ABSTRACT

MASHAYEKHI, MEHDI. Regulation of Open Normative Multiagent Systems. (Under the direction of George List and Munindar Singh.)

Multiagent Systems (MASs) are computerized systems built of autonomous software agents that interact and sometimes work together to resolve complex problems. Regulation and coordination of agents' activities are key issues within the MAS. MAS community has commonly made use of *norms* for coordination of the activities of the agents. In a MAS, a (social) norm describes what the agents may expect from each other in specific situations. The importance of norms arises from their promoting both autonomy (in general, an agent need not comply with a norm) and heterogeneity (an agent need not comply with a norm). Researchers have studied norm emergence through social learning where the agents interact repeatedly in a closed system (e.g., Graph structure).

This dissertation focuses on norm emergence in an *open* system. In an open MAS, the population of agents and their memberships change dynamically. Also behavior of the agents is not known a priori. These specific situations make it challenging the regulation of agents' behaviors within a MAS.

Considering this background, in this research we propose two new computational frameworks for regulation of open multiagent systems.

Our first proposed framework which we dubbed *Silk*, is a hybrid *bottom-up top-down* framework which is composed of two kinds of agents. It has a unique *generator* with a global view which monitors interactions among two or more *members*. The generator institutes *laws*, i.e., hard integrity constraints that ensure certain undesirable events do not occur. It recommends norms, i.e., soft recommendations that if respected would lead to conflict avoidance. Members must follow the laws but each member reasons individually

on whether to violate or fulfill a recommended norm. Thus, members develop their norms in a bottom-up manner to resolve conflicts.

The second proposed framework which we dubbed *Cha* is a decentralized (or bottom-up) framework. In this framework we equipped members with more capabilities, so there is no central agent (i.e., Generator). Also, we only have considered soft regulation. Each member itself anticipates potential conflicts, and creates norms if necessary. Also, unlike previous studies, the normative system can change dynamically based on continual changes in its environment.

We evaluate both proposed frameworks in transportation domain via the regulation of traffic in an intersection in a simulated system. Our results show that social norms that lead to conflict resolution emerge in both frameworks.

Overall, this research advances in the state of the art of regulation of open multiagent systems by proposing two novel computational frameworks.

Regulation of Open Normative Multiagent Systems

by
Mehdi Mashayekhi

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Civil Engineering

Raleigh, North Carolina

2016

APPROVED BY:

Yahya Fathi	Emily Berglund
George List Co-chair of Advisory Committee	Munindar Singh Co-chair of Advisory Committee

DEDICATION

To my family, my lovely parents, my beloved grandparents, my bother, and my two sisters. Without their support and love, none of this would be possible. I am truly blessed.

BIOGRAPHY

Mehdi Mashayekhi is a PhD candidate in transportation systems at North Carolina State University. He received his BS degree in Civil Engineering from Iran University of Science & Technology in 2008 and his MS degree in Transportation Systems from Iran University of Science & Technology in 2011 in Tehran, Iran.

In Fall 2012, he joined Civil Engineering program at North Carolina State University to pursue his PhD degree. His research interests include intelligent transportation systems and logistics, simulation, artificial intelligence, multiagent systems, machine learning, data science, and social computing.

ACKNOWLEDGEMENTS

This dissertation is the end of my PhD journey. It would not have been possible to finish this journey without the support and help of many people. I would like to take this opportunity to recognize them. Somehow, all of them have left an indelible mark on me.

First and foremost, I would like to express my deepest gratitude to my advisors. There is no doubt that I would not be able to complete this work without their support, and guidance. I am deeply grateful for having the opportunity to collaborate with them during my time at the North Carolina State University. To Professor George List, who has been a tremendous mentor during this journey. His guidance, dedication, kindness, and patience have contributed immensely to my development as an independent researcher and a problem solver. I am forever indebted to him for believing in me, being extremely supportive, and teaching me the importance of good research. To Professor Munindar Singh, one of the most brilliant minds I have ever known, for letting me be a part of his team, co-advising me along this way, and assisting me in shaping as a student and researcher. I have learned so many things from you. Thank you for teaching me to think abstractly, to explain things clearly, and to be rigorous.

I am also grateful to Professor Yahya Fathi, and Professor Emily Berglund for accepting to serve on my doctoral committee and generously sharing their knowledge with me.

I also had the honor of collaborating with Professor Billy Williams and Professor Nagui Rouphail in the transportation research group at North Carolina State University. Thanks to them for helping me have a better understanding of transportation domain problems.

I would also like to express my gratitude to Professor Richard Kim for supporting me as a teaching assistant during the first two years of my PhD.

Last but not least, I am so blessed and grateful to have many true and real friends all over the world. Many thanks go to Mohammad Sabouri, Yuriy Veytskin, Ehsan Dousmohamadi, Amir Bahmani, Reza Shamsaei, Mohammad A. Tutunchian, Hooman Amid, Siddharth Shah, Hadi Moheb-Alizadeh, Maryam Delavarrafiee, Ashtad Javanmardi, Amirhossein Norouzi, Alireza Abbasian, Bahman Moghimi, Pooya Najaf, Meysam Naderi, Farzad Karimzad, Masoud Ghavami, Sahar Talebi, Pavan Chigullapally, Nirav Ajmeri, Elizabeth Williams, Bo Yang, Javon Marcell Adams, Jeong Hyuk Im, and Mohammad Ilias for extreme kindness, heartfelt friendship, and lifting my spirits when I needed it the most.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Introduction

A social norm characterizes sound or “normal” interactions among members of a social group, reflecting their mutual expectations [37] . Norms are a powerful means to regulate the interactions of autonomous agents. For example, following a “first-come first-served” norm in an uncontrolled intersection (with no signs and signals) helps determine which vehicle has right of way. Norms are commonly used in resource sharing settings, such as those used for information sharing on social media, and for road sharing by autonomous vehicles. Specifically, there has been an increasing realization of norms for characterizing acceptable behaviors and achieving sustainable collaboration. In essence, norms can characterize a social architecture that promotes prosocial behavior [36]. Previous studies on norms, such as the EMIL project [5, 24], has shown encouraging outcomes on modeling and simulating real-life problems such as regulating e-communities, article authorship in Wikipedia, and financial decisions making.

During the past decades, the growth of technology has promoted the advent of multiagent systems (MAS), these computing systems in which autonomous agents meet and interact to solve problems that are beyond the individual abilities of each problem solver. Within a MAS, agents are usually assumed to have goals that may be in conflict with one and other. In order to successfully interact, they need some mechanism in which they can coordinate their actions. Motivated on human societies, researchers have extensively applied norms in order to *regulate* agents' actions in multiagent systems [34]. Regulation should be in such a way that each agent can successfully accomplish its goals without precluding other agents from achieving theirs; and also some global, system-level goals can be reached. The problem of generating norms to reach regulation in a multiagent system has attracted significant attention of many researchers. Resolving this problem is particularly difficult in the case of *open* multiagent systems, which are typically characterized by their dynamics and uncertainty. Briefly, In an open MAS, the population of agents and their memberships change dynamically. Also behavior of the agents is not known a priori.

Whereas social learning [3] assumes that agents interact repeatedly in a closed system (i.e., With neighbors in a specified graph), we consider an open multiagent system, namely, one wherein agents can dynamically join and leave [26]. The leaving agents share their experience with joining members, thus promoting learning. This configuration is compatible with a wide range of real-life scenarios. For instance, in many cybersecurity problems, the parties are autonomous and constantly changing.

Considering this background, in this research we propose two novel computational frameworks for regulation of open multiagent systems. Our first proposed framework which we dubbed *Silk*, is based on the metaphor of the famous Silk Road—agents enter, transact with each other, and exit. The second proposed framework which we dubbed *Cha*, meaning tea, refers to ritual tea ceremony in human societies, in which agents get together, interact, develop social norms while having tea, and exit. Newcomers learn such norms from oldtimers.

1.2 Motivation

An open multiagent System (MAS) is a system in which the member agents are developed and designed independently (maybe by different parties) and serve different, often conflicting interests. In open MAS, the behavior of an agent cannot be anticipated in advance [6]. Also, the membership in the system changes dynamically and agents can freely join and leave the system at any time. A few instances of this open MAS are electronic marketplaces, virtual organizations and digital media rights management applications.

As pointed out in previous section, existing studies of norm emergence through social learning, e.g., [3], assume the agents interact in a closed system. In contrast, we consider an open system and a setting wherein outgoing members share their experiences with incoming members. An assumption of a closed system is rarely appropriate. Even a mundane setting, such as cybersecurity within an organization, presupposes openness because the parties concerned are autonomous and changing. Defining and generating norms that will help regulate an open MAS remains an open problem.

In the literature, three approaches to the regulation of MASs have been studied: (1) bottom-up or decentralized (norms arise via interaction among member agents); (2) top-down or centralized (a central agent or governor determines the norms); and (3) a hybrid of the two. Most of the existing studies have been focusing on the top-down approaches which somehow endangers the autonomy of member agents. However, in our study, Silk offers a hybrid framework, while Cha offers a decentralized framework for regulation of open MASs.

Interaction of autonomous member agents in an open multiagent system (MAS), might lead to *Integrity* violation and *Conflict*. *Integrity* refers to a hard requirement in the domain of interest (e.g., Collision in traffic), which a MAS may or may not allow members to violate. *Conflict* refers to a potential undesirable situation (e.g., An unsafe move in traffic), which typically a MAS allows its members to proceed with (even though it is undesirable) so as to support greater autonomy. Following Jones and Sergot [19], we differentiate *regimentation* (technical control to ensure integrity that members cannot override) from *regulation* (social control to promote cooperation that members can disregard).

Silk is a hybrid framework (i.e., top-down and bottom-up) which is composed of two kinds of agents. Silk accommodates both regimentation and regulation. It does so via *laws* and *norms*, respectively. It has a unique *generator* with a global view which monitors interactions among *members*. The generator institutes *laws*, i.e., hard integrity constraints that ensure certain undesirable events do not occur. It recommends norms, i.e., soft recommendations that if respected would lead to conflict avoidance. Members must follow the laws but each member reasons individually on whether to violate or fulfill a recommended norm. Thus, members develop their norms in a bottom-up manner to resolve conflicts. Think of a law as being realized in the technical infrastructure of an MAS. For example, a locking mechanism implements a law prohibiting concurrent access to a resource. The generator also anticipates conflicts, defines a potential norm that would avert the conflict, and broadcasts the norm to the members as a recommendation. Think of a norm as being realized in the social architecture. For example, a norm would be for users to avoid a resource when others are working on it. A recommendation from the generator is not quite a "norm"—a norm based on different members accepting or rejecting the recommendation may come into being.

Cha is a decentralized (or bottom-up) framework. In this framework, we equipped members with more capabilities, so there is no central agent. Also, we only have concentrated and considered soft regulation, since that is the more challenging aspect. In Cha, each member itself anticipates potential conflicts, and creates norms if necessary.

Moreover, almost all existing social learning approaches for norm emergence assume a static environment, where actions have fixed consequences. Silk also supports static environment. However, there are MASs in which this assumption might not hold and require social norms to change when environmental conditions change (which static settings do not support). Cha offers a dynamic setting to address this. In dynamic setting, the environment's properties can change and lead to varied consequences of actions.

Finally, the other important issue which is tackled in this dissertation is the issue of *fairness* and social optimality in multiagent systems. Due to individual rationality of agents, addressing fairness is a non-trivial issue to address. To the best of our knowledge fairness has not got enough attention in MAS community. In this dissertation, both proposed frameworks; Silk and Cha; tackle the issue of fairness and social optimality in MASs.

1.3 Research Hypotheses

As previously discussed, regulation of open multiagent systems remains an open problem in the literature. There needs to be computational frameworks which could efficiently regulate agents' interactions, so that they can successfully interact. From this discussion, we study the following high-level research hypotheses, which we map into lower-level hypotheses for evaluation on our testbed that realizes Silk and Cha:

- *Conflict:* Both Silk and Cha frameworks can resolve conflicts in open MASs.

- *Efficient-conflict:* Cha framework can resolve conflicts in an efficient manner.

- *Dynamic-adaption:* Agents in Cha's framework can adapt their decision making based on the changes to their environment.

- *Fairness:* Both Silk and Cha frameworks support achieving fairness, which indicates that resource allocation is not grossly imbalanced.

1.4 Contribution

This dissertation contributes to the research hypotheses posed in Section 1.3 by proposing two novel computational frameworks for norm emergence which enhance fairness in resource sharing.

In particular, *Silk* offers the following contributions:

- Supports regulation of *open* MAS in a hybrid fashion.

- A general, domain-independent framework to accommodate online norm generation. Where a central agent called *generator* monitors the system at runtime and ensures *Integrity violations* are averted by imposing *laws*. Also, it anticipates *conflicts*, and recommends norms to the *members* for conflict resolution. Also Silk does not require previous knowledge of agents' behaviors and their potential interactions.

- Despite the fact that the norm community has long mentioned the suitability of combining both bottom-up and top-down approaches to norm regulation, as far as we know, no other works have actually gone as far as this one. Silk offers a hybrid approach while considering, at the same time, *fairness*. Fairness is achieved relying on the generator.

And *Cha* offers the following contributions:

- Supports regulation of *open* MAS in a bottom-up fashion.
- A general, domain-independent framework to accommodate online norm generation within an open MAS and regulation of the members. In this framework members reason individually on what norm to develop and follow. Also Like Silk's framework, Cha does not require previous knowledge of agents' behaviors and their potential interactions.
- A dynamic normative framework, in which the system reacts to the dynamically changing environment. If the environment condition changes, it allows for decay of existing normative set and emerging of new one.
- Cha's framework supports achieving fairness. In this framework, fairness is achieved by the members based on a variation of inequity-aversion theory (described later).

Finally, in this dissertation, we have used transportation domain to evaluate the performance of the proposed frameworks. As you would most likely agree, building effective and user-friendly transportation systems is one of the big challenges for engineers in the 21st century [18]. Transportation domain can be viewed as an open multiagent system in which their participants have diverse goals and motivations, and their behaviors are unpredictable. Social norms seem to be very suitable to regulate such a system. This dissertation is among the few studies (if not the only one) who tackle the problem of regulation of traffic form the view point of normative multiagent systems.

Below is a detailed list of publications derived from this dissertation.

