UNIVERSITÄT MANNHEIM

DOKTORARBEIT

Selbstlernende restriktionsbasierte Steuerung von Multiagentensystemen

---

Autor:
Michael OESTERLE

Betreuer:
Prof. Dr. Heiner STUCKENSCHMIDT

Inauguraldissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften der Universität Mannheim

26. Februar 2024

Dekan: Prof. Dr. Claus HERTLING

Gutachter: Prof. Dr. Heiner STUCKENSCHMIDT, Universität Mannheim
Prof. Guni SHARON, Texas A&M University
Dr. Christian BARTELT, Universität Mannheim

Datum der Disputation: 3. Juli 2024

UNIVERSITY OF MANNHEIM

DOCTORAL THESIS

Self-Learning Restriction-Based Governance of Multi-Agent Systems

Author:
Michael OESTERLE

Supervisor:
Prof. Dr. Heiner STUCKENSCHMIDT

A thesis submitted in fulfillment of the requirements for the degree of Dr. rer. nat. at the University of Mannheim

February 26, 2024

Abstract

The purpose of this thesis is the scientific investigation of a specific form of governance for multi-agent systems: The dynamic restriction of action spaces for achieving a system-level objective.

Governance in multi-agent systems addresses the well-known challenges associated with managing and coordinating the behavior of autonomous agents. Particularly in competitive systems, self-interested individual optimization often leads to outcomes that deviate from socially optimal results.

There are two major existing approaches to solve this problem: Rewarding or sanctioning certain behaviors through monetary incentives (called *reward shaping*) or providing agents with special capabilities for cooperation (we call this approach *cooperative capabilities*). However, both approaches exhibit certain failure modes; reward shaping assumes inter-agent comparability of rewards and unlimited governance means, while cooperative capabilities require that the agents' action policy can be altered and that agents actually want to cooperate. Another crucial factor that causes difficulties for existing governance approaches is fairness in the face of heterogeneous agents.

This motivates a novel approach to multi-agent governance, based solely on restricting action spaces in reaction to observations of the system. Such governance does not need to know about or influence the agents' inner workings, nor does it have to hand out rewards to steer agent behavior. As the prime example of "improvement through restriction", Braess' Paradox—the fact that closing a road can improve traffic flow in a congested network—, serves as a recurring illustration of the power of restriction-based governance.

We develop a unified theoretical framework, called *Action-Space Restricted Multi-Agent System* (ARMAS), which can be applied to any system modeled as a Partially Observable Stochastic Game. In this model, we propose various governance learning mechanisms for subclasses of ARMAS since the general problem of learning an optimal governance policy—being equivalent to Reinforcement Learning in non-stationary environments—cannot be expected to admit an efficient solution.

In addition to the learning algorithms, we propose an implementation of ARMAS, which is compatible with major multi-agent learning frameworks, and we evaluate our approach concerning efficacy and fairness in comparison to reward shaping. Our results demonstrate that restriction-based governance can indeed manage and coordinate the behavior of autonomous agents, leading to significant enhancements in social welfare compared to a baseline approach that does not employ action space restrictions, while avoiding problems associated with reward-based governance approaches.

Zusammenfassung

Das Ziel dieser Dissertation ist die wissenschaftliche Untersuchung einer spezifischen Form der Steuerung von Multiagentensystemen: die dynamische Einschränkung von Aktionsräumen zur Erreichung eines systemübergreifenden Ziels.

Governance im Kontext von Multiagentensystemen befasst sich mit den Herausforderungen, die mit der Steuerung und Koordination des Verhaltens autonomer Agenten verbunden sind. Insbesondere in wettbewerbsorientierten Systemen führt individuelle eigennützige Optimierung häufig zu Ergebnissen, die von sozial optimalen Resultaten abweichen.

Es existieren zwei vorherrschende Ansätze zur Lösung dieses Problems: das Belohnen oder Sanktionieren bestimmter Verhaltensweisen durch monetäre Anreize (bekannt als *Reward Shaping*) oder die Ausstattung der Agenten mit speziellen Koordinationsfähigkeiten (wir nennen diesen Ansatz *Cooperative Capabilities*). Beide Ansätze weisen jedoch bestimmte Probleme auf; Reward Shaping setzt die Vergleichbarkeit von Belohnungen zwischen den Agenten sowie unbegrenzte Steuerungsmittel voraus, während Cooperative Capabilities erfordern, dass die Handlungsstrategien der Agenten geändert werden können und dass die Agenten tatsächlich kooperieren wollen. Ein weiterer entscheidender Faktor, der bestehende Steuerungsansätze erschwert, ist Fairness im Hinblick auf heterogene Agenten.

Dies motiviert einen neuen Ansatz zur Steuerung von Multiagentensystemen, der allein auf der Einschränkung von Aktionsräumen als Reaktion auf Beobachtungen des Systems basiert. Eine solche Governance muss die interne Funktionsweise der Agenten nicht kennen oder beeinflussen, noch muss sie Belohnungen verteilen, um das Verhalten der Agenten zu steuern. Als Paradebeispiel für „Verbesserung durch Einschränkung“ dient das Braess-Paradoxon – die Tatsache, dass das Sperren einer Straße den Verkehrsfluss in einem Netzwerk verbessern kann – wiederholt dazu, das Potential einer einschränkungsbasierten Steuerung zu illustrieren.

Wir entwickeln ein einheitliches theoretisches Framework, genannt *Action-Space Restricted Multi-Agent System* (ARMAS), das auf jedes System angewendet werden kann, das als Teilweise Beobachtbares Stochastisches Spiel (*Partially Observable Stochastic Game*) modelliert wird. In diesem Modell konstruieren wir verschiedene Steuerungslernmechanismen für Subklassen von ARMAS, da das allgemeine Problem des Erlernens einer optimalen Steuerung – äquivalent zu Reinforcement Learning in nicht-stationären Umgebungen – nicht effizient lösbar sein dürfte.

Neben den Lernalgorithmen zeigen wir eine Implementierung von ARMAS, die mit wichtigen Multiagenten-Lernframeworks kompatibel ist, und wir evaluieren unseren Ansatz hinsichtlich Wirksamkeit und Fairness im Vergleich zu Reward Shaping. Unsere Ergebnisse zeigen, dass die einschränkungsbasierte Governance das Verhalten autonomer Agenten tatsächlich koordinieren kann, was zu signifikanten Verbesserungen der sozialen Wohlfahrt im Vergleich zu einem Referenzansatz führt, der keine Handlungsraumeinschränkungen verwendet, und dabei Probleme vermeidet, die mit belohnungsbasierten Steuerungsansätzen verbunden sind.

Acknowledgements

As I bring this significant chapter of my academic journey to a close, I find myself reflecting on the incredible support and guidance I have received. It's a privilege to express my profound gratitude to those who have been instrumental in the pursuit and completion of my PhD.

First and foremost, my deepest appreciation goes to my supervisors, Dr. Christian Bartelt and Prof. Dr. Heiner Stuckenschmidt. Their expertise and mentorship have been the cornerstone of my research journey; their guidance has not only shaped this thesis but has also profoundly influenced my personal and professional development.

I extend my heartfelt thanks to Prof. Guni Sharon, whose support before, during, and after my research visit at the Texas A&M University was invaluable. This experience has significantly enriched my academic perspective and contributed greatly to this thesis.

My sincere gratitude also goes to my co-authors, Stefan Lüdtke and Tim Grams. Collaborating with them has been an enlightening experience, and their contributions have been critical to the success of our joint research endeavors.

I would like to acknowledge my colleagues at the Institute for Enterprise Systems (InES), as well as Jakob Kappenberger from the DWS group at the University of Mannheim, for many stimulating discussions. Their diverse perspectives and support have been a constant source of inspiration and motivation.

Special thanks are due to the Fulbright Program for sponsoring a five-month research visit at Guni Sharon's lab, an opportunity that has been one of the highlights of my PhD journey, offering me invaluable international experience and exposure to new academic environments and ideas.

On a more personal note, my deepest gratitude is for my family. To my wife, Andrea, and my son, Luca: your love, patience (in Andrea's case), and unwavering support have been my anchor. This journey would have been unimaginably harder without your constant encouragement and sacrifices.

Contents

List of Figures

List of Tables

To L&B

Chapter 1

Introduction

Interactions between autonomous, self-interested decision-making entities—also called *agents*—commonly cause a well-known challenge: Conflicting goals lead to competitive behavior, which, in turn, often culminates in suboptimal outcomes. This dynamic, albeit deceptively simple in appearance, has been the subject of active research for at least 70 years, beginning with von Neumann and Morgenstern's seminal work that laid the foundation of *Game Theory*: The mathematical study of strategic interactions among rational agents (von Neumann and Morgenstern, 1947).

For an intuitive understanding of the main challenge addressed in this thesis, let us start with a very generic schema of interaction between autonomous decision-makers. It is customary to consider a number of *agents* and an *environment* as shown in Figure 1.1. This preliminary conceptualization of a *Multi-Agent System* (MAS) permits interactions between agents and the environment to unfold in arbitrary ways[1].

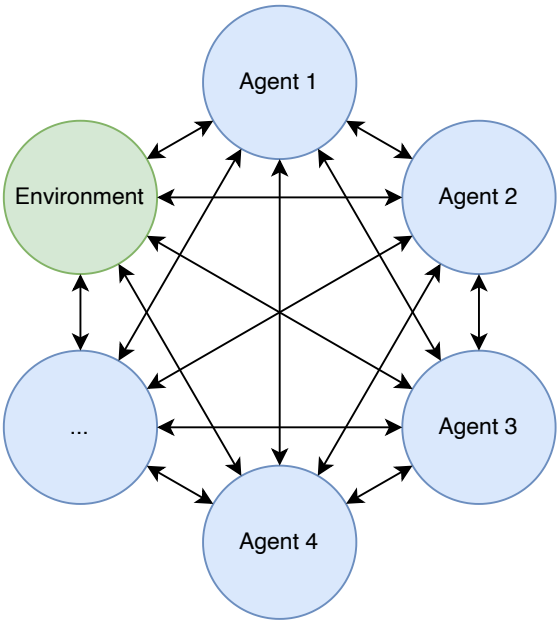


FIGURE 1.1: Generic interaction schema of agents and environment.

However, this initial model leaves much to be desired in terms of precision, given its vague definitions concerning the types, frequencies, sequence, and consequences of interactions. To address this, a more refined and sequential model, as illustrated in Figure 1.2, is commonly used, allowing for a more detailed examination of such scenarios. In this model, a single cycle of perception followed by action constitutes

[1]Even though we use technical terms such as *agent*, *utility*, or *strategy* in this introductory chapter, we defer a rigorous definition to Chapters 2 and 3 and, instead, appeal to the intuition of the reader to gain a first understanding of the problem statement and the scope of this work.

a *time step*. The introduction of time steps instills a sense of synchrony among the agents; they all make their decisions simultaneously, and the collective impact of their actions on the environment is assessed in unison[2].

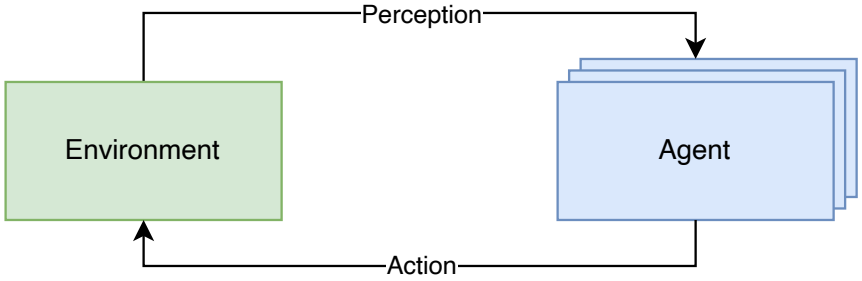


FIGURE 1.2: Interaction loop of agents and environment.

Using the terminology of this model, agents are in a continuous loop of *perceiving* their environment and then *acting* based on their observations. The whole sequence of interaction and communication among the agents happens via the environment, creating a precise and complete representation of the process in terms of observations, actions, and the *state* of the environment.

Given that the changes in the environmental state (the *transitions*) at every time step are dictated by the collective actions of all the agents, it becomes crucial for each agent to consider what the others might do when plotting their own moves (or sequences of moves) to reach a specific goal. This is where things start to get tricky, and strategies get complex: Agents constantly adjust their plans based on their observations and beliefs about their opponents, who are simultaneously doing the same thing. Often, this intricate interplay settles into a stable state where each agent has figured out their best strategy, and there is no longer any incentive to change things up. However, as we will see later, no ironclad convergence guarantee exists in *non-stationary* systems where behaviors can shift over time.

The central issue addressed in this thesis is the fact that a stable state resulting from the interactions of multiple self-interested agents does not inherently equate to an optimal outcome, especially when evaluated against system-level objectives such as social welfare[3] or fairness. Using the more general notion of a *governance utility* to represent the system-level objective, this thesis proposes and critically examines the incorporation of a governance entity within a multi-agent system. The governance's role is to interact with the system, aiming to increase the governance utility to an optimal level.

From Figure 1.2, it is clear that any governance that is not part of the environment or part of an agent can only interfere with such a system at two points (see Figure 1.3): Either it changes the way the agents perceive the environment, or it changes the way they act on it.

As we will see in Chapter 2 when we look at the commonly used mathematical models for multi-agent systems, these two interception points translate to a number of tangible methods of intervention, one of which—restriction of action spaces—is the main focus of this thesis.

[2]It is worth noting that different concepts of (a)synchronicity can be captured through various *temporal logics*, such as Linear-Time Temporal Logic (LTL) (Pnueli, 1977), Computation-Tree Logic (CTL) (EMERSON, 1990), or Alternating-time Temporal Logic (ATL) (Alur, Henzinger, and Kupferman, 2002). For a comprehensive overview, the reader is referred to Hoek and Wooldridge, 2012.

[3]We adopt the widely accepted meaning of the term *social welfare*, which refers to the aggregate utility of all agents in a system.

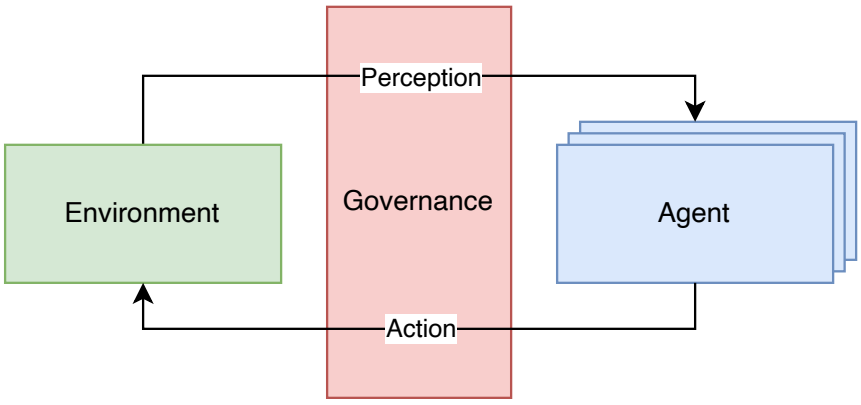


FIGURE 1.3: Interaction loop of agents, environment, and governance.

Let us now look closer at the two fundamental building blocks of governed multi-agent systems: Autonomous agents and the governance entity.

1.1 Autonomous agents and multi-agent systems

1.1.1 Autonomous agents

An *agent* is any entity that is able to make decisions in a goal-oriented way with some degree of autonomy while perceiving and interacting with an environment. Standard definitions from the research literature include "Autonomous agents are systems capable of autonomous, purposeful action in the real world" (Brustoloni, 1991), "Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed" (Maes, 1995) and "An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" (Franklin and Graesser, 1996).

For the problems that we are concerned with in this thesis, it is not relevant how such an agent observes its surroundings, what its physical nature is, or how it makes the decision to choose a specific action at a given point in time; all that counts is that the agent can act and that these actions, in some way, influence the environment. This implies that humans, as well as artificial devices, can be seen as autonomous agents in the context of decision-making and action selection: Like autonomous agents in artificial intelligence, humans have the ability to perceive their environment, reason about their internal state, and make decisions about how to act in order to achieve their goals. Humans are able to autonomously navigate complex and dynamic environments, such as cities, workplaces, and social situations, using a combination of learned and innate behaviors.

One of the critical challenges in designing artificial autonomous agents is developing effective decision-making algorithms that can handle uncertainty, incomplete information, and changing environments. Reinforcement Learning (RL) (see Sutton and Barto, 2018) has emerged as the gold standard for developing such algorithms, as it allows agents to learn optimal decision-making policies through trial-and-error

interactions with their environment[4]. RL has been successfully applied in single-agent scenarios, where only one agent interacts with the environment, but extending RL to multi-agent scenarios introduces new challenges.

