.

In the price curves (figures on the second rows), the problems are subdivided according to three levels of price of anarchy, in order to clearly display the trade-off between the quality loss and the communication cost. Interestingly, the results showed the thin middle-ground problems generally incurred higher communication costs than those with the high price of anarchy. This observation can be interpreted that a larger subgroup is formed when it is difficult to make an improvement.

In summary, this set of results showed that IMPRES is scalable with respect to problem size.

5.5.5 On dynamically changing population

This set of experiments examines the robustness of the IMPRES algorithm when the constituents of an agent population gradually change over time. The motivation for this set of experiments is to model how a social norm is maintained when the individual members of a society are gradually replaced with a new generation over time. Note that the properties of social learning are defined for a static population that does not change over time.

In this experiment, a randomly selected agent was replaced with a new agent every d^{th} iteration for $d = \{1, 5, 10\}$. A set of 40 randomly generated games were used; this set satisfies the three properties of social learning under a static population assumption. The minimax-safety property still holds under moderate population changes. The ϵ values are listed in Table 5.8. The results in Figure 5.18 show that the performance was stable with gradual population changes in this set of problems.

While I find this result promising, a more in-depth evaluation will be necessary to make a more general remark for the case of dynamically changing populations. Furthermore, the learning of newcomers leads to my future work on other types of social learning algorithms; for instance, a newcomer may be able to effortlessly learn to act rationally by observing what the current members of a society has already learned.

5.6 Two-player matrix games

Although this thesis mainly focuses on symmetric congestion games, the notion of social learning can be applied to more general problems beyond this class. This section demonstrates that IMPRES in self-play converges to an optimal Nash equilibrium of repeated games in some of the well-known 2-player games. In order to be consistent with congestion games, the payoffs of all the games presented in this section should also be interpreted as cost (penalty) instead of rewards.

The thin middle-ground problems with low price of anarchy ($\$^A < 1.3$)

ID	minimax	worst	ϵ
h1p3-0	0.495	0.497	0.002
h1p3-1	0.970	0.870	0.000
h1p3-2	0.436	0.400	0.000
h1p6-0	0.303	0.324	0.021
h1p6-1	0.748	0.595	0.000
h1p6-2	0.624	0.662	0.038
h1p10-0	0.394	0.406	0.012
h1p10-1	0.840	0.817	0.000
h1p10-2	0.768	0.757	0.000
h1p15-0	0.859	0.914	0.055
h1p15-1	0.907	0.861	0.000
h1p15-2	0.862	0.823	0.000

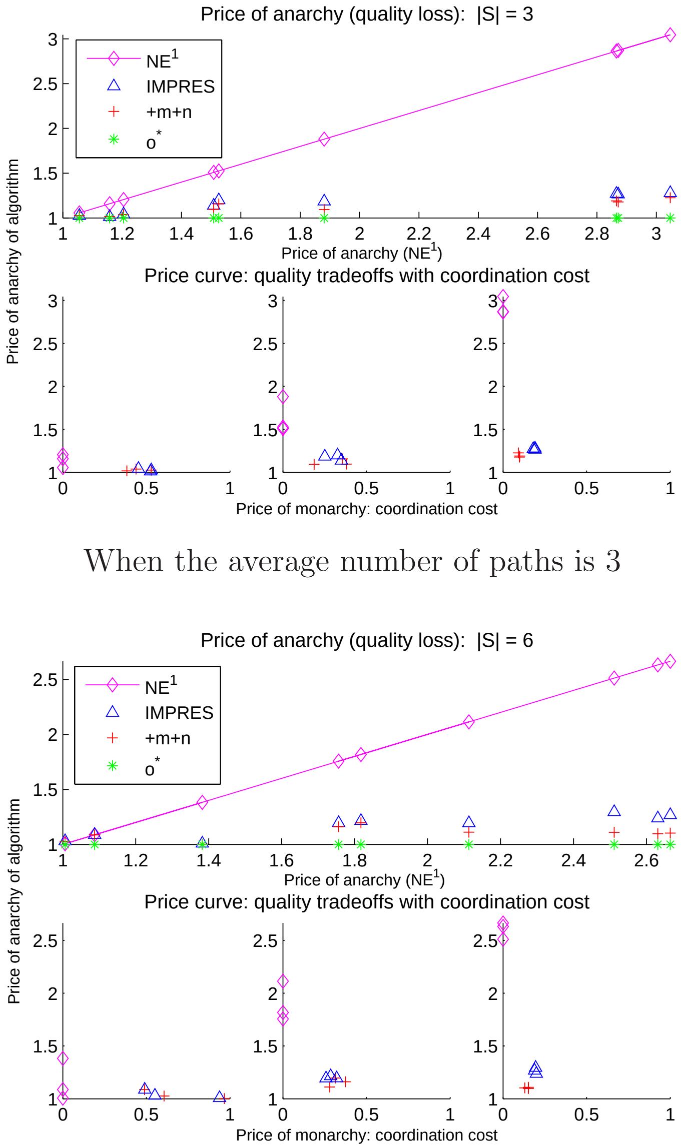
Modest price of anarchy ($1.3 \leq \$^A < 2$)

ID	minimax	worst	ϵ
h2p3-0	0.287	0.232	0.000
h2p3-1	0.180	0.157	0.000
h2p3-2	0.392	0.267	0.000
h2p6-0	0.426	0.270	0.000
h2p6-1	0.210	0.163	0.000
h2p6-2	0.199	0.148	0.000
h2p10-0	0.289	0.235	0.000
h2p10-1	0.190	0.152	0.000
h2p10-2	0.222	0.137	0.000
h2p15-0	0.978	0.533	0.000
h2p15-1	1.000	0.679	0.000
h2p15-2	0.353	0.206	0.000

High price of anarchy ($\$^A > 2$)

ID	minimax	worst	ϵ
h3p3-0	0.981	0.457	0.000
h3p3-1	0.792	0.403	0.000
h3p3-2	0.854	0.441	0.000
h3p6-0	0.467	0.267	0.000
h3p6-1	0.525	0.296	0.000
h3p6-2	0.580	0.318	0.000
h3p10-0	1.002	0.582	0.000
h3p10-1	0.762	0.388	0.000
h3p10-2	0.811	0.435	0.000
h3p15-0	0.931	0.499	0.000
h3p15-1	0.934	0.501	0.000
h3p15-2	0.725	0.372	0.000

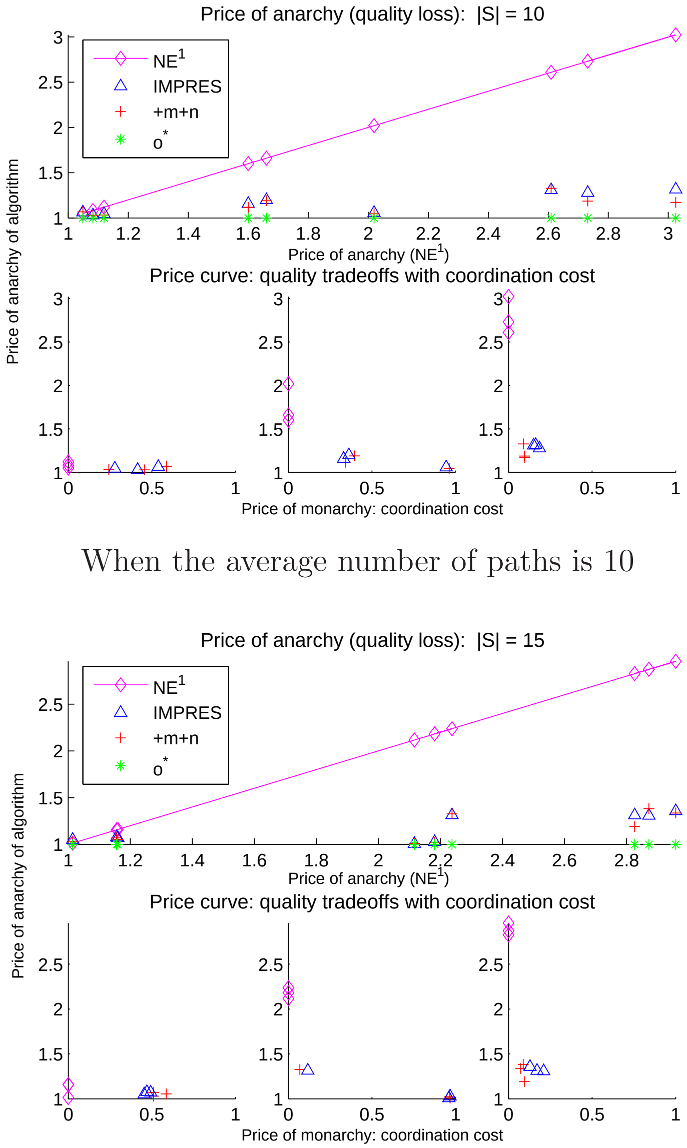
Table 5.7: The minimax-safety property on the problem with various size



When the average number of paths is 3

When the average number of paths is 6

Figure 5.16: Small problems



When the average number of paths is 10

When the average number of paths is 15

Figure 5.17: Large problems

ID	ϵ_{10}	ϵ_5	ϵ_1
mix40-0	0.000	0.026	0.126
mix40-1	0.016	0.012	0.012
mix40-2	0.117	0.123	0.128
mix40-3	0.063	0.058	0.059
mix40-4	0.000	0.000	0.005
mix40-5	0.092	0.095	0.092
mix40-6	0.040	0.043	0.053
mix40-7	0.000	0.004	0.033
mix40-8	0.000	0.000	0.049
mix40-9	0.000	0.000	0.000
mix40-10	0.000	0.000	0.000
mix40-11	0.000	0.000	0.000
mix40-12	0.000	0.000	0.000
mix40-13	0.000	0.000	0.000
mix40-14	0.000	0.000	0.059
mix40-15	0.000	0.000	0.000
mix40-16	0.000	0.000	0.066
mix40-17	0.000	0.000	0.000
mix40-18	0.117	0.131	0.317
mix40-19	0.000	0.000	0.000
mix40-20	0.000	0.000	0.000
mix40-21	0.000	0.000	0.000
mix40-22	0.000	0.000	0.000
mix40-23	0.000	0.000	0.000
mix40-24	0.000	0.000	0.000
mix40-25	0.000	0.000	0.000
mix40-26	0.000	0.000	0.000
mix40-27	0.000	0.000	0.000
mix40-28	0.000	0.000	0.000
mix40-29	0.000	0.000	0.000
mix40-30	0.000	0.000	0.000
mix40-31	0.000	0.000	0.000
mix40-32	0.000	0.000	0.000
mix40-33	0.000	0.000	0.000
mix40-34	0.000	0.000	0.000
mix40-35	0.000	0.000	0.000
mix40-36	0.000	0.000	0.000
mix40-37	0.000	0.000	0.000
mix40-38	0.000	0.000	0.000
mix40-39	0.000	0.000	0.000

Table 5.8: The minimax-safety property when a randomly selected agent is replaced every $(10, 5, 1)^{th}$ round

5.6.1 Inner-learning algorithm for 2-player matrix games

The best-response algorithm in Section 3.3.3 is focused on network congestion games. In this section, an algorithm for computing best response strategy is given for 2-player matrix games. In 2-player games, only two configurations are feasible: an anarchy where each agent makes independent decisions, or a monarchy where one is a strategist while the other is a subscriber.