- *Published conference paper:*

M. Mashayekhi, H. Du, G.F. List, M.P. Singh “ Silk: A Simulation Study of Regulating Open Normative Mutiagent Systems ”, 25th International Joint Conference on Artificial Intelligence (IJCAI), New York, July 2016, pages 373-379, 2016.

- *Published workshop paper:*

  M. Mashayekhi, G.F. List “ A Multiagent Auction-based Approach for Modeling of Signalized Intersections ”, (IJCAI Workshops), The 2nd Workshop on Synergic between Multiagent Systems, Machine Learning and Complex Systems (TRI) Buenos Aires, Argentina, 2015.

- *Submitted conference paper:*

  M. Mashayekhi, R. Beheshti, G.F. List, M.P. Singh “ A Dynamic Framework for Decentralized Norm Emergence in Open Multiagent Systems ”, Submitted, 2016.

Also, the motivation behind this dissertation is influenced by the following paper:

- G.F. List, M. Mashayekhi “ A Modular Colored Stochastic Petri Net for Modeling and Analysis of Signalized Intersections ”, IEEE Transactions on Intelligent Transportation Systems, Volume: 17, Issue: 3, 2016.

1.5 Organization

The rest of this dissertation is organized as follows.

Chapter 2 provides required background needed in this study and present a set of general definitions. We finish by presenting a summary of related literature in multiagent systems.

Chapter 3 presents a set of specific definitions, terms and assumptions useful for remaining chapters. Also, we introduce and present the architecture and workflow of both proposed frameworks, namely Silk and Cha.

Chapter 4 introduces a scenario that can be modeled as an open multiagent system, in which norms can be used to regulate the activities of their members. The described scenario will be used to empirically evaluate the proposed frameworks.

Chapter 5 presents the results of our experiments for the scenario described in Chapter 4 and in relation to our research hypotheses presented in Chapter 1.

Chapter 6 finishes with conclusions derived from this study, and discuss future research directions.

Chapter 2

Foundations

2.1 Introduction

The research community in multiagent systems (MASs) has studied various problems in relation to norms and normative multiagent systems (NMAS). One of the most important and related problems (also related to the problem in this dissertation) is how to create norms to effectually regulate a given MAS. Considering this background, section 2.2 of this chapter aims to provide a brief required knowledge needed for the rest of this dissertation. In section 2.3, the second part of this chapter, we provide a summary of relevant literature related to the problem of norm emergence in MASs.

2.2 Background

2.2.1 Types of Norms

A number of definitions for norm exists because of its multi-disciplinary interest. Singh [37], identifies the norm as sound or “normal” interactions among the participants of a social group, reflecting their mutual expectations. Researchers have divided norms into various categories.

Mahmoud et al. [25] has categorized norms into the following categories (Figure 2.1):

- *Conventions* Conventions are natural norms that come to exist without any enforcement. Young [46] defined conventions as “a pattern of behavior that is customary, expected, and self-enforcing. Everyone conforms, everyone expects others to conform, and everyone wants to conform given that everyone else conforms.”
- *Essential Norms:* Essential norms are divided into three categories:
  - Regulative Norms: Regulative norms describe *obligations*, *prohibitions* and *permissions* of actions [12]. Mahmoud et al. [25] propose a new type of regulative norms, which they call *recommendation* norms. This is the type of norm that “rewards an agent if it is exercised by the agent but is not penalized otherwise.” Regulative norms are influenced by reward and penalty.
  - Constitutive Norms: Which describe that something counts as something else for a given institution [12]. Boella and van der Torre [12] differentiated regulative norms and constitutive norms with an example; for instance, if the regulative norm states that cars are forbidden in the park, then the constitutive norm is that “bicycles are also counted as vehicles in the park.”
  - Procedural Norms: Procedural norms are those which are objective and subjective. Objective procedural norms characterize the rules that convey how decisions are made in a normative system, whereas subjective procedural norms define the instrument for individuals functioning in a system.

In this research we focus on regulative norms. Thus, a norm is assumed to be a formal specification of a deontic statement that imposes prohibitions, obligations, or permissions on the behavior of agents. For more information regarding norm categories, please refer to the cited references.

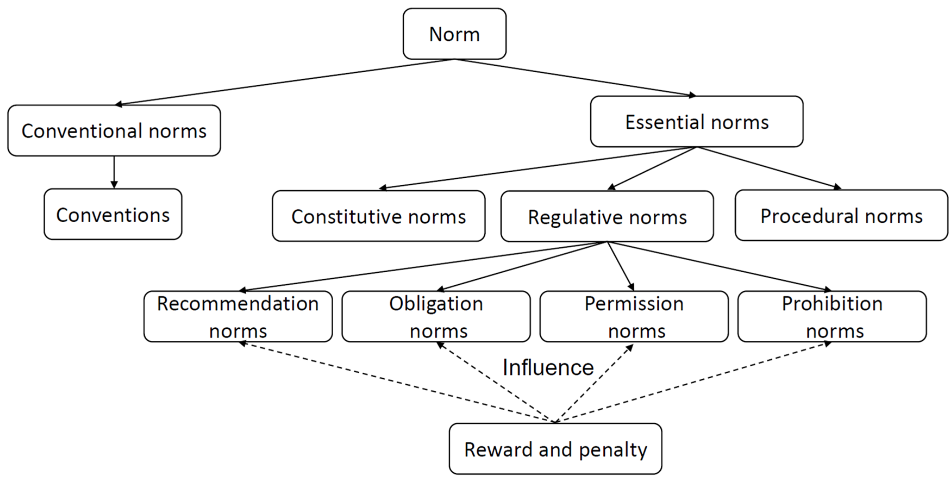


Figure 2.1: Norm classification architecture.

2.2.2 Norm Design

Although norms are pervasive around us; they are complicated and challenging to be studied and designed. In the literature, there are two approaches for designing norms. First approach is *offline*, and second approach is *online*.

offline design follows the following steps: (i) characterizing all the probable states of a MAS at design time; (ii) recognizing those states that must be regulated to reach coordination; (iii) designing norms to regulate such states; and (iv) hardwiring norms into the agent decision making process. Example of an offline design, is the work by Shoham and Tennenholtz [35] in which norms are designed offline to regulate the navigation of robots in a grid. Offline approach suffers from the fact that, some characteristics may not be known at the design time, and agent goals may be constantly changing. Also, complex systems are hard to predict.

On the other hand there is *online* approach where, norms are constructed for a system while that system is actually running. Online method, has the benefit that they do not suffer from the computational complexity issues of offline method, since there is no need to know or compute the system state space at design time. In online approach, norm emergence is based on the agents' ability to converge to a solution from a space of alternate solutions. A norm is considered to have emerged when a large proportion of the agent population has followed to some common behavior. For example, in a traffic scenario with driverless cars (car agents), two alternate norms might emerge: driving on the left, or driving on the right side of the road. When majority of the agents, drive on the same side, it is said that a norm has emerged.

In this research we focus on online norm emergence.

2.2.3 Agent-based Modeling

Bringing together the area of agent-based modeling (ABM) and multiagent systems (MAS), provides an opportunity for creating multiagent based modeling (MABM).

The focus of agent-based modeling is on simulating and synthesizing social behaviors in order to understand observed social systems of autonomous agents (e.g., Humans). The key idea in ABM is to define a population of agents, and to define rules regulating the behavior of agents, in order to allow complex notions that are hard to model, emerge within the system. ABM has been successfully used to a long list of various domains such as water resources planning and management [10, 47].

Multiagent systems (MAS), on the other hand, emphases on the solution of hard engineering problems linked to the construction, deployment and effectual operation of multiagent based systems. ABM and MAS has learned much from each other and provided an opportunity for multiagent based simulation community (MABM).

Tools and Repast Symphony

A fairly big set of tools and frameworks have evolved around the various disciplines. The tools and platforms on the agent-based modeling usually stand out by their fast prototyping abilities. Their goal is to let the fast implementation of simulations that allow the analysis of emerging behaviors developed by agents' interactions. Popular ABM simulation environments are NetLogo [1], Swarm [2], MASON[3], and Repast Symphony [4].

In this research we have used Repast Simphony. Repast Simphony has made a big step toward assisting users with tools to trace the simulation. The simulation environment offers an easy way to generate graphical output of the running simulation, runtime control (like step by step execution, debugging, etc.) and has interfaces to export data to a broad range of statistical and mathematical programs like MatLab.

2.2.4 Normative Multiagent Systems

A normative multiagent system is the result of bring together norms into a multiagent system to regulate agents' interactions [42]. Lately, Boella et al. [11] defined a normative multiagent system (NMAS) as a "*multiagent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify and enforce norms, and mech-*

[1] $http://ccl.northwestern.edu/netlogo$
[2] $http://www.swarm.org/wiki/Swarm_main_page$
[3] $http://cs.gmu.edu/\ eclab/projects/mason/$
[4] $http://repast.sourceforge.net/$

anisms to deliberate about norms and detect norm violation and fulfilment".

Currently, the NMAS research community is heading towards the direction in which the norms are emerged built upon agents' interactions in the system. At this level, norms are emergent standards of behavior that arise within the agent population.

In this dissertation we emphasis on the emergence of norms based on the agents' interactions (explained in the following). So, therefore that this dissertation is suited within the current state of the art of NMAS research.

2.2.5 Two Lines of Research Directions on Normative Models

Typically normative models can be categorized into two parts: *Cognitive* and *Learning* models. Figure 2.2 shows these two approaches and their sub-approaches derived from existing literature. Neumann [31] differentiates between these two groups as being divided into models that are based on architectures of cognitive agents with some roots in artificial intelligence (i.e., Cognition-based models), and the ones which are inspired by the conceptual terminology of game theory and models (i.e., Learning). This dissertation lies into learning models and specifically interaction-based models. In the following we briefly introduce these two categories.

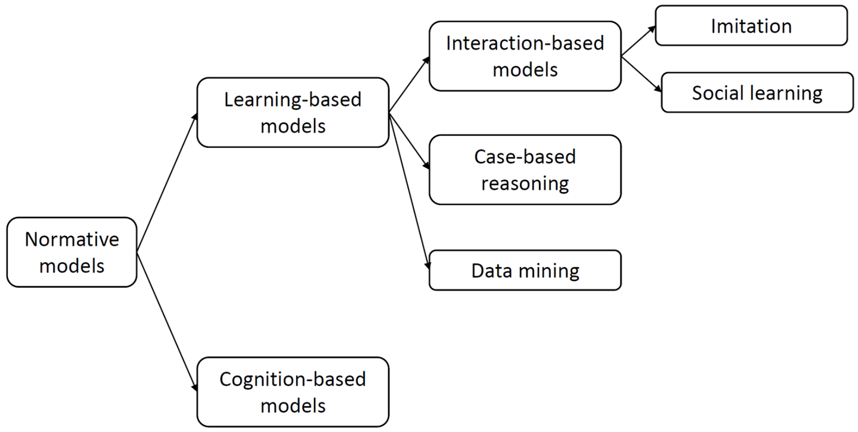


Figure 2.2: Modeling approaches of norms.

Cognition-based Models

In contrast with the learning-based models explained in next section, this category depends less on the usage of reward and punishment to encourage norm emergence.

Cognitive-based models can model norm internalization in which agents manifest behaviors, not because of existing rewards or penalties in the environment, but as a personal goal. In these models, norms emerge mostly as an internal process that occurs inside agents' minds when they enter new communities and delibrate about norms. Norm internalization via emotions is an example of these kinds of models [15]. A famous example in this category is the approach proposed by Andrighetto et al. as part of the EMIL project [5].

Learning-based Models

Learning-based model itself divides into three categories: *Interaction-based models*, *Cased-based reasoning*, and *Data mining*.

Interaction-based methods build models that can lead to norm emergence based on what agents see and perceive in the environment and their interactions with other agents. Often the agents are equipped with the capacity to learn from experience, and interactions among agents are modeled as repeated games with payoff matrices. The simplest interaction method is *imitation* approach in which agents mimic other agents in the environment —"when in Rome, do as the Romans do."

Social learning, offers a rich model for norm emergence in interaction-based category. Social norms are characterized by the strategies that the agents use to make their decisions while interacting in the system and competing against one or more adversaries. Briefly, the rules of the game and an agent's decisions yield to a specific payoff matrix. The goal of agent is to maximize its total reward, by computing a control policy. The policy is a function, that maps from each possible state of the environment to an action. Usually the assumption is that the agents do not know a priori what the rewards are for states where norms apply, nor exactly which states are affected by norms. After sufficient enough interactions, it is said that a norm arises when a majority number of agents end up using the very same strategy.

Reinforcement learning (RL) is a popular technique which is typically been used in social learning and in interaction-based models in general [41]. The progression of RL can be defined as follows. Agent first perceives the initial state of the environment and then selects and executes an action. The environment then responds with a numerical reward and a new state. The agent makes its second action based on the reward it received in the first step and the new state. This process repeats until the agent reaches the end state or it cannot proceed further.

A key problem in RL, is balancing exploration and exploitation of the state space environment. For each step, the agent needs to decide whether to follow the best action given by its learned policy (i.e., Exploitation) or randomly select an action (i.e., Exploration).

Case-based reasoning (CBR) is a supervised Machine Learning method that resolves new problems (i.e., cases) by retrieving similar problems from a knowledge base (i.e., a case base) and using their solutions. In case there is no similar approach, a new solution is suggested. Example of this method is the study done by Campos et al. [13], and Koeppen et al. [20] in which they used CBR as a learning technique to decide how to adapt domain-level norms that depend on current system status.

Data Mining. Some studies in norms modeling arisen from data mining techniques. Data mining involves searching through data archives in databases to detect significant patterns that are valuable for a decision-making procedure. Amongst the data mining techniques such as classification or clustering, association rule mining is one specific technique that extracts desired information structures like correlations, frequent patterns, associations and casualness between sets of items in transaction databases or other supplies of data [25].

2.2.6 Traditional Control Strategies for Regulation of Traffic

Here in this section, we briefly describe the fully actuated controller strategy as a common method for regulation of traffic at signalized intersections. We have used fully actuated control strategy as a base line to compare the performance of the two proposed frameworks.