1.1.2 Multi-agent systems

In their standard work *Multi-agent Systems: Algorithmic, Game-Theoretic, and Logical Foundations* (Shoham and Leyton-Brown, 2009), the authors define multi-agent systems as "those systems that include multiple autonomous entities with either diverging information or diverging interests, or both". Therefore, agents must learn to make decisions while interacting with other agents, who may have different objectives and strategies. This introduces the potential for cooperation and competition among agents and the need for agents to reason about the beliefs, intentions, and strategies of other agents.

Scientific research about multi-agent systems typically models a very restricted set of characteristics of a real-world system, focusing on a specific aspect while ignoring irrelevant noise. This aspect can simply be the agents' manipulation of an environment without considering the agents' inner workings, as in Figure 1.2. It might also include details of the perception process, an agent's inner model of its environment and competitors, or the communication schemes that allow the agents to exchange information.

Multi-agent Reinforcement Learning (MARL) aims to develop RL algorithms for multi-agent scenarios, considering the unique challenges and opportunities of these settings.

1.2 Governance

In its most general form, governance refers to the process of decision-making and management of a system. It involves establishing rules, policies, and procedures to guide behavior and ensure the system operates effectively and efficiently. The concept of governance typically applies to systems involving decision-making, control, and coordination, and it is often discussed in systems with autonomous agents. As such, it can be applied to social systems like governments and corporations, as well as to artificial agents like robots and software programs[5].

For systems involving humans, whether nations, corporations, or smaller groups, governance involves establishing systems, structures, and processes that regulate behavior, coordinate actions, and make collective decisions. One common aspect of governance is developing and enforcing legal and regulatory frameworks. These frameworks define rights, responsibilities, and acceptable behavior within the community, providing a basis for maintaining order and resolving conflicts. Additionally, institutional structures play a vital role in governance. Governments, administrative bodies, and corporate boards are examples of institutions with authority and responsibility for decision-making, policy formulation, and oversight.

[4]Usually, the number of trials required to learn optimal decisions in any reasonable domain is huge (or, in other words, the *sample efficiency* is low). Therefore, RL models are most commonly trained in simulated environments over thousands or even millions of episodes.

[5]Governance can also be meaningful in systems without autonomous agents, although the nature and implementation of governance in such contexts may differ: Examples include centralized control systems like operating systems, organizational and institutional governance, or technical protocols which govern networking and communication systems.

Governance is influenced by social norms, ethics, and unwritten rules that shape behavior and interactions within the community. These informal mechanisms promote cooperation, trust, and shared values.

It is clear that governance in this sense has a much broader scope than in the context of well-defined artificial agents within a mathematical framework. Nevertheless, governance of artificial systems is ultimately inspired by and borrows fundamental concepts and goals from its human counterpart.

For our purposes, two aspects of governance are essential: The *governance objective* defines what the governance wants to achieve, while the *governance mechanism* defines how it can achieve this objective.

1.2.1 Governance objective

The governance objective describes the goal(s) of the governance, that is, which states of the system[6] are desired or undesired. Just as the agents participating in a multi-agent system have preferences with respect to the system's states, allowing them to decide which actions they want to choose and which ones to avoid, the governance also needs a way of distinguishing good and bad to steer its behavior. This objective can depend on the agents' objectives, include other factors, or be completely unrelated.

A prevalent principle is that governance should serve the agents—after all, the governance of a human community is often devised and established by the very parts of this community. Operationalizing this principle leads to the concept of *social welfare*, which refers to the collective well-being or utility of all agents in the system. It is often measured by a quantitative social utility function that takes into account the (also quantified) individual utilities of all agents and how they interact with each other. As a result of conflicting agent goals, the actions of one agent may affect the well-being of other agents and, therefore, the overall welfare of the system. Social welfare seeks to optimize the system's overall well-being rather than just individual agents' interests.

The social welfare function can take many different forms, depending on the system's context and the agents' preferences. It may be a linear or nonlinear combination of individual utilities and incorporate external factors such as environmental conditions or resource constraints.

Maximizing social welfare is an essential goal in multi-agent systems, as it can lead to more efficient and equitable outcomes for all agents involved. However, achieving social welfare may require cooperation and coordination among the agents, which can be challenging without a central authority or governance mechanism. This is why research into multi-agent systems often focuses on developing mechanisms for incentivizing cooperation and promoting social welfare.

It should be noted that social welfare is an objective that naturally arises from ethical considerations about multi-agent systems. Still, it is by no means the only imaginable governance objective. More generally, any assignment of utility values to system states can be used as a governance objective function.

[6] As we will see in Chapter 2, usually all information about the system is captured in the environmental state.

1.2.2 Governance mechanism

The governance mechanism describes how the governance influences the multi-agent system in order to reach its objective. Therefore, the term includes all behaviors of the governance entity that change anything about the system.

Static governance mechanisms, also known as *rule-based* or *predetermined* governance, involve establishing fixed rules, regulations, or constraints that govern agents' behavior within the system. These rules are determined in advance and remain unchanged during the system's operation. Static governance mechanisms prescribe specific actions or behaviors that agents must adhere to, often based on predefined policies or guidelines. The rules may be designed to ensure fairness, resource allocation, or coordination among agents. However, static governance approaches may lack adaptability and struggle to cope with changing environments or evolving agent dynamics. They do not have the ability to learn from past experiences or adjust their rules in response to observed agent behavior.

Dynamic governance, on the other hand, is a more flexible and adaptive approach that allows the governance mechanism to evolve and adjust over time based on the observed behavior of agents and the system's performance. Dynamic governance mechanisms actively monitor and assess the behavior and interactions of agents and make real-time adjustments to guide or influence their actions. These adjustments can be based on various factors such as performance metrics, social welfare, or other objectives. Dynamic governance can involve learning algorithms, feedback mechanisms, or optimization processes that iteratively refine the governance strategies in response to the system's dynamics. This adaptability enables dynamic governance to handle better changes in agent behavior, environmental conditions, or system requirements. Within dynamic governance, we can further distinguish between reactive governance, which operates as a response to some observed behavior, and proactive governance, which anticipates agent behavior and acts based on its predictions.

In economics and game theory, *Mechanism Design* (Hurwicz and Reiter, 2006) is the study of suitable governance schemes for a given multi-agent system. It requires understanding the agents' preferences, capabilities, and information availability, as well as the potential conflicts or trade-offs that may arise between individual and collective goals. Naturally, mechanism design has also been the subject of automation efforts, using observations and learning capabilities to adapt the governance mechanism to the situation at hand.

1.2.3 User Equilibrium and Social Optimum

We have initially mentioned the discrepancy in outcome between systems where agents simply pursue their individual goals through selfish optimization and systems where social welfare maximization is also taken into account. To make this dichotomy more crisp, we borrow two terms from transportation networks: *User Equilibrium* and *Social Optimum*. These two concepts are used to describe the behavior of users in a transportation network:

- User equilibrium refers to a situation where each user of the transportation network selects the shortest or fastest path that minimizes their own travel time or cost. In other words, each user makes a selfish decision to optimize their own travel experience without considering the impact of their decision on other users. However, in a system with many users making similar decisions,

this can lead to congestion and delays as the network becomes overloaded and travel times increase.

- Social optimum, on the other hand, refers to a situation where all transportation network users make decisions that collectively minimize the total travel time or cost for everyone. This requires users to take into account not only their own preferences but also the preferences of others and the overall efficiency of the network. In a social optimum, the resources of the network are used in the most efficient way possible, minimizing congestion and delays.

More generally, the difference between user equilibrium and social optimum can be expressed as the trade-off between individual and collective optimization. In the present context of governance in multi-agent systems, this corresponds to an ungoverned system—where the governance does not act at all—on the one hand, and an optimally governed system—where the governance pursues its own objective—on the other hand.

1.2.4 Criteria for successful governance

Using the concept of governance utility (see Section 1.2.1), the primary measure for the success of a governance mechanism is the utility value it achieves over time. At the same time, there are a number of other considerations that can also be thought of as defining a good or successful governance:

- Is the goal of the governance clear?
- Is the governance mechanism transparent, such that the agents or outside observers can understand it?
- Is the governance (both goal and mechanism) perceived as fair? Does it conform to certain ethical standards?
- Can the governance mechanism react to unforeseen changes in the multi-agent system? If so, how fast can it react?
- Does the governance need to learn from interaction, or is its mechanism independent of the system?
- If it learns, how fast does it learn, and is it guaranteed to reach an optimum?

For a formal definition of governance as the optimization process of finding the best governing actions, it is, of course, necessary to quantify and weigh these criteria. In Chapter 3, we define the governance utility as a mathematical function that captures all the criteria that determine the success of the governance and, therefore, serves as the sole performance measure for optimization. As with all mathematical models, such a function necessarily abstracts away many layers of complexity of real-world systems, focusing on a small set of well-defined criteria within the context of the MAS model.

1.2.5 Autonomy and restriction

Connected to the fairness and ethics discussion of the previous section is a point that is particularly pressing when multi-agent systems consist of human agents: What justifies the governance's right to restrict the agents' autonomy by intervening in the

system? And even if other goals generally justify intervention, how severe should it be allowed to be?

Apart from ethical concerns, there can also be objective reasons for keeping the governance influence to a minimum: If agent preferences (and, therefore, their utility functions) are unknown to the governance, achieving social welfare is impossible without agent autonomy. Moreover, participating in a system in order to achieve individual goals (e.g., in a market) might become less attractive when the freedom of action is restricted. Finally, agent autonomy enables distributed optimization, which might be hard or impossible for the governance to replicate as a single optimizer, especially when a large number of agents are involved.

In line with human intuition, it is, therefore, a reasonable assumption that all else being equal, more agent autonomy and less governance intervention is preferable. This tenet can be used to inform the governance policy, for example, by choosing a mechanism that selects the most minor intervention among all utility-maximizing mechanisms.

1.2.6 Why do we need governance?

It seems at first glance that governance is merely a crutch needed by agents who fail to find a cooperative joint strategy by themselves that achieves the social optimum. After all, one could argue that rational agents with unlimited reasoning abilities should be able to figure out that their current selfish strategy is only resulting in the (sub-optimal) user equilibrium and then come up with a better solution.

However, this common-sense argument falls apart when we look at a particular type of multi-agent system: In a *Social Dilemma*, it is always better for an agent to defect from the cooperative strategy. Hence, even when an agent knows that everyone will be worse off if cooperation breaks down, it still pays off to defect, no matter what the other agents do. The Prisoner's Dilemma (Rapoport and Chammah, 1965) is one of the simplest games exhibiting this structure and arguably the most famous and widely known game-theoretic setting. There are two options for an agent to achieve cooperation as a dominant strategy in the Prisoner's Dilemma: Either have the opportunity for punishment in future games[7], or have the ability to agree on and then enforce a specific behavior. The former requires a credible threat for punishment, i.e., a strategy that punishes a defector *and still gives a higher overall return than ignoring the defection*. The latter requires both a method of agreeing on cooperative behavior and some guarantee for future actions[8].

Crucially, both options are not available to the agent in the basic one-off setting: It can only choose cooperation or defection without being able to learn from repeated interaction or forcing the opponent to choose the same strategy. It is precisely this limitation that makes a social dilemma a dilemma, preventing agents from simply solving it from within.

It is undoubtedly true that human agents, with their innate ability to devise complex interaction schemes—built on concepts like reciprocity, trust, punishment, contracts, and others[9]—can find ways to make the social optimum the game outcome.

[7] The immensely fruitful idea of an *iterated* prisoner's dilemma (with far-reaching consequences for the optimal strategy) was investigated in detail by Axelrod, 1984.

[8] Such a guarantee is sometimes called a *commitment device* and constitutes a powerful instrument not only for multi-agent coordination but also for delaying gratification in individual humans.

[9] It is striking that all these concepts are built around language as their medium; in Section 10.3 we discuss possible uses of language for our particular governance approach.

On the other hand, there are many domains in which this has failed, despite decade-long global efforts and gigantic investments of time and money.

This being said, agents in well-defined, simplified interaction frameworks (see Section 2.1) do not have the capabilities necessary to align social optimum and user equilibrium without external support. Therefore, governance can be seen as precisely a formal term for whatever mechanisms human communities use to coordinate their actions in a way that lets them achieve outcomes beyond selfish optimization.

1.2.7 Why do we propose restriction-based governance?

As hinted at in the abstract, there are two major existing paradigms for the governance of multi-agent systems: *Reward shaping*, as a form of centralized governance[10], incentives or disincentivizes certain agent behaviors through artificial rewards which are applied to the "natural" rewards provided by the environment in the standard multi-agent interaction frameworks (see Section 2.1). The other paradigm, which we call *cooperative capabilities*, consists of enhancing the agents' own capabilities so that they *can and want to cooperate* without any external guidance. As such, it represents a decentralized governance mechanism.

These approaches are capable of aligning user equilibrium and social optimum of a system, but both of them rely on specific assumptions which, as we argue, are often not satisfied:

Reward shaping To effectively apply reward shaping, governance rewards need to be chosen such that (unknown) agents change their behavior. This requires that agents react (approximately) equally to rewards, a requirement known as *inter-agent reward comparability*. Additionally, the governance must have access to sufficient reward (or sanction) means to maintain or escalate its policy until the desired behavioral change is achieved. These assumptions are violated in many relevant application domains, and even when they are satisfied, they create a high risk of manipulability by agents who anticipate and exploit the governance policy by behaving strategically.

Cooperative capabilities To overcome selfish optimization (e.g., in a social dilemma) and achieve a social optimum distinct from the user equilibrium, agents need additional capabilities. As explained in Section 1.2.6, it is not enough to see through the user equilibrium as a globally sub-optimal outcome as long as there is no device to bind the competing agents to a different policy. If such control or guarantees are absent, it is always preferable for agents to deviate from cooperation towards individually optimal policies. Therefore, emergent cooperation can only be achieved if the governance designer has access to the agents' action policies and can control them in a way that guarantees adherence to cooperative strategies even in the face of a (selfishly) better alternative.

These conditions are intimately tied to the respective mechanisms and are not necessarily required for other approaches. In particular, *restriction-based governance*, as defined and investigated in this thesis, leaves the agents as they are—it does not

[10] In this context, a *centralized* governance is a distinct entity which interacts with the MAS, much like the schema shown in Figure 1.3. Such a governance can, of course, exhibit a distributed architecture but forms a single logical unit in relation to the MAS. In contrast, a *decentralized* governance means that the governance mechanism is distributed among the agents, with no need for such an entity.

change their inner workings, and does not even require knowledge of their reward functions or action policies. It therefore seems worth examining in detail the applicability, performance, and scalability of this novel approach to multi-agent governance.

1.3 The learning problem

A multi-agent system is usually considered from two perspectives at the same time: First, the *descriptive* viewpoint states how agents and environment interact, and second, the *prescriptive* viewpoint asks how agents can choose an interaction strategy that leads to optimal (cumulative) reward. While not explicitly part of the interaction framework (see Figure 1.2 and Section 2.1), the latter perspective—the *learning problem*—is crucial for an agent to *successfully* act in a multi-agent system, and therefore has attracted most of the research effort in the field over the last decades[11]. Illustratively, the agent-environment loop is complemented with a learning (meta-)loop (see Figure 1.4), designed to alter an agent's action policy in response to the observed interactions and the agent's performance, as measured by its reward function. This learning loop is not necessarily synchronized with the interaction loop as long as there is a valid policy for action selection at any time step.

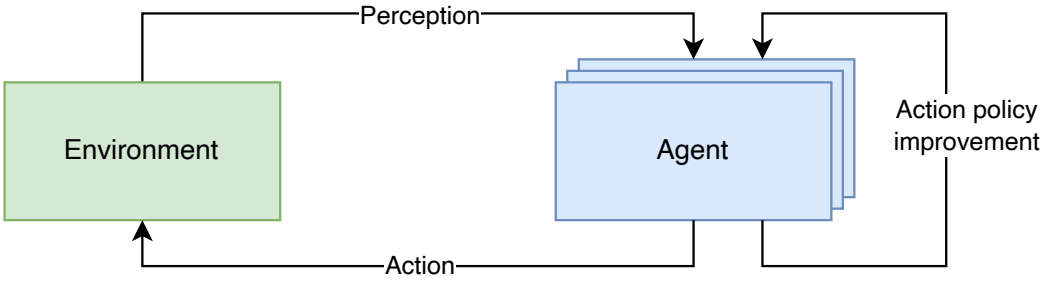


FIGURE 1.4: Illustration of the agent learning loop. Independent of its interaction with the environment, an agent can change and improve its strategy based on past observations and future predictions.