Let R and C denote a set of actions available to the row and to the column players, respectively. An agent's strategy is represented as a probability distribution over all pairs of joint actions (r, c) where $r \in R, c \in C$. Initially, the algorithm evenly distributes probability mass among all joint strategies.

In an anarchy configuration, each agent selects a strategy pair according to the probability distribution of its current strategy, plays its part from the pair, receives a payoff, and updates the probability using Boltzmann exploration according to

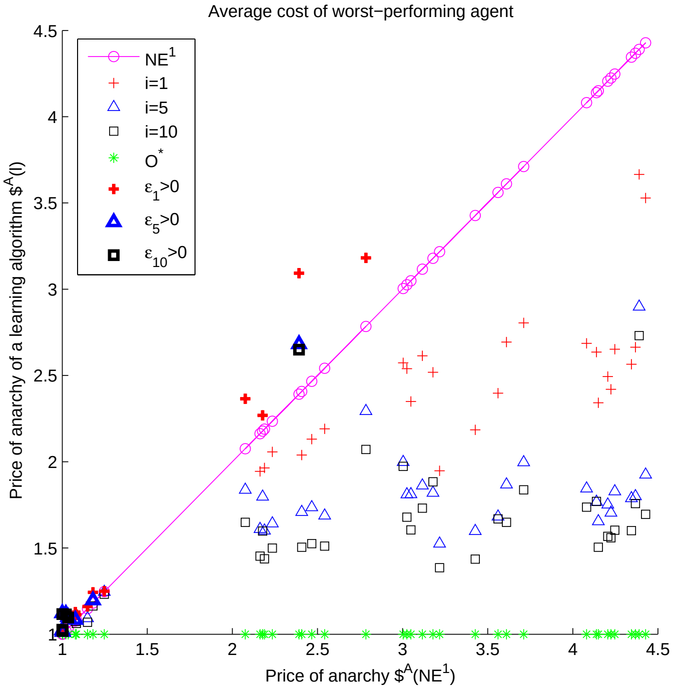


Figure 5.18: Dynamically changing population (1 new agent in every i^{th} round): IMPRES generally achieves the properties of social learning under moderately dynamic population. The problems in which IMPRES scored positive ϵ_i values are marked in boldface (ϵ_i denotes the positive offset from the minimax value for $i \in \{1, 5, 10\}$.

Equation 5.4.

When the two agents are in a monarchy configuration, a strategist computes a joint strategy that is a best response to an environment. Since there are no other agents, the strategist computes an optimal solution for both agents. The best-response algorithm for this case is exhaustive, and requires that the learner knows the payoff matrix of both players. The algorithm finds a strategy as follows: 1) find a set of socially-optimal strategy pairs in terms of the sum of both agents' costs; 2) evenly distribute probability mass among the optimal pairs in the set; and 3) the remaining pairs of joint strategies (that are not socially optimal) are assigned zero probability. The resulting strategy is thus not only socially optimal, but also fair to both agents.

Here the roles of the two learning layers are carefully separated. By choosing socially-optimal best-responses as opposed to individual best-responses, the expected cost of the strategist can increase. The purpose of the inner-learning algorithm is to discover alternative strategies (without fully speculating potential outcomes); and the vetting process for the newly found strategy is conducted in the meta-layer where the agent makes a higher level decision. Therefore, if the socially-optimal solution is in fact unfair to the extent that either agent performs worse than in an anarchy configuration, the correlated strategy cannot be sustained.

The following subsections discuss empirical results. The figures from this section display the results of a sample run of a repeated game in order to verify the fairness of the algorithms, a point that has been less emphasized in previous sections. Additionally, a more general analysis over 30 trials is also discussed.

5.6.2 Iterative prisoner's dilemma (IPD)

Let us revisit the prisoner's dilemma game; this time, the penalty matrix is re-scaled to the $[0, 1]$ range (shown in table below). This game possesses a dominant strategy equilibrium where both agents defect; that is, regardless of the other prisoner's choice, defecting is always a better option. In the $+m+n$ algorithm, the agents are given this dominant-strategy equilibrium strategy profile (`D, D`) and its average cost of 0.7.

row,column	Cooperate	Defect
Cooperate	0.4, 0.4	1.0, 0.0
Defect	0.0, 1.0	0.7, 0.7

Cost matrix

row,column	Cooperate	Defect
Cooperate	1.0	0.0
Defect	0.0	0.0

Optimal correlated strategy

Table 5.9: Prisoner's dilemma

The results are shown in Figure 5.19 where the left figures show how the row player progresses over time, and the figures on the right exhibit the counterparts for the column player. In brief, the -m algorithm converges to a dominant-strategy equilibrium in this example after several hundreds rounds; and the resulting average cost is 0.7. On the other hand, when both agents adopt the IMPRES algorithm, the agents learned to coordinate their actions to play optimally; and the $+m+n$ variation also performed optimally. The figures on the second row (for each algorithm) displays how the values of meta-strategies changed over time. For instance, after approximately 500 rounds, the meta-strategy of the row player adopting IMPRES converges to α, while the column player similarly converges to β. Thus, the resulting configuration was a monarchy.

5.6.3 Game of chicken

The game of chicken is a symmetric coordination game where an optimal outcome is achieved when each agent takes an alternative move; for instance, when one agent chooses to `go`, the best response of the other agent is to `stop`. Thus, both (`go, stop`) and (`stop, go`) are pure-strategy Nash equilibria. At the same time, both profiles are Pareto optimal, meaning that one cannot be better off without making the other worse off. In addition, this game has a mixed-strategy equilibrium where each player chooses to `go` with probability $\frac{2}{9}$. With respect to social welfare, the mixed-strategy is the worst Nash equilibrium in this case due to a positive probability (0.05) of collision. On the other hand, the pure-strategy NE are unfair since the payoff is better for the agent that chooses to `go`.

row,column	Stop	Go
Stop	0.2, 0.2	0.3, 0.0
Go	0.0, 0.3	1.0, 1.0

Cost matrix

row,column	Stop	Go
Stop	0.0	0.5
Go	0.5	0.0

Optimal correlated strategy

Table 5.10: A game of chicken

The performances of learning algorithms are compared for the game of chicken in Figure 5.20. The following results are consistent with all 30 trials. When agents are all independent (-m), their strategies always converged to one of the two unfair Pareto-optimal pure-strategy NE. The $+m$-n variation formed a monarchy around the fair-NE strategy that was given, while IMPRES agents self-organized into a monarchy, achieving optimal solutions. This result demonstrates that IMPRES agents are more inclined to a fair outcome when Pareto-optimal pure-strategy Nash equilibria are unfair.

5.6.4 Coordination game

The coordination game is *asymmetric* matrix game that reflects agents' preferences over the set of actions; for instance, given two actions of watching a movie or a baseball game, an agent may prefer watching a movie to a baseball game. Similarly with the game of chicken, the coordination game has three Nash equilibria: two pure-strategy equilibria and a mixed strategy equilibrium.

One of the pure-strategy Nash equilibrium strategy profiles is given to the $+m+n$ variation; thus, the NE option is not as fair as in the symmetric games case. Nevertheless, the outcome of $+m+n$ converges to the given NE strategy. Suppose that

row,column	Movie	Baseball
Movie	0.0, 0.5	1.0, 1.0
Baseball	1.0, 1.0	0.5, 0.0

Cost matrix

row,column	Stop	Go
Stop	0.0	0.5
Go	0.5	0.0

Optimal correlated strategy

Table 5.11: Coordination game

the given strategy profile is (movie, movie). Because there is only one strategy profile for the NE signal, whenever an agent chooses to follow the NE signal, `movie` is the realized action; hence `movie` is played more frequently during the early learning period than `baseball` (because other meta-strategies are more likely to explore both actions stochastically in the beginning). Given that, the agent that prefers `movie` will put more probability mass to following the NE signal (becoming more stationary). Subsequently, the other player gives in to play its part of the Nash strategy.

This result from the $+m+n$ variation illustrates that it can be more difficult to establish a mutually beneficial solution when a concrete but unfair norm exists, because the existing norm privileges some subset of agents with a headstart so that the norm can be stabilized more quickly.

In contrast, IMPRES agents established a fair and optimal outcome through a successful self-organization. The results were consistent in 30 trials.

5.6.5 Discussion

Learning rate : In the three well-known 2-player 2-action matrix games (symmetric and asymmetric), agents adopting IMPRES learned to play optimally within 500 rounds. As noted earlier, there exist other algorithms that learn mutually beneficial outcomes in 2-player games. In terms of learning rate, IMPRES is efficient when compared to other approaches: Table 5.12 compares approximate number of iterations until convergence for various algorithms.

Algorithm	Number of iterations
IMPRES	500
Sen et al. [54]	500
Stimpson et al. [58]	5,000
Crandall and Goodrich [14]	40,000

Table 5.12: Learning rate: approximate number of iterations until convergence in 2-player games

Fairness : Recall that both the chicken and the coordination games possess unfair but optimal NE[1]. Although decision making under the IMPRES algorithm is strictly self-interested, empirical results exhibit that IMPRES agents are inclined to fair solutions especially when persistent exploration is used.

The learning parameters used for the presented results are consistent with the main experiments (Table 5.3), thus employ persistent exploration. With decaying exploration, the algorithm more frequently results in a single-shot Nash equilibrium of game; Table 5.13 compares the performances of algorithm when different parameter settings are used.

Exploration, Game	IPD	Chicken	Coord
Setting 1 ($z = 0, T_l = 0.002$)	0	13	2
Setting 2 ($z = 0, T_l = 0.01$)	40	5	7
Setting 3 ($z = 0.01, T_l = 0.002$)	0	6	7
Setting 4 ($z = 0.01, T_l = 0.01$)	0	2	0

Table 5.13: The number of times the game resulted in a single-shot Nash equilibrium out of 100 trials.

5.7 Summary

Theoretically, IMPRES agents learn to behave as if they are in a Nash equilibrium of a repeated game. The notion of Nash equilibria of a repeated game, however, comprises a wide spectrum of solutions with respect to the quality of solution. In this chapter, a comprehensive set of empirical results was presented that the IMPRES agents learn close-to-optimal solutions; and their coordination cost is generally far less than in a centrally administered system.