Fully actuated control refers to intersections for which all phases (i.e., Combination of nonconflicting movements) are actuated and hence, it needs detection for all traffic movements [2]. Fully actuated control allows all the movement green times to vary in response to the arriving traffic. For fully actuated control, a pre-determined, repeating sequence for phase choice is followed. Phases can be skipped if there is no traffic to be serviced. Also, the durations of the movement greens are determined by *minimum green times*, *maximum green times*, and a *gap timer/out* (wait state). If no actuations happen during the wait state, the signal stays green for the current phase or for the main phase (i.e., Main street)[21].

Breifly, according to FHWA [2]:

Minimum green is a parameter that delineates the shortest permissible duration of the green interval.

Maximum green is the amount of time a phase is allowed for service after the minimum green have timed out.

Gap out is a type of actuated action for a specified phase where the phase terminates due to a absence of vehicle calls within a particular period of time (passage time).

For more information regarding the logic of this controller, please refer to the references.

2.3 Related Literature Review

This section is divided into two parts. First part of this section covers related literature regarding norm emergence in multiagent systems and are closely related to this dissertation. And, the second part briefly covers related researches in transportation domain (which is the domain of application in this dissertation). Related literature in both sections, have been briefly analyzed. The analysis below is not meant to be extensive, but to identify the important issues that exist in the related state of the art.

2.3.1 Norms and Multiagent Systems

Social norms are a useful means for regulating agents' behaviors to achieve coordination. As mentioned earlier an important, open problem in normative multiagent system research is norm emergence. This norm emergence problem, which in fact is the problem that we have focused in this dissertation, has been studied from various angles. In the following we present a brief review of the research on the problem of norm emergence from two important perspectives.

Social learning studies of closed MASs. These types of studies lie in the interaction-based category described earlier. Much of the social learning (or game-theoretic) approaches to social norm emergence consider agents in a graph structure (ie., Closed system) with fixed payoffs [3, 33, 44]. In these studies, agents are nodes of a graph. They learn their policies by playing repeated games with randomly selected agents that they are connected to, potentially leading to norms emerging. Network structure can greatly influence norm emergence [39]. Sugawara [38] applied reinforcement learning on a modified version of the Narrow Road Game, where two agents traveling in opposite directions try to decide whether to go forward when they come to any narrow part of a road which only allows one agent to go through. Sugawara showed that though norms emerge, their stability depends on the characteristic of the agents. For example, norms where most agents are selfish are robust. These works involve a fixed set of agents, two of whom interact at a time, and fixed payoffs. In contrast, Silk, and Cha consider a system that is open (i.e., agents continually leave and enter), and address fairness.

Centralized regulation of open MASs. Other works consider norm emergence in a setting where a central agent with complete knowledge produces norms. Morales et al. [28] proposed a mechanism called IRON for the on-line synthesis of norms. IRON employs designated agents and observes interactions of members in order to synthesize conflict-free norms in a top-down manner while trying to avoid over-regulation. Morales et al. proposed SIMON [29] to improve compactness (minimality and simplicity) by providing a finer granularity while generalizing norms, and LION [30] to maximize agents' freedom while maintaining compactness of the generated norms. LION achieves this by identifying substitutable and complementary norms and using them to synthesize liberal normative systems during run-time. Silk, and Cha are fundamentally different from these approaches. While they use a central norm learner, Cha uses a dynamic decentralized mechanism for norm generation, and Silk uses a hybrid mechanism. Moreover, Silk, and Cha tackle the problem of fairness.

2.3.2 Norms and Agent-based Simulation in Traffic Domain

Recently, agent-based approaches to traffic simulation have shown that they are able to capture necessary details at an entity level, and can reproduce relevant realistic phenomena. Agents can be applied to represent drivers, vehicles, or other entities. They can be active, heterogeneous elements in an environment representing the road network, where they may exhibit arbitrary complex information processing and decision-making. Their behavior, specifically the agents involved in simulating movement, can be visualized, monitored, and validated at an individual level, leading to new possibilities for analyzing, debugging, and illustrating traffic phenomena [8, 22].

A specific application of agent-based approaches in traffic domain is modeling of auctions. As an example, Mashayekhi and List constructed an auction-based traffic controller where the agents learn a bidding strategy using Reinforcement Learning [27]. In [27] traffic streams are represented by agents (or *movement managers*). Movement managers participate in an auctions to determine which pair of safe movements (i.e., phases) should be permitted to pass through the intersection. Auctions take place only when there is the possibility of a collision occurring between two or more movements.

Dresner and Stone have put forward the idea of an advanced urban road traffic infrastructure that lets for autonomous guided vehicles to individually reserve space and time at an intersection from *intersection managers* so as to be able to safely pass through the intersection [16].

Vasirani and Ossowski have also demonstrated that a distributed, market-inspired, mechanism can be developed for the management of an urban road network, where drivers trade with the infrastructure agents in a virtual marketplace, purchasing reservations to cross intersections [43].

Bazzan and Klugl [8] provide an overview of agent-based technology for transportation systems in more detail. In summary we can conclude that, few to none studies tackle the intersection management problem from the norms viewpoint.

2.4 Conclusions

In this chapter we provided a short necessary background, and briefly reviewed the state of the art of normative multiagent systems. In the literature, most of the existing researches focus on the closed systems (e.g., Graph structure) and no computational framework has been proposed for norm generation in open multiagent systems. The assumption of a closed system is rarely appropriate in multiagent systems and most MASs require and presuppose openness. Also, we provided a brief literature review regarding the application of normative multiagent systems in transportation domain. Although this domain is very critical and important domain of interest for researchers and for sure is a good example of open MASs, there is relatively few works who studies this domain of interest from the view point of open normative MASs. So, this dissertation is also novel in pushing the frontier of open normative multiagent system in transportation domain.

Chapter 3

Framework and Methodology

3.1 Introduction

In the previous chapter we claimed that the norm emergence problem within an open multiagent system, still remains an open research problem. Considering this background, this chapter is focused on three main important subjects. First, we start with section 3.2 in which we provide some definitions and terms to better understand the problem. In section 3.3, we present the computational architecture of the both proposed frameworks for regulation of open multiagent systems. In section 3.4, the workflow of each framework is covered.

3.2 Definitions

In this section, we provide some definitions to better understand the problem and the architecture of both proposed frameworks.

Definition 3.2.1. *(Integrity). Integrity refers to a hard requirement in the domain of* *interest (e.g., Collision in traffic), which is typically not allowed to happen in a multiagent system.*

Definition 3.2.2. *(Conflict). Conflict refers to a potential undesirable situation (e.g., An unsafe move in traffic). Even though it is detrimental for achieving coordination. A multiagent system typically allows its members to proceed with (even though it is undesirable) so as to support greater autonomy.*

Definition 3.2.3. *(Conflicting agents). The set of agents which are involved in a conflicting state.*

To better understand the definitions introduced so far, Figure 3.1 shows a simplified version of a traffic intersection understood as a multiagent system. In the Figure 3.1, the intersection is composed of four lanes, modeled as a 6×6 grid, with three car agents (car agent a, car agent b, and car agent c) passing through the intersection. Available actions to the car agents are $Ac = \{Go; Stop\}$. Specifically, action Go means that a car moves forward towards its target direction at a constant speed of one cell per move; and action Stop means that a car stays stopped at its location. State s_1, shows a conflicting state between car agent a and b (i.e., Conflicting agents). In sate s_1, both car agent a and b decide to Go which leads to an unsafe state (s_2). Car agent c's default action was to Go (since it does not sense any conflict). In sate (s_2), both car agent b and c decide to Go which leads to an integrity violation (Collision in intersection) in State s_3. Car agent a's default action was to Go.

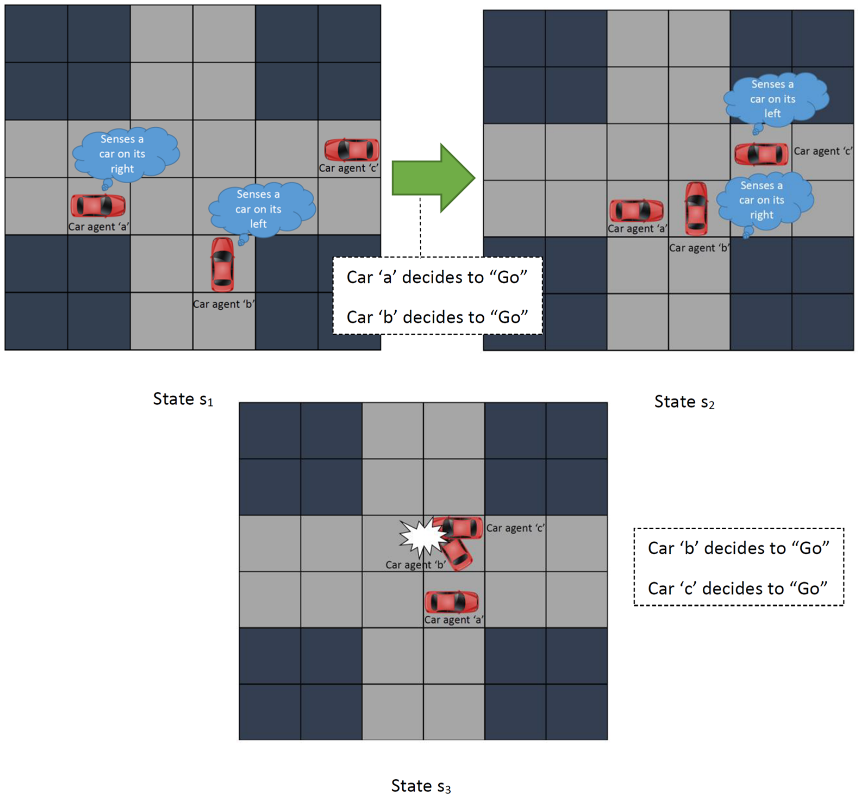


Figure 3.1: *Conflict*, and *integrity* violation among cars in a traffic intersection understood as a MAS. State s_1, shows a conflicting state between car agent a and b (i.e., *Conflicting agents*). State s_3 shows an integrity violation (i.e., Collision) between car agent b and c.

Definition 3.2.4. *(Law). Hard requirements on the behavior of members in a multiagent system. Think of a law as being realized in the technical infrastructure of MASs. For example, a locking mechanism implements a law prohibiting concurrent access to a resource*

Definition 3.2.5. *(Norm). Soft requirements on the behaviors of the members in a multiagent system. Think of a norm as being realized in the social architecture. For example, a norm would be for users to avoid a resource when others are working on it.*

Definition 3.2.6. *(Normative system). Set of active norms and laws (possibly empty) in a multiagent system.*

Definition 3.2.7. *(Norm and Law Representation)* For norm and law representation, we adopt grammar G from García-Camino et al. [17]. This grammar defines a norm or law as:

$$Norm ::= \langle antecedent, consequent \rangle$$

$$Law ::= \langle antecedent, consequent \rangle$$

$$antecedent ::= \phi$$

$$consequent ::= \Theta(ac)$$

$$\Theta := may \mid per \mid obl \mid prh$$

where *antecedent* constitutes a representation of parts of the current state of system (i.e., ϕ) which, if they hold, will cause the *consequent* to be initiated. *Consequent* is composed of Θ, which is a deontic operator applied on an action (i.e., ac). In the deontic operator set, *may* indicates a may (neutral deontic), *per* indicates a permission (also neutral deontic), *obl* indicates an obligation, and *prh* indicates a prohibition.

Next we present the architecture of both proposed frameworks.

3.3 Architectures

3.3.1 Silk

Figure 3.2 shows Silk's architecture: we explain its main components along with the associated algorithms below.

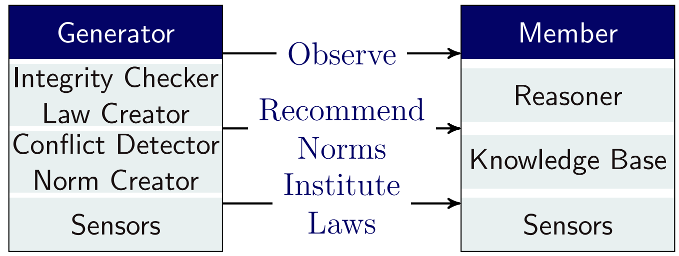


Figure 3.2: The Silk architecture in schematic terms.

As is illustrated in the figure, Silk's architecture is composed of two kinds of agents, namely, *generator* and *member*. Each agent has its own special core elements.

The *generator* is in charge of monitoring the system using its sensors, and check for integrity violations, and conflicts. The core elements of the *generator* are:

- *Sensors*. The generator has several sensors that provide a partial view of the multi-agent system environment. Each sensor monitors agents' interactions and generates descriptions of what it perceives. Perception of a "specific" multiagent system, needs domain-dependent data. So, sensors need a domain-dependent function to map the state of a specific MAS to description of the state in terms of the language of MAS. The language of MAS used here is first-order logic.

- *Integrity Checker and Law Creator*. Respectively, these two technical components of the generator, are designed to detect integrity violations and to provide hard constrains (i.e, Laws) to prevent them.

- *Conflict Detector and Norm Creator*. Respectively, these two technical components of the generator, are designed to detect conflicts and to provide soft constrains (i.e, Norms) to resolve them.

The other main element of Silk's architecture, is its *members*, which are participants in the MAS. The core elements of a member are:

- *Sensors*. Which provide local views of the state of the system that the agent is part of in terms of the language of the agent (i.e., First-order logic).
- *Knowledge Base*. The knowledge base component keeps information shared by all other components of the member. Think of it as the global memory for all the components of the member, and can be used to store information such as existing norms and laws within the multiagent system.
- *Reasoner*. Reasoner is the main functional component of a member which governs the decision making process for the member such as deciding what norms to follow, or what actions to take.

In this dissertation norms are intended to avoid conflicts. In particular, the proposed frameworks are not aimed to avoid any possible conflict from the very start. That would be impossible in an open MAS, since it would need to know all the possible conflict that may rise. Instead, our proposed frameworks detect when conflicts arise at runtime. Then, they synthesize norms aimed at avoiding such conflicts in the future. The proposed frameworks can be utilized by allowing for an initially empty normative system, or by considering an initial normative system with norms/laws.

Next, we describe the workflow of both proposed frameworks.

3.3.2 Cha

The architecture of an agent in Cha is shown abstractly through its main components in Figure 3.3. The components of Cha agent is similar to the Silk's architecture except some of the capabilities of the generator including, *Conflict Detection* ,and *Norm Creation*, are transferred form the generator to the agent itself. The transferring of these capabilities is necessary to make the framework decentralized. Because of the similarity of the components, for brevity, we omit describing the components.