This two-level model of interacting and learning is in close analogy with the governance mechanism with which we are concerned in this work:

As outlined in Section 1.2.2, a governance that is static, i.e., predefined before the execution of the MAS, cannot react to any unforeseen development but needs to account for any possible (combination of) agent behavior from the beginning. Since this is unrealistic for all but the most trivial systems—particularly when agents themselves improve their action policies through learning—, we mirror the agent learning paradigm by allowing the governance to change its policy in response to its own observations, as shown in Figure 1.5.

The resulting model is again both descriptive and prescriptive: On the former level, it defines the mechanism with which the governance intervenes in a multi-agent system. On the latter level, it includes the governance learning method, which improves the governance mechanism over time with respect to its objective and, therefore, allows it to act purposefully.

[11] In contrast, the *environmental model* has not changed much, such that POMDPs (see Section 2.1.2) are still the most-used model, more than 50 years after their invention by Åström, 1965.

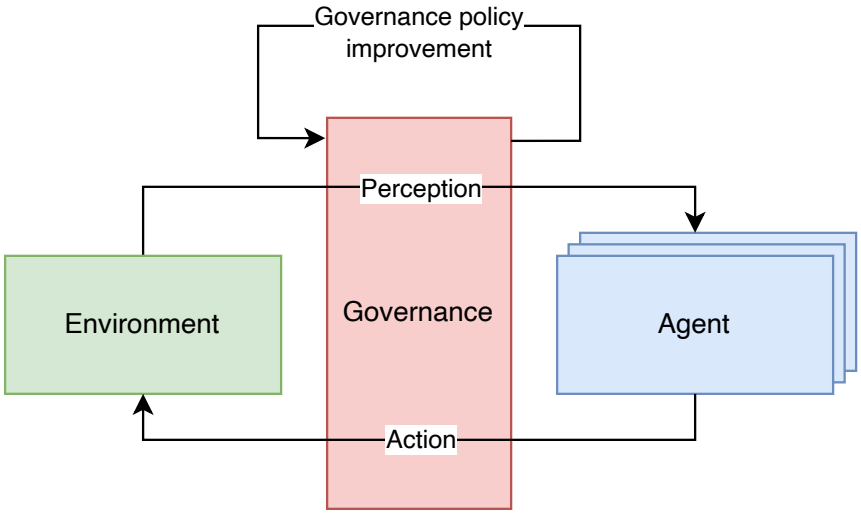


FIGURE 1.5: Illustration of the governance learning loop. By observing how the governance's current strategy performs in connection with the agents' own decisions and the corresponding environmental effects and learning from these observations, the governance can change its strategy over time.

1.4 Scope and contribution

On the highest level, this thesis investigates a specific approach of governance for multi-agent systems: The dynamic restriction of action spaces.

One level deeper, we look at *Action-Space Restricted Multi-Agent Systems* as a formal extension of existing MAS models and investigate the governance learning problem for various settings and solution approaches.

Even more specifically, the content of the thesis is as follows: After having motivated the problem in the introduction within the context of multi-agent systems and governance, we next provide the theoretical background of MAS and the optimization methods that we will apply later to the governance learning problem (Chapter 2), and we present the restriction-based governance model that we will use throughout the thesis (Chapter 3). We give a comprehensive overview of the relevant existing work in Chapter 4, and then use our model to investigate various facets of restriction-governed multi-agent systems (Chapters 5 to 9). As a first step, Chapter 5 proposes a heuristic for governance policy optimization in matrix games via successive elimination of actions. Chapter 6 then uses Reinforcement Learning for governance policy optimization in discrete stochastic games. Chapter 7 proposes yet another governance policy optimization method, this time based on tree search for continuous normal-form games. Chapter 8 describes an integration of governance capabilities into computational multi-agent reinforcement learning frameworks, and Chapter 9 evaluates the efficacy and fairness of two opposing governance paradigms. We round the thesis off with a discussion of results and limitations (Chapter 10) before concluding the work (Chapter 11).

Our work contributes to the understanding and advancement of governed multi-agent systems by:

- Defining a formal model for *Action-Space Restricted Multi-Agent Systems* (AR-MAS),

- providing original solution approaches to identify optimal restrictions for different subclasses of MAS,
- proposing a reference implementation for the integration of ARMAS in the widely used *PettingZoo* (Terry et al., 2021) MARL framework, and
- demonstrating the potential of restriction-based governance for performance and fairness.

1.5 Previous publications

Major parts of the contents have been published in peer-reviewed journals and presented at international scientific conferences. In particular, the following chapters correspond to specific publications[12]:

Chapter 5 Michael Pernpeintner, Christian Bartelt and Heiner Stuckenschmidt (2021). *Governing Black-Box Agents in Competitive Multi-Agent Systems*. In: Proceedings of the 18th European Conference on Multi-Agent Systems (EUMAS 2021).

Chapter 6 Michael Oesterle, Christian Bartelt, Stefan Lüdtke and Heiner Stuckenschmidt (2022). *Self-learning Governance of Black-Box Multi-Agent Systems*. In: Proceedings of the International Workshop on Coordination, Organizations, Institutions, Norms and Ethics for Governance of Multi-Agent Systems (COINE 2022).

Chapter 7 Michael Oesterle and Guni Sharon (2023). *Socially Optimal Non-Discriminatory Restrictions for Continuous-Action Games*. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI 2023).

Chapter 8 Michael Oesterle, Tim Grams and Christian Bartelt (2024). *DRAMA at the PettingZoo: Dynamically Restricted Action Spaces for Multi-Agent Reinforcement Learning Frameworks*. In: Proceedings of the 57th Hawaii International Conference on System Sciences (HICSS 2024).

Chapter 9 Michael Oesterle, Tim Grams, Christian Bartelt and Heiner Stuckenschmidt (2024). *RAISE the Bar: Restriction of Action Spaces for Improved Social Welfare and Equity in Traffic Management*. To appear in: Proceedings of the 23rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2024).

Other publications with ideas that have been developed and refined throughout the PhD research and included in this thesis but do not appear here in their original form are:

- Michael Pernpeintner (2019). *Collaboration as an Emergent Property of Self-Organizing Software Systems*. In: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W 2019).

[12] For each of these publications, I explicitly state my own contributions at the beginning of the chapter.

- Michael Pernpeintner (2020). *Achieving Emergent Governance in Competitive Multi-Agent Systems (Doctoral Consortium)*. In: Proceedings of the 19th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2020).
- Michael Pernpeintner (2021). *Self-Learning Governance of Competitive Multi-Agent Systems*. In: Organic Computing Doctoral Dissertation Colloquium (OC-DDC 2020).
- Michael Pernpeintner (2021). *Toward a Self-Learning Governance Loop for Competitive Multi-Attribute MAS (Extended Abstract)*. In: Proceedings of the 20th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2021).

From all publications, the models, related work sections, and general discussions have been grouped and condensed into the respective chapters of this thesis. Note that the motivational storylines for the individual chapters, while building upon each other, have evolved naturally over the course of the PhD work. To make this process transparent and allow the reader to follow the chronological progress comfortably, we have deliberately left the motivation sections of Chapters 5 to 9 essentially unchanged. A more stringent overall motivation—looking backward from the final product and connecting the dots—is provided in a few paragraphs at the beginning of each chapter.

The code for all experiments, while not constituting a part of the formal submission, can be found at

`https://github.com/michoest/thesis`,

allowing for traceability and reproducibility of our results.

Chapter 2

Background

To understand precisely how agents interact in a multi-agent system and how governance can influence it to achieve its goals, we need to formalize the schematic models from Chapter 1 and recap existing solution methods. This background chapter introduces existing frameworks and methods widely used in multi-agent research and relevant to our work; most originate in game theory, machine learning, and mathematical optimization.

2.1 Interaction frameworks

This thesis's novel content builds upon several frameworks that have been used for many decades to formally describe decision-making scenarios. In this section, we provide the necessary models using a unified notation that can easily be extended in Chapter 3 to incorporate our own contributions. The two main complexities are (a) the interaction between an agent and an environment and (b) the interaction among multiple agents. These two challenges are formalized in the concepts of Markov Decision Processes (MDPs) and Normal-Form Games (NFGs), respectively. The Stochastic Game (SG) model combines the above frameworks, and the notion of *partial observability* provides additional support for incomplete information.

2.1.1 Markov Decision Processes

Decision-making in artificial and real-world scenarios often involves navigating complex and uncertain environments. Whether it is a robot exploring an unfamiliar terrain, a financial investor making investment decisions in a volatile market, or a healthcare provider determining treatment strategies, the ability to make optimal decisions in the face of uncertainty is paramount.

Markov Decision Processes (MDPs) offer a structured and formal approach to tackle such decision problems. By explicitly considering the probabilistic nature of events and incorporating the consequences of different actions, MDPs enable us to reason about the long-term implications of decisions and identify optimal strategies. The power of MDPs lies in their ability to capture the dynamics of a system over time. By defining states, actions, transitions, and rewards, we can model the system's behavior and study the interplay between decision-making and (uncertain) outcomes. This allows us, for example, to design intelligent agents that can adapt their actions based on observed states and thus maximize their expected cumulative rewards.

MDPs are used in Reinforcement Learning and other decision-making approaches to formalize and study problems in which the outcome of each decision is uncertain. They are particularly useful in scenarios where an agent needs to make

a sequence of decisions over time in a dynamic environment, and the outcome of each decision is probabilistic and depends on the current state of the environment.

Definition 1. *A* Markov Decision Process *(MDP) is a tuple*

$$(\mathcal{S}, A, r, \delta) ,$$

where $\mathcal{S}$ is the set of environmental states, A is the agent's action space, $r : \mathcal{S} \times A \times \mathcal{S} \to \mathbb{R}$ is the agent's reward function, and $\delta : \mathcal{S} \times A \to \Delta_{\mathcal{S}}$ is the (stochastic) transition function[1].

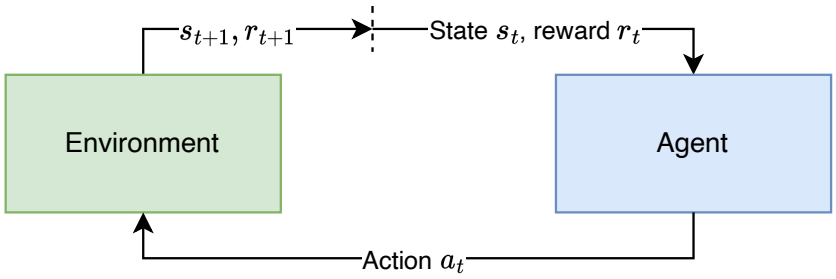


FIGURE 2.1: Illustration of the agent-environment interaction in an MDP.

Remark 1. *If the MDP is deterministic, i.e.,*

$$\forall s \in \mathcal{S}, a \in A \; \exists s' \in \mathcal{S} : \mathbb{P}[\delta(s, a) = s'] = 1 ,$$

the notation of reward and transition functions can be simplified by writing $r : \mathcal{S} \times A \to \mathbb{R}$ and $\delta : \mathcal{S} \times A \to \mathcal{S}$.

The state transition function δ describes the probability of moving from one state to another given a particular action. Specifically, $\delta(s, a)$ represents the probability distribution of the next state s' from state s by taking action a. r is a reward function that maps each triple of state, action, and next state to a numerical reward signal, indicating the desirability of taking that action in that state[2].

The (stochastic) action policy of the agent, i.e., the mechanism used by the agent to choose an action when given a state, is given as a function $\pi : \mathcal{S} \to \Delta_A$. Using this notation, the interaction between agent and environment can be succinctly described by the *evolution formula*

$$s_{t+1} = \delta(s_t, \pi(s_t)) , \tag{2.1}$$

where the output of the stochastic functions is sampled according to the respective distribution[3].

The iterative application of Equation (2.1) to some initial state $s_0 \in \mathcal{S}$ leads to a *trajectory* $(s_0, s_1, s_2, ..., s_T)$ of states over discrete time steps, from the start ($t = 0$) to the end ($t = T$) of an *episode*. Adding information about the actions taken and the rewards received gives a complete description

$$(s_0, a_0, r_0, s_1, a_1, r_1, ..., s_{t-1}, a_{t-1}, r_{t-1}, s_T)$$

[1] For a (finite or infinite) set X, Δ_X denotes the set of probability distributions over X, i.e., the set of all functions $p : X \to [0, 1]$ with $\sum_{x \in X} p(x) = 1$.

[2] Note that, as the action affects both the reward and the next state, it is not necessarily optimal to greedily choose the action with the highest (expected) reward at each step.

[3] The use of a time step t as a subscript indicates that a variable or function evolves over time.

of an episode, called its *history*.

The learning behavior of the agent, i.e., the fact that the action policy π can change over time in order to maximize the agent's cumulative reward is not described by the MDP [4].

From Definition 1, we can immediately see that the *Markovian property* holds for an MDP: The future state of the environment depends only on the current state and the action taken and not on any previous states or actions. This property, which is often stated as "the future is independent of the past given the present", is essential for the applicability of most RL algorithms.

2.1.2 Partially Observable Markov Decision Processes

Partially Observable Markov Decision Processes (POMDPs) extend MDPs by incorporating uncertainty in the form of partial observability. In a POMDP, the agent does not have complete information about the environmental state but instead only has access to partial observations or noisy signals of the actual state. To represent this, a set of possible observations and an observation function are added to the model:

Definition 2. *A* Partially Observable Markov Decision Process *(POMDP) is a tuple*

$$(\mathcal{S}, \mathcal{O}, \sigma, A, r, \delta) \ ,$$

where $\mathcal{S}$ is the set of environmental states, $\mathcal{O}$ is the set of possible observations, $\sigma : \mathcal{S} \to \mathcal{O}$ is the agent's observation function, A is the agent's action space, $r : \mathcal{S} \times A \times \mathcal{S} \to \mathbb{R}$ is the agent's reward function, and $\delta : \mathcal{S} \times A \to \Delta_{\mathcal{S}}$ is the transition function.

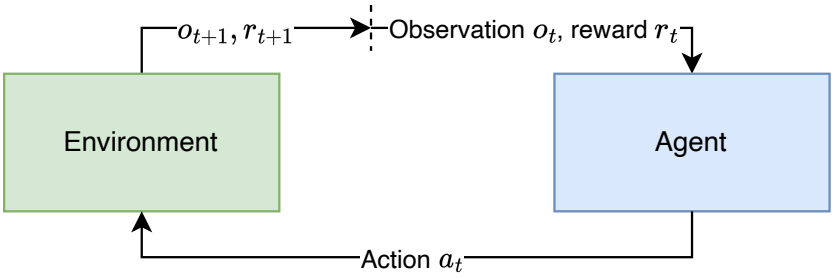


FIGURE 2.2: Illustration of the agent-environment interaction in a POMDP.

Accordingly, the evolution formula analogous to Equation (2.1) is

$$s_{t+1} = \delta(s_t, \pi^{(t)}(\sigma(s_t))) \ . \tag{2.2}$$

2.1.3 Normal-Form Games

In various aspects of life, from economics and politics to social interactions and biology, we encounter situations where multiple individuals or entities are faced with choices that impact their own outcomes as well as the outcomes of others. Normal-Form Games (NFGs), or strategic-form games, offer a structured approach to studying and understanding such strategic interactions. At its core, an NFG represents

[4] When describing the interaction of agent(s) and environment, we will, from now on, write $\pi^{(t)}$ instead of π to emphasize the fact that the action policy does not have to remain the same function over time.

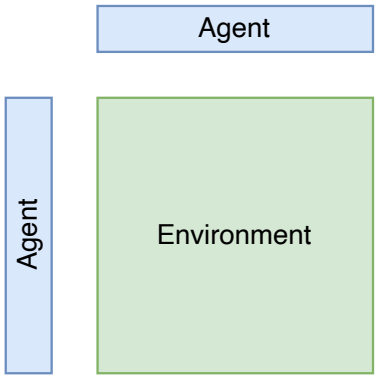
a set of players, their possible actions, and the associated payoffs or outcomes. By mapping out the strategic choices and the resulting outcomes, we can analyze the game's dynamics and uncover insights into decision-making strategies, cooperation, competition, and the overall equilibrium of the system.