Generally, IMPRES achieves the following properties of social learning:

I. in self-play, every agent adopting IMPRES keeps its average cost below its approximate minimax value, which verifies behavioral equivalency to an approximate Nash equilibrium;

II. with respect to social welfare standard, the use of social learning (in the meta-layer) always improves the performance; and

III. IMPRES agents successfully learn mutually beneficial strategies without an explicit threat and the performance is comparable with the version of algorithm that has an absolute criterion.

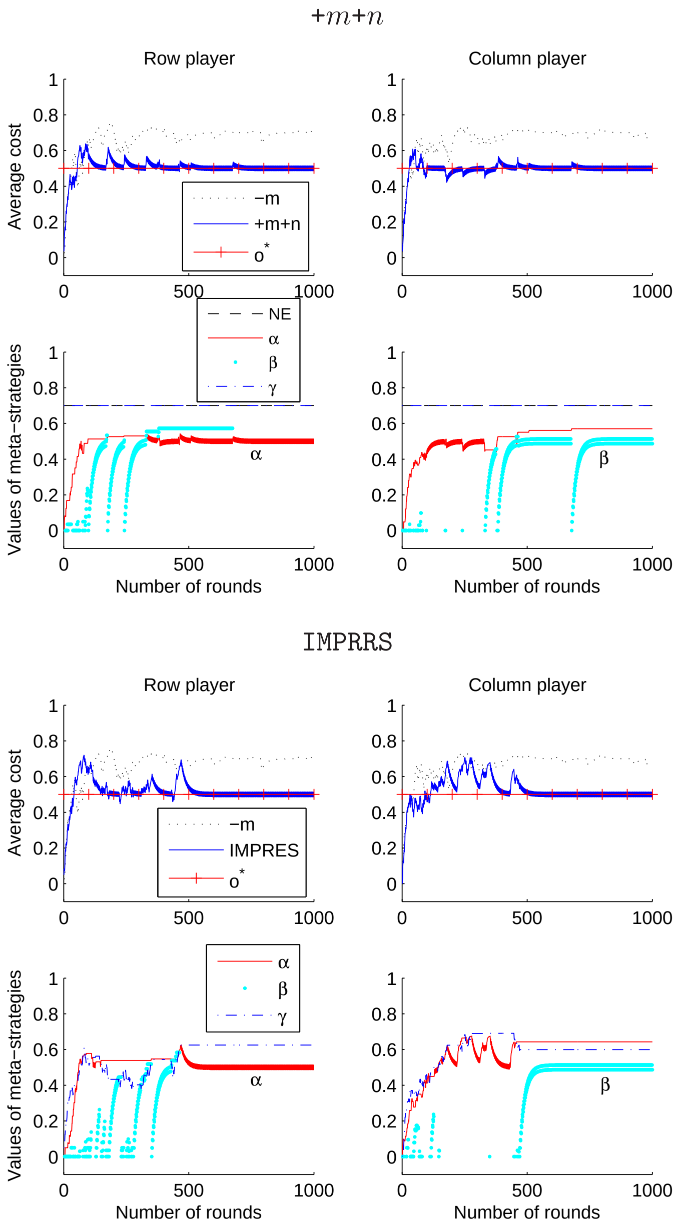
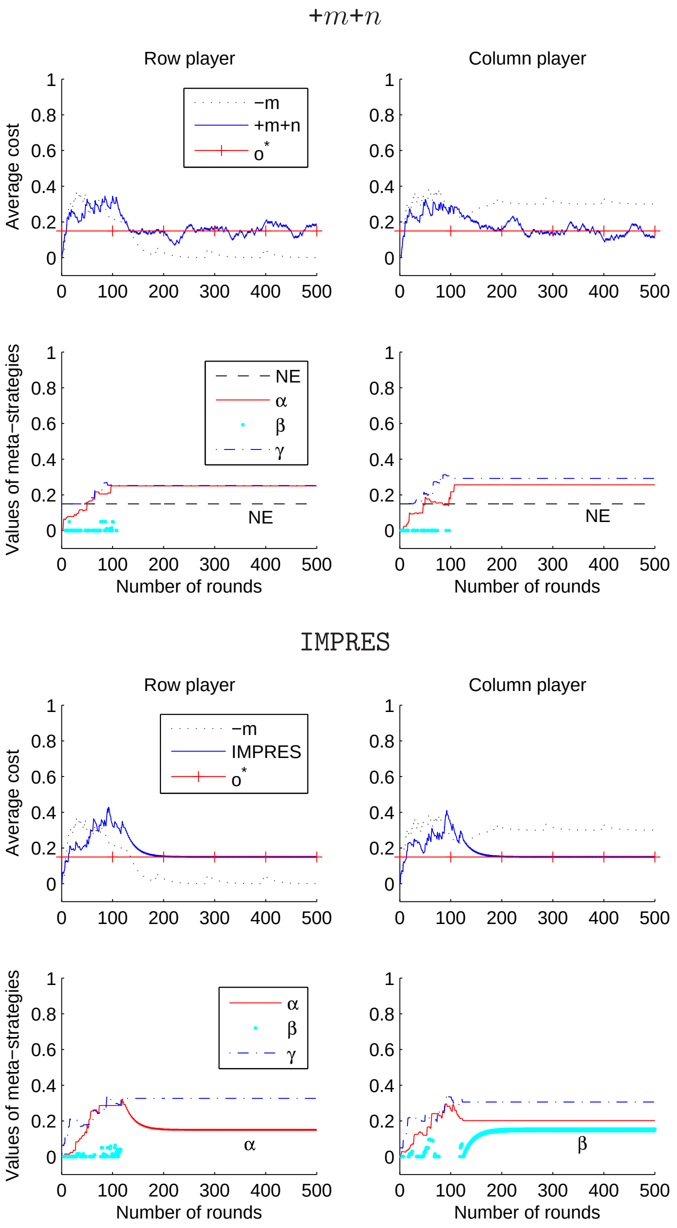


Figure 5.19: Iterative prisoner's dilemma



+*m*+*n*

IMPRES

Figure 5.20: Game of chicken

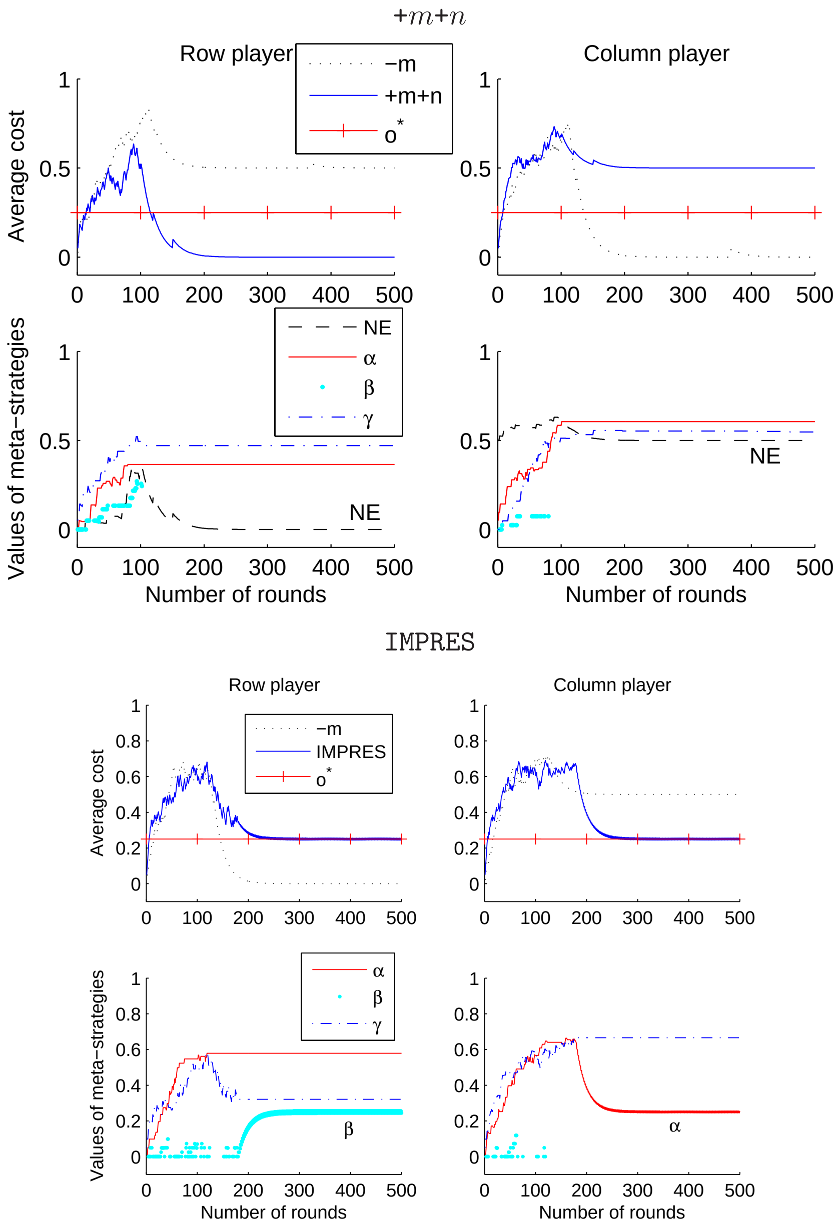


Figure 5.21: Coordination game

Furthermore, the empirical results showed that the IMPRES algorithm scales well with respect to problem size and population size; and exhibits robustness against modest population changes. A set of newly introduced learning parameters and their impacts on learning performance were also examined. Finally, the experiments on 2-player games stressed that IMPRES agents are also inclined to fair solutions.

Chapter 6

Conclusions

> “The effectiveness of the docility mechanism would be impaired if individuals could discriminate perfectly proper behaviors that were ‘for their own good’ from those that were altruistic. But people can discriminate only very imperfectly between beneficial and altruistic behaviors. Moreover, much of the value of docility to the individual is lost if great effort is expended evaluating each bit of social influence before accepting it. Acceptance without full evaluation is an integral part of the docility mechanism, and of the mechanisms of guilt and shame.” – Herbert A. Simon, 1997

6.1 Motivations

Studies of multiagent learning can generally be categorized as either descriptive or prescriptive theories [55]. The former applies to case where multiagent learning is used to model certain social or natural phenomena, or to predict likely outcomes of such phenomena; in this sense, it is important that the learning converges to solution concepts. The latter focuses on how a self-interested agent should act in a dynamic environment filled with other agents; thus, a performance objective such as minimizing expected cost becomes a more important criterion for evaluating a learning algorithm when the environment is dynamic.

The initial motivation for this research was prescriptive: to design a learning algorithm to perform well against various types of other agents. From a prescriptive framework, the case where the strategies of other agents are stationary is less intriguing, since the learning task differs little from a single-agent setting where classical reinforcement learning algorithms can learn optimal solutions. When other agents in an environment are simultaneously learning and thus changing strategies, however, many interesting scenarios can take place. For instance, if other agents are adaptive such that the learner can lead them to a strategy profile that is more advantageous to the learner, then the learner should exploit this option.