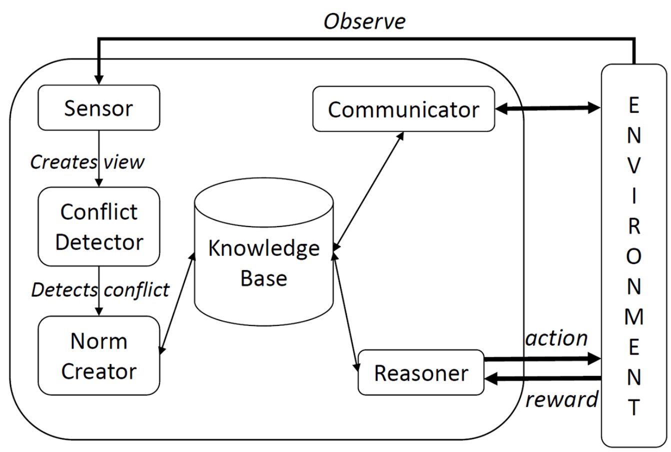


Figure 3.3: The Cha architecture in schematic terms.

3.4 Workflow

In this section, we provide the algorithms which explain the workflow of the both proposed frameworks. In Section 3.4.1 the workflow of norm generation and interaction protocol in Silk's framework are explained. In Section 3.4.2, the workflow of norm generation in Cha's framework is described.

3.4.1 Workflow in Silk

Algorithm 3 shows how the generator generates norms and laws. Its required inputs are: (1) simulation time period T; (2) a list of *views* $[\langle v_t^0 \rangle, ..., \langle v_t^m \rangle]$ where v_t^i is the i-th view of the particular system state at time t; (3) a grammar G; (4) a function f_{conflict} to detect potential conflicts; and (5) a function $f_{\text{violation}}$ to detect potential integrity violations.

Here, a view is a projection of the system state at a given time. For example, in a traffic scenario a possible view is the traffic state in one part of the road at time t. We express views using predicates describing the system state. Each member has a limited view and can see parts of its environment whereas the generator can see the entire environment.

For each view v_t^i, the generator predicts the next view and detects whether a conflict or an integrity violation would occur based on current laws and recommended norms. For each detected conflict and integrity violation, it generates new norms and laws.

In Algorithm 3, the function $f_{\text{violation}}$ (Line 2) uses the current view (v_t^i) to predict if an integrity violation would occur in the next timestep. If so, the *createL* operator (Line 5) generates a law to prevent that integrity violation. Likewise, the function f_{conflict} (Line 6) uses the current view (v_t^i) to predict if a conflict may occur in the next timestep. If so, the *createN* operator (Line 9) generates a norm. The creation of a law requires the process of selecting an integrity violation source. During such operation, generator chooses an agent as responsible for integrity violation. The assumption here would be, whoever agent who already has accepted the recommendation form the generator to avoid the conflict, and is involved in an integrity violation, would be responsible for the integrity violation.

Both createL and createN place the current view (i.e., v_t^i) as the antecedent (*antecedent* in Definition 3.2.7) of the new norm or law and employ an appropriate deontic operator for the consequent. Specifically, a recommended norm uses permission (per), which indicates neutrality, whereas a law uses prohibition (prh), which indicates a hard restriction. A member's policy (i.e., mapping of states to actions) may end up adopting or rejecting a recommendation, thereby treating the recommended norm as if it were an obligation or prohibition, respectively.

Norm and law generation follows the assumption that if a law or norm would avoid an integrity violation or conflict in the present state, then it would do so in the similar or same future state. Figure 3.4 shows two similar situations which lead to integrity violations (i.e., Collisions) in the traffic domain. Creating a law for each of the situations would preclude integrity violation in the similar case.

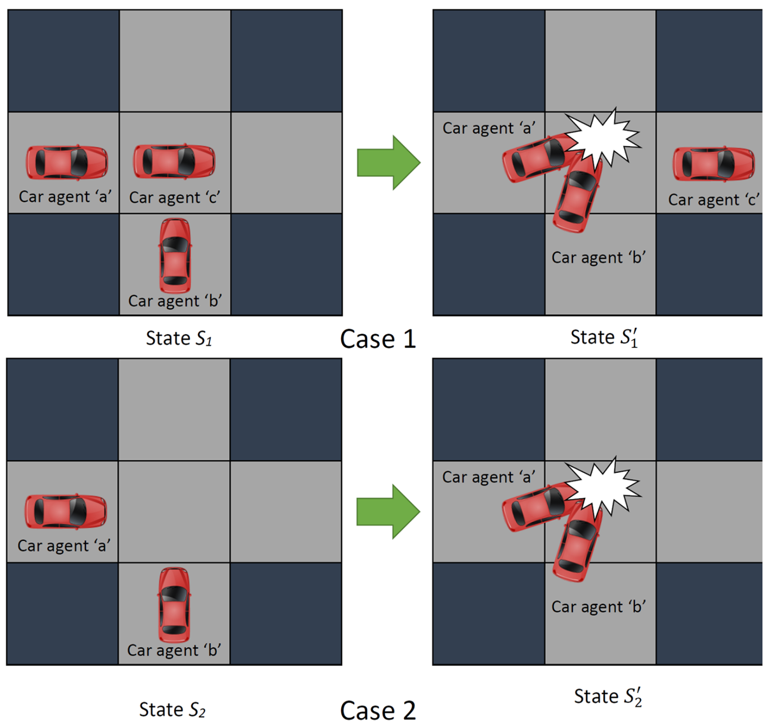


Figure 3.4: Two similar situations which lead to integrity violations.

Each member has a private payoff matrix and a learning algorithm to reason about

Algorithm 1 Norm and law generation in Silk

function Π(views, G, f_{conflict}, $f_{\text{violation}}$, T)
    violations $\leftarrow$ violationDetection(v_t^i, $f_{\text{violation}}$);
    for all v_t^i in violations **do**
        if there is no law to prevent the violation **then**
            law $\leftarrow$ createL(G, v_t^i);
        end if
    end for
    conflicts $\leftarrow$ conflictDetection(v_t^i, f_{conflict});
    for all members in conflicts **do**
        if there is no norm to prevent the conflict **then**
            norm $\leftarrow$ createN(G, v_t^i);
        end if
    end for
end function

its actions. Members apply reinforcement learning to learn behaviors based on their interactions with other members. A member knows neither the identity of the members it interacts with nor their payoff matrices, but can observe their actions.

Basically members use Q-learning algorithm proposed by Watkins [45] for learning norms. Briefly, the core of the Q-learning algorithm is a Q-table and an algorithm/formula for updating the Q-table and choosing actions. A Q-table *Q(s, a)* is a matrix indexed by state s and action a, which is the expected discounted reinforcement of taking action a in state s. At each time, an agent is assumed to be in a certain state s, and it chooses an action a according to the Q-table to interact with the environment. Then the agent receives a reward r from performing action a and observes a new state s'. Here, think of utilities as the Q values.

Members use an ϵ-greedy strategy [40] in order to explore the state and action environment. That is, a random action is selected with probability (ϵ) and the action with the highest utility is chosen with probability (1-ϵ). The exponential function (e^{-Em}) is used to estimate ϵ, where E is a constant and m is the number of times members of the same type experience the same situation. Under this approach, the members mainly explore initially, as they have no prior knowledge to exploit, and gradually increase their extent of exploitation.

Each type of member has two utilities for a norm, for its violation and fulfillment, respectively. Let $U(n,t,a)$ and $U(n,t-1,a)$ be a member's utilities of norm n for action a at times t and $t-1$, $0 \le \alpha \le 1$ the learning rate, and $r(n,t,a)$ a reward from the payoff matrix. The utilities are updated as:

$$U(n,t,a) = (1-\alpha) \times U(n,t-1,a) + \alpha \times r(n,t,a). \tag{3.1}$$

Algorithm 2 shows how the members interact. The generator broadcasts laws and norms. Each member reasons whether to violate or respect each recommended norm and selects an action (Line 3). The members' payoff depends upon their joint actions. They pass on their experiences, represented as average utilities associated with states (antecedent of norms) and actions (violation or fulfillment of norms), to incoming members of the same type (Line 6). In other words, an incoming member can obtain experience from any member who has experience to share: those are members who have been through the current intersection. Sharing experience is thus similar to the situation in human society where old people pass on their experience to young people.

3.4.2 Workflow in Cha

Algorithm 3 shows the pseudo-code of the workflow employed by Cha agents. The algorithm is composed of the norm generation, reasoning, updating, and sharing phases.

Norm generation phase. The agent first uses its sensors to perceive the environ-

Algorithm 2 Interaction protocol

```
1: generator announces laws and norms;
2: for all members in Conflict do
3:     member selects and performs an action;
4:     compute joint action rewards from payoff matrices;
5:     member updates the utility;
6:     member shares its experience with new members of
       the same type;
7: end for
```

ment and construct a local *view*—a projection of the system snapshot to what the agent can observe (Line 2). For example, in a traffic scenario, a possible view is the traffic in one part of the road at time t. Like Silk's framework in Cha, we express views using predicates representing system state. For each view v_t^i, the agent anticipates the next view, and checks whether a conflict would happen using the function f_{conflict} (Line 3). Normative set (*NS*) is initialized to the set of existing norms (Line 4). If no norm in *NS* can prevent the conflict, *createN* generates a norm based on the current view (v_t^i) as its *antecedent* (in Definition 3.2.7), and initially with a *may* operator applied to the action (Line 6). If so, norm is inserted into *NS* (Line 7).

Again, norm generation follows the assumption that if a norm would avoid a conflict in the present state, then it would do so in the similar or same future state.

Reasoning phase. Each agent retrieves its policy, selects an action, and performs that action (Lines 9-11). Each agent applies reinforcement learning (Q-learning algorithm described earlier), and ϵ-greedy strategy for exploration and exploitation of the environment. It selects a random action with probability (ϵ) and an action with the highest utility with probability (1-ϵ). It estimates ϵ via the exponential function (e^{-Em}), where E is a constant and m is the number of times the same situation has arisen (Like Silk's framework). Because of the exponential decay of ϵ, the agents initially primarily explore,

Algorithm 3 Pseudocode of the algorithm executed by Cha agents

for $ticks = 1$ **to** *SimulationLength* **do**
// **Norm Generation Phase:**
  $v_t^i \leftarrow$ perceiveEnvironment (*Sensors*);
  Conflict $\leftarrow$ conflictDetection($v_t^i, f_{\text{conflict}}$);
  NS $\leftarrow$ normAcquisition (knowledge base);
  if there is no *norm* in *NS* to prevent *conflict* **then**
    norm $\leftarrow$ createN(G, v_t^i);
    NS $\leftarrow$ *NS* + *norm*;
  end if
// **Reasoning, Updating, Communicating:**
  for all agents in *Conflict* **do**
// **Reasoning:**
    $P \leftarrow$ getPolicy();
    $a \leftarrow$ actionSelection(P, ϵ-*greedy*);
    execute (a);
// **Updating:**
    $R \leftarrow$ jointActionReward(*Payoffs*);
    updatePolicy (R, a);
// **Sharing:**
    sharePolicy();
  end for
end for

as they have no prior knowledge to exploit, and then gradually increase their extent of exploitation.

Updating phase. Based on the joint actions of conflicting agents, immediate rewards are obtained according to the payoff matrix (Line 12) and utilities are updated (Line 13) via the following equation:

$$U(s,t,a) = (1-\alpha) \times U(s,t-1,a) + \alpha \times r(s,t,a). \tag{3.2}$$

Here, $U(s,t,a)$ and $U(s,t-1,a)$ stand for the utility of the state s for the action a at the time t and $t-1$ accordingly; $0 \leq \alpha \leq 1$ is a parameter to trade off exploitation (of the utility obtained so far until time $t-1$) with exploration (of the utility reward obtained from time $t-1$ to t) in the cumulative utility; and $r(s,t,a)$ is a reward based on the payoff matrix.

Sharing phase. This step is also similar to Silk's framework. Each agent passes on its experiences, i.e., average utilities associated with states observed and actions performed in them, to incoming agents of the same type (Line 14). In other words, an incoming agent can obtain experience from any agent who has experience to share. Sharing experience is similar to the situation in human societies where elderly individuals pass on their experience to younger generations. Sharing experience is also related to Axelrod's [7] concept of *learning from observation*, whereby agents learn from each others' interactions.

3.5 Conclusions

In the previous chapter we claimed that the norm emergence problem within an open multiagent system, still remains an open research problem. In this chapter, we presented the architecture and framework of two abstract computational frameworks for regulation of open normative multigent systems; namely Silk and Cha.

Silk's framework which is a hybrid framework (top-down and bottom-up) is composed of two main components; namely, *generator* and *member*. The core elements of the generator are its Sensors, Integrity Checker, Law Creator, Conflict Detector, and Norm Creator. The generator is in charge of monitoring the system using its sensors, and check for integrity violations, and conflicts. The other main element of Silk's architecture, is its members, which are participants in the MAS. The core elements of a member are its Sensors, Knowledge Base, and Reasoner. In this framework, the generator institutes laws, i.e., hard integrity constraints that ensure certain undesirable events do not occur. It recommends norms, i.e., soft recommendations that if respected would lead to conflict avoidance.

Second proposed framework is Cha, which is a decentralized (or bottom-up) framework. In this framework we equipped members with more capabilities (i.e., Conflict detection and Norm Creation), so there is no central agent. The core elements of Cha agents are its Sensors, Conflict Detector, Norm Creator, Knowledge Base, Reasoner, and Communicator. In Cha, we only have considered soft regulation since that is more challenging aspect. In Cha each member itself anticipate potential conflicts and create norms if necessary.

Chapter 4

Domain of Study

4.1 Introduction

In this chapter we introduce a scenario that can be modeled as a multiagent system and in which norms can be used to regulate the activities of its members. We have chosen the transportation domain as the domain of study to evaluate the proposed frameworks.

Traffic networks are open systems in which their participants have diverse goals and motivations, and their behaviors are unpredictable. Also, the traffic setting is extremely dynamic. Due to these characteristics, many applications in this domain can be adequately modeled as multiagent systems. Transportation domain as a complex and important domain, has attracted some attention from multiagent systems community [1, 16], but not from a normative multiagent point of view.