By studying NFGs, we can gain valuable insights into various real-world scenarios: Whether analyzing pricing strategies in a competitive market, understanding the dynamics of political elections, or unraveling the complexities of evolutionary biology, NFGs offer a versatile tool for modeling and analyzing strategic interactions among rational agents. NFGs are the mathematical representation of one-shot decision-making problems. They are used to model scenarios where two or more players make simultaneous decisions, with each player trying to maximize their own payoff based on all players' (expected) decisions.

Definition 3. *A* Normal-Form Game *(NFG) is a tuple*

$$(I, \boldsymbol{A}, \boldsymbol{r}) \,,$$

where I is the set of agents, $\boldsymbol{A} = (A_i)_{i \in I}$ are the agents' action spaces[5]*, and $r_i : \boldsymbol{A} \to \mathbb{R}$ is agent i's reward function.*



	a	b	c
a	$0, 0$	$0, 1$	$0, 0$
b	$1, 0$	$2, 2$	$0, 3$
c	$0, 0$	$3, 0$	$1, 1$

FIGURE 2.3: Schematic illustration and exemplary payoff matrix of a two-player NFG. More players can be added by adding dimensions to the matrix.

In an NFG, each player chooses their action simultaneously without knowing the actions selected by the other players. The payoff received by each player depends on the actions chosen by all players, and each player's goal is to choose the action that maximizes their expected payoff, given the other players' choices.

Normal-form games with finite action spaces can be represented in a matrix form, called the payoff matrix (see Figure 2.3), where each row represents an action available to the first player, each column represents an action available to the second player (and higher-dimensional matrices are used for more than two players), and the entries in the matrix cells are $|I|$-tuples representing the payoffs for each player, given the actions chosen by all players for the respective cell. This matrix representation is commonly used to analyze and solve normal-form games[6].

Remark 2. *The terms* agent *(more common in the multi-agent community) and* player *(more common in the game theory community) can be used interchangeably in this context. By convention, we call the decision-making entities "agents" for the remainder of this work.*

[5] By convention, we write vectors or sequences of variables in boldface.

[6] For action spaces with infinitely many elements, the reward functions are usually given in analytical form instead of a matrix (see Chapter 7, where we consider continuous-action games).

2.1.4 Stochastic Games

Stochastic Games (SGs) are the logical conjunction of Markov Decision Processes and Normal-Form Games. They describe settings where multiple agents interact with each other over time in a dynamic and uncertain environment. Stochastic games can, therefore, be seen as a multi-agent generalization of MDPs, in which the environment is influenced by the actions of all agents, and each agent's reward function depends not only on the state and action of that agent but also on the actions of the other agents. Vice versa, they can be seen as "stateful" versions of NFGs, where the environmental state influences the reward dynamics.

Definition 4. *A* Stochastic Game *(SG) is a tuple*

$$(I, \mathcal{S}, \boldsymbol{A}, \boldsymbol{r}, \delta) \ ,$$

where I is the set of agents, $\mathcal{S}$ is the set of environmental states, $\boldsymbol{A} = (A_i)_{i \in I}$ are the agents' action spaces, $r_i : \mathcal{S} \times \boldsymbol{A} \times \mathcal{S} \to \mathbb{R}$ is agent i's reward function, and $\delta : \mathcal{S} \times \boldsymbol{A} \to \Delta_{\mathcal{S}}$ is the transition function.

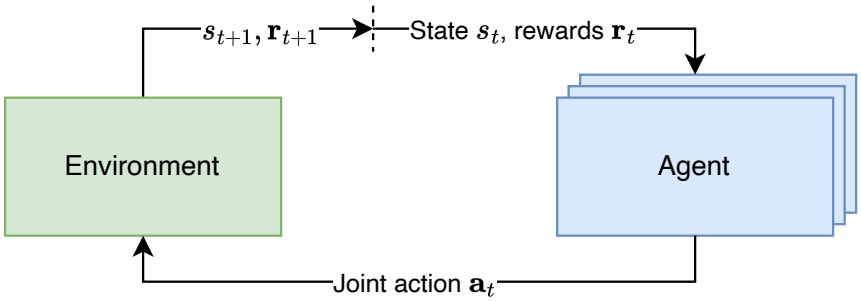


FIGURE 2.4: Illustration of the agent-environment interaction in a Stochastic Game.

As above, each agent has an action policy $\pi_i : \mathcal{S} \to \Delta_A$ defining its behaviour; the evolution of the system is therefore described by

$$s_{t+1} = \delta(s_t, \boldsymbol{\pi}^{(t)}(s_t)) \ . \tag{2.3}$$

Again, the reward and transition functions can be simplified for deterministic settings.

2.1.5 Partially Observable Stochastic Games

As with MDPs, Stochastic Games can be extended to account for partial observability:

Definition 5. *A* Partially Observable Stochastic Game *(POSG) is a tuple*

$$(I, \mathcal{S}, \boldsymbol{\mathcal{O}}, \sigma, \boldsymbol{A}, \boldsymbol{r}, \delta) \ ,$$

where I is the set of agents, $\mathcal{S}$ is the set of environmental states, $\boldsymbol{\mathcal{O}} = (\mathcal{O}_i)_{i \in I}$ are the sets of observations for each agent, $\sigma_i : \mathcal{S} \to \mathcal{O}_i$ is agent i's observation function, $\boldsymbol{A} = (A_i)_{i \in I}$ are the agents' action spaces, $r_i : \mathcal{S} \times \boldsymbol{A} \times \mathcal{S} \to \mathbb{R}$ is agent i's reward function, and $\delta : \mathcal{S} \times \boldsymbol{A} \to \Delta_{\mathcal{S}}$ is the transition function.

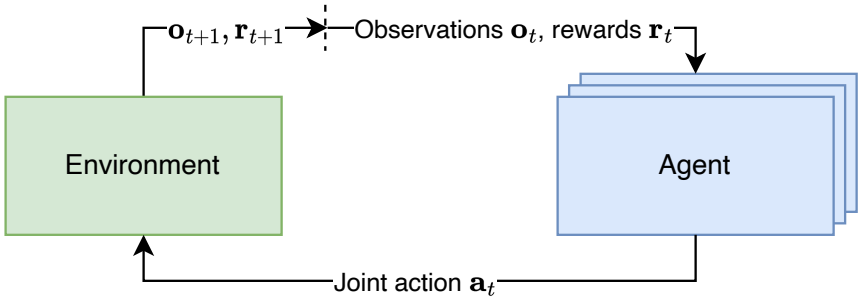


FIGURE 2.5: Illustration of the agent-environment interaction in a POSG.

The essence of a POSG is captured in its evolution function

$$s_{t+1} = \delta(s_t, \boldsymbol{\pi}^{(t)}(\sigma(s_t))) \ . \tag{2.4}$$

Due to the lack of full observability of the environmental state, the agents must make decisions based on their current observations and beliefs about the true state of the game, which may be incorrect due to the presence of noise or uncertainty. This introduces another level of complexity and strategic depth to the game, as agents must consider not only their current actions but also how those actions might affect the observations of the other agents and their own beliefs about the state of the game.

POSGs are a powerful generic framework for modeling real-world decision-making scenarios with inherent uncertainty and incomplete information. They can be used to study a wide range of applications, including robotics, economics, and social science. For these reasons, we build our restriction-based governance approach upon this model.

2.2 Action policies

In the above interaction frameworks, the agents' part is represented by their action policies, i.e., their way of mapping an observation to an action, which is then processed by the environment, resulting in a transition to another state. An action policy can be deterministic or stochastic; formally, in the case of a POSG, this means that the policy of agent i is defined as

$$\pi_i : \mathcal{O}_i \to A_i$$

or

$$\pi_i : \mathcal{O}_i \to \Delta_{A_i} \ ,$$

respectively.

The action policy, as the agent's only means of interacting with the world, can be thought of as containing all of its current knowledge, beliefs, and goals. Therefore, a great deal of research has been dedicated to finding optimal action policies for a wide range of tasks, particularly to learning optimal action policies from iterative observation and interaction. The predominant paradigm for agent learning in the setting of Equation (2.4) is *Reinforcement Learning* (see Section 2.3).

Remark 3. *In the context of a stateless game (e.g., a Normal-Form Game), the action policy reduces to a probability distribution over the action space and is called a* strategy.

However, before we turn to how agents can learn, let us review some concepts of *rational agents*, which are independent of any specific learning approach.

2.2.1 Agent rationality

Agents, by definition, want to achieve a specific goal, expressed as a *utility function*[7]. In a decision process or game, this function is specified as the reward given by the environment at each step. An agent is said to be *rational* if its action always reflects the best choice toward this goal, given its current information. As we will see in Section 2.3, it is not always easy to define what the best choice is since there is the inherent trade-off of *exploring* an unknown environment to gather information and *exploiting* this information to make informed decisions. In simple settings like Normal-Form Games with known utility functions, however, the concept of a *best response* is sufficient for rational behavior.

2.2.2 Best response

When an agent knows how the other agents will act in a game (i.e., when the action policies $\boldsymbol{\pi}_{-i}$ are fixed and known[8]), the maximization of its reward function is an ordinary optimization problem over the agent's action space. The notion of *best responses* in an NFG formalizes this situation without providing any information on how to find such an action:

Definition 6 (Best Response)**.** *Let $\boldsymbol{a} \in \boldsymbol{A}$ be a joint action. Then*

$$\mathcal{B}_i(\boldsymbol{a}_{-i}) := \arg\max_{a \in A_i} r_i(a, \boldsymbol{a}_{-i}) \subseteq A_i$$

denotes the set of all best responses *(BRs) of agent i to the other agents' given actions $\boldsymbol{a}_{-i}$.*

For a stateful setting, the analogous concept would select the strategies which yield the highest *expected reward*, taken over the uncertainty of the transition function.

2.2.3 Nash equilibrium

A *Nash Equilibrium* represents a state of a game in which each agent's strategy is a best response to the strategies of the other agents. Formally, a Nash Equilibrium is a set of strategies, one for each agent, such that no agent can improve their payoff by unilaterally changing their strategy, given the strategies of the other agents.

Definition 7 (Nash Equilibrium)**.** *A joint action $\boldsymbol{a} \in \boldsymbol{A}$ is a (pure)* Nash Equilibrium *(NE) if each individual action $a_i \in \boldsymbol{a}$ is a best response to the other agents' actions. $\mathcal{N}$ denotes the set of all Nash Equilibria of a game:*

$$\mathcal{N} := \{\boldsymbol{a} \in \boldsymbol{A} : \ a_i \in \mathcal{B}_i(\boldsymbol{a}_{-i}) \, \forall i \in I\} \ .$$

In addition to pure Nash Equilibria, given by a deterministic action for each agent, the same concept exists for stochastic strategies and dynamic policies. If any

[7] Depending on the context, the terms *reward*, *preference*, or *cost* function are also used; the name implicitly defines the direction of optimization.

[8] For a vector $\boldsymbol{x} \in X^n$ over a set X, let $\boldsymbol{x}_{-i} := (x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) \in X^{n-1}$ denote the vector obtained by removing x_i. By convention, we also use the concatenation $\boldsymbol{x} = (x_i, \boldsymbol{x}_{-i})$.

agent were to change their strategy from a Nash Equilibrium unilaterally, their payoff would decrease, given the fixed strategies of the other agents. Once all agents have chosen their strategies according to the equilibrium, no agent is incentivized to change their strategy unilaterally, making a Nash Equilibrium a "stable" strategy. However, a game can have multiple Nash equilibria, in which case it may not be clear which one will actually be played in practice by rational agents.

2.2.4 Optimal policy

In stateful settings, an agent gets a reward for every action it takes, but the actions also determine (or at least influence) the state where the agent will end up next. Therefore, a policy needs to take into account both the short-term reward and the *future expected reward*. To balance these two goals, the objective function of an agent is usually defined as the *expected return* over a given number of steps (also called the *time horizon* T):

Definition 8. *The* return *of a policy π is the sum of future rewards when following the policy, discounted by a factor $\gamma \in [0, 1]$:*

$$R = \sum_{t=0}^{T} \gamma^t r_t : \; a_t = \pi(s_t) \tag{2.5}$$

When there is uncertainty, e.g., with respect to the stochastic policy π, the transition function δ, or other agents' actions, the expected return

$$R = \mathbb{E}_{a_t \sim \pi(s_t)} \left[\sum_{t=0}^{T} \gamma^t r_t \right] \tag{2.6}$$

is taken instead.

The discount factor γ determines the weight assigned to future rewards relative to immediate rewards: A discount factor of 0 means that only the reward at the next step is considered, while a discount factor of 1 means that all future rewards are given equal importance.

Remark 4. *For an infinite time horizon, i.e., $T = \infty$, the return is only well-defined if $\gamma < 1$ since the sum might not converge otherwise.*

Naturally, we call a policy *optimal* if no other policy yields a higher expected return.

2.3 Reinforcement Learning

Reinforcement learning (RL) (Sutton and Barto, 2018) is currently the most popular paradigm for developing agents that can learn to make decisions based on feedback from their environment[9]. RL agents improve their action policy through iterative interaction with the environment, as defined in Section 2.1.1. Ideally, the policy converges to an optimal value after some training and can then be applied to navigate the environment without further learning.

There are two main categories of RL algorithms: value-based and policy-based. Value-based methods learn to approximate the "value" of different actions in a given

[9] This capability, according to some definitions, already constitutes a form of intelligence.

state. These methods use a value function to represent the expected reward that an agent will receive when it takes a particular action in a specific state. The agent can then choose the action with the highest value in a given state, leading to optimal decision-making. Common value-based methods include Q-learning (Watkins, 1989)—with its deep learning counterpart, Deep Q Networks (DQN) (Mnih et al., 2013)—, and SARSA (Rummery and Niranjan, 1994). Policy-based methods, in contrast, learn to optimize the agent's policy directly as a stochastic mapping from states to actions. These methods use a policy function to represent the probability of taking a particular action in a given state. The agent can then update its policy based on the rewards it receives, thereby improving its decision-making. Common policy-based methods are (variations of) REINFORCE (Williams, 1992) and actor-critic (Konda and Tsitsiklis, 1999).

Since value-based and policy-based methods have their respective strengths and weaknesses, recent research has focused on combining the two approaches to create hybrid methods that leverage the benefits of both. These hybrid methods, including Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2016) and Proximal Policy Optimization (PPO) (Schulman et al., 2017), have shown promising results in complex decision-making tasks such as robotics and game playing.

When representing the value function or policy, early RL approaches often used tables, where each entry corresponds to a state (or state-action pair) and its associated value (or policy). However, this tabular representation becomes impractical for environments with a large number of states or continuous state spaces. Modern RL methods often use neural networks as function approximators to overcome this issue. These networks can generalize across similar states, enabling RL to handle complex, high-dimensional environments, such as those encountered in robotics or video games. The use of neural networks in RL has given rise to the sub-field known as Deep Reinforcement Learning (Mnih et al., 2013), which has achieved significant breakthroughs in various domains in recent years.

For single-agent RL, there are some algorithms that have been proven to converge to an optimal policy (Sutton and Barto, 2018): For example, value iteration and policy iteration are guaranteed to converge to the optimal policy if (a) the MDP has a finite number of states and actions, (b) the model, i.e., the transition dynamics and rewards, is known, and (c) the discount factor γ for the cumulative rewards is less than one. Q-Learning, on the other hand, is proven to converge to the optimal Q-values, and consequently, the optimal policy, if (a) every state-action pair is visited infinitely often, and (b) the learning rate[10] α over the training time satisfies the *Robbins-Monro conditions* $\sum \alpha = \infty$ and $\sum \alpha^2 < \infty$. Similar conditions hold for the convergence of SARSA: (a) the policy follows an ε-greedy strategy with decaying ε, (b) the learning rate conditions of Q-Learning hold, and (c) all state-action pairs are infinitely explored. In contrast to value and policy iteration, Q-Learning and SARSA do not require a known model of the environmental dynamics (in other words, they are *model-free*).

The details of specific algorithms are not our primary interest here, but we will make use of the model-free algorithms A3C (Mnih et al., 2016), PPO, and DQN and, therefore, provide a few key points of interest about them.

A3C, or *Asynchronous Advantage Actor-Critic*, is a nuanced variant of the actor-critic approach. In the traditional actor-critic paradigm, both a policy (termed the *actor*) and a value function (the *critic*) are maintained. A3C elevates this setup by introducing an element of asynchrony. It deploys multiple agent instances to explore

[10]The learning rate of an algorithm generally defines how fast it reacts to new data.

the environment concurrently. This parallel exploration breaks the usual correlations arising from single-threaded experience collection. Furthermore, A3C is notable for its use of the advantage function, which gauges how much better a particular action is compared to an average action in a given state. This advantage measure guides policy updates, pushing the agent towards actions yielding higher-than-average rewards.