On the other hand, if the other agent attempts to lead the game, the learner should avoid being exploited, by moving the game's momentum so that the outcome is at least a draw. In order for this to happen, the learner may at times be required to change its objective at least temporarily; for instance, the learner may threaten other agents by maximizing their costs, instead of playing a best response with respect to its original objective of minimizing learner cost. This type of objective-shifting decision making can be represented as a non-stationary strategy that provides a set of contingent actions, making possible the learner's swift response. The learning of sophisticated non-stationary strategies constitutes one of the main contributions of this thesis.

The basic premise of an artificial agent's learning - whether in single-agent or in multi-agent setting - is that the agent is self-interested. Considering theories in natural and social sciences, this premise also applies to a biological agent's learning. As Dawkins [17] states, "anything that has evolved by natural selection should be selfish." Herbert Simon elaborates on this idea, arguing that altruistic (human) society is realized due to "enlightened selfishness" referred to as "docility" [56]. The notion of docility does not mean that an agent is simply adaptive and gives in; rather, its direct meaning is *teachability* such that a docile agent readily assumes social influence without full evaluation if on average in the long term the agent is better off by doing so.

Returning to the idea of the artificial agent's learning, the notion of multiagent social learning is based on a similar premise: agents in an environment may be neither stationary nor simply adaptive, but exhibit a docile nature; as such, some agents may be willing to try the strategies given by other agents if in the long run their performances improve by doing so.

In particular, when individually rational outcomes known as selfish equilibria are suboptimal due to negative externalities that each agent unintentionally introduces to the common welfare, mutually beneficial outcomes can only be achieved through explicit coordination. Hence, some agents must take altruistic acts. In this context, this thesis contributes to a descriptive theory by discovering an interesting structure of a social norm that can enforce mutually beneficial outcomes among self-interested constituents of a society.

6.2 Summary of thesis

This thesis investigates the notion of social learning in a multiagent learning context, and proposes a specific example of social learning algorithm known here as IMPRES (IMPlicit REciprocal Strategy learning). This idea is a break from earlier thinking in several ways. First, when many intelligent agents exist in an environment where individual agents have only myopic views, some agents naturally become more advantageous than others by incidentally being exposed to privileged experiences. Nonetheless, the majority of work in game theory is based on the general assumption that all agents are equally rational. The IMPRES algorithm in contrast fully exploits the asymmetry in agents' rationalities.

In multiagent learning literature, especially in prescriptive theories, other agents are addressed as "the opponents" or sources of uncertainty. My approach sheds new light on the existence of other agents, and advances the possibility that other agents may be additional sources of new knowledge that can improve learner performance. In certain problems, this may of course be a dangerous idea that may cause learner performance to subsequently degrade. The premise of social learning is that agents are sufficiently rational such that they can revert back to another decision if on average their performances do not improve. On the other hand, when mutually beneficial solutions are actually feasible, IMPRES agents should be able to establish such solutions as stable outcomes.

It is in fact generally believed that mutually beneficial outcomes can be stabilized if a tangible threat can be made clear to all constituents such that any deviator from the mutually beneficial strategy profile will be - without exception – severely punished. Generally, a thread of ideas similar to this is dubbed the folk theorem. I took a different view, stating that rationally bounded agents can stabilize a mutually beneficial outcome without an explicit notion of threat.

In fact, Kalai and Lehrer [32, 33] have made a similar suggestion, that a set of rational agents under perfect monitoring can learn to behave like they are in a Nash equilibrium; they characterized the solution concept as subjective equilibria. As discussed earlier, however, their rational learning algorithm requires a strong assumption that from the beginning the agents must have a pretty good idea about the strategies of other agents. Another striking difference is that their algorithm explicitly disregards the possibility of correlated strategies; in contrast, IMPRES specifically searches for a correlated strategy. That is, there exists a set of strictly correlated strategies that IMPRES can learn, but that cannot be supported by the independent strategy profiles learned by rational learning. Furthermore, under the imperfect monitoring assumption (used in this thesis) rational learning can fail to learn rational behaviors in some problems [61] including the El Farol bar problem discussed in Section 1.5.2.

The IMPRES algorithm expands the general idea that rational agents must be able to learn solutions beyond selfish equilibria that are rational only from a short-sighted view. Generally, such improved solutions (rather than selfish equilibria) can be characterized as Nash equilibria of a repeated game. Chapters 4 and 5 established the main claim of this thesis both theoretically and empirically, that IMPRES agents learn to find mutually beneficial solutions that are behavior-equivalent to Nash equilibria of a repeated game; and furthermore, the quality of solution, both with respect to social welfare and fairness standards, generally approach the optimal solutions.

With regard to the structure of learned strategies, perhaps the most relevant work is Littman and Stone's model, where the strategies are modeled as a pair of automata in 2-player games [36]. As noted, work to this aim has to date been generally restricted to 2-player games. On the contrary, IMPRES is designed to scale well with respect to the number of agents; and the scalability is also supported by a set of empirical results in Sections 5.5.4 – 5.5.3. The structure of the IMPRES meta-strategy resembles that of classical reinforcement learning; yet the IMPRES model is a significant extension to existing models. The major difference is that a set of available actions change dynamically in the IMPRES model.

Let us reconsider the IMPRES model as an extension to classical reinforcement learning using a familiar example. Consider the multi-arm bandit problem where the learner is trying to maximize the expected payoff. Suppose that the learner owns one arm (so that it is always available to the learner but the reward from the arm is determined by the environment) and some large set of arms that comes and goes. If the learner pulls certain arms more frequently, then the probability of the arm being available increases. If the learner continues to search the entire space, it may be able to find the optimal solution. Suppose that the learner finds an arm that performs significantly better than the arm that is always available. Given that the search space is infinitely large, and that the learner is uncertain about the probability of finding a better arm, pulling the arm more frequently to ensure that it remains is a reasonable strategy for the learner.

In such a case, the learner's choice of selecting the best option among what is available may not be objectively rational under the strict notion of rationality discussed in Section 4.3.2; that is, an agent is objectively rational only when it is choosing the best among a complete set of available options. Suppose now that there are other agents in the environment of the multi-arm bandit problem, such that the changing set of arms has been introduced to the environment by some other agents. Let an arm's probability of remaining be some monotonically increasing function of the average number of agents that pull the arm. When the other agents reason in a manner similar to the learner's and choose better arms than the one available to each individual agent, a set of good arms that are desirable by the majority of constituents can be sustained so long as a sufficient number of agents continue to pull those arms. The IMPRES algorithm model is designed to solve this sort of problem where the learner's current strategy (together with those of other agents) influences the set of actions that will be available in the future.

Lastly, this thesis contributes to bridging the gap between theories of decision making from social sciences to artificial intelligence. The notion of social learning is prevalent in human learning; and it plays a major role in reinforcing social norms. This thesis initiates an effort to establish the general idea of "learning from others" in the multiagent learning context. Specifically, this thesis investigates IMPRES - an example of social learning algorithm – in depth, and proves that the use of social learning enables artificial agents to accomplish desirable solutions. In return, this mathematical model of social learning can be used to model interesting (human) social behaviors. This leads me to future work.

6.3 Limitations

The IMPRES algorithm aims at middle-ground solutions; that is, the outcome of IMPRES may be more desirable than selfish solutions in terms of average payoffs (costs), and its coordination overhead may be smaller than that of a centrally administered approach. As discussed in Chapter 4, however, the target solution space does not exclude selfish solutions (an anarchy configuration) nor centrally administered solutions (a monarchy configuration). Although the empirical study demonstrated promising results, IMPRES can result in either an anarchy or a monarchy configurations in the worst case.

The algorithm is also limited to symmetric games (or games where every agent knows the cost functions of other agents as in the coordination game in Section 5.6.4). In addition, computing optimal joint strategy in general can be a computationally challenging task. In symmetric network congestion games, the algorithm for computing joint strategy is polynomial with respect to the number of agents that participate in the joint strategy. In problem domains where joint-strategy computation is expensive, computational complexity may also be incorporated into coordination overhead.

6.4 Future work

Short-term goals concentrate on further analysis of implicit reciprocal strategy learning. My long-term goal is to advance the idea of "learning from others" in a broader context.

6.4.1 Free riders phenomenon

Although the optimal solution may only be achieved when every agent follows a centralized signal, e.g., the traffic light model, it is generally the case that a close to optimal solution can be achieved when some subset of agents take altruistic acts. That being said, there can be free riders that receive the same benefits as those altruistic ones in spite of not engaging in their fair share of altruistic acts. An interesting observation from empirical studies on the IMPRES model is that the number of free riders varies for different problems. Understanding the free rider phenomenon is important to explaining real-life examples.

6.4.2 Tradeoffs of various criteria

Decision making at a meta-strategy level requires that the agents will reason about multiple criteria. For example, an agent may trade off a quality gain for a lower communication cost. On the other hand, when an agent is performing multiple tasks, the agent may distribute its computational cycles according to its task priority; in this case, the agent may elect to follow the strategies of other agents for low priority tasks because it may consume less computational power than learning the strategy for itself. The IMPRES model currently supports a tradeoff between a quality gain and a communication cost, and the use of a new criterion for learning costs is proposed.

Weighted-mixture of multiple criteria

The priorities of various criteria depend on problem domains, and one can consider a weighted-evaluation function that combines multiple criteria into a single measure. For instance, the total price $\$_{total}$ of a solution can be defined to combine quality loss (φ) and coordination overhead (ς) as follows:

$$\$_{total} = \frac{\lambda\varsigma_l + (1-\lambda)\varphi_l}{\lambda\varsigma_A + (1-\lambda)\varphi_{o^*}},$$

where λ is a weight parameter representing the relative priority of quality objective (e.g. average cost) over coordination overhead.

Learning budget

Given that subscriber-agents do not have to learn how to choose actual actions, a good portion of IMPRES agents save their computational cycles by substituting complex learning problems with simpler ones; therefore, the overall learning utility of the multiagent system is enhanced. From this observation, one can consider a new evaluation criterion for those agents that are performing multiple tasks simultaneously.

Suppose an agent is assigned to perform two independent tasks: A and B, thus the computing resources of the agent are shared by task A and task B. Suppose the agent decides to use learning to improve its performance on task A. Since learning also requires time and computing resources the performance of the other task B may degrade by some Δ. The agent has to trade off the performance gain from task A and the relative loss from task B in order to determine how much learning it can afford on task A.