Although challenging, combination of social norms and multiagent systems seems very natural to be applied to regulate such a system that could potentially lead to efficient coordination in a traffic network.

In the following, we introduce the traffic scenario used in this dissertation which is an *intersection*. An intersection is a prominent feature of existing infrastructure, where roads cross each other and the need to coordinate access to the intersection is vital for preventing collisions and increasing efficiency of the whole network. Intersection considered here, is a similar version of the commonly used traffic intersection scenario by other researchers [16]. The modeled traffic scenario is understood as a multiagent system, and is used throughout the rest of this dissertation.

4.2 Scenario in Silk

For concreteness and simplicity of exposition, we evaluate Silk on an intersection understood as an open multiagent system where members continually arrive and depart. The *intersection manager (IM)* is the generator and each *car* is a member.

As Figure 4.1 shows, we map an intersection and its vicinity to a grid of cells. Traffic flows in each of the four directions. The *(intersection) zone* in the middle is composed of four cells. Cars travel along the grid at the speed of one cell per timestep. Cars can randomly turn left or right at an intersection.

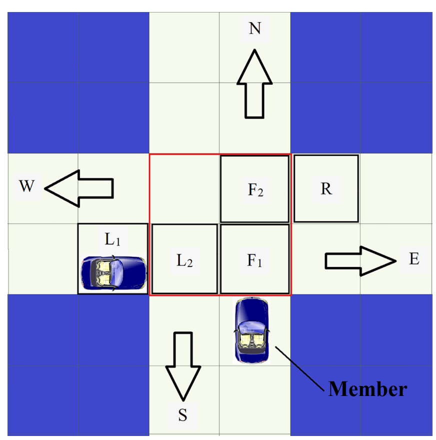


Figure 4.1: An intersection and (intersection) zone.

A conflict arises when cars moving in orthogonal directions are about to enter the intersection zone at the same time. These conflicting cars can observe each other's actions but not each other's internal policies. An integrity violation corresponds to a collision: when two cars occupy the same cell. We consider the possibility of collision only within the zone.

Imagine four cars are about to enter the zone, one from each direction. If each proceeds they would occupy distinct cells: hence the IM predicts no integrity violation and produces no law. But it anticipates a conflict (being in the zone at the same time, which we understand as unsafe here), creates a norm, and recommends it to the cars. Each car decides whether to adopt or reject the recommendation. If all reject it, a conflict would occur. Conversely, if all accept it, no progress would occur. We interpret Go as the violation of a recommended norm generated by the IM and Stop as the fulfillment of the generated recommended norms by the IM.

The *antecedent* of a norm or a law (Definition 3.2.7) refers to the values of five cells identified in Figure 4.1 with respect to the car entering from below: L_1, L_2 to its left, F_1, F_2 to its front, and R to its right. These cells constitute the local view of that car. Our grammar can specify cells with one of six values: $>$ (car heading east), $<$ (car heading west), $\vee$ (car heading south), $\wedge$ (car heading north), $-$ (nothing), and $*$ (car in any heading).

An example norm is $L_1(>)$, per(Go): if a car perceives another car in cell L_1 heading east ($>$), then it is permitted to go (since per is a neutral deontic, the car also has the option to stop). Figure 4.1 shows a relevant case where L_1 is occupied but other cells are empty. An example of a law is $L_2(>)$, prh(Go): if a car perceives another car in cell L_2 heading east ($>$), then it is prohibited to go.

As pointed out earlier, this dissertation lies into interaction-based model and specifically game theoretic approaches. In these models, the rules of the game and an agent's decisions yield to a specific payoff matrices.

Tables 4.1 and 4.2 show the payoff matrices used in our simulation for Silk's framework. These matrices characterize a *moderate* society, where there are higher rewards (1) for north- and southbound cars to Go than to Stop, and (2) for eastbound and westbound cars to Stop than to Go. Table 4.1 shows the payoff matrix for an east- or westbound car that perceives two cars on its left (L) and right (R), traveling north or south. Table 4.2 gives the payoffs for a north- or southbound car with eastbound or westbound cars on its left and right. When a car conflicts with only one other car, its payoffs are projections of the matrices in Tables 4.1 and 4.2 to the first and fourth columns where both Go or both Stop.

Table 4.1: East-west car's payoffs given two conflicting cars.

	$L_{\text{Go}}, R_{\text{Go}}$	$L_{\text{Stop}}, R_{\text{Go}}$	$L_{\text{Go}}, R_{\text{Stop}}$	$L_{\text{Stop}}, R_{\text{Stop}}$
Go	-5.0	-2.0	-2.0	0.5
Stop	4.0	1.0	1.0	0.0

Table 4.2: North-south car's payoffs given two conflicting cars.

	$L_{\text{Go}}, R_{\text{Go}}$	$L_{\text{Stop}}, R_{\text{Go}}$	$L_{\text{Go}}, R_{\text{Stop}}$	$L_{\text{Stop}}, R_{\text{Stop}}$
Go	-5.0	-2.0	-2.0	4.0
Stop	0.5	0.5	0.5	0.0

4.3 Scenario in Cha

Like Silk, we evaluate Cha on an intersection understood as an open multiagent system, where agents continually arrive and depart. As Figure 4.2 illustrates, we map an intersection and its vicinity to a grid of cells. Traffic flows in each of the four directions. The *intersection zone (izone)* in the middle is composed of eight cells (highlighted in the Figure; ignore the dead cell in the center). Unlike the intersection considered in Silk's framework, the one used in Cha provides more storage for the car agents in the shared resource (izone). This is because we are only considering soft regulation in the Cha's framework and giving more *freedom* to the agents, and to make it reasonable for the car agents to Stop in the izone. However, note here that, Silk's framework can be applied in this traffic scenario as well.

Similar to the intersection modeled in Silk's framework, cars travel along the grid at the speed of one cell per time-step. A car may continue on a straight path or randomly turn left or right in an intersection (but not against the flow of traffic). A car may move only into an empty cell.

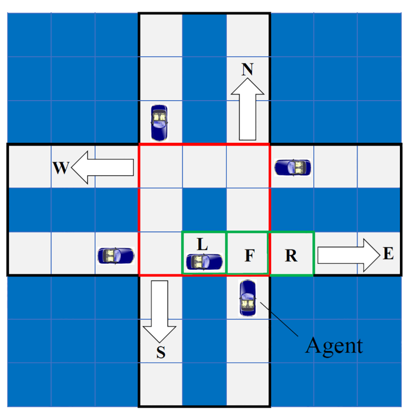


Figure 4.2: Center of the modeled intersection.

A conflict arises when agents moving in different directions are about to occupy the same izone cell at the same time. Such agents are called conflicting agents and opponents of each other. We consider the possibility of conflict only within the izone. Opponents can observe each other's actions but not internal policies.

If there is no norm in a conflict situation, each agent creates a norm, initially with a neutral deontic operator (i.e., *may*). While interacting in the system and gaining experience, the deontic operator will move to the extreme of the ranges which are obligation (i.e, *obl*) or prohibition (i.e., *prh*). As mentioned earlier, each agent applies RL to find its optimal policy. A norm emerges when a majority of agents adopt compatible policies. Ideally, we would like to have a norm like yielding to the agent on the right or left.

In Cha, the *antecedent* (Definition 3.2.7) of a norm refers to the values of three cells identified in Figure 4.2 with respect to the car entering from bottom of the intersection (indicated as *Agent* in the Figure): L, to its left, F, to its front, and R to its right. These cells constitute the local view of that car. Our grammar can specify cells with one of six values: $>$ (car heading east), $<$ (car heading west), $\vee$ (car heading south), $\wedge$ (car heading north), $-$ (nothing), and $*$ (car in any heading). An example norm is $L(>)$, prh(Go), which means that if a car perceives another car in cell L heading east ($>$), then it is prohibited to proceed. Figure 4.2 shows a similar case where L is occupied but the other cells are empty.

Table 4.3 shows the payoff matrix used in our simulation for Cha's framework. The payoff matrix characterize a *selfish* society. The highest positive payoff refers to the situation where one agent chooses a selfish action (i.e., Go), and the conflicting agent chooses the unselfish action (i.e., Stop). The highest negative payoff is when both agents are selfish (i.e., both decide to Go), hence a collision can happen. The payoff of an unselfish action is denoted by CU (cost of being unselfish) and is calculated by equation 4.1:

Table 4.3: Payoff matrix

	Go	*Stop*
Go	–6.0, –6.0	5.0, CU
Stop	CU, 5.0	CU, CU

$$CU = \max(f(se), -6) \tag{4.1}$$

$$f(se) = -d^p \tag{4.2}$$

Equation 4.1 is composed of a function which depends on the current state of the environment, i.e., *se*. In the traffic domain *se* corresponds to "*delay*" (i.e., d). Here in our simulation, a power function is used to calculate it. Intuitively, when the delay is high, the penalty for unselfishness (i.e., stopping) should be higher. The exponent p is set to 1.1. The delay refers to the total delay time agent experienced so far. It is determined by accumulating the number of stops until the agent needs to decide about entering the izone. Note that, because of the max function in Equation 4.1, the payoff of stopping cannot be less than the payoff of the (*Go, Go*) action pair, which is the worst-case scenario.

Note here that introducing such a payoff matrix itself is a contribution to the state of the art. Since unlike other studies which use static payoffs, which is not very reasonable, offers dynamic payoffs for members in the game.

4.4 Conclusions

Transportation domain as a complex and important domain, is a good example of an open multiagent system, in which their participants have diverse goals and motivations, and their behaviors are unpredictable. Also, the traffic setting is extremely dynamic. Modeling traffic as a normative multiagent system seems natural and promising to potentially have a more efficient and resource-saving network. However, due to existence of variety of dynamics in traffic, it is also challenging to regulate such a system from a norm point of view. This dissertation is among the few studies who have tackled regulation of traffic from a norm point of view.

Considering this background, in this chapter we introduced the traffic scenario which has been used in both proposed frameworks along with the rules of the game and characteristics of the societies (i.e., Payoff matrices) studied in this dissertation. The modeled intersections used in this dissertation and introduced here, are similar versions of the commonly used traffic intersection scenario by other researchers [16], which are understood and modeled as open multiagent systems in this dissertation.

Chapter 5

Experimental Results

5.1 Introduction

In this chapter we present the results of our experiments in relation to our research hypotheses presented in the Chapter 1. First we present the results of Silk's framework, and second we present the results of our experiments for Cha's framework in the transportation domain scenario described in Chapter 4.

5.2 Silk's Experimental Results

5.2.1 Simulation Setup in Silk

As stated before our simulated traffic system is a simulation environment in Repast [32, 28]. We consider an intersection where each lane has 20 cells: a total of 76 cells with four cells in the zone (Figure 4.1). We initialize utilities to zero at $t = 0$ and set α in formula (3.1) to 0.2. We set $E = 0.05$ in the exponential function (e^{-Em}), which is used for the ϵ-greedy approach. The data shows average over 1,000 trials.

5.2.2 Norm Emergence

Table 5.1 shows the norms (first three rows) and law (last row) that IM generates. Left-Right, Right, Left refer to the kind of conflicts they address. Laws come as prohibitions and norm recommendations as permissions. To verify whether conflict-resolving norms can emerge in Silk, we compute the change in average utility for the society for these norms.

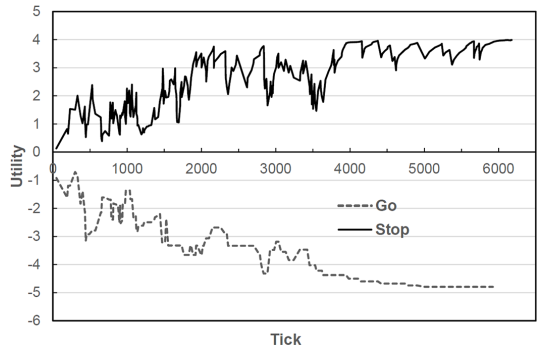
Table 5.1: Generated norms and laws.

	Antecedent	**Modality**
Left-Right	$L_1(>) \wedge L_2(*) \wedge F_1(*) \wedge F_2(*) \wedge R(<)$	$\mathsf{per}(\text{Go})$
Right	$L_1(-) \wedge L_2(*) \wedge F_1(*) \wedge F_2(*) \wedge R(<)$	$\mathsf{per}(\text{Go})$
Left	$L_1(>) \wedge L_2(*) \wedge F_1(*) \wedge F_2(*) \wedge R(-)$	$\mathsf{per}(\text{Go})$
Collision	$L_1(*) \wedge L_2(>) \wedge F_1(*) \wedge F_2(*) \wedge R(*)$	$\mathsf{prh}(\text{Go})$

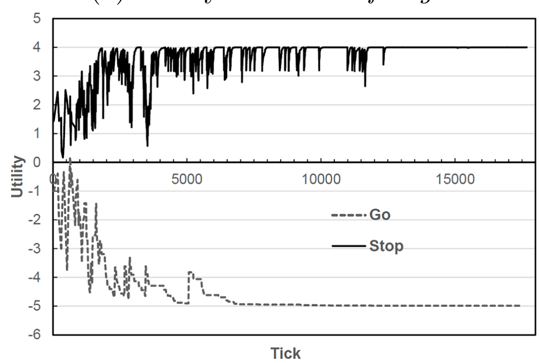
According to their directions of travel, the cars converge to a policy on whether to adopt or reject a norm recommendation. As Figure 5.1 shows, eastbound cars gain utility by stopping in the case of a conflict (whether left, right, or both). All these members acquired the corresponding policy by $t = 12,500$. Figures 5.2 shows similar results for northbound cars. (The wobbles in utility are due to random selection of actions according to ϵ-greedy strategy). Trends for west- and southbound cars are shown in Figures 5.3, and 5.4; they have the same outcomes as east- and northbound cars, respectively.

Table 5.2 summarizes the converged behaviors. Each row describes the matching behavior of cars heading in the specified directions. For example, Row 1 describes what is learned in the conflict situation of Figure 4.1. The learned behavior is for the eastbound car to yield (Figure 5.1b) and the northbound car to go (Figure 5.2c). That is, referring to Table 5.1, eastbound cars learn to accept IM's recommendation, Right, whereas northbound cars learn to reject IM's recommendation, Left.