In *Proximal Policy Optimization* (PPO), we find a method that enhances policy gradient techniques. One of the challenges with traditional policy gradient methods is the possibility of large policy updates that can destabilize learning. PPO mitigates this by introducing a clipped objective function. This function penalizes deviations in the policy that veer too far from a previous iteration, thereby preventing excessively aggressive updates. An additional feature of PPO is its approach to data efficiency: instead of updating the policy once for each data sample, PPO reuses each sample across multiple epochs.

On the other end of the spectrum lies DQN, or *Deep Q-Network*, which is rooted in value-based RL. Here, the focus is on approximating the Q-values, representing the expected cumulative reward for taking a specific action in a particular state. Deep neural networks handle this approximation of the value function. DQN incorporates two primary mechanisms to stabilize the learning process: experience replay and a target network. Experience replay involves storing history snippets (comprising state, action, reward, and next state) in a buffer. During learning, random batches from this buffer are sampled, ensuring that updates are not biased by recent experiences and thus breaking harmful temporal correlations. The target network, a separate neural network that updates more slowly than the primary network, aids in calculating expected Q-values, further contributing to stable learning.

2.3.1 Multi-Agent Reinforcement Learning

Unlike traditional single-agent RL, where an agent interacts with a stationary[11] environment, in Multi-Agent Reinforcement Learning (see Canese et al., 2021 for a recent survey), agents interact with environments that are also affected by other learning agents, leading to non-stationarity in the observed environment dynamics. Non-stationarity implies that past learnings cannot simply be extrapolated to guide future behavior since the learned connection between actions, environmental transitions, and rewards can change unpredictably over time. MARL approaches, therefore, concentrate on two core issues, compared to classical RL: Non-stationarity, by force of unknown and dynamic agents, and scalability, since joint action spaces usually grow exponentially in the number of agents.

If an agent simply sees all other agents as part of the environment without further distinguishing them from the "true" environmental dynamics, non-stationarity prevents guaranteed convergence: There is currently no Reinforcement Learning algorithm that always converges in non-stationary environments (Padakandla, K. J., and Bhatnagar, 2020), and consequently no optimality guarantee for independent reinforcement learners in multi-agent settings (Lee, Subramanian, and Crowley, 2021).

MARL algorithms can still improve learning by considering multiple agents simultaneously, i.e., by sharing, for example, observations, episodes, or learned policies (Tan, 1997). Some common paradigms have evolved with regard to information

[11] In this context, *stationary* refers to an environment where the probabilities of transitions and the rewards for each state-action pair remain constant over time.

shared between agents at training and execution time: *Centralized training with decentralized execution* (CTDE) involves training a single agent that has access to the observations and actions of all agents during training, but each agent executes its own policy independently during execution. *Decentralized training with centralized execution* (DTCE) means training each agent independently but using a centralized controller to select actions during testing. In contrast, fully decentralized (i.e., conventional) learning does not allow any coordination or communication between agents during training or execution.

To name a few specific algorithms, Value-Decomposition Networks (VDN) (Sunehag et al., 2018) decompose a joint value function into individual value functions and train agents with a joint Q-function while acting with decentralized policies. Multi-Agent Deep Deterministic Policy Gradient (MADDPG) (Lowe et al., 2017) is a CTDE extension of the DDPG algorithm to multi-agent environments, while COMA (Counterfactual Multi-Agent Policy Gradients) (Foerster et al., 2018) trains decentralized policies with a centralized critic, using counterfactual reward baselines.

2.4 Governance

All MARL approaches have in common that they try to minimize the divergence between user equilibrium and social optimum by enabling cooperation *within the agents*, without resorting to an external entity.

In contrast, the *governance* approach aims at influencing a MAS to maximize the governance utility of stable results (therefore generalizing the social welfare objective of cooperative MARL) while the agents keep maximizing their own reward. By convention, any actions guiding the system towards this objective are termed as "governance", no matter how they influence the MAS.

The interaction loop of the POSG model allows for governance intervention at four different points or any combination of them (see Figure 2.6):

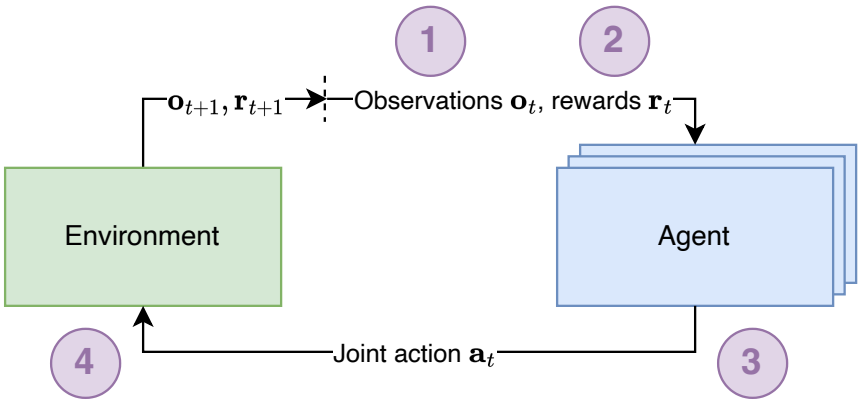


FIGURE 2.6: Potential governance intervention points in a POSG

1. *Observation*: The governance can change what agents observe, given the environmental state

2. *Reward*: The governance can change what rewards agents receive for a given state/action pair

3. *Action space*: The governance can change what actions agents can take

4. *Transition*: The governance can change what the next environmental state is, given the current state and joint action

While interventions at the observation and transition level are somewhat artificial and thus have minimal applicability in real-world systems and research, interventions with respect to rewards and action spaces are common in the literature. Chapter 4 will provide details about existing governance approaches on both intervention points—action-space interventions are the base of our own work, whereas reward interventions are a natural comparison for the performance of a restriction-based governance.

Chapter 3

Action-Space Restricted Multi-Agent Systems (ARMAS)

The POSG model (Definition 5) defines a fixed action space for each agent, and the system dynamics of such a MAS are based on the fact that all agents optimize their strategies with respect to these available actions.

This chapter introduces an extension of the model, allowing us to describe the interaction between agents, environment, and a governance with the ability to (dynamically) restrict the set of actions available to each agent in the system.

3.1 Model

Our overall goal is to allow a governance entity to influence the MAS to achieve its objective. Specifically, we propose a governance that can restrict the agents' action spaces such that the system dynamics in the restricted MAS lead to higher values of the governance utility function. To do so, we extend a POSG with two components, as motivated in Section 1.2: The governance utility function and a *restriction policy* as its mechanism. Restrictions, in this context, are simple subsets of an action space; they are defined by the governance and communicated to the agents.

Definition 9. *A* restriction *is any subset $R \subseteq A$ of an action space A, denoting the set of* allowed actions[1].

Definition 10. *An* Action-Space Restricted Multi-Agent System *(ARMAS) is a tuple*

$$(I, \mathcal{S}, \boldsymbol{\mathcal{O}}, \sigma, \boldsymbol{A}, \boldsymbol{\rho}, \boldsymbol{r}, \mathfrak{u}, \delta) \,,$$

where $(I, \mathcal{S}, \boldsymbol{\mathcal{O}}, \sigma, \boldsymbol{A}, \boldsymbol{r}, \delta)$ is a POSG, $\boldsymbol{\rho} = (\rho_i)_{i \in I}$ with $\rho_i : \mathcal{S} \to 2^{A_i}$ are the restriction functions (or restrictors*) that are applied to each agent, respectively*[2]*, and $\mathfrak{u} : \mathcal{S} \to \mathbb{R}$ is the governance utility function. Accordingly, agents' action policies are now defined as $\pi_i : \mathcal{O}_i \times 2^{A_i} \to \Delta_{A_i}$ with the requirement that $\operatorname{supp}(\pi_i(o, R)) \subseteq R$ for any observation $o \in \mathcal{O}_i$ and restriction $R \subseteq A_i$*[3]*. This requirement ensures that any action not in R (i.e., any forbidden action) is taken with probability zero*[4]*.*

[1] Since restrictions for individual agents are mutually independent, the *joint restriction* $\boldsymbol{R}$ has the form of a cartesian set $\boldsymbol{R} = \prod_{i \in I} R_i$; we write its relation to the full joint action space $\boldsymbol{A}$ as $\boldsymbol{R} \sqsubseteq \boldsymbol{A}$.

[2] We denote with $2^S := \{S' : S' \subseteq S\}$ the power set (i.e., the set of all subsets) of an arbitrary set S, both finite and infinite.

[3] For a real-valued function $f : X \to \mathbb{R}$, $\operatorname{supp}(f) := \{x \in X : f(x) \neq 0\}$ denotes the *support* of f.

[4] In other words, we assume restrictions to be *hard constraints* in the terminology of Shoham and Tennenholtz, 1995 (see Chapter 4).

The evolution function in this model is

$$s_{t+1} = \delta\left(s_t, \boldsymbol{\pi}^{(t)}\left(\boldsymbol{\sigma}(s_t), \boldsymbol{\rho}^{(t)}(s_t)\right)\right) .$$

As with the action policy in an MDP, the evolution and optimization of the restriction policy is not part of the model; we will return to this crucial part of governance-based MAS in Section 3.1.1. For now, the governance restriction functions are simply a static canvas on which the multi-agent system is executed (see Figure 3.1).

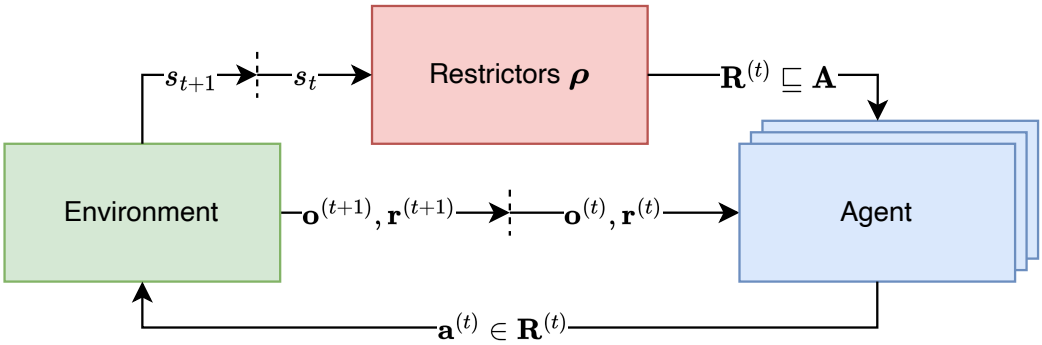


FIGURE 3.1: Agent-Restrictor-Environment loop of ARMAS.

The ARMAS approach does not directly alter the action spaces but instead defines subsets of the action spaces as allowed actions. This design choice ensures compatibility with neural network architectures that rely on fixed input and output shapes. In Chapter 8, where an integration of ARMAS with existing MARL implementations is showcased, the advantages will become evident.

As opposed to other authors (Balke et al., 2013), we do not distinguish between legal and physical power[5]: An agent can, as a matter of fact, only choose from the set of currently allowed actions, which might change from step to step. From an agent's viewpoint, an ARMAS is a regular MAS, with the difference that not all actions are available at each time step.

3.1.1 Governance learning

There is an intentional analogy between agent learning (see Figure 1.4) and our approach to governance learning: Just like an agent acting in a MAS can alter its action policy to maximize its reward, the governance in an ARMAS can alter its restriction policy to maximize its utility. The overall learning scheme, being agnostic about the kind or frequency of policy changes, is shown in Figure 3.2. Given a concrete definition of the observation and action space of the governance, this architecture lends itself well to trail-and-error optimizers like RL; Chapter 6 exploits this by training an end-to-end RL algorithm as the governance of a discrete-action MAS. However, other learning algorithms can also be used to update the governance's restriction policy.

[5]This distinction has been the subject of much debate, especially in the field of *normative systems*; the different viewpoints are described in Chapters 4 and 6, respectively.

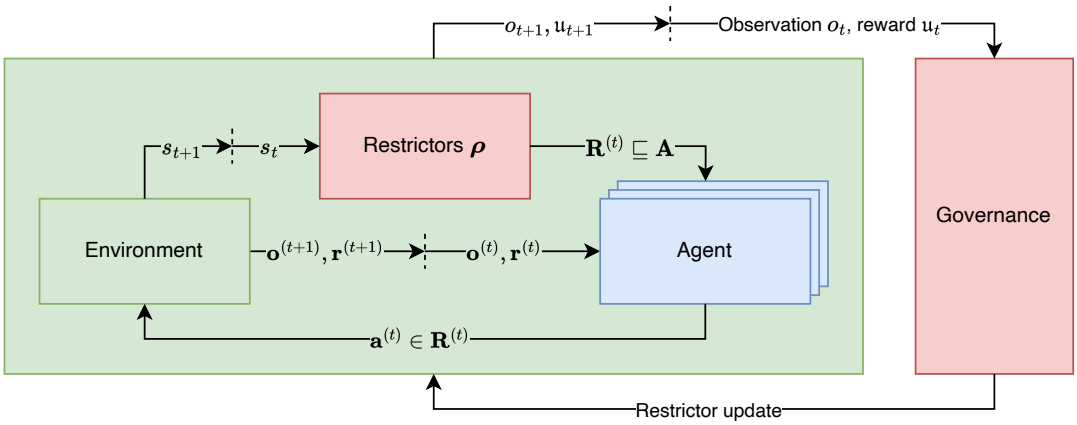


FIGURE 3.2: Generic governance learning scheme based on the ARMAS model. Note that the time steps of the inner and outer loops are not necessarily linked; in fact, it seems plausible that the governance should only act (i.e., change the restrictors $\boldsymbol{\rho}$) after observing the effect of the current restriction policy for a sufficient number of steps.

3.2 Notable subclasses

In many cases, the full feature set of ARMAS is not exploited. Accordingly, there are a number of common simplifications of the model; we will make use of the following reductions in later chapters:

3.2.1 Statelessness

A stateless ARMAS is basically a "governed normal-form game" (see Section 2.1.3): $|\mathcal{S}| = 1$, and therefore the environmental state, observations and transition function can be omitted.

3.2.2 Determinism

Determinism in an ARMAS refers to the transition function, meaning that the current environmental state and the joint action uniquely define the next state:

$$\forall s \in \mathcal{S}, \boldsymbol{a} \in \boldsymbol{A}\ \exists s' \in \mathcal{S} : \mathbb{P}[\delta(s, \boldsymbol{a}) = s'] = 1\ .$$

However, the agents' action policies can still be stochastic.

3.2.3 Uniformity

In a uniform ARMAS, the restriction functions of all agents are equal: $\rho_i = \rho_j\ \forall i, j \in I$, implying that $A_i = A_j\ \forall i, j \in I$[6].

3.2.4 Conditionality

A conditional ARMAS has uniform action spaces and observation functions, and a (uniform) restriction policy ρ which is conditioned on an agent's observation: $A_i = A_j = A\ \forall i, j \in I$, $\mathcal{O}_i = \mathcal{O}_j = \mathcal{O}\ \forall i, j \in I$ and $\rho : \mathcal{S} \times \mathcal{O} \to 2^A$.

[6] In case of uniform action spaces, restrictions and observations, we use a simplified notation: For example, A (without an index) denotes the action space of any single agent, while $\boldsymbol{A} := A^I$ denotes the joint action space.

Chapter 4

Related Work

Addressing the governance learning problem in multi-agent systems can draw inspiration from a variety of existing research. Just like multi-agent learning mainly derives its methods from single-agent learning and adapts them to address the specific challenges of agent interaction, our governance learning approach builds on both agent learning and traditional optimization techniques.

This chapter gives an overview of the literature, limited to work relevant to any of the subsequent chapters of the present work. Containing theoretical, conceptual, and domain-specific work, it is divided into three major parts: Agent learning, governance, and the research of "improvement by restriction" in Braess' Paradox.

4.1 Agent learning

The study of agents who learn to act optimally in an unknown environment is almost entirely dominated by the Reinforcement Learning (RL) paradigm (Sutton and Barto, 2018). Designed as a framework to optimize the selection of actions in an MDP (Section 2.1.1), RL serves as an umbrella term for a wide variety of algorithms and is still heavily researched. As the name indicates, the basic intuition behind RL is that, by interacting with the environment and learning from these interactions, historically successful actions should be *reinforced* over time, eventually leading to a policy that always selects the optimal action for any given state.

For our purposes, the relevant topics in the learning literature are the adherence to action-space restrictions in RL algorithms, as well as the agent-interaction challenges in Multi-Agent Reinforcement Learning, which appear in a similar form in the governance learning problem.