In this context, the cost of a learning algorithm can be defined as a function of quality loss on the other tasks. Let $tasks(i, t)$ denote a set of tasks assigned to agent i at time t. Let f be a flag vector of a length $|tasks(i, t)|$ such that each element of f_j holds a flag $+L(-L)$ if the learning mode of a corresponding task j is on (or off) such that a task utilizes learning only when the flag is $+L$. Let $jps(j, f)$ denote the job processing speed of task j. Let us define the quality loss of task j' due to the learning employed by task j, denoted by $\Delta_{j',j}$, as the difference in the job processing speed of task j' with the learning mode of task j off and on as follows:

$$\Delta_{j',j} = jps(j', \begin{pmatrix} \ldots \\ j : -L \\ \ldots \end{pmatrix}) - jps(j', \begin{pmatrix} \ldots \\ j : +L \\ \ldots \end{pmatrix})$$

The cost of a learning algorithm l employed by task j at time t, denoted by $cost(l, j, t)$, is formally defined as:

$$cost(l, j, t) = \sum_{j' \in tasks(i,t), j' \neq j} \Delta_{j',j}$$

Given a learning algorithm, let *learning budget* denote a criterion for setting the maximum budget on computing resources that the algorithm can use. In general, the asymptotic bounds of a learning algorithm are defined from an algorithm's perspective; that is, an agent is expected to have a certain degree of computing resources to run complex algorithms. On the other hand, learning budget is defined from an agent's perspective of how much computational resource it can afford for each task.

The learning of high-level decision makings, such as learning budget, can be another generalization of this thesis; this will be of interest in the design of complex intelligent agents.

6.4.3 General social learning models

The IMPRES model provides an example of social learning where agents learn to act more rationally by using the strategies *given* by others. Other social learning models worth exploring include: simply copying the actions of others and learning from the mistakes of others.

6.4.4 Multidisciplinary research

Lastly, I also anticipate opportunities for multidisciplinary research in relevant fields including social science, cognitive science, public policy, transportation science, and complex adaptive systems.

For example, consider an intelligent traffic system where each automobile is equipped with an intelligent agent assisting a human driver. Each agent can collect realtime traffic information about a certain area that it has driven by, that might be useful to other agents that are heading towards the area. Suppose that some agent provides its traffic knowledge to another agent. In this scenario, the information recipient will be benefited, e.g. by avoiding congested areas. On the other hand, the information provider does not gain anything; rather proving information to others may hurt its performance by sparing the communication bandwidth that could have been used for receiving useful traffic information for itself. In a selfish solution, therefore, there will be no information providers. In this context, the notion of social learning can be used so that the agents learn to share information to accomplish mutually beneficial outcomes.

6.5 Contributions

The main contributions of this thesis can be summarized as follows:

I. Initiated an effort for applying the notion of social learning to multiagent learning.

II. Developed implicit reciprocal strategy learning (IMPRES) – the first algorithm that learns the non-stationary structure of reciprocal strategies for more than 2 players.

III. Proved formally that the outcome of IMPRES is behavior-equivalent to a Nash equilibrium of a repeated game.

IV. Evaluated the IMPRES algorithm empirically through a set of controlled experiments in symmetric network congestion games with nondecreasing cost functions, and a set of experiments on 2-player matrix games.

   (i) Presented empirical evidence to support the theoretical result that IMPRES agents learn to act as if they are in a Nash equilibrium of a repeated game.

   (ii) Demonstrated empirically that the outcome of IMPRES in self-play is close to optimal in terms of system-wide solution quality (e.g., average performance of all agents).

   (iii) Evaluated that the algorithm scales well with respect to the number of agents and to network size.

   (iv) Evaluated the algorithm's robustness relative to gradual changes in agent population.

   (v) Analyzed the impacts of newly introduced social learning parameters.

   (vi) The results on 2-player games stressed that IMPRES agents are inclined to fair outcomes.

V. Proposed the use of a set of general criteria for evaluating the system-wide performance of multiagent learning algorithms in self-play; and introduced two types of plots that can succinctly present the performance tradeoffs:

   (i) The price of anarchy plot allows a visual comparison of the system-wide quality of solutions found by various algorithms. Depending on how the objective function value is defined, this plot can be used to visualize social welfare or fairness of a learning algorithm.

   (ii) The social learning price curve visualizes the tradeoffs between the quality loss (due to selfish decision making) and the communication cost (due to coordination effort).

VI. Proposed a set of desired properties of social learning. Whereas the first two properties are required, the last property is preferable.

   (i) the minimax-safety property: every agent must be better off than its minimax value.

(ii) the collusion-safety property: the social welfare must be improved from independent strategy profiles.

(iii) the comparability property: the algorithm with an implicit criterion should be comparable with one using an absolute standard.

6.6 Conclusion

When individual constituents of a general population possess a high-level goal of minimizing their long-term average costs, rationally bounded agents that are willing to take chances on the exploration of potentially better solutions can learn to take reasonably rational actions in repeated games. The notion of reasonable rationality refers to the fact that the agents make (at least apparently) altruistic acts, due to their enlightened selfishness that the acts will pay off later in time, as long as such subjective beliefs are realized in their long-term average payoffs. In the natural sciences, this property is known as docility and appears as an advantage in species survival.

In this thesis, I present a computational model of reasonably rational learning; and propose the IMPRES algorithm – the first algorithm that learns a reciprocal strategy profile in self-play that is behavior-equivalent to a Nash equilibrium of a repeated game. Further, I demonstrate both theoretically and empirically that the IMPRES algorithm accomplishes the basic premise of social learning that every agent achieves its minimax value; and the social welfare of IMPRES agents is generally close to optimal.

Appendix A

CMRadar's learning of bumping probability

This section describes how CMRadar room-finder updates local room owners' bumping probability. Let n denote the number of bumping requests sent to a room owner, and let y denote the number of times when the room owner accepts the request. Based on an assumption that a room owner makes decision following a Bernoulli distribution, i.e., the owner accepts with a probability θ, and rejects with the probability $1 - \theta$, CMRadar uses a Bayesian learning method to estimate the expected value of bumping probability θ as follows.

The likelihood of seeing y acceptance over n trials given a room owner's bumping probability θ is

$$p(y|\theta, n) = \left(\begin{array}{c} n \\ y \end{array} \right) \theta^y (1-\theta)^{n-y}. \tag{A.1}$$

By applying Bayes rule, the probability of bumping being θ given y acceptance over n trials is

$$p(\theta|y, n) = \frac{p(y|\theta, n)p(\theta|n)}{p(y|n)} \tag{A.2}$$

where $p(\theta|n)$ denotes a prior. By taking a uniform prior,

$$= \frac{p(y|\theta, n)}{p(y|n)}$$

Bayes proved that the normalization factor, i.e., the denominator in Equation (A.2), is a function of n.

$$p(y|n) = \int_0^1 p(y|\theta, n)p(\theta|n)\, d\theta = \frac{1}{n+1} \tag{A.3}$$

By substituting the normalization factor and the likelihood in Equation (A.2) with Equation (A.3) and (A.1), respectively, the probability of bumping can be rewritten as

$$p(\theta|y, n) = (n+1)\left(\begin{array}{c} n \\ y \end{array}\right)\theta^y(1-\theta)^{n-y}.$$

Then, the expected value of θ given y and n is

$$\begin{aligned} E(\theta|y, n) &= \int_0^1 \theta p(\theta|y, n)\, d\theta \\ &= (n+1)\int_0^1 \theta \left(\begin{array}{c} n \\ y \end{array}\right)\theta^y(1-\theta)^{n-y} \\ &= (n+1)\frac{y+1}{n+1}\underline{\int_0^1 \left(\begin{array}{c} n+1 \\ y+1 \end{array}\right)\theta^{y+1}(1-\theta)^{(n+1)-(y+1)}}. \end{aligned}$$

Again, by using Equation (A.3), the underlined part can be simplified as

$$\begin{aligned} &\int_0^1 \left(\begin{array}{c} n+1 \\ y+1 \end{array}\right)\theta^{y+1}(1-\theta)^{(n+1)-(y+1)} \\ &= \int_0^1 p(y+1|\theta, n+1)p(\theta|n+1)\, d\theta \\ &= p(y+1|n+1) \\ &= \frac{1}{(n+1)+1}. \end{aligned}$$

Therefore, the expected value of probability of bumping given y acceptance over n trials is

$$E(\theta|y, n) = \frac{y+1}{n+2}.$$

Appendix B

Complete results

Problem		Price of anarchy (monarchy)			
ID	$\lvert S\rvert$	PNE	-m	+m+n	IMPRES
linear-0	9	1.30	1.30	1.10 (0.22)	1.09 (0.27)
linear-1	2	1.32	1.32	1.07 (0.33)	1.07 (0.32)
linear-2	2	1.31	1.30	1.06 (0.33)	1.06 (0.32)
linear-3	2	1.31	1.30	1.07 (0.31)	1.07 (0.34)
linear-4	2	1.31	1.30	1.07 (0.31)	1.06 (0.32)
linear-5	4	1.08	1.09	1.03 (0.54)	1.03 (0.56)
linear-6	2	1.12	1.13	1.02 (0.39)	1.02 (0.50)
linear-7	8	1.00	1.01	1.01 (0.50)	1.01 (0.57)
linear-8	4	1.01	1.13	1.00 (0.53)	1.00 (0.54)
linear-9	13	1.01	1.65	1.03 (0.60)	1.03 (0.59)

Table B.1: Results on symmetric network congestion games with linear cost functions

Problem		Price of anarchy (monarchy)			
ID	$\lvert S\rvert$	PNE	-m	+m+n	IMPRES
poly-0	5	2.82	2.88	1.22 (0.10)	1.31 (0.10)
poly-1	5	3.10	2.86	1.49 (0.08)	1.48 (0.08)
poly-2	5	3.24	2.92	1.26 (0.09)	1.30 (0.10)
poly-3	3	2.88	2.72	1.18 (0.13)	1.29 (0.14)
poly-4	3	3.02	2.81	1.26 (0.10)	1.31 (0.13)
poly-5	6	1.62	1.74	1.35 (0.37)	1.36 (0.36)
poly-6	3	2.25	2.20	1.08 (0.18)	1.18 (0.14)
poly-7	4	2.23	2.31	1.12 (0.13)	1.21 (0.11)
poly-8	3	2.00	2.05	1.11 (0.18)	1.21 (0.15)
poly-9	4	2.30	2.09	1.34 (0.07)	1.28 (0.08)
poly-10	4	1.02	1.09	1.02 (0.20)	1.03 (0.18)
poly-11	5	1.06	1.13	1.02 (0.26)	1.02 (0.37)
poly-12	7	1.20	1.20	1.04 (0.41)	1.05 (0.47)
poly-13	3	1.06	1.24	1.02 (0.52)	1.02 (0.55)
poly-14	4	1.02	1.84	1.03 (0.24)	1.03 (0.19)

Table B.2: Results on symmetric network congestion games with polynomial cost functions