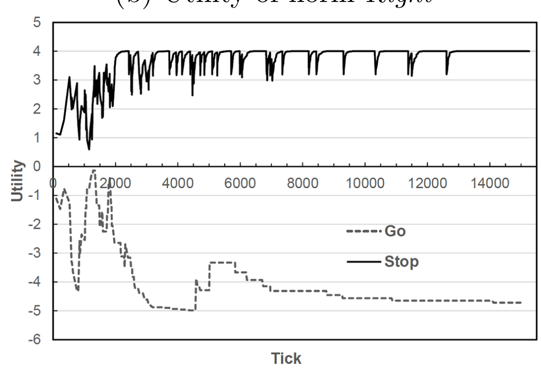
The obtained results prove that the hypothesis *Conflict* holds in the Silk's framework.



(a) Utility of norm *Left-right*

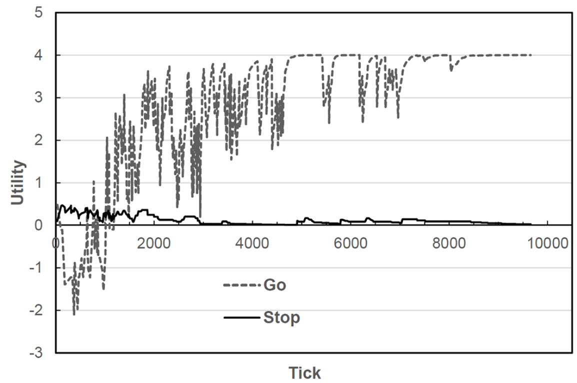


(b) Utility of norm *Right*

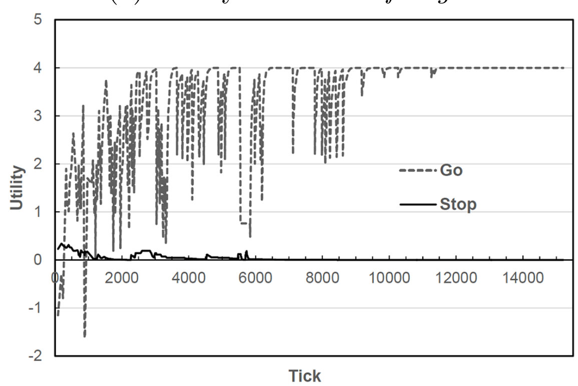


(c) Utility of norm *Left*

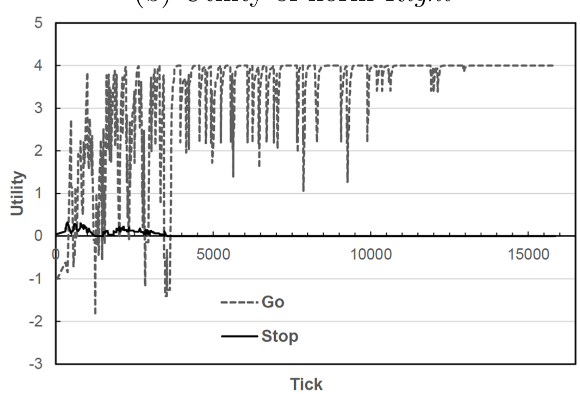
Figure 5.1: Utility of norms for eastbound cars.



(a) Utility of norm *Left-right*

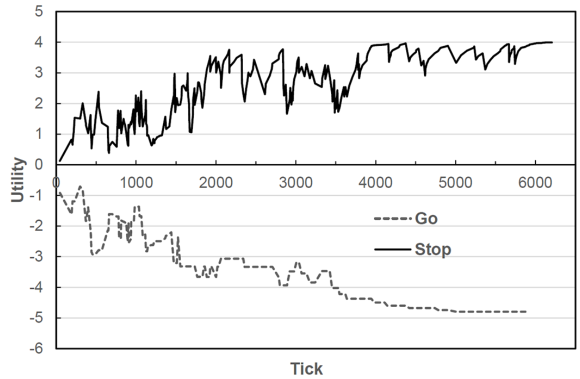


(b) Utility of norm *Right*

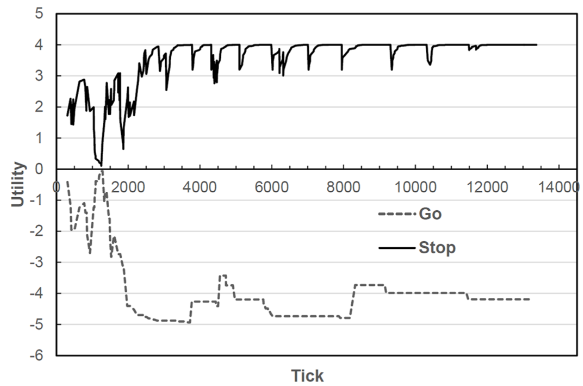


(c) Utility of norm *Left*

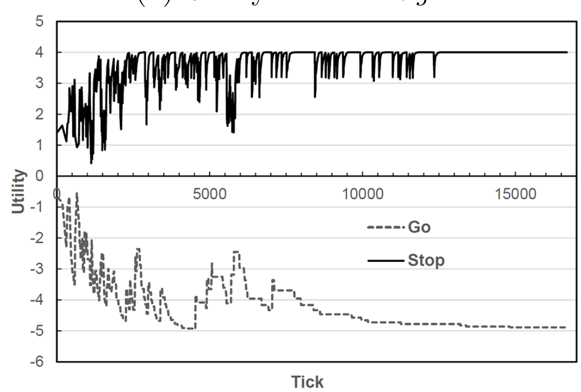
Figure 5.2: Utility of norms for northbound cars.



(a) Utility of norm *Left-right*

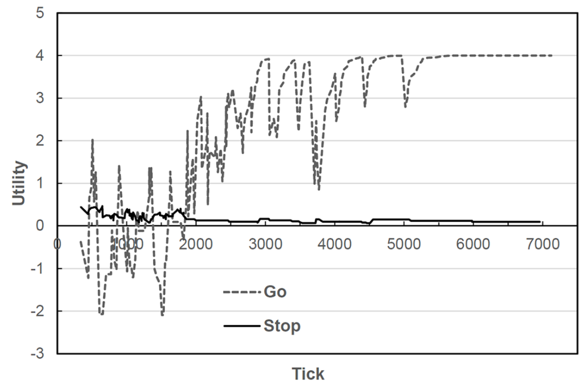


(b) Utility of norm *Right*

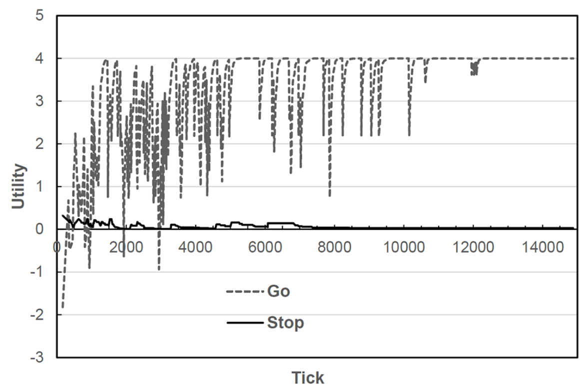


(c) Utility of norm *Left*

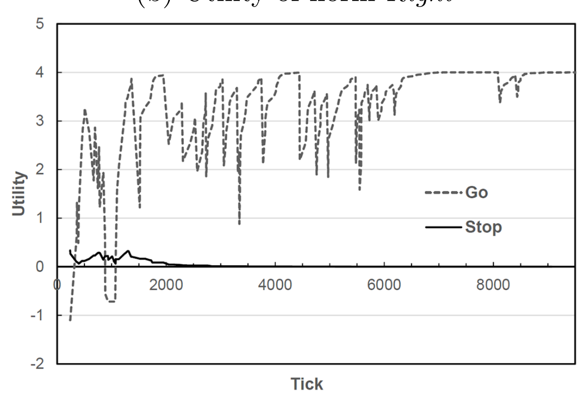
Figure 5.3: Utility of norms for westbound cars.



(a) Utility of norm *Left-right*



(b) Utility of norm *Right*



(c) Utility of norm *Left*

Figure 5.4: Utility of norms for southbound cars.

Table 5.2: Converged behaviors for cars in different directions.

#	East	West	North	South
1	Yield to right		Go	
2	Yield to left			Go
3		Yield to left	Go	
4		Yield to right		Go
5	Yield to right	Yield to left	Go	
6	Yield to left	Yield to right		Go
7	Yield to both		Go	Go
8		Yield to both	Go	Go
9	Yield to both	Yield to both	Go	Go

5.2.3 Fairness of the Framework

In Table 5.2, the system converges to the behavior in which the east-west cars stop and north-south cars go. This may not be *fair* [14] to the east-west cars: in heavy north-south traffic, they may be forced into long waits. Therefore, we propose that the generator monitors performance and intervenes if necessary to improve fairness. For example, in our traffic scenario, the IM would monitor delays suffered by the cars and act accordingly. If some cars experience high delays (i.e., Five ticks here), indicating that they have learned the behavior to stop in a conflict situation, the IM would intervene to let those cars pass the zone by temporarily stopping their conflicting cars from entering the zone. The IM can rely upon the law mechanism to carry this out. We accomplish this by reversing the converged behavior for one tick.

5.2.4 Social Performance Improvement in Silk

Now we compare Silk with a fully actuated control strategy [2] as an alternative way of managing traffic which was described in Section 2.2.6. In the implemented fully actuated strategy, north- and southbound approaches are paired to form the first phase, and east- and westbound approaches are paired to form the second phase.

To make the fully actuated control strategy as similar as possible to Silk, minimum green is set to one tick and the gap time is set to zero. When a gap out occurs for one phase, which makes the corresponding signals change from green to red, the signals for the other phase can turn green if there is demand (i.e., cars are detected). However, the next phase cannot be implemented until the zone becomes empty. Maximum green is considered five ticks. The described fully actuated logic corresponds to the learned behavior as in Table 5.2. Since cars can change direction and turn left, in order to prevent collision, we incorporate the law of Table 5.1 in the logic of the fully actuated controller. We compare Silk with and without the fairness property with traffic signals as well as with and without maximum green. Our metric is average travel time, calculated by adding expected travel time and average delay, where delay is the total number of stops for all cars in two successive ticks. Since both Silk and traffic lights are effective in preventing collisions, so it makes sense to compare them in terms of average travel time.

Table 5.3 shows the average travel time for Silk and the fully actuated controller. As shown, the improvement on travel time gained by using Silk is 7.4% without the fair property and 16.2% with fair property.

Table 5.3: Silk versus fully actuated control.

	Unfair/No max green		*Fair/Max green*	
	Expected	Average	Expected	Average
Silk	20	25.70	20	24.65
Fully actuated	20	27.75	20	29.41
Improvement		7.40%		16.20%

5.2.5 Selfish Society

We now investigate the effects of payoff matrices on the converged behaviors. To this end, and to compare with the moderate society we studied above, we study a *selfish* society, as characterized by Table 5.4. The payoff matrix shows that a member profits from selfishness unless everyone chooses the selfish alternative, in which case everyone loses. Table 5.4 represents the payoffs for an east-west or north-south car when it perceives two cars on its left and right. When a car conflicts with only one other car, its payoffs are projections of the above matrix to the first and fourth columns where cars from the other two orthogonal directions both go or both stop.

Table 5.4: East-west and North-South car's payoffs given two conflicting cars.

	$L_{\text{Go}}, R_{\text{Go}}$	$L_{\text{Stop}}, R_{\text{Go}}$	$L_{\text{Go}}, R_{\text{Stop}}$	$L_{\text{Stop}}, R_{\text{Stop}}$
Go	-5.0	-2.0	-2.0	4.0
Stop	4.0	1.0	1.0	0.0

We ran our simulation as before except that we use the above payoff matrices. About half the time (502 times out of 1,000 simulation runs), the north-south cars learn to go and the east-west cars learn to stop (i.e., yield), and in the other half of the time (498 times out of 1,000 runs of simulation), the reverse behavior emerges. Depending on which direction initially explore the go action or the stop action more, the population converges to one or the other behavior.

This outcomes shows that Silk promotes norm emergence even under radically different payoffs.

5.3 Cha's Experimental Results

5.3.1 Simulation Setup in Cha

Like Silk, simulation environment is developed in Repast [28, 32]. Here, we consider an intersection where each lane has 19 cells: a total of 72 cells with eight cells in the izone (Figure 4.2). Other parameters are set like Silk's framework. For completeness we repeat them here. We initialize utilities to zero at $t = 0$ and set α in equation (3.1) to 0.2. We set $E = 0.05$ in the exponential function (e^{-Em}), which we use for the ϵ-greedy approach. The data shows the average of 1,000 trials. Also, we consider that Cha has converged if the utilities associated with the actions are converged to asymptotic values.

5.3.2 Norm Emergence

In this section, we describe two experiments and present their results for norm emergence.

The first experiment is designed to test Hypotheses *Conflict*, and *Efficient-conflict*; that is, to check whether our framework can resolve conflict and lead to generation of efficient norms for conflict resolution. And, the second experiment is designed to test Hypothesis *Dynamic adaption*; that is, to check whether the proposed mechanism is able to reflect and react to the changes in a dynamic environment setting.

5.3.2.1 Static Setting

We start with a case where N-S has higher traffic comparing to E-W (i.e., 30% more). The converged norms are not known a priori. Based on the defined payoff matrix (i.e., Table 4.3), we would expect that N-S agents learn to Go and E-W agents learn to Stop. The results of this experiment are shown in Table 5.5. Figure 5.5 also shows the total number of collisions per tick of simulation. Since there are four points of potential conflict, maximum number of collisions can be four. Roughly speaking, after 20,000 ticks, number of collisions decreases dramatically, and we can say that car agents have learned norms of the intersection. As the converged norms shown in the Table, E-W cars have learned to Stop in conflicting situations with N-S cars (which has highest traffic and consequently higher cost of being unselfish). Since the N-S approach has higher traffic, the converged norms are efficient comparing to the reverse norms in which E-W cars learn to Go, and N-S cars learn to Stop. That is, the evidence to support Hypotheses *Conflict* , and *Efficient-conflict*.

Table 5.5: Converged norms.