4.1.1 Learning with restricted action spaces

Almost all state-of-the-art RL algorithms, including the three algorithms described in Section 2.3, rely on fixed neural architectures that can only process flat input vectors and output arrays of constant size (notably, algorithms without function approximators, like tabular Q-learning, do not have this restriction). Although preprocessing techniques can be employed to flatten complex spaces, they also require data of fixed shape.

However, RL environments often possess complex space structures, ranging from simple discrete and continuous spaces (Brockman et al., 2016) to mixed discrete-continuous variants (Neunert et al., 2020) and parametric spaces (Hausknecht and Stone, 2016; Fan et al., 2019). Several solutions have been proposed to reconcile fixed input and output for RL agents and variable action spaces for RL environments. These solutions can be categorized as *masking*, where the agent is first informed of valid actions and then selects from this set, or *replacement*, where

(A) Masking: The subset of valid actions is given to an agent *before* it chooses its action.

(B) Replacement: Invalid actions are replaced by valid ones *after* an agent has chosen its action.

FIGURE 4.1: Intervention points for action space restrictions (cf. Krasowski et al., 2023).

an invalid action chosen by the agent is later replaced with a valid one (following some replacement strategy), as discussed by Krasowski et al., 2023 and illustrated in Figure 4.1. Note that we only consider the *environment perspective* here; of course, an agent can also mask or replace actions internally before outputting an action to the environment.

The most commonly used masking approach for discrete action spaces is invalid action masking, which employs a Boolean masking vector to provide the mask from the environment (Vinyals et al., 2017; Huang and Ontañón, 2022). There is, to the best of our knowledge, no analogous method for infinite or continuous spaces, such that continuous environments need to be discretized for masking (Uther and Veloso, 1998; Sinclair et al., 2020). As for replacement approaches, various alternatives have been proposed, including random replacement and projection (see Krasowski et al., 2023), and penalization (Dietterich, 2000). However, penalty-based RL methods have been shown not to scale well for a large number of invalid actions (Huang and Ontañón, 2022).

Numerous methods exist for handling irregular action spaces *within* an agent's action policy. Actor-critic methods (Konda and Tsitsiklis, 1999) can internally penalize the choice of invalid actions while ensuring that the final selected actions are valid. Dulac-Arnold et al., 2016 propose embedding discrete action spaces into continuous spaces using nearest-neighbor methods. Conversely, Tang and Agrawal, 2020 suggest discretizing continuous spaces for masking purposes. Zahavy et al., 2018 train an Action Elimination Network (AEN) to reduce the set of feasible actions, and Kanervisto, Scheller, and Hautamäki, 2020 enhance learning through action space shaping. Discarding invalid actions and re-sampling is another straightforward method that can be implemented either within an agent's action policy or as a feature of the environment (by setting $\delta(s, a^*) := s$ for any invalid action a^*). However, this method scales poorly when the ratio of invalid actions is high.

Recently, Grams, 2023 have proposed steps toward built-in consideration of action-space restrictions in RL algorithms. They use interval-union restrictions (see Equation (7.1) in Chapter 7) as part of the agent's observation and exploit this information in two distinct model architectures, called Parameterized Action Masking (PAM), based on the Parameterized DQN algorithm (Xiong et al., 2018), and Multi-Pass Scaled TD3 (MPS-TD3), based on TD3 (Fujimoto, Hoof, and Meger, 2018). However, this approach is quite restrictive with respect to the shape of action spaces, and its performance does not yet allow for the problem of restricted RL to be considered solved.

Existing RL libraries typically have limitations in supporting dynamic observation and action spaces due to the aggregation of trajectories into batches, which require homogeneous tensors (as explicitly mentioned, for example, in the documentations of Tianshou (Weng et al., 2022) and RLlib (Liang et al., 2018)). Padding is

often the only method used to handle heterogeneous data. Chapter 8 deals with the problem of integrating ARMAS with commonly used libraries, showing a flexible workaround for these limitations.

4.1.2 Multi-agent learning

Most of the MAS literature focuses on the agents' perspective, attempting to improve their cooperative learning behavior (see, e.g., the surveys of Nowé, Vrancx, and De Hauwere, 2012; Rizk, Awad, and Tunstel, 2018). As shown in Section 2.1, the underlying model, the Stochastic Game, can be derived as an extension of an MDP to multiple agents and as an extension of a Normal-Form Game to multiple states. Hence, methods from both Stochastic Processes and Game Theory have been adapted to this setting, mostly with additional assumptions that the proposed algorithms can exploit.

When agents have a common reward function—in other words, under a strong assumption of intrinsic cooperation—, Claus and Boutilier, 1998 show convergence to (optimal and suboptimal) Nash equilibria through suitable exploration strategies in (a simple form of) Q-learning. However, they do not provide any guarantees or theoretical bounds but instead, propose optimistic exploration strategies to "increase the likelihood of convergence to an optimal equilibrium". Doan, Maguluri, and Romberg, 2019 provide a finite-time analysis for the convergence of the distributed TD(0) algorithm when the communication network between the agents is time-varying. They obtain an explicit upper bound on the convergence rate as a function of the network topology and the discount factor.

In another cooperative setting with jointly observed state-action pairs and private local rewards, Wai et al., 2018 propose a double averaging scheme, where each agent iteratively performs averaging over both space and time to incorporate neighboring gradient information and local reward information, respectively. They prove that the proposed algorithm converges to the optimal solution at a global geometric rate.

For competitive multi-agent learning, Hoen et al., 2006 and Hernandez-Leal, Kartal, and Taylor, 2019 identify two main research streams: Game theoretical approaches, including auctions and negotiations, and Multi-Agent Reinforcement Learning. The latter adds a layer of complexity to classical Reinforcement Learning since competitive agents all evolve at the same time and, therefore, disturb the learning process of their opponents (*moving-target problem*) (Nowé, Vrancx, and De Hauwere, 2012).

Mazumdar and Ratliff, 2018 show that competitive agents can get stuck in periodic orbits. They introduce a new subclass of MAS, called *Morse-Smale games*, for which they can provide guarantees that competitive gradient-based learning almost surely does not get stuck at critical points.

Game theory for multi-agent learning often deals with small, well-defined (and mostly contrived) scenarios (Bade, 2005; Stirling and Felin, 2013; Gutierrez, Perelli, and Wooldridge, 2018) like two-player games with a fixed payoff matrix, which can be formally examined and sometimes also completely solved in terms of optimal responses and behavioral equilibria. What these solutions lack is widespread applicability to real-world settings where environments are large, information is incomplete, and agents do not behave nicely. Therefore, the gap between academic use cases, on the one hand, and industrial and societal applications, on the other hand, is still significant.

Again, there is vast literature for multi-agent learning from an agent's perspective (Sutton and Barto, 2018). Durugkar, Liebman, and Stone, 2020 specifically look at the balance between individual preferences and shared objectives but only consider cooperative agents.

When only a single agent in a multi-agent setting is considered, we must deal with non-stationarity (see Section 2.3.1). Facing a lack of stationary transition probabilities, multiple methods for model-free learning have been proposed, for example, Q-learning (Watkins, 1989), DQN (Mnih et al., 2015) and A3C (Mnih et al., 2016), but none of them comes with a convergence guarantee. In contrast, Lecarpentier and Rachelson, 2019 employ a model-based approach for non-stationary environments, assuming a continuous, bounded evolution of both transition and reward.

To solve the scalability issue, Majeed and Hutter, 2020 apply sequentialization to RL problems with large action spaces at the expense of an increased time horizon. Their technique of binarizing the action space into sequential decisions lends itself particularly well to spaces that are binary themselves, for example, all subsets of a fixed set. As we will see, this is the structure that we face in the governance learning problem if action spaces are discrete. Other methods for the reduction of state spaces or action spaces include ϵ-reduction (Dean, Givan, and Leach, 1997; Asadi and Huber, 2004) as well as exploitation of symmetry (Lüdtke et al., 2018) and policy structure (Liu, A. Chattopadhyay, and U. Mitra, 2019). Kim, M. E. Lewis, and C. C. White, 2005 apply such techniques to the problem of stochastic shortest paths, while Relund Nielsen, Jørgensen, and Højsgaard, 2011 use them for embedding biological state space models into an MDP.

An alternative is provided by many successful approaches to the multi-agent learning problem, which introduce new concepts for equilibria (e.g., correlated equilibria (Greenwald and Hall, 2003) and cyclic equilibria (Zinkevich, Greenwald, and Littman, 2005)) or make additional assumptions: Among others, agents can learn optimal strategies if all agents receive the same rewards (Team Markov Games (Wang and Sandholm, 2002)), if the game is a Zero-Sum Game (Littman, 1994), if all opponents are stationary (Conitzer and Sandholm, 2003), or if the "rate of non-stationarity" is bounded by a *variation budget* (Cheung, Simchi-Levi, and Zhu, 2020).

The general problem of finding an optimal strategy in a model-free, general-sum Stochastic Game, however, is still an open challenge (Zhang, Yang, and Basar, 2019). As a consequence, researchers have introduced additional support for the learning agents. This support can either directly apply to the interaction between the agents or involve another entity besides the agents. For the first type, agents are usually allowed to exchange additional information in order to find optimal strategies (Hwang, Jiang, and Chen, 2015; Cacciamani et al., 2021) (see also the recent MARL surveys of Zhang, Yang, and Basar, 2019 and Gronauer and Diepold, 2021). The second type is part of the next section.

4.2 Governance

Multi-agent learning, particularly MARL, is concerned with distributed agents achieving a common goal, which can either be given as a shared reward function or as a set of related reward functions. The learning and cooperative behavior resides within the agents or a specialized training process (e.g., training with a centralized critic who ensures cooperation and mitigates conflicts).

The concept of governance contrasts this approach, introducing an additional agent or supervisor in the multi-agent system[1].

4.2.1 The general governance problem

We have defined *governance* in Chapter 1 as any interference with a multi-agent system with the goal of achieving some system-level objective in addition to the agent objectives. This general idea has been brought up in the literature multiple times:

- Shoham and Tennenholtz, 1995 see the governance goal as "guaranteeing the successful coexistence of multiple programs", and note that it requires both a measure for "success" and an instance which can evaluate and possibly influence the degree of success. The authors make the point that designers of multi-agent systems can use social laws to make agents cooperate without formally controlling them. They describe an approach to defining such laws offline and keeping them fixed for the entire run-time of the system and mention the possibility that the agents do not always obey their laws.

- Weyns, Brückner, and Demazeau, 2007 pose the problem in the following way: "When designing a system that is based only on local interactions in the environment and the emergent properties resulting from these interactions, it is a difficult research problem on the one hand to obtain the required global behavior of the system and on the other hand to avoid undesired global properties". As a solution, they suggest to "off-load some of the agent complexity into the processes of the dynamic agent environment", calling their approach *Environment-Mediated Multi-Agent Systems* (EMMAS).

- Noriega and Jonge, 2016 propose *Electronic Institutions* (EI) as "coordination artefacts that serve as an interface between the internal decision making of individuals and their (collective) goals.". In contrast to conventional institutions, they envision electronic institutions "to work on-line and [...] involve the participation of humans as well as software agents".

None of these approaches necessarily requires a physically separate governance component; instead, they all propose a logical unit with governance capabilities to be added to a multi-agent system. Those capabilities can sit in a separate entity or be distributed to the agents, e.g., via norm-awareness within the agents' models (see Section 4.2.2 below).

The literature on multi-agent governance, based on the above problem statements, is marked by a number of related ideas and paradigms that are not mutually exclusive: *Normative Multi-Agent Systems* (NorMAS) focus on norms (or "social conventions") which guide agent behavior and can be violated by agents. *Mechanism Design* creates static rules, thereby defining the playing field on which the agents interact. *Self-organization* aims to equip agents with emergent coordination or self-governance capabilities. *Electronic Institutions*, as mentioned above, propose a multi-faceted regulation instance, which includes both normative and restrictive features. *Reward shaping* addresses the agents' rewards to change their behavior. Finally, *restrictions* define hard constraints on the agents' behavior without room for violations.

[1] We call the governance entity an *agent* because it satisfies the characteristics of an autonomous agent as defined in Chapter 1. However, its actions have a particular structure and meaning; namely, it governs the MAS by intervening in the agent-environment loop.

Many works touch more than one of these paradigms, bridging gaps between some and clearly separating others. Therefore, we follow here the structure of the broad research streams, pointing out connections and differences.

4.2.2 Norms and normative systems

Norms are a very common approach for achieving system goals in MAS. The distinction between norms and rules—for example, Balke et al., 2013 state that "[norms] are a concept of social reality [...] Therefore, it is possible to violate them"—has been made many times in the literature. They have been called "social conventions" and "explicit prescriptions" (Conte, Falcone, and Sartor, 1999), "legalistic view of norms" and "interactionist view of norms" (Boella, Torre, and Verhagen, 2008), "norms" and "regimented norms" (Balke et al., 2013), "norms" and "hard constraints" (Frantz and Pigozzi, 2018; Mellema, Jensen, and Dignum, 2021), or "hard norms" and "soft norms" (Rotolo, 2011; Rotolo and Torre, 2011).

Normative Multi-Agent Systems (Conte, Falcone, and Sartor, 1999; Boella, Torre, and Verhagen, 2006; Andrighetto et al., 2013) embrace the idea that agent communities can self-regulate their interactions without a controlling force. Therefore, the field focuses on (violable) norms, their creation or emergence, observation, revision, adherence or violation, and sanctioning mechanisms. Rotolo and Torre, 2011 argue that "achieving compliance by design can be very hard" due to various reasons, among them norm consistency and enforcement complexity. In their view, NorMAS are therefore more suitable for open and distributed environments. The lack of hard obligations requires alternative concepts like sanctions, norm revision, norm conflict resolution, and others. NorMAS have been researched from various perspectives and with a host of theoretical frameworks, among them formal languages and logics (García-Camino et al., 2006; Bulling and Dastani, 2016; Perelli, 2019), Bayesian networks for the analysis of effectiveness (Dell'Anna, Dastani, and Dalpiaz, 2019), bottom-up norm emergence (Morris-Martin, De Vos, and Padget, 2019) and on-line norm synthesis (Morales, 2016). Whether normative rewards and sanctions are imposed onto the agents from an outside entity (Morales et al., 2013; Neufeld et al., 2021) or emerge from within the agent community (Morris-Martin, De Vos, and Padget, 2021), there is always a need for the agents to be *norm-aware* and to use normative capabilities in their action policy (Cramton, 2006).

Most authors working on NorMAS follow the convention that norms are "a concept of social reality [which does] not physically constrain the relations between individuals. Therefore it is possible to violate them." (Balke et al., 2013). However, this convention is far from being unambiguous; for instance, Perelli, 2019 use the term "Normative Synthesis" for the *enforcement* of certain equilibria.

Like general multi-agent learning capabilities, normative capabilities in MAS can either be part of the agents (Riad and Golpayegani, 2021), or part of an additional entity (Aires and Meneguzzi, 2017), or both. While early work defined static norms at design-time (Shoham and Tennenholtz, 1995; Barbuceanu, 1997), the field has since evolved towards run-time norm creation, synthesis and adaptation (Morales, 2016), applying methods like Automated Theorem Proving (Neufeld et al., 2021) or Deep Learning (Aires and Meneguzzi, 2017) to NorMAS.

4.2.3 Electronic Institutions

Electronic Institutions (EI) (Noriega, 1997; Esteva et al., 2001) propose an *institution* as the entity which regulates agent interactions, among many other features. They

do not commit themselves to using either norms or rules but provide support for both approaches, described as an "implementation of the control functionality of the institution infrastructure [which] takes care of the institutional enforcement".

The EI framework itself does not only describe rule-setting capabilities but also agents, *roles*, a *performative structure*, and *normative rules*, and other features (Esteva et al., 2001). The same holds for alternative models for social coordination, e.g., ANTE (Lopes Cardoso et al., 2016), or INGENIAS (Gomez-Sanz and Fuentes Fernandez, 2016); details of all these frameworks can be found in Aldewereld et al., 2016.

The original implementation of EI and its development environment EIDE (Noriega and Jonge, 2016) envisaged a clear distinction between rule/norm creation at design-time and agent interaction at run-time (i.e., all rules/norms are given independently of the agents and do not change during execution). A logical next step was the Autonomic Electronic Institutions (AEI) approach (Bou, López-Sánchez, and Rodríguez-Aguilar, 2007; Arcos, Rodríguez-Aguilar, and Rosell, 2008): Acknowledging the fact that static norms are not always sufficient for dealing with self-adapting agents, it moved norm creation from the design time to the run-time and allowed for dynamic changes. EI was therefore extended to include an evolutionary norm adaptation mechanism (e.g., a genetic algorithm).