Problem		Price of anarchy (monarchy)			
ID	$\lvert S \rvert$	PNE	-m	+m+n	IMPRES
exp-0	2	2.07	2.03	1.09 (0.18)	1.10 (0.19)
exp-1	4	2.06	2.00	1.09 (0.18)	1.10 (0.19)
exp-2	2	2.01	2.06	1.10 (0.18)	1.10 (0.20)
exp-3	2	1.96	2.00	1.09 (0.18)	1.10 (0.20)
exp-4	2	1.96	2.00	1.10 (0.19)	1.10 (0.19)
exp-5†	2	1.01	9.22	1.17 (0.54)	1.16 (0.51)
exp-6†	6	1.02	4.23	1.10 (0.59)	1.10 (0.54)
exp-7†	7	1.08	4.68	1.09 (0.53)	1.08 (0.55)
exp-8†	2	1.10	4.89	1.11 (0.50)	1.10 (0.48)
exp-9†	3	1.01	1.73	1.01 (0.52)	1.01 (0.49)
exp-10	6	3.16	2.66	1.12 (0.06)	1.13 (0.06)
exp-11	6	3.56	3.00	1.65 (0.04)	1.60 (0.03)
exp-12	5	3.62	3.01	1.34 (0.05)	1.37 (0.04)
exp-13	4	3.98	3.59	1.26 (0.06)	1.33 (0.07)
exp-14	7	3.13	2.86	1.55 (0.04)	1.54 (0.03)

Table B.3: Results on symmetric network congestion games with exponential cost functions

Problem		Price of anarchy (monarchy)			
ID	$\|S\|$	PNE	-m	+m+n	IMPRES
h1p3-0	2	1.06	2.12	1.03 (0.48)	1.03 (0.52)
h1p3-1	2	1.16	1.16	1.02 (0.42)	1.03 (0.38)
h1p3-2	3	1.20	1.20	1.04 (0.34)	1.03 (0.52)
h1p3-3	4	1.25	1.25	1.04 (0.39)	1.05 (0.44)
h1p3-4	2	1.18	1.17	1.06 (0.33)	1.06 (0.41)
h1p3-5	2	1.04	1.58	1.02 (0.59)	1.01 (0.63)
h1p3-6	4	1.11	1.10	1.02 (0.51)	1.02 (0.51)
h1p3-7	2	1.25	1.68	1.04 (0.28)	1.05 (0.34)
h1p3-8	4	1.25	1.27	1.04 (0.30)	1.10 (0.50)
h1p3-9	4	1.04	1.04	1.02 (0.63)	1.03 (0.63)
h2p3-0	4	1.51	1.55	1.10 (0.33)	1.11 (0.30)
h2p3-1	2	1.53	1.51	1.17 (0.40)	1.13 (0.32)
h2p3-2	3	1.88	5.33	1.08 (0.20)	1.17 (0.22)
h2p3-3	4	1.91	6.79	1.37 (0.16)	1.24 (0.20)
h2p3-4	2	1.72	1.80	1.07 (0.25)	1.20 (0.23)
h2p3-5	5	2.15	7.83	1.09 (0.18)	1.30 (0.20)
h2p3-6	4	1.51	1.48	1.10 (0.28)	1.15 (0.33)
h2p3-7	2	1.74	3.06	1.09 (0.30)	1.15 (0.28)
h2p3-8	2	1.90	1.98	1.14 (0.19)	1.22 (0.17)
h2p3-9	5	1.91	1.82	1.09 (0.15)	1.18 (0.22)
h3p3-0	3	3.05	3.16	1.32 (0.11)	1.34 (0.10)
h3p3-1	2	2.87	2.83	1.21 (0.14)	1.34 (0.14)
h3p3-2	3	2.87	2.73	1.24 (0.13)	1.35 (0.12)
h3p3-3	2	2.53	2.47	1.26 (0.14)	1.31 (0.12)
h3p3-4	2	2.92	2.88	1.33 (0.11)	1.30 (0.13)
h3p3-5	3	2.52	2.57	1.12 (0.21)	1.24 (0.17)
h3p3-6	5	3.00	3.07	1.20 (0.13)	1.32 (0.12)
h3p3-7	2	2.48	2.55	1.32 (0.12)	1.32 (0.12)
h3p3-8	3	2.51	2.59	1.30 (0.12)	1.32 (0.14)
h3p3-9	2	3.09	2.93	1.18 (0.15)	1.31 (0.13)

Table B.4: Complete results on problems with $|S| \simeq 3$

Problem		Price of anarchy (monarchy)			
ID	$\|S\|$	PNE	-m	+m+n	IMPRES
h1p6-0	6	1.09	4.92	1.08 (0.29)	1.06 (0.41)
h1p6-1	6	1.38	2.91	1.00 (0.98)	1.02 (0.96)
h1p6-2	6	1.01	1.38	1.02 (0.71)	1.02 (0.60)
h1p6-3	4	1.15	1.40	1.08 (0.21)	1.08 (0.24)
h1p6-4	7	1.10	1.16	1.02 (0.47)	1.03 (0.48)
h1p6-5	4	1.19	2.79	1.13 (0.41)	1.13 (0.39)
h1p6-6	5	1.08	1.18	1.03 (0.46)	1.03 (0.42)
h1p6-7	5	1.00	1.00	1.00 (0.66)	1.00 (0.70)
h1p6-8	5	1.16	1.45	1.06 (0.56)	1.08 (0.56)
h1p6-9	5	1.06	2.66	1.04 (0.59)	1.07 (0.55)
h2p6-0	4	2.11	2.16	1.07 (0.26)	1.22 (0.20)
h2p6-1	4	1.76	1.72	1.07 (0.33)	1.14 (0.40)
h2p6-2	4	1.82	1.87	1.19 (0.33)	1.25 (0.31)
h2p6-3	4	1.54	1.53	1.16 (0.44)	1.22 (0.43)
h2p6-4	4	1.77	1.80	1.13 (0.24)	1.18 (0.20)
h2p6-5	4	1.54	1.65	1.08 (0.32)	1.15 (0.40)
h2p6-6	4	1.87	1.93	1.12 (0.35)	1.19 (0.42)
h2p6-7	4	1.82	1.75	1.14 (0.26)	1.23 (0.26)
h2p6-8	6	1.65	1.70	1.14 (0.28)	1.18 (0.37)
h2p6-9	4	2.39	2.46	1.08 (0.23)	1.27 (0.16)
h3p6-0	4	2.51	2.59	1.14 (0.17)	1.34 (0.14)
h3p6-1	4	2.63	2.49	1.17 (0.21)	1.28 (0.19)
h3p6-2	4	2.67	2.53	1.23 (0.13)	1.32 (0.16)
h3p6-3	5	2.60	2.48	1.18 (0.18)	1.29 (0.15)
h3p6-4	4	2.63	2.75	1.17 (0.18)	1.29 (0.19)
h3p6-5	4	2.57	2.44	1.18 (0.15)	1.28 (0.15)
h3p6-6	5	3.12	2.94	1.31 (0.12)	1.43 (0.10)
h3p6-7	8	3.04	6.38	1.37 (0.10)	1.45 (0.13)
h3p6-8	4	3.51	3.27	1.38 (0.10)	1.38 (0.10)
h3p6-9	4	3.05	3.15	1.27 (0.12)	1.38 (0.10)

Table B.5: Complete results on problems with $|S| \simeq 6$

Problem		Price of anarchy (monarchy)			
ID	$\|S\|$	PNE	-m	+m+n	IMPRES
h1p10-0	10	1.05	2.08	1.13 (0.56)	1.07 (0.59)
h1p10-1	11	1.12	3.08	1.05 (0.29)	1.05 (0.21)
h1p10-2	10	1.08	1.44	1.02 (0.40)	1.02 (0.39)
h1p10-3	11	1.13	1.27	1.07 (0.42)	1.06 (0.38)
h1p10-4	10	1.32	2.26	1.19 (0.21)	1.12 (0.34)
h1p10-5	11	1.02	2.79	1.04 (0.62)	1.04 (0.50)
h1p10-6	9	1.08	2.53	1.08 (0.54)	1.09 (0.49)
h1p10-7	12	1.13	6.78	1.15 (0.28)	1.34 (0.38)
h1p10-8	8	1.11	1.73	1.10 (0.66)	1.08 (0.64)
h1p10-9	8	1.03	2.34	1.02 (0.67)	1.03 (0.59)
h2p10-0	9	1.60	1.56	1.11 (0.32)	1.14 (0.28)
h2p10-1	12	1.66	1.70	1.12 (0.36)	1.19 (0.42)
h2p10-2	12	2.02	2.34	1.39 (0.18)	1.65 (0.14)
h2p10-3	11	1.64	1.63	1.09 (0.33)	1.17 (0.31)
h2p10-4	12	1.64	5.10	1.12 (0.27)	1.18 (0.37)
h2p10-5	11	1.52	14.92	1.27 (0.33)	1.31 (0.30)
h2p10-6	11	1.61	2.09	1.27 (0.23)	1.24 (0.26)
h2p10-7	9	1.81	1.86	1.10 (0.35)	1.18 (0.31)
h2p10-8	9	1.59	1.91	1.09 (0.30)	1.21 (0.30)
h2p10-9	12	1.75	1.82	1.24 (0.12)	1.23 (0.15)
h3p10-0	10	2.61	4.72	1.40 (0.11)	1.36 (0.11)
h3p10-1	10	3.02	2.90	1.20 (0.12)	1.28 (0.14)
h3p10-2	9	2.73	2.63	1.22 (0.13)	1.34 (0.10)
h3p10-3	8	3.23	2.98	1.20 (0.13)	1.38 (0.11)
h3p10-4	10	3.05	2.95	1.00 (0.98)	1.01 (0.94)
h3p10-5	12	3.30	3.06	1.40 (0.10)	1.40 (0.09)
h3p10-6	8	2.79	2.57	1.59 (0.07)	1.44 (0.05)
h3p10-7	9	3.51	3.27	1.79 (0.06)	1.50 (0.06)
h3p10-8	8	3.30	3.08	1.47 (0.09)	1.45 (0.07)
h3p10-9	12	2.50	3.39	1.17 (0.71)	1.62 (0.50)

Table B.6: Complete results on problems with $|S| \simeq 10$

Problem		Price of anarchy (monarchy)			
ID	$\|S\|$	PNE	-m	+m+n	IMPRES
h1p15-0	17	1.01	3.10	1.03 (0.51)	1.05 (0.45)
h1p15-1	13	1.16	2.14	1.05 (0.59)	1.07 (0.49)
h1p15-2	13	1.16	2.91	1.07 (0.51)	1.08 (0.47)
h2p15-0	14	2.12	4.99	1.01 (0.97)	1.01 (0.96)
h2p15-1	17	2.24	2.10	1.33 (0.07)	1.31 (0.12)
h2p15-2	16	2.18	4.87	1.02 (0.97)	1.03 (0.97)
h3p15-0	15	2.96	2.76	1.34 (0.07)	1.36 (0.13)
h3p15-1	16	2.87	5.48	1.38 (0.09)	1.31 (0.21)
h3p15-2	13	2.83	2.86	1.19 (0.09)	1.31 (0.17)
h1p15-0	17	1.01	3.10	1.03 (0.51)	1.05 (0.45)
h1p15-1	13	1.16	2.14	1.05 (0.59)	1.07 (0.49)
h1p15-2	13	1.16	2.91	1.07 (0.51)	1.08 (0.47)
h2p15-0	14	2.12	4.99	1.01 (0.97)	1.01 (0.96)
h2p15-1	17	2.24	2.10	1.33 (0.07)	1.31 (0.12)
h2p15-2	16	2.18	4.87	1.02 (0.97)	1.03 (0.97)
h3p15-0	15	2.96	2.76	1.34 (0.07)	1.36 (0.13)
h3p15-1	16	2.87	5.48	1.38 (0.09)	1.31 (0.21)
h3p15-2	13	2.83	2.86	1.19 (0.09)	1.31 (0.17)