	Precondition	**Modality**
East-and Westbound	$L(*) \wedge F(*) \wedge R(<)$	$prh(\text{Go})$
	$L(>) \wedge F(*) \wedge R(*)$	$prh(\text{Go})$
South-and Northbound	$L(*) \wedge F(*) \wedge R(<)$	$obl(\text{Go})$
	$L(>) \wedge F(*) \wedge R(*)$	$obl(\text{Go})$

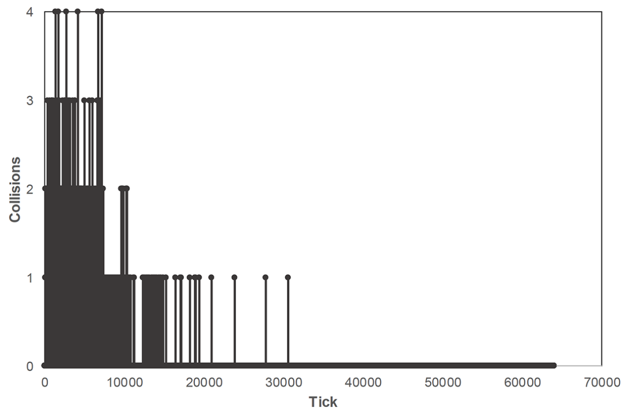


Figure 5.5: Total number of collisions per tick.

5.3.2.2 Dynamic Setting

We now study whether the system is sensitive to the changing environment. Specifically, would the norms learned in the static setting persist even if the traffic pattern changes? Preferably, the existing norms would decay and new norms would emerge.

Following the static setting example, if the E-W traffic increases dramatically relative to N-S traffic, we would expect new norms whereby E-W agents Go and N-S agents Stop. In order to study this, we look into the changes in average utilities. We start this experiment like static setting but will reverse the traffic pattern after convergence to see if the framework is capable of developing a new normative system in response to the changes in the environment.

Figure 5.6 shows the change in average utility for eastbound cars. By $t \approx 22,000$, the norm is for E-W cars to Stop in case of conflict (higher utility for Stopping than Going, shown in the box in the Figure). By $t = 40,000$, the traffic pattern is reversed, and by $t \approx 50,000$, the new norm is for E-W cars to Go in case of conflict. Figure 5.7 shows similar pattern for northbound cars. That is, initially the norm for N-S cars is to Go, and once the traffic pattern changes, the new norm is to Stop in conflicting situation with

E-W cars. Figures 5.8 and 5.9 show changes in average utility for west- and southbound cars: they have the same outcomes as east- and northbound cars, respectively.

Table 5.6 summarizes the results of this experiment. The results indicate support for Hypothesis *Dynamic-adaption*, namely, that Cha responds appropriately to its dynamically changing environment.

Table 5.6: Converged norms.

	Precondition	**Modality**
East-and Westbound	$L(*)\wedge F(*)\wedge R(<)$	obl(Go)
	$L(>)\wedge F(*)\wedge R(*)$	obl(Go)
South-and Northbound	$L(*)\wedge F(*)\wedge R(<)$	prh(Go)
	$L(>)\wedge F(*)\wedge R(*)$	prh(Go)

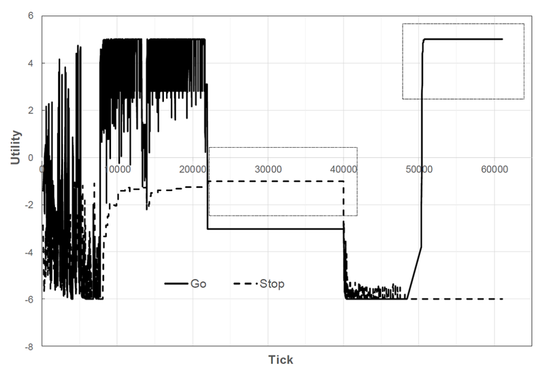


Figure 5.6: Utilities of eastbound cars

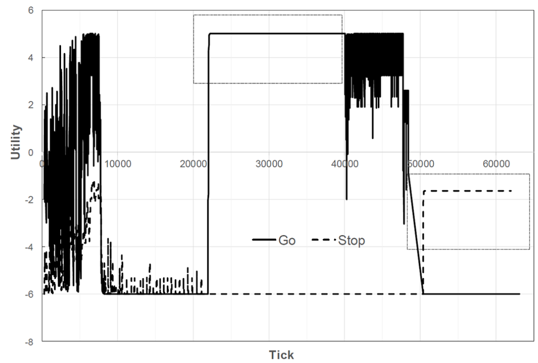


Figure 5.7: Utilities of northbound cars

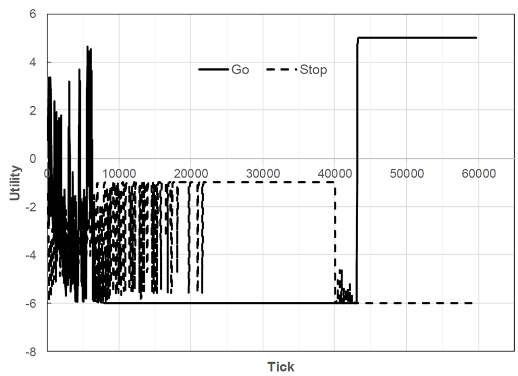


Figure 5.8: Utilities of westbound cars

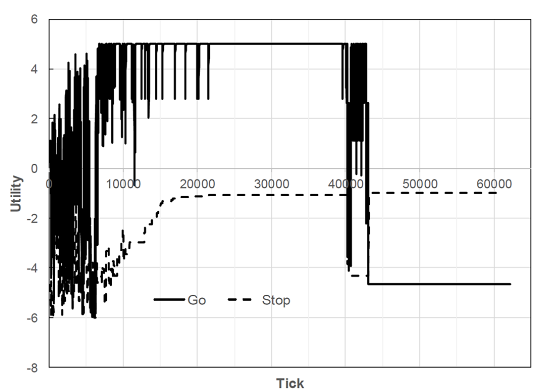


Figure 5.9: Utilities of southbound cars

5.3.3 Fairness in Cha

This section relates to Hypothesis *Fairness*. As mentioned before, our approach to incorporating fairness in Cha is motivated by the inequity-aversion theory. We propose a method based on a variation of this theory and also the ideas for simulating *guilt* put forward by Lorini and Mühlenbernd [23]. We define a fair system as one in which no agent experiences starvation (i.e., High delays) for long periods of time (here, five ticks). Note that we assume that fairness is a *privilege-driven* phenomenon and is controlled by the agent who has the power under certain conditions. We investigate fairness after the system has learned a meaningful norm. Imagine that agent i has learned to Go and agent j has learned to Stop. Agent i can decide to be fair to j, and consider agent j's situation in its decision making. We propose the following process for modeling guilt.

Guilt disutility. The guilt disutility of the state s for the agent i is defined in Equation 5.1. The assumption made here is that conflicting agents broadcast their delays to each other.

$$\Delta U_{-,i}(s,t,a) = -\beta_i\{(f_j(d) - f_i(d) - 4)\} \tag{5.1}$$

Where $f\ (d)$ is defined as in Equation 4.2, and parameter β_i represents agent i's suffering or guilt *attitude*. This equation describes the utility when the difference between delays of agents j and i is more than four ticks. Algorithm 4 summarizes the fairness learning strategy in Cha. Agent i initializes its fairness utility (i.e., U_i^f) as its converged learned utility (Lines 1-3). As long as agent i has right of way (Line 4), it receives agent j's delay (Line 6), and in any interaction in which the difference in delays is greater than four ticks, and agent i's policy is to Go (Line 7), it feels a negative cost associated with the feeling of guilt (Lines 9-10) until it learns to be fair to its conflicting agent j (i.e., to yield to the conflicting agent j). After learning to be fair, agent i signals to the conflicting agent j to Go (Line 12). As stated before, parameter β_i represents agent i's guilt attitude. β=0 leads no fairness in the system, and β=1 leads to constant yielding. In our experiments β was set to 0.5.

Algorithm 4 Fairness learning strategy for agent i

1: *Initialize*:
2: $U_i^f(s, t, Go) \leftarrow U_i^c(s, t, Go)$
3: $U_i^f(s, t, Stop) \leftarrow U_i^c(s, t, Stop)$
4: **while** $U_i(s, t, Go) > U_i(s, t, Stop)$ **do**
5: **for** each tick before conflict state **do**
6: *Receive* agent j's $f_j(d)$ in the system
7: **if** $f_j(d) - f_i(d) > 4$ **and** $U_i^f(Go) > U_i^f(Stop)$ **then**
8: *Perform* Go action
9: $\Delta U_- = \beta_i\,(f_j(d) - f_i(d) - 4)$
10: $U_i^f(s, t, Go) = U_i^f(s, t-1, Go) - \Delta U_-$
11: **else if** $f_j(d) - f_i(d) > 4$ **and** $U_i^f(Stop) > U_i^f(Go)$ **then**
12: *Perform* Stop action **and** *Signal* agent j to Go
13: **else**
14: *Follow* the learned policy by U_i
15: **end if**
16: **end for**
17: **end while**

5.3.4 Social Performance Improvement in Cha

Like Silk's framework, we now compare Cha with a fully actuated control strategy [2] as an alternative way of regulating traffic which was described in Section 2.2.6. Our metric for comparison is average travel time, calculated by adding expected travel time and average delay (average number of stops for all cars in two successive ticks). Average travel time is calculated for the whole intersection.

We run this experiment for the case where N-S has 30% higher traffic than E-W. Because of the comparative performance of two approaches, this hypothesis is amenable to statistical hypothesis testing (assume the corresponding null hypothesis is defined as that the performance of Cha and the fully actuated controller are indistinguishable from the evidence.) As our results in Figure 5.10 and Table 5.7 show, Cha and fully actuated control are different with an improvement gained by Cha is 18.38%. In addition, the null hypothesis can be rejected with a p-value of under 0.001. That is, Cha and the fully actuated controller are statistically different.

Table 5.7: Cha versus fully actuated control.

	Expected	Average	p value
Cha	19	20.03	–
Fully actuated	19	24.53	–
Improvement	–	18.30%	–
Fully actuated vs Cha	–	–	< 0.001

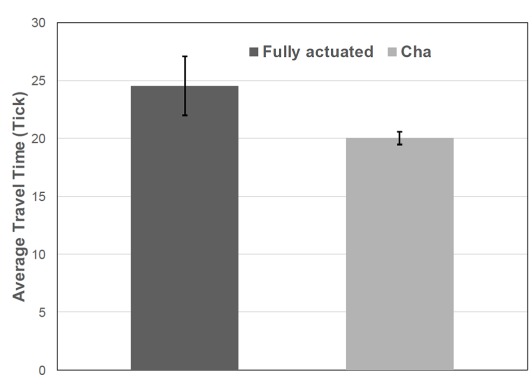


Figure 5.10: Average travel time.

Now we study fairness learning strategy as described in Section 5.3.3. For this study we use the same setting as the static setting above, in which N-S has 30% more traffic volume comparing to E-W. As explained before, in that setting E-W cars learned to Stop in conflicting situations with N-S cars. Now, we want to see whether N-S car agents can also be fair to E-W car agents or not. For this, as before, we look into changes in average "fair" utilities as defined before. Figure 5.11 shows the change in the average fair utility of southbound cars. Fairness is activated on at $t \approx 40,000$ (could be activated any time after convergence), and by $t \approx 51,000$, these agents have learned to be fair to their opponents. For brevity, we omit the figure for northbound cars, they have the same outcomes as southbound cars. We evaluate Cha's performance in terms of delays, with and without fairness. Table 5.8 shows the percentiles of delays (i.e., the percentage of times that delay falls below the time shown in the table) for the individual cars. Fairness affects the percentiles of car agents at the tail end (specifically, from 99.7 to 100). Further, fairness does not influence the total average delay substantially, when comparing Cha with and without fairness.

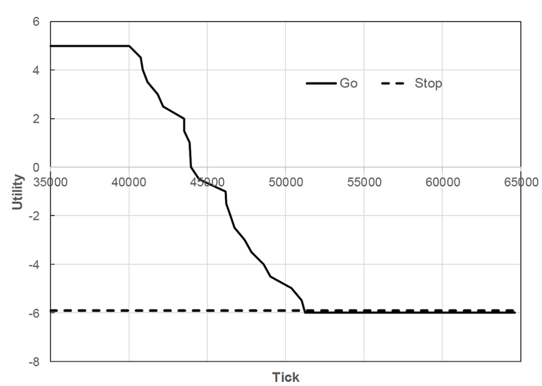


Figure 5.11: Fair utilities of southbound cars.

Table 5.8: Delays with Cha, with and without fairness.

	Percentiles							
	50	85	95	99	99.5	99.7	99.9	100
w/o Fair	0	1	2	3	4	5	6	6
w/ Fair	0	1	2	3	4	4	5	6

5.4 Conclusions

In this chapter we have empirically evaluated the both proposed frameworks in the transportation domain scenario described in Chapter 4 in relation to our research hypotheses presented in the Chapter 1.

The results of our experiments for Silk's framework shows that social norms promoting conflict resolution emerge across a variety of societies, ranging from moderate to selfish. Silk also yields improved outcomes over a traditional, fully actuated approach both in settings where fairness is ensured and where it is not.

To evaluate Cha's frameworks, we have designed two experiments to test the related hypotheses. In the static setting experiment, and in relation to conflict and efficient-conflict hypotheses, our results show that, Cha leads to emergence of norms which are effectual. In other words, in the described traffic scenario and the designed experiment, since the N-S approach has higher traffic (and consequently higher cost of being stopped), the converged norms are efficient comparing to the reverse norms in which E-W cars learn to Go, and N-S cars learn to Stop. In the dynamic setting experiment, and in relation to dynamic-adaption hypothesis, our results show that Cha responds appropriately to its dynamically changing environment. In other words, in the described traffic scenario and the designed experiment, Cha is sensitive to traffic distribution changes, and allows the existing norms to decay and new norms to evolve in favor of car agents with high traffic.

In relation to hypothesis fairness, we proposed a new methodology based on the variation of inequity averse theory. Our results show that, Cha would lead to a fair system in which no individual car agent suffer resource starvation for long period of time (i.e., Five ticks in the described traffic scenario). Finally, we compared Cha's performance in terms of the average travel time with fully actuated control strategy, our results show that, Cha yields to significant improvement over fully actuated control strategy.