4.2.4 Reward shaping

Normative systems and reward shaping are closely connected by NorMAS' tenet that norms can be violated and, therefore, require a structure of rewards and sanctions as the consequences for obeying and disregarding a norm, respectively.

In general, reward shaping (Mataric, 1994) addresses the agents' rewards to change their behavior, relying on the fact that maximizing the expected new reward will result in a different action policy. Centralized reward shaping can follow the structure of, for example, a Vickrey–Clarke–Groves (VCG) mechanism (Nisan and Ronen, 2004). However, instead of letting the agents optimize their policies, a VCG mechanism performs the outcome selection itself, computing concrete best actions for the agents. This leads to well-known computability issues, for example, when solving the NP-hard problem of optimal allocation in a combinatorial auction. The VCG-based method of Marginal-Cost Pricing (MCP) (Turvey, 1969) has been successfully applied to Braess' Paradox (Ding and Song, 2012) and real-world traffic networks (Sharon et al., 2017b; Sharon et al., 2017a; Sharon et al., 2018; Sharon et al., 2019; Hanna et al., 2019). While presenting promising theoretical and experimental results, these solutions rely on the assumption that agents' utility functions can be manipulated in a discriminatory way. Moreover, the effectiveness of rewards and sanctions fundamentally depends on the agents' susceptibility to this kind of (dis-)incentives, rendering the approach useless when agents simply do not react to sanctions whatsoever.

4.2.5 Restrictions

The ability to restrict the space of available actions for the participating agents has been described as an essential part of an Electronic Institution: "An electronic institution defines a set of rules that structure agent interactions, establishing what agents are permitted and forbidden to do" (Esteva et al., 2008). Aldewereld et al., 2016 emphasize that "organisational objectives are not necessarily shared by any of

the individual participants, but can only be achieved through their combined action", and that "one cannot make any assumptions about the inner workings of participants. [...] Rather, external aspects of the participants (actions, interactions, etc.) have to be leveraged to create the required coordination structures".

Mittelmann et al., 2022 propose a logic-based method (*Automated Synthesis of Mechanisms*) for automated mechanism design but focus on optimizing the transition function while keeping the action space fixed. This is, in a way, a complementary approach to restriction-based governance. Kanervisto, Scheller, and Hautamäki, 2020 use action space shaping to improve learning, focusing on a single agent's observation and action spaces in video games such as Atari, StarCraft, and Dota. Kalweit et al., 2021 shape the action space of a DQN agent in the domain of autonomous driving by defining a cost function for actions and then restricting the action space to those whose cost is below a fixed threshold. A similar approach is used by Achiam et al., 2017 to directly shape the policy space of an RL agent. Tang, 2017 and Cai et al., 2018, on the other hand, use *Reinforcement Mechanism Design* to automate the design of e-auctions, restricting bidders' actions based on past behavior. Therefore, their restrictions are imposed from an outside entity (as in our approach) but not optimized over a utility function.

The development in NorMAS towards dynamic, objective-driven norm creation and adaptation has, to our knowledge, not yet taken place for rules (i.e., hard constraints): Prior to our own work (Pernpeintner, Bartelt, and Stuckenschmidt, 2021, see Chapter 5), we are not aware of any work on action space shaping as a means of aligning user equilibria and social optima in a multi-agent setting.

4.2.6 Fairness

The notion of fairness in algorithmic decision-making (Barocas, Hardt, and Narayanan, 2019; Mehrabi et al., 2021) has become an important metric next to traditional performance indicators like reward, loss, or prediction error, particularly for mechanisms whose output directly affects people. It emerged as an important requirement to guarantee that ML predictive systems do not discriminate against specific individuals or entire sub-populations, particularly minorities (Makhlouf, Zhioua, and Palamidessi, 2020).

While *fairness by unawareness* (Grgic-Hlaca et al., 2016) emphasizes the importance of a fair process, this does not imply a fair (i.e., balanced) outcome. The concept of *equity*, defined by the World Health Organization (WHO) as "the absence of unfair, avoidable or remediable differences among groups of people" (World Health Organization, 2023) and widely used in the context of health (Rajkomar et al., 2018; Rychetnik et al., 2002), can help judge the fairness of an algorithm or a governance mechanism by evaluating the impact the mechanism has on different groups.

Oneto and Chiappa, 2020 discuss the limitations of current methods for ensuring fairness in machine learning and propose using causal Bayesian networks and optimal transport theory to address these limitations. Corbett-Davies and Goel, 2018 argue that formal definitions of fairness, such as anti-classification and classification parity, suffer from statistical limitations and propose treating similarly risky people in a similar way based on accurate risk estimates. Barrio, Gordaliza, and Loubes, 2020 review fairness definitions and methodologies from a mathematical perspective, focusing on the performance degradation in fair algorithms compared to possibly unfair ones. Joseph et al., 2016 propose a technical definition of fairness modeled after Rawls' notion of "fair equality of opportunity" and present an algorithm that

satisfies this constraint while still being able to learn at a comparable rate to non-fair algorithms.

4.3 Braess' Paradox

Restrictions as a means of improving social welfare are widely found in traffic management and, more specifically, traffic routing in congested graph networks. Braess' Paradox is *the* example of social welfare improvement through restriction of multi-agent systems, and it has been extensively studied from the perspectives of network design, graph theory, game theory, and others after it had been first described by Braess, 1968. Most early (and some later) work focuses on the original four-node network structure, examining criteria for the occurrence of the paradox in terms of latency functions and traffic rate (Pas and Principio, 1997; Penchina, 1997; Zverovich and Avineri, 2012). As a second focus area, Roughgarden and Tardos, 2002 show that the *price of anarchy*, defined as the ratio between the user equilibrium and the social optimum and therefore an upper bound for the improvement achievable through edge restrictions, is $\leq \frac{4}{3}$ for affine latency functions, regardless of the underlying graph. For general latency functions, particularly for polynomials of unlimited degree, Lin et al., 2011 demonstrate that Braess' Paradox can be arbitrarily severe, and Roughgarden, 2006 prove various inapproximability and hardness results for the problem of identifying the edges causing the paradox. A third line of research deals with the occurrence of Braess' Paradox in random and real-world networks: Steinberg and Zangwill, 1983 derive a likelihood of 50% via a non-constructive proof, and Valiant and Roughgarden, 2010 argue that for large Erdős-Rényi graphs with certain assumptions on edge density and latency functions, the paradox occurs with high probability *for some carefully chosen traffic rate*. At the same time, Pas and Principio, 1997; Friedman, 2004 provide evidence suggesting that Braess' Paradox is much less likely to occur with randomly chosen traffic rates compared to adversarial rates.

Single-commodity networks (i.e., there is exactly one source-sink pair) with constant traffic rates allow for a static solution while changing demand and multiple commodities can require adaptive strategies. The multi-commodity case of Braess' Paradox, albeit with constant traffic rates, has been examined in Roughgarden and Tardos, 2002; Roughgarden, 2006; Lin et al., 2011; Eickmeyer and Kawarabayashi, 2013. It turns out that the worst-case behavior of congested networks can be much worse than in a single-commodity scenario: The price of anarchy for the maximum latency objective can grow exponentially with the network size (Lin et al., 2011).

While most of the research on Braess' Paradox models the traffic flow on a macroscopic level, it can also be shown to occur in microscopic (Bazzan and Klügl, 2005; Bittihn and Schadschneider, 2021) and mesoscopic (Pala et al., 2012) models.

Chapter 5

Finding optimal restrictions via action elimination

In this chapter, we look at a MAS with discrete action spaces and learn a governance policy using a frequentist approach: The governance observes the joint actions taken by the agents, takes this distribution as a predictor for future actions, and successively eliminates low-utility actions from the joint action space until the (probability-weighted) expected utility exceeds a given threshold.

Personal Contribution. *I defined the model, built the algorithm, designed and conducted the experiments, and was the sole author of the text. The definition of the research question and the experimental domain selection were undertaken jointly with Christian Bartelt and Heiner Stuckenschmidt.*

5.1 Motivation

One of the most intriguing and challenging characteristics of a MAS is the fact that its environmental changes depend simultaneously on the actions of all agents, such that a single agent can never simply choose an action and confidently predict the resulting transition. This mutual influence leads to strategic and sometimes even seemingly erratic agent actions—particularly when human agents are involved—, and at the same time decouples *intended* and *observed* system behavior:

Example 1. *Consider a MAS consisting of two agents X and Y, two states A (the initial state) and B, and two actions* 0 *and* 1 *for each agent. This results in the joint action space $\boldsymbol{A} = \{00, 01, 10, 11\}$, where the joint action* 10 *means that the first agent, X, takes action* 1*, while the second agent, Y, takes action* 0*. The transition function of this MAS is shown in Figure 5.1.*

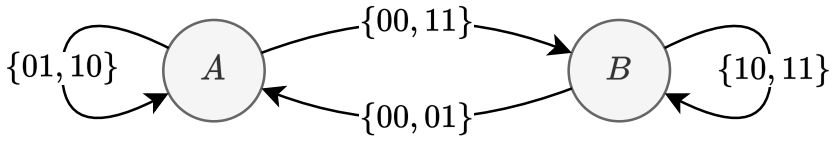


FIGURE 5.1: Transition graph of the MAS defined in Example 1.

Imagine now an observer who sees the following sequence of actions and transitions:

$$A \xrightarrow{10} A \xrightarrow{01} A \xrightarrow{00} B$$

Suppose the observer does not know anything about the inner workings of X and Y. In this case, it cannot distinguish whether X wanted to stay at state A and changed its action from 1 *in the first step to* 0 *in the second step because it anticipated Y's second action*

or X wanted to reach state B, observed the uselessness of its first action and then tried another strategy to reach B (and failed again). This shows that intentions are not immediately linked to observable behavior, and no preference order over the environmental states can be concluded with certainty.

Several existing methods like, for instance, preference elicitation using CP-nets (Koriche and Zanuttini, 2010), rely on the fact that preferences can *be observed. However, this requires additional assumptions about the link between actions and preferences.*

In this chapter, we propose a governance approach that uses action space restrictions to achieve a system goal, only basing its policy on observed (joint) actions without trying to deduce the agents' goals or preferences. By assumption, the agents are purely self-interested and strategically pursue their (confidential) individual goals without an inherent desire for cooperation.

Throughout the chapter, we use a smart home scenario for both illustration and evaluation. In the simplest case (Example 2), we only observe the agents' behavior, while later on we show how the governance can intervene in this system (Examples 3 to 5).

Example 2. *Consider a smart home environment consisting of seven binary variables that fully describe the system's state:*

$$\mathcal{S} = T \times O \times W \times B \times H \times L \times A \,,$$

where the variables denote Time (day/night), Occupancy (occupied/empty), Window (open/closed), Blinds (open/closed), Heating (on/off), Lights (on/off), and Alarm (on/off), respectively. The agents, who each have their individual preferences over the environmental state, can now choose to change at most one of the variables W, B, H, L or A at each step (they cannot, however, influence the Time or the Occupancy of the house).

Assume that there are three agents acting upon the environment with identical action sets $A_i = \{\varnothing, w, b, h, l, a\}\ \forall i \in I$. Time and Occupancy would, of course, be controlled by external forces, but this is omitted here for simplicity.

An exemplary progression of this system could be

$$\begin{aligned} 1100101 &\xrightarrow{wa\varnothing} 1110100 \xrightarrow{blb} 1111110 \\ &\xrightarrow{\varnothing\varnothing h} 1111010 \xrightarrow{hlb} 1110100 \xrightarrow{bwl} 1101110 \,, \end{aligned}$$

where states are written as binary numbers, and transitions, together with the respective chosen actions, connect subsequent states.

While total control on the part of an outside authority contradicts the multi-agent property of such a system, some level of control and cooperation can still be achieved by a suitable governance approach. Under the assumption that there are some "global desirable properties" (Rotolo and Torre, 2011) which are to be fulfilled in addition to the natural, uncontrolled agent behavior, the governance works toward this objective without destroying the multi-agent property of the system.

The simplest way of achieving a system objective would be, of course, to set fixed rules that have to be obeyed by all agents, as in off-line rule design (Shoham and Tennenholtz, 1995). While this can be an effective approach, it necessarily suffers from at least one of two drawbacks: Either the agents are so heavily constricted that they lose their autonomy altogether (Fitoussi and Tennenholtz, 2000), or the system is unable to cope with unforeseen strategies dynamically. Therefore, we propose to make use of the knowledge that can be collected by observing how agents behave in the system in order to update and refine the governance interventions successively.

Following the line of thought given in Example 1, we do not reason in terms of agent preferences or utilities, but rather in terms of actions and transitions. Naturally, there is a conflict between control and autonomy, requiring a relative weighting of the two objectives. We strive here for minimal restriction, subject to a constraint on the expected value of the system objective.

This chapter proposes a first solution method for the governance learning problem in an ARMAS, i.e., for optimizing the restriction function ρ with respect to the governance utility u. The high-level approach is shown in Figure 5.2: The governance component observes agent actions and subsequent state transitions to refine its predictions about future agent actions in real time. The current prediction is stored as an internal governance state, which, at each time step, is used to compute the optimal restriction. By looking only at the observable behavior, we avoid the fallacies described in Section 5.1, which arise when directly concluding preferences over the environmental states.

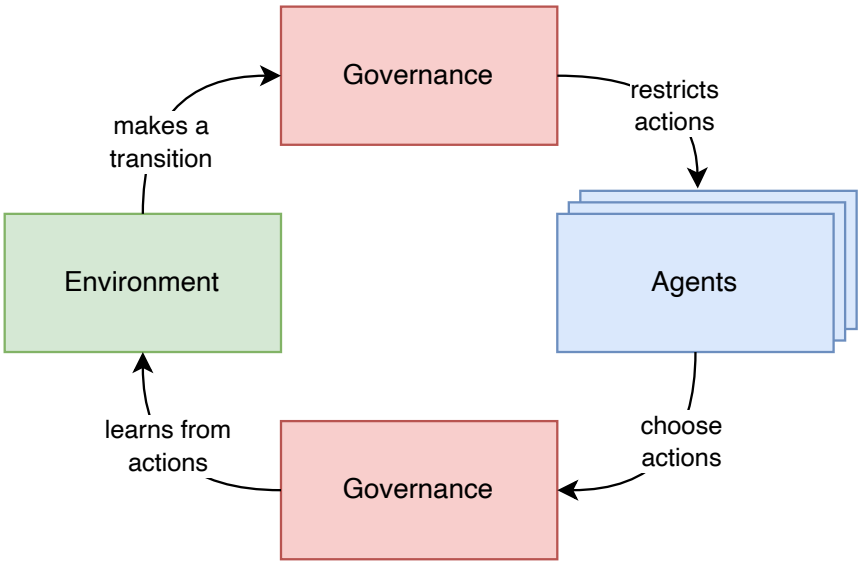


FIGURE 5.2: Governance loop, showing the sequence of acting (i.e., restricting) and learning intertwined with the MAS.

We present a practical approach and corresponding algorithm to immediately turn observations about the history of the MAS into suitable restrictions, such that the governance utility is maximized while agent autonomy is preserved as much as possible. Our proposed algorithm focuses on multi-attribute MAS with binary attributes, but the results carry over quite naturally to attributes with arbitrary finite domains since the model does not assume any particular structure of agents and environment.

The experimental evaluation in Section 5.3 indicates that the approach yields an effective governance and indeed avoids the problem mentioned above induced by "observing preferences".

5.2 Governance approach

5.2.1 Model assumptions

We limit our investigation here to binary multi-attribute environments, i.e., $\mathcal{S} = \mathbb{B}^m$[1] for some fixed $m \in \mathbb{N}$. For consistency reasons, each agent i has a *neutral action* $\varnothing_i \in A_i$ which cannot be deleted from the set of allowed actions. Agents can change one attribute per time step (or choose the neutral action), and an attribute is toggled when at least one agent chooses to change it[2].

5.2.2 Governance utility

There are two common types of system objectives: Either minimize (or maximize) a numerical value, which can directly be expressed by $\mathfrak{u}$, or distinguish between *valid* states $\mathcal{S}_+$ and *invalid* states $\mathcal{S}_- := \mathcal{S} \setminus \mathcal{S}_+$. In the latter case,

$$\mathfrak{u}(s) := -\mathbb{1}_{\mathcal{S}_-}(s) = \begin{cases} 0 & \text{if } s \in \mathcal{S}_+ \\ -1 & \text{if } s \in \mathcal{S}_- \end{cases} \tag{5.1}$$

describes a system objective that strictly prefers all valid states over all invalid states but makes no further distinction.