Table B.7: Complete results on problems with $|S| \simeq 15$

Problem		Price of anarchy (monarchy)			
ID	Population	PNE	-m	+m+n	IMPRES
pop-0	100	2.71	2.69	1.24 (0.08)	1.32 (0.07)
	500	2.81	2.79	2.05 (0.01)	1.97 (0.01)
	1000	2.81	2.81	2.35 (0.01)	2.28 (0.01)
pop-1	100	2.04	2.13	1.68 (0.12)	1.67 (0.02)
	500	2.12	2.15	2.06 (0.94)	1.97 (0.01)
	1000	2.13	2.16	2.10 (0.97)	2.07 (0.00)
pop-2	100	2.23	2.12	1.27 (0.06)	1.30 (0.05)
	500	2.23	2.12	1.85 (0.01)	1.79 (0.01)
	1000	2.23	2.12	1.98 (0.01)	1.99 (0.01)
pop-3	100	1.00	1.01	1.00 (0.43)	1.01 (0.03)
	500	1.00	1.01	1.00 (0.95)	1.01 (0.01)
	1000	1.00	1.01	1.00 (0.98)	1.01 (0.00)
pop-4	100	1.10	1.11	1.05 (0.08)	1.04 (0.08)
	500	1.10	1.11	1.08 (0.21)	1.09 (0.01)
	1000	1.10	1.11	1.09 (0.32)	1.10 (0.00)
pop-5	100	1.47	1.51	1.05 (0.14)	1.08 (0.13)
	500	1.53	1.55	1.27 (0.02)	1.25 (0.02)
	1000	1.53	1.72	1.37 (0.01)	1.36 (0.01)

Table B.8: On growing population size

Problem[1]		Price of anarchy (monarchy)			
ID	$\|S\|$	PNE	-m	+m+n	IMPRES
mix40-0	6	1.18	1.20	1.14 (0.12)	1.14 (0.12)
mix40-1	3	1.00	1.01	1.00 (0.12)	1.00 (0.13)
mix40-2	4	1.00	1.02	1.02 (0.12)	1.02 (0.12)
mix40-3	4	1.00	1.00	1.00 (0.13)	1.00 (0.13)
mix40-4	6	1.25	1.23	1.20 (0.12)	1.20 (0.13)
mix40-5	7	1.02	1.04	1.03 (0.12)	1.04 (0.12)
mix40-6	5	1.04	1.21	1.03 (0.12)	1.03 (0.12)
mix40-7	3	1.08	1.64	1.05 (0.12)	1.05 (0.12)
mix40-8	5	1.08	1.16	1.05 (0.13)	1.05 (0.13)
mix40-9	3	1.15	1.20	1.05 (0.13)	1.05 (0.12)
mix40-10	6	2.19	2.16	1.31 (0.11)	1.30 (0.11)
mix40-11	3	2.47	2.33	1.34 (0.10)	1.33 (0.10)
mix40-12	3	2.54	2.49	1.33 (0.10)	1.28 (0.10)
mix40-13	3	2.41	2.32	1.38 (0.11)	1.39 (0.11)
mix40-14	7	2.08	2.23	1.32 (0.08)	1.34 (0.08)
mix40-15	3	2.23	2.25	1.39 (0.12)	1.37 (0.12)
mix40-16	6	2.78	3.00	1.47 (0.05)	1.49 (0.05)
mix40-17	3	2.16	2.07	1.30 (0.11)	1.29 (0.10)
mix40-18	5	2.39	2.49	1.50 (0.05)	1.51 (0.05)
mix40-19	5	2.18	2.29	1.35 (0.10)	1.34 (0.10)
mix40-20	3	3.05	2.90	1.41 (0.11)	1.40 (0.11)
mix40-21	3	3.56	2.92	1.17 (0.06)	1.29 (0.06)
mix40-22	4	3.22	2.61	1.19 (0.10)	1.17 (0.09)
mix40-23	3	3.12	2.90	1.24 (0.05)	1.29 (0.06)
mix40-24	3	3.18	2.72	1.26 (0.05)	1.39 (0.05)
mix40-25	6	3.00	3.17	1.83 (0.11)	1.84 (0.11)
mix40-26	7	3.03	2.80	1.18 (0.06)	1.21 (0.06)
mix40-27	3	3.71	3.29	1.23 (0.05)	1.28 (0.05)
mix40-28	4	3.61	3.19	1.38 (0.09)	1.38 (0.09)
mix40-29	6	3.43	2.90	1.18 (0.09)	1.21 (0.09)
mix40-30	6	4.22	3.35	1.19 (0.08)	1.24 (0.07)
mix40-31	4	4.37	3.43	1.18 (0.06)	1.35 (0.06)
mix40-32	3	4.34	3.70	1.22 (0.08)	1.26 (0.08)

[1] This table continues on the next page.

mix40-33	3	4.14	3.34	1.20 (0.06)	1.32 (0.06)
mix40-34	3	4.25	3.74	1.31 (0.09)	1.33 (0.09)
mix40-35	5	4.08	3.41	1.21 (0.07)	1.31 (0.06)
mix40-36	3	4.21	3.58	1.20 (0.08)	1.27 (0.08)
mix40-37	4	4.39	4.51	2.49 (0.11)	2.51 (0.11)
mix40-38	6	4.43	3.92	1.29 (0.07)	1.36 (0.07)
mix40-39	4	4.15	3.32	1.23 (0.09)	1.22 (0.09)

Table B.9: On dynamic population: when a new agent is added every 10^{th} iteration ($i = 10$)

Problem[2]		Price of anarchy (monarchy)			
ID	$\|S\|$	PNE	-m	+m+n	IMPRES
mix40-0	6	1.18	1.22	1.17 (0.08)	1.17 (0.08)
mix40-1	3	1.00	1.01	1.00 (0.08)	1.00 (0.08)
mix40-2	4	1.00	1.02	1.02 (0.08)	1.02 (0.08)
mix40-3	4	1.00	1.00	1.00 (0.08)	1.00 (0.08)
mix40-4	6	1.25	1.23	1.22 (0.08)	1.22 (0.08)
mix40-5	7	1.02	1.04	1.04 (0.08)	1.04 (0.08)
mix40-6	5	1.04	1.31	1.03 (0.07)	1.03 (0.08)
mix40-7	3	1.08	1.80	1.07 (0.08)	1.06 (0.07)
mix40-8	5	1.08	1.12	1.07 (0.08)	1.06 (0.08)
mix40-9	3	1.15	1.27	1.08 (0.08)	1.08 (0.08)
mix40-10	6	2.19	2.18	1.49 (0.08)	1.48 (0.08)
mix40-11	3	2.47	2.33	1.61 (0.08)	1.58 (0.07)
mix40-12	3	2.54	2.53	1.52 (0.07)	1.52 (0.07)
mix40-13	3	2.41	2.32	1.60 (0.07)	1.60 (0.07)
mix40-14	7	2.08	2.30	1.54 (0.07)	1.51 (0.07)
mix40-15	3	2.23	2.26	1.56 (0.07)	1.55 (0.07)
mix40-16	6	2.78	3.01	1.61 (0.06)	1.67 (0.06)
mix40-17	3	2.16	2.07	1.51 (0.07)	1.51 (0.08)
mix40-18	5	2.39	2.49	1.55 (0.06)	1.58 (0.06)
mix40-19	5	2.18	2.33	1.57 (0.07)	1.56 (0.07)
mix40-20	3	3.05	2.91	1.70 (0.07)	1.69 (0.07)
mix40-21	3	3.56	2.92	1.28 (0.06)	1.33 (0.06)
mix40-22	4	3.22	2.61	1.33 (0.07)	1.31 (0.07)
mix40-23	3	3.12	2.90	1.44 (0.07)	1.45 (0.06)
mix40-24	3	3.18	2.72	1.33 (0.05)	1.43 (0.06)
mix40-25	6	3.00	3.18	1.90 (0.08)	1.88 (0.08)
mix40-26	7	3.03	2.89	1.34 (0.07)	1.37 (0.07)
mix40-27	3	3.71	3.29	1.38 (0.06)	1.42 (0.06)
mix40-28	4	3.61	3.19	1.75 (0.08)	1.65 (0.07)
mix40-29	6	3.43	2.90	1.41 (0.07)	1.41 (0.07)
mix40-30	6	4.22	3.35	1.38 (0.07)	1.38 (0.07)
mix40-31	4	4.37	3.43	1.34 (0.06)	1.38 (0.06)
mix40-32	3	4.34	3.70	1.50 (0.07)	1.47 (0.07)

[2] This table continues on the next page.

mix40-33	3	4.14	3.34	1.32 (0.06)	1.40 (0.06)
mix40-34	3	4.25	3.74	1.65 (0.07)	1.59 (0.07)
mix40-35	5	4.08	3.41	1.44 (0.06)	1.46 (0.07)
mix40-36	3	4.21	3.58	1.50 (0.07)	1.50 (0.07)
mix40-37	4	4.39	4.51	2.72 (0.08)	2.70 (0.08)
mix40-38	6	4.43	4.05	1.65 (0.07)	1.63 (0.07)
mix40-39	4	4.15	3.32	1.44 (0.07)	1.40 (0.07)

Table B.10: On dynamic population: when a new agent is added every 5th iteration ($i = 5$)