Chapter 6

Conclusions and Future Work

This chapter presents the conclusions of this dissertation. In Section 6.1 we present the contributions made during this thesis. In Section 6.2 we present conclusions obtained. Section 6.3 concludes with a discussion of some future research directions.

6.1 Contribution

Multiagent systems are one of the best means to study, model, and understand human societies and behaviors. As a matter of fact, such systems are the systems of the future. Norms are an important part of human social systems (and multiagent systems), governing a lot of aspects of group decision-making. Discovering the details about how social norms arise in societies, and how they influence human activities allows us to have a better understanding of human behaviors in general. Specially, developing normative frameworks that can be used in designing life-like simulations has many applications in domains such as public policy management, clinical health promotion and advertising [9].

Recently, the problem of the emergence of social norms as keys to coordination mechanism in multiagent systems, has gained increasing attention. In the literature, a great extent of problems have been studied regarding the use of norms in multiagent systems. However, most of the existing researches focus on the closed systems (e.g., Graph structure) and no computational framework has been proposed for norm generation in open multiagent systems. The assumption of a closed system is rarely appropriate in multiagent systems and most MASs require and presuppose openness. Even a mundane setting, such as cybersecurity within an organization, presupposes openness because the parties concerned are autonomous and changing.

Fairness, a fundamental social concept is another important characteristic in evaluating interactions in MAS. Unfortunately fairness has not got enough attention in the normative MAS community.

Considering this background, in this dissertation, we have proposed two frameworks for coordination of *open* MAS, which we dubbed *Silk* and *Cha*.

Silk offers the following contributions:

- Supports regulation of *open* MAS in a hybrid fashion.

- A general, domain-independent framework to accommodate online norm generation. Where a central agent called *governor* monitors the system at runtime and ensures *Integrity violations* are averted by imposing *laws*. Also, it anticipates *conflicts*, and recommends norms to the *members* for conflict resolution. Also Silk does not require previous knowledge of agents' behaviors and their potential interactions.

- Despite the fact that the norm community has long mentioned the suitability of combining both bottom-up and top-down approaches to norm regulation, as far as we know, no other work has actually gone as far as this one. Silk offers a hybrid approach while considering, at the same time, *fairness*.

Existing studies of norm emergence in MASs, concentrate on top-down or hybrid approaches [3]. However, top-down and hybrid schemes are vulnerable, in some situations, to failure of the centralized portion. Also, existing social learning approaches for norm emergence, including Silk, assume a static environment, where actions have fixed payoffs. Although this assumption is reasonable in some circumstances, it could be violated in some MASs. There exits cases where, social norms need to change when environmental conditions change.

To fill in these gaps, *Cha* offers the following contributions:

- Supports regulation of *open* MAS in a bottom-up fashion.

- A general, domain-independent framework to accommodate online norm generation in which members reason individually on what norm to develop and follow.

- A dynamic normative framework, in which the system reacts to the dynamically changing environment. If the environment condition changes, it allows for decay of existing normative systems and emerging of new ones.

Finally, in this dissertation, we have used transportation domain to evaluate the performance of the proposed frameworks. As you would most likely agree, building effective and user-friendly transportation systems is one of the big challenges for engineers in the 21st century. Transportation domain can be viewed as an open multiagent system in which their participants have diverse goals and motivations, and their behaviors are unpredictable. Social norms seem to be very suitable to regulate such a system. This dissertation is among the few studies (if not the only one) who tackle the problem of regulation of traffic form the view point of normative multiagent systems.

6.2 Summary and Conclusion

In this dissertation, we have introduced Cha and Silk frameworks for online synthesis of norms and regulation of open multiagent systems.

Silk is a hybrid bottom-up top-down framework for norm emergence. Silk is expressly suited to systems that are open (changing membership and fluid interactions determined on the fly). It accommodates regimentation and regulation in multiagent systems. We have performed an empirical evaluation of Silk on a simplified traffic intersection scenario. Our results show that, Silk promotes the emergence of norms across a variety of societies, ranging from moderate to selfish. Silk yields improved outcomes over a traditional, fully actuated approach both in settings where fairness is ensured and where it is not.

Cha, is a bottom-up dynamic framework for norm emergence, wherein each agent reasons individually about which norms to develop and follow. Cha is also expressly suited to systems that are open. In Cha, we have concentrated on regulation since that is the more challenging aspect. We have performed an empirical evaluation of Cha on a simplified traffic intersection scenario. Our results show that Cha supports emergence of norms in a decentralized (or bottom-up) manner of norms. It outperforms traditional fully actuated approach both in settings where fairness is ensured and where it is not. More importantly, unlike previous studies, it is also sensitive to changes in the environment, such as traffic distribution changes, and facilitates existing norms to decay and new norms to emerge.

An important application of the proposed frameworks is in the regulation of self-driving cars controlled by agents that communicate with each other or the infrastructure and learn what actions to take, thus relieving humans from driving and making decisions when approaching shared spaces, such as intersections.

6.3 Future Work Beyond the Dissertation

While this dissertation has pursued to contribute to the regulation of open normative multiagent systems, it also opens several paths for future research. In the following, some research directions where we think further students and improvement could be useful, are presented.

- One possible direction for future research is to do more sensitivity analysis on some parameters of the proposed frameworks such as the payoff matrices, and fairness model in cha (possibly taking into account heterogeneous agents).

- Another possible direction for future research could be to enrich the current scenario to include further traffic elements like pedestrians, cars driving at different speeds, or even traffic lights. For this, it is important to come up with a new grammar which allows to include these new features. We believe that it is worth pursuing this line of research.

- Next possible direction is to explore the generation of group norms (i.e., norms referring or applying to a group, in opposition to norms applying to a single individual [4]). Existing studies rely on individual decision making rather than group decision making which might not be efficient. As an example, in the transportation domain and specifically in the world of connected vehicles, which is an emerging trend, vehicles (or car agents) sharing the same space, can share information with each other and can decide actions and move together as a platoon of vehicles (i.e., creating a group norms). We believe that group decision making can help to improve throughput in the system.

- Another important direction is implementing Silk and Cha for other domains. Silk and Cha can be potentially applied in any open MAS that satisfies these criteria: (1) its members compete over resources, which may lead to conflicts and/or integrity violations, (2) at least one agent can observe all the members, (3) members can observe their neighbors and can communicate. An example of other domains is online communities in which conflicts may concern privacy and information sharing (e.g., Is sharing pictures of gunshot victims and perpetrators appropriate?), which may involve sanctions via complaints and direct criticism.

REFERENCES

[1] Jeffrey L. Adler, Goutam Satapathy, Vikram Manikonda, Betty Bowles, and Victor J. Blue. A multi-agent approach to cooperative traffic management and route guidance. *Transportation Research Part B: Methodological*, 39(4):297 – 318, 2005.

[2] Federal Highway Administration. Traffic signal timing manual, 2013.

[3] S.; Airiau, S.; Sen and D. Villatoro. Emergence of conventions through social learning. *Autonomous Agents and Multi-Agent Systems*, 28(5):779–804, 2014.

[4] V.; Vasconcelos W. Aldewereld, H.; Dignum. We ought to; they do; blame the management! a conceptualisation of group norms. *In: Balke, T., Dignum, F., van Riemsdijk, M.B., Chopra, A.K. (eds.) COIN 2013. LNCS (LNAI)*, 8386:195–210, 2014.

[5] Giulia Andrighetto, Rosaria Conte, Paolo Turrini, and Mario Paolucci. Emergence in the loop: Simulating the two way dynamics of norm innovation. In Guido Boella, Leon van der Torre, and Harko Verhagen, editors, *Normative Multi-agent Systems*, number 07122 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[6] Marek; Pitt Jeremy; Busquets Dídac; Riveret Régis Artikis, Alexander; Sergot. *Specifying and Executing Open Multi-agent Systems", BOOKTITLE="Social Coordination Frameworks for Social Technical Systems.* Springer International Publishing, 197–212, 2016.

[7] Robert Axelrod. An evolutionary approach to norms. *The American Political Science Review*, 80(4):1095–1111, 1986.

[8] A. L. C. Bazzan and F. Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29:375–403, 2014.

[9] R. BEHESHTI. *MODELING SOCIAL NORMS IN REAL-WORLD AGENT-BASED SIMULATIONS*. PhD thesis, University of Central Florida, 2015.

[10] Emily Zechman Berglund. Using agent-based modeling for water resources planning and management. *Journal of Water Resources Planning and Management*, 141(11):04015025, 2015.

[11] Guido Boella, Leendert Torre, and Harko Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1):1–10, 2008.

[12] Guido Boella and Leendert van der Torre. Regulative and constitutive norms in normative multiagent systems. In *IN PROCS. OF KR04*, pages 255–265. AAAI Press, 2004.

[13] Jordi Campos, Maite López-Sánchez, and Marc Esteva. *A Case-Based Reasoning Approach for Norm Adaptation*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[14] H.; Centeno, R.; Billhardt and R. Hermoso. Persuading agents to act in the right way: An incentive-based approach. *Engineering Applications of Artificial Intelligence*, 26(1):198–210, January 2013.

[15] N. Criado, E. Argente, P. Noriega, and V. Botti. Human-inspired model for norm compliance decision making. *Information Sciences*, 245:218 – 239, 2013.

[16] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31(1):591–656, January 2008.

[17] J.A. García-Camino, A.; Rodríguez-Aguilar. Constraint rule-based programming of norms for electronic institutions. *Autonomous Agents and Multi-Agent Systems*, 18(1):186–217, February 2009.

[18] P.; Lavasani S.; Isaai, M.; Najaf and H. Nezamianpour Jahromi. Intelligent evaluation of transportation management policies for metropolitan areas. *92nd Annual Meeting of Transportation Research Board*, 2013.

[19] Andrew J. I. Jones and Marek J. Sergot. On the characterisation of law and computer systems: The normative systems perspective. In John-Jules Ch. Meyer and Roel J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 275–307, Chichester, United Kingdom, 1993. John Wiley and Sons.

[20] Jan Koeppen, Maite López-Sánchez, Javier Morales, and Marc Esteva. Learning from experience to generate new regulations. In Marina De Vos, Nicoletta Fornara, JeremyV. Pitt, and George Vouros, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, volume 6541 of *Lecture Notes in Computer Science*, pages 337–356. Springer Berlin Heidelberg, 2011.

[21] G. F. List and M. Mashayekhi. A modular colored stochastic petri net for modeling and analysis of signalized intersections. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):701–713, March 2016.

[22] G. F. List, M. Mashayekhi, Robert L Bertini, and Brandon Nevers. Development of a transportation system simulation manual (tssm) framework: From theory to practice. *Transportation Research Record (TRR), Journal of the Transportation Research Board*, 2016.

[23] Emiliano Lorini and Roland Mühlenbernd. The long-term benefits of following fairness norms: A game-theoretic analysis. In *PRIMA 2015: Proceedings of the 18th International Conference on Principles and Practice of Multi-Agent Systems*, pages 301–318, Bertinoro, Italy, 2015. Springer International Publishing.

[24] Ulf Lotzmann, Michael Möhring, and Klaus G. Troitzsch. Simulating the emergence of norms in different scenarios. *Artificial Intelligence and Law*, 21(1):109–138, 2013.

[25] M. S.; Mohd Yusoff M. Z.; Mahmoud, M. A.; Ahmad and A. Mustapha. A review of norms and normative multiagent systems. *The Scientific World Journal*, 2014:1–23, 2014.

[26] Mehdi Mashayekhi, Hongying Du, George F. List, and Munindar P. Singh. Silk: A simulation study of regulating open normative multiagent systems. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, IJCAI, pages 373–379, 2016.

[27] Mehdi Mashayekhi and George List. A multiagent auction-based approach for modeling of signalized intersections. In *In Second Workshop on Synergies Between Multiagent Systems, Machine Learning and Complex Systems (TRI 2015)*, pages 13–24, Buenos Aires, Argentina, 2015.

[28] M.; Rodríguez-Aguilar J. A. Morales, J.; López-Sánchez. Automated synthesis of normative systems. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 483–490, 2013.

[29] M.; Rodríguez-Aguilar J. A. Morales, J.; López-Sánchez. Minimality and simplicity in the on-line automated synthesis of normative systems. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 109–116, 2014.

[30] M.; Rodríguez-Aguilar J. A. Morales, J.; López-Sánchez. Synthesising liberal normative systems. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 433–441, 2015.

[31] Martin Neumann. *A Classification of Normative Architectures*. Springer Japan, Tokyo, 2010.

[32] Michael J. North, Nicholson T. Collier, Jonathan Ozik, Eric R. Tatara, Charles M. Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with Repast Simphony. *Complex Adaptive Systems Modeling*, 1(1):1–26, December 2013.

[33] S. Sen and S. Airiau. Emergence of norms through social learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence(IJCAI)*, pages 1507–1512, 2007.

[34] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, New York, NY, USA, 2008.

[35] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 73(1):231 – 252, 1995.

[36] M. P. Singh. Cybersecurity as an application domain for multiagent systems. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1207–1212, 2015.

[37] Munindar P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Trans. Intell. Syst. Technol.*, 5(1):21:1–21:23, January 2014.

[38] T. Sugawara. Cybersecurity as an application domain for multiagent systems. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 371–378, 2011.

[39] T. Sugawara. Emergence of conventions for efficiently resolving conflicts in complex networks. In *roceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pages 222–229, 2014.

[40] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, 1998.

[41] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning.* Cambridge, MA, USA, 1st edition, 1998.

[42] Guido Boella; Leendert van der Torre; and Harko Verhagen. Introduction to normative multiagent systems. *Computational and Mathematical Organization Theory*, 12(2):71–79, October 2006.

[43] M. Vasirani and S. Ossowski. A market-inspired approach to reservation-based urban road traffic management. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 617–624, 2009.

[44] J.; Villatoro, D.; Sabater-Mir and S. Sen. Social instruments for robust convention emergence. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 420–425, 2011.

[45] C. J. C. H. Watkins and P. D. Dayan. Q-learning. Technical report, 1992.

[46] H. Peyton Young. Evolution of conventions. *ECONOMETRICA*, 61(1):57–84, 1993.

[47] Emily M. Zechman. Integrating evolution strategies and genetic algorithms with agent-based modeling for flushing a contaminated water distribution system. *Journal of Hydroinformatics*, 15(3):798–812, 2013.