Problem[3]		Price of anarchy (monarchy)			
ID	$\lvert S\rvert$	PNE	-m	+m+n	IMPRES
mix40-0	6	1.18	1.23	1.22 (0.03)	1.22 (0.03)
mix40-1	3	1.00	1.01	1.00 (0.02)	1.00 (0.02)
mix40-2	4	1.00	1.02	1.02 (0.02)	1.02 (0.02)
mix40-3	4	1.00	1.00	1.00 (0.03)	1.00 (0.03)
mix40-4	6	1.25	1.23	1.23 (0.03)	1.23 (0.03)
mix40-5	7	1.02	1.04	1.04 (0.02)	1.04 (0.02)
mix40-6	5	1.04	1.37	1.05 (0.02)	1.05 (0.02)
mix40-7	3	1.08	1.78	1.12 (0.02)	1.12 (0.02)
mix40-8	5	1.08	1.10	1.10 (0.03)	1.10 (0.03)
mix40-9	3	1.15	1.22	1.16 (0.03)	1.15 (0.03)
mix40-10	6	2.19	2.23	1.89 (0.02)	1.87 (0.02)
mix40-11	3	2.47	2.33	2.08 (0.02)	2.05 (0.02)
mix40-12	3	2.54	2.59	2.06 (0.02)	2.04 (0.02)
mix40-13	3	2.41	2.32	2.02 (0.02)	1.99 (0.02)
mix40-14	7	2.08	2.40	2.11 (0.02)	2.05 (0.02)
mix40-15	3	2.23	2.32	2.02 (0.02)	2.00 (0.02)
mix40-16	6	2.78	3.07	2.52 (0.02)	2.48 (0.02)
mix40-17	3	2.16	2.07	1.88 (0.02)	1.86 (0.02)
mix40-18	5	2.39	2.49	2.12 (0.02)	2.08 (0.02)
mix40-19	5	2.18	2.39	2.08 (0.02)	2.04 (0.02)
mix40-20	3	3.05	2.96	2.31 (0.02)	2.28 (0.02)
mix40-21	3	3.56	2.92	2.26 (0.02)	2.08 (0.02)
mix40-22	4	3.22	2.61	1.89 (0.02)	1.83 (0.02)
mix40-23	3	3.12	2.90	2.25 (0.02)	2.22 (0.02)
mix40-24	3	3.18	2.72	2.32 (0.02)	2.25 (0.02)
mix40-25	6	3.00	3.11	2.54 (0.03)	2.47 (0.03)
mix40-26	7	3.03	3.02	2.09 (0.02)	2.06 (0.02)
mix40-27	3	3.71	3.29	2.36 (0.02)	2.29 (0.02)
mix40-28	4	3.61	3.19	2.62 (0.03)	2.58 (0.03)
mix40-29	6	3.43	2.90	2.09 (0.02)	1.96 (0.02)
mix40-30	6	4.22	3.35	2.30 (0.02)	2.17 (0.02)
mix40-31	4	4.37	3.43	2.37 (0.02)	2.35 (0.02)
mix40-32	3	4.34	3.70	2.41 (0.02)	2.32 (0.02)

[3] This table continues on the next page.

mix40-33	3	4.14	3.34	2.42 (0.02)	2.34 (0.02)
mix40-34	3	4.25	3.74	2.56 (0.02)	2.52 (0.02)
mix40-35	5	4.08	3.41	2.50 (0.02)	2.37 (0.02)
mix40-36	3	4.21	3.58	2.39 (0.02)	2.28 (0.02)
mix40-37	4	4.39	4.51	3.49 (0.03)	3.49 (0.03)
mix40-38	6	4.43	4.52	3.27 (0.03)	3.13 (0.03)
mix40-39	4	4.15	3.32	2.21 (0.02)	2.10 (0.02)

Table B.11: On dynamic population: when a new agent is added every iteration ($i = 1$)

Index

Bibliography

[1] W. Brian Arthur. Inductive reasoning and bounded rationality (the El Farol problem). *American Economic Association Annual Meeting, Complexity in Economics Theory*, 1994.

[2] Robert J. Aumann. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica*, 55(1):1–18, 1987.

[3] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Adam Tauman Kalai, Vahab Mirrokni, and Christos Papadimitriou. The myth of the folk theorem. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pages 365–372, New York, NY, USA, 2008. ACM.

[4] Michael H. Bowling and Manuela M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[5] Gearge W. Brown. Some notes on computation of games solutions. Technical report, RAND CORP, Santa Monica, CA, April 1949.

[6] Jan-P. Calliess and Geoffrey J. Gordon. No-regret learning and a mechanism for distributed multiagent planning. In *Proceedings of the Seventh international joint conference on Autonomous agents and multiagent systems*, pages 509–516, Estoril, Portugal, 2008. IFAAMAS.

[7] Deeparnab Chakrabarty, Aranyak Mehta, and Viswanath Nagarajan. Fairness and optimality in congestion games. In *Proceedings of the Sixth ACM conference on Electronic commerce*, pages 52–57, New York, NY, USA, 2005. ACM.

[8] Xi Chen and Xiaotie Deng. Settling the complexity of 2-player Nash-equilibrium. In *Proceedings of the Forty-Seventh Annual IEEE Symposium on Foundations of Computer Science*, pages 261–272, 2006.

[9] George Christodoulou and Elias Koutsoupias. Coordination mechanisms. *Lecture Notes in Computer Science*, 3142:345–357, 2004.

[10] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, 1998.

[11] Vincent Conitzer and Tuomas Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

[12] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to algorithms*. The MIT Press, Cambridge, Massachusetts, 1 edition, 1990.

[13] Richards Cornes and Todd Sandler. *The theory of externalities, public goods, club goods*. Cambridge University Press, New York, 1996.

[14] Jacob W. Crandall and Michael A. Goodrich. Learning to compete, compromise, and cooperate in repeated general-sum games. In *Proceedings of the Twenty-Second International Conference on Machine learning*, pages 161–168, New York, NY, USA, 2005. ACM.

[15] Laura A. Dabbish and Robert E. Kraut. Email overload at work: an analysis of factors associated with email strain. In *Proceedings of the Twentieth Anniversary Conference on Computer Supported Cooperative Work*, pages 431–440, New York, NY, USA, 2006. ACM.

[16] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, pages 71–78, New York, NY, USA, 2006. ACM.

[17] Richard Dawkins. *The selfish gene*. Oxford University Press, 1976.

[18] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *In Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 604–612. ACM, 2004.

[19] Julie Farago, Amy Greenwald, and Keith Hall. Fair and efficient solutions to the Santa Fe bar problem. In *Grace Hopper Celebration of Women in Computing*, 2002.

[20] Eugene Fink, P. Matthew Jennings, Ulas Bardak, Jean Oh, Stephen F. Smith, and Jaime G. Carbonell. Scheduling with uncertain resources: Search for a near-optimal solution. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2006.

[21] Dean P. Foster and H. Peyton Young. On the impossibility of predicting the behavior of rational agents. Working Papers 01-08-039, Santa Fe Institute, August 2001.

[22] Michael Freed, Jaime Carbonell, Geoff Gordon, Jordan Hayes, Brad Myers, Dan Siewiorek, Stephen Smith, Aaron Steinfeld, and Anthony Tomasic. Radar: A personal assistant that learns to reduce email overload. In *Proceedings of the Twenty-Third Conference on Artificial Intelligence*, pages 1287–1293, Menlo Park, CA, USA, 2008. AAAI.

[23] Drew Fudenberg and Eric Maskin. The folk theorem in repeated games with discounting or with incomplete information. *Econometrica*, 54(3):533–554, May 1986.

[24] Amy R. Greenwald. *Learning to play network games: Does rationality yield Nash equilibrium?* PhD thesis, New York University, 1999.

[25] James Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 1957.

[26] Garrett Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[27] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[28] Sergiu Hart and Andreu Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 1:26–54, 2001.

[29] Ara Hayrapetyan, Éva Tardos, and Tom Wexler. The effect of collusion in congestion games. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 89–98, New York, NY, USA, 2006. ACM.

[30] Junling Hu and Michael P. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[31] David S. Johnson, Christos H. Papadimtriou, and Mihalis Yannakakis. How easy is local search? *Computer and System Sciences*, 37(1):79–100, 1988.

[32] Ehud Kalai and Ehud Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, 61(5):1019–45, September 1993.

[33] Ehud Kalai and Ehud Lehrer. Subjective equilibrium in repeated games. *Econometrica*, 61(5):1231–40, September 1993.

[34] Yannis A. Korilis, Aurel A. Lazar, and Ariel Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking*, 5:161–173, 1997.

[35] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. *Lecture Notes in Computer Science*, 1563:404–413, 1999.

[36] Michael Littman and Peter Stone. A polynomial-time Nash equilibrium algorithm for repeated games. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 48–54, 2003.

[37] Michael L. Littman. Friend-or-Foe Q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 322–328, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[38] Carol Meyers. *Network flow problems and congestion games: Complexity and approximation results*. PhD thesis, Massachusetts Institute of Technology, June 2006.

[39] Igal Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13:111–124, 1996.

[40] Igal Milchtaich. Social optimality and cooperation in nonatomic congestion games. *Journal of Economic Theory*, 114:56–87, January 2004.

[41] Pragnesh Jay Modi, Manuela Veloso, Stephen F. Smith, and Jean Oh. CMRadar: A personal assistant agent for calendar management. *Agent Oriented Information Systems II. Lecture Notes in Computer Science*, 3508:169–181, 2005.

[42] Dov Monderer and Lloyd Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

[43] John Forbes Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

[44] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, editors. *Algorithmic game theory.* Cambridge University Press, first edition, September 2007.

[45] Jean Oh and Stephen F. Smith. Learning user preferences in distributed calendar scheduling. *Practice and Theory of Automated Timetabling V, Lecture Notes in Computer Science*, 3616:3–16, 2005.

[46] Jean Oh and Stephen F. Smith. A few good agents: Multi-agent social learning. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 339–346, Estoril, Portugal, 2008. IFAAMAS.

[47] Martin J. Osborne and Ariel Rubinstein. *A course in game theory.* The MIT Press, Cambridge, Massachusetts, 1994.

[48] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and Systems Sciences*, 48(3):498–532, 1994.

[49] Arthur Cecil Pigou. *The economics of welfare.* Macmillan, 1920.

[50] Rob Powers, Yoav Shoham, and Thuc Vu. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning*, 67(1-2):45–76, 2007.

[51] Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[52] Tim Roughgarden. Stackelberg scheduling strategies. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 104–113, New York, NY, USA, 2001. ACM Press.

[53] Tim Roughgarden. Selfish routing and the price of anarchy (survey). *OPTIMA*, 74, 2007.

[54] Sandip Sen, Stephane Airiau, and Rajatish Mukherjee. Towards a pareto-optimal solution in general-sum games. In *Proceedings of the Second Intenational Joint Conference on Autonomous Agents and Multiagent Systems*, pages 153–160, 2003.

[55] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-Theoretic, and logical foundations.* Cambridge University Press, New York, NY, 2008.

[56] Herbert A. Simon. *Administrative behavior.* The Free Press, New York, NY, fourth edition, 1997.

[57] Herbert A. Simon. *Models of bounded rationality: Empirically grounded econimic reason*, volume 3. MIT Press, Cambridge, MA, 1997.

[58] Jeff L. Stimpson, Michael A. Goodrich, and Lawrence C. Walters. Satisficing and learning cooperation in the prisoners dilemma. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 535–540, 2001.

[59] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction.* MIT Press, Cambridge, MA, 1998.

[60] Katja Verbeeck, Johan Parent, and Ann Now. Homo egualis reinforcement learning agents for load balancing. *Lecture Notes in Computer Science*, 2564:81–91, 2003.

[61] H. Peyton Young. The possible and the impossible in multi-agent learning. Economics Series Working Papers 304, University of Oxford, Department of Economics, 2007.