In addition to this governance utility function, we make one more assumption which relates to the discussion of autonomy and restriction from Section 1.2.5:

Assumption 1. *Between policies that yield the same (expected) governance utility, it is desirable to choose one that imposes minimal restrictions on the agents.*

In adherence to this assumption, the governance we propose will pursue a valid state with minimal restriction of the agents.

Example 3. *As a continuation of Example 2, consider an ARMAS where* $\mathfrak{u}$ *is defined as in Equation* (5.1) *with*

$$\mathcal{S}_+ = \left\{ s \in \mathcal{S} : \left(\overline{w}(s) \vee \bar{h}(s)\right) \wedge (a(s) \vee o(s)) \wedge \left(\bar{l}(s) \vee o(s)\right) \right\} ,$$

meaning that the governance wants to make sure that (a) the window is not open while the heating is turned on, (b) the alarm is on when the house is empty, and (c) the lights are off when there is nobody at home.

It is now the task of the governance to impose minimal restrictions on the agents while keeping $s_t \in \mathcal{S}_+ \; \forall t \geq 0$.

5.2.3 Governance state

The governance's knowledge about past agent behavior is stored in a data structure similar to a Q-table (Watkins, 1989), such that acquisition of new knowledge from observations as well as conclusions about conflicts and optimal action restrictions can be performed as part of an on-line governing cycle.

Let n be the number of agents, m the number of binary attributes, and k the number of actions for each agent (we assume the same fundamental action space for all agents). Then the governance state space is $\mathfrak{S} := \mathbb{N}_0^{n \times 2^m \times k}$, i.e., a simple counter of observed actions per agent per environmental state. Note that the second index of

[1] $\mathbb{B} := \{0, 1\}$ denotes the *Boolean field*.

[2] This setup corresponds to the *deterministic* subclass of ARMAS, as defined in Section 3.2.2.

the governance state can naturally be identified with an environmental state s since $\mathcal{S} = \mathbb{B}^m \cong \{0, ..., 2^m - 1\}$.

This gives rise to an (observed) probability distribution

$$P_i^{(t)}(s) := \left(\frac{\mathfrak{s}_{(i,s,1)}}{\mathfrak{s}_{(i,s)}}, ..., \frac{\mathfrak{s}_{(i,s,k)}}{\mathfrak{s}_{(i,s)}} \right) \in \mathbb{P}_k \text{ , where } \mathfrak{s}_{(i,s)} = \sum_{j=1}^{k} \mathfrak{s}_{(i,s,j)}$$

for each agent i and environmental state s, which reflects the knowledge about the agents' past actions up to step t and thus contains the governance's best guess for the actions at step $(t+1)$. It is customary to set $P_i(s) := \left(\frac{1}{k}, ..., \frac{1}{k}\right)$ if $\mathfrak{s}_{(i,s)} = 0$, or to use another initial distribution[3].

Example 4. *In the setting of Example 3, the governance state $\mathfrak{s}$ is a three-dimensional matrix of size $3 \times 2^7 \times 6$ (i.e., agents $\times$ states $\times$ actions). Slicing this matrix along its second axis, i.e., at a specific environmental state, gives a (3×6) matrix; at step $t = 5$, the transition sequence from Example 2 would result in*

$$\mathfrak{s}_{(5)}\,(\square, 1110100, \square) = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

and consequently

$$\begin{aligned} P_1(1110100) &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \\ P_2(1110100) &= \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \end{pmatrix} \\ P_3(1110100) &= \begin{pmatrix} 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix} . \end{aligned}$$

5.2.4 Observation and learning

As soon as all agents have made their choice of action $\boldsymbol{a}^{(t)} = (a_i^{(t)})_i \in \boldsymbol{R}^{(t)} = \boldsymbol{\rho}^{(t)}(s_t)$ and the environment has proceeded to the next state s_{t+1}, the governance can use the newly observed transition $(s_t, \boldsymbol{a}^{(t)}, s_{t+1})$ to learn about the agents and the effectiveness of the restrictors $\boldsymbol{\rho}^{(t)}$. This learning step is expressed as an update of the governance's internal state $\mathfrak{s}$, which, in turn, will be used by $\boldsymbol{\rho}$ in the next step.

Writing the *governance learning function* as $\lambda : \mathfrak{S} \times S \times \boldsymbol{A} \to \mathfrak{S}$, we have

$$\mathfrak{s}^{(t+1)} = \lambda \left(\mathfrak{s}^{(t)}, s_t, \boldsymbol{a}^{(t)} \right) .$$

Specifically, λ tracks the actions chosen by the agents by incrementing the respective elements of the governance state after observing the tuple $(s_t, \boldsymbol{a}^{(t)})$:

$$\lambda(\mathfrak{s}, s_t, \boldsymbol{a}^{(t)}) = \mathfrak{s}', \text{ where } \mathfrak{s}'_{(i,s,j)} := \begin{cases} \mathfrak{s}_{(i,s,j)} + 1 & \text{if } a_i^{(t)} = j \wedge s_t = s \\ \mathfrak{s}_{(i,s,j)} & \text{else} \end{cases} .$$

[3] $\mathbb{P}_k := \left\{ x \in \mathbb{R}^k, 0 \le x_i \le 1, \|x\|_1 = 1 \right\}$ denotes the set of probability vectors with k elements. Similarly, let $\mathbb{P}_k^n$ be the set of n-dimensional matrices with size k in each dimension, whose entries lie within $[0, 1]$ and sum up to 1.

5.2.5 Restriction of action spaces

We make two independence assumptions regarding the probability of choosing an action: First, the relative probability of choosing an action a_i over another action b_i does not change when a third action is forbidden:

Assumption 2. *Let $R_i, R_i' \subseteq A_i$ be two restrictions. Then*

$$\forall a_i, b_i \in R_i \cap R_i' : \ \frac{P_{R_i}(a_i)}{P_{R_i}(b_i)} = \frac{P_{R_i'}(a_i)}{P_{R_i'}(b_i)} \in \mathbb{R} \cup \{\infty\} \ .$$

Therefore, we can remove individual actions from $P_i(s)$ and still have a valid distribution (up to normalization) for the remaining actions.

Second, agents exclusively communicate by observing each other's actions, such that their actions are independent of each other at a single time step. Interactions between agents, therefore, require at least one step between an action and the corresponding reaction.

Assumption 3. *Let $P_i^{(t)}(s)$ be agent i's action probability distribution for state s at time t. Then*

$$P^{(t)}(s) = \prod_i P_i^{(t)}(s) \in \mathbb{P}_k^n$$

is the probability distribution for the joint action of all agents.

This product rule holds for accurate knowledge about the probabilities as well as the governance's estimate thereof.

5.2.6 Algorithm

Putting the above considerations into action, we can now present and analyze an algorithm to integrate observations into the current knowledge (learning step) and then turn this knowledge into a minimal restriction of action sets (restriction step).

The governance loop (see Algorithm 1) works as follows: The n-dimensional matrix $P(s)$ of state $s \in \mathcal{S}$ represents a function $P(s) : \boldsymbol{A} \to \mathbb{R}$ which assigns to each joint action $\boldsymbol{a} \in \boldsymbol{A}$ the (expected) probability of being chosen. Element-wise multiplication of the matrix with the utility values of the resulting states gives an *expected cost matrix* $C(s) \in \mathbb{R}^{k^n}$ with entries

$$C(s)_{\boldsymbol{a}} := \mathfrak{u}\left(\delta(s, \boldsymbol{a})\right) \cdot \prod_{i \in I} P_i(s)_{a_i} \ \forall \boldsymbol{a} \in \boldsymbol{A} \ ,$$

where an action $\boldsymbol{a}$ is identified with its n-dimensional position in $C(s)$.

Each hyperplane of $C(s)$ along axis i corresponds to an action of agent i, and the sum of entries at this hyperplane is the expected cost of agent i taking the respective action. We can, therefore, see from the expected cost matrix which actions from which agents have the highest expected costs. The matrix $C(s)$ can now be reduced by successively removing maximum-cost hyperplanes (each corresponding to an action by an individual agent) until the sum of all remaining entries drops below a given cost threshold $\alpha \in \mathbb{R}$. Forbidding the removed actions ensures that the expected cost in the next step is less or equal to α and that no unnecessary restrictions are made. Note that α can be chosen arbitrarily, as long as it is large enough to allow the neutral action to be selected (see Section 5.2.1). The value of α defines the balance between optimizing restriction and cost.

For an expected cost matrix C and a subset $\boldsymbol{R} \sqsubseteq \boldsymbol{A}$ of joint actions, we write $\|C\|_{\boldsymbol{R}} := \sum_{\boldsymbol{a} \in \boldsymbol{R}} C_{\boldsymbol{a}}$ for the cumulative cost of actions in $\boldsymbol{R}$ (this is equivalent to the element-wise sum norm of C, restricted to entries corresponding to actions in $\boldsymbol{R}$). As a shorthand for the full matrix, we use $\|C\| := \|C\|_{\boldsymbol{A}}$. Moreover, we use the element-wise product $X \circ Y := (X_i \cdot Y_i)_i$ for matrices X, Y with the same shape, and the component replacement $(\boldsymbol{x}_{-i}, y_i) := (x_1, x_2, ..., x_{i-1}, y_i, x_{i+1}, ..., x_n)$ for vectors $\boldsymbol{x}, \boldsymbol{y}$.

Algorithm 1: Restricting agent actions via successive elimination

Data: Governance utility function $\mathfrak{u}$, joint action space $\boldsymbol{A}$, cost threshold α
Input: Probability distributions $P_i(s_t)$ for the actions of all agents at the current state s_t
Output: Restricted action space $\boldsymbol{R}$

$P(s_t) := \prod_i P_i(s_t) \in \mathbb{P}_k^n$;
$C := P(s_t) \circ \mathfrak{u}(s_t) \in \mathbb{R}^{k^n}$;
$\boldsymbol{R} := \boldsymbol{A}$;
while $\|C\|_{\boldsymbol{R}} > \alpha$ **do**
  $(i, j) := \arg\max_{\boldsymbol{a} \in \boldsymbol{R}, a_j \in R_i \setminus \{\varnothing\}} C_{(\boldsymbol{a}_{-i}, a_j)}$;
  $R_i := R_i \setminus \{a_j\}$;
  Slice C to remove the corresponding hyperplane;
end

Theorem 1. *Let C be an expected cost matrix, and assume that $\|C\|_{\{\varnothing\}} \leq \alpha$ for some $\alpha > 0$. Then Algorithm 1 produces a joint restriction $\boldsymbol{R} \sqsubseteq \boldsymbol{A}$ of actions such that*

$$\|C\|_{\boldsymbol{R}} \leq \alpha \,. \tag{5.2}$$

This restriction is Pareto minimal, i.e., $\nexists \boldsymbol{R}' \sqsupset \boldsymbol{R}$ with the same property.

Proof. Termination and threshold: At each step of the **while** loop, an action (which is not the neutral action) is removed for one of the agents. Therefore, the loop exits after at most $n \cdot (k-1)$ passes. Since $\alpha \geq \|C\|_{\{\varnothing\}}$, the cost is guaranteed to fall below α at some point, and the loop does not break until this has happened. Hence, Equation (5.2) is satisfied at the end of the algorithm.

Minimality: Let C^* be the cost matrix corresponding to $\boldsymbol{R}$ as returned by Algorithm 1, and assume that $C^* \neq C$. Then C^* was derived from C by successively deleting hyperplanes, i.e., individual actions $a_j \in A_i$. Let such a deleted action be denoted by the two defining indices (i, j) (the j-th action of agent i). Then, the sequence of deletions can be written as

$$C = C_0 \xrightarrow{(i_1, j_1)} C_1 \to \cdots \to C_{x-1} \xrightarrow{(i_x, j_x)} C_x = C^* \,,$$

where $x > 0$, $\|C_{x-1}\| > \alpha$ and $\|C_x\| \leq \alpha$.

Assume now that there is a restriction $\boldsymbol{R}' \sqsupset \boldsymbol{R}$ whose expected cost matrix C' satisfies $\|C'\| \leq \alpha$. Then $\exists y \leq x$ such that action (i_y, j_y) lies in $\boldsymbol{R}'$. From $\|C_{x-1}\| > \alpha$ we can conclude that $y \neq x$ (otherwise C' would be equal to C_{x-1}, which is a contradiction) and therefore $y < x$.

This means that (i_y, j_y) was removed from C before (i_x, j_x), thus $C_{(i_y, j_y)} \geq C_{(i_x, j_x)}$ and consequently

$$\|C\|_{\boldsymbol{R}} = \|C'\| > \|C_{x-1}\| > \alpha \,,$$

contradicting the above assumption. □

If $\mathfrak{u}$ has the structure of Equation (5.1)—valid states have utility 0, invalid states have utility -1—, then α is precisely an upper bound for the probability of ending up at an invalid state.

Example 5. *Coming back to Example 4 one last time, we see that* $s_1 = 1110100$ *incurs utility* $\mathfrak{u}(s_1) = -1$ *since* $s_1 \notin \mathcal{S}_+$*. While the governance probably cannot anticipate and prevent this transition between* $t = 0$ *and* $t = 1$ *due to lack of experience, it might be able to do so at a later time when enough information has been gathered. For example, at* $t = 3$*, the governance could forbid action* $h \in A_1$ *such that the joint action hlb cannot be taken. If agent 1 now chooses action w instead,* $s_4 = \delta(s_3, wlb) = 1100000 \in \mathcal{S}_+$*, and the governance has successfully prevented an undesirable transition with this restriction.*

5.2.7 Computational complexity

The time complexity of Algorithm 1 with regard to the input sizes n, m and k can be straightforwardly derived from the pseudo-code:

- Initialization of P, C, and $\boldsymbol{R}$: $\mathcal{O}(k^n) + \mathcal{O}(k^n) + \mathcal{O}(kn)$
- **while** loop: $\mathcal{O}(kn)$ passes
  - Checking the break condition: $\mathcal{O}(k^n)$
  - Finding the arg max: $\mathcal{O}(kn \cdot k^n)$
  - Reducing $\boldsymbol{R}$: $\mathcal{O}(1)$
  - Reducing C: $\mathcal{O}(k^n)$

Altogether, this results in a worst-case time complexity of $\mathcal{O}\left(n^2 \cdot k^{(n+2)}\right)$.

5.3 Evaluation

To test the validity and efficacy of our approach, we compare unrestricted and restricted runs of the smart home use case from Examples 2 to 5: In the unrestricted case, agents simply act according to their action policies, having the full range of actions at their disposal all the time. The restricted case adds a governance that employs the governance loop from Section 5.2.

5.3.1 Evaluation metrics

There is a natural trade-off between achieving the system objective and preserving agent freedom: The more actions the governance forbids, the higher its level of control over the agents—in the extreme case, only a single action is allowed for any given observation, resulting in a fully deterministic trajectory[4]. On the other end of the spectrum, the governance always allows all actions, which means there is no improvement of the governance utility.

In addition to the change in governance utility with and without restrictions, it is therefore reasonable to measure the *degree of restriction*, i.e., the relative number of forbidden actions:

[4] If the restriction policy is optimal, the prescribed joint action is indeed a governance optimum.

Definition 11. *For an individual agent $i \in I$ and time step $t \in \mathbb{N}_0$, the* degree of restriction *is defined as*

$$\mathfrak{r}_i^{(t)} := 1 - \frac{\left|\rho_i^{(t)}(s_t)\right|}{|A_i|} \in [0,1]\,.$$

The overall degree of restriction $\mathfrak{r}^{(t)} := \frac{1}{|I|}\sum_{i\in I}\mathfrak{r}_i^{(t)}$ is simply the mean over all agents. The higher the degree of restriction, the lower the agents' autonomy.

Remark 5. *For finite action spaces, as in the present setting, the function $|\square|$ in Definition 11 can simply be the cardinality (i.e., the number of elements) of the respective sets. In Chapter 7, we will consider continuous action spaces and, therefore, need to redefine the degree of restriction, using a suitable measure on these spaces.*

Remark 6. *It should be noted that real-world agents sometimes cannot choose every action at every step due to physical or other environmental constraints. Instead, only a subset of actions is* feasible*, depending on the environmental state (*parametric action spaces*). In this case, the degree of restriction should be defined as the ratio between forbidden and feasible actions.*

The average cost over time is shown for unrestricted and restricted simulations, while the degree of restriction only applies to the restricted case.

5.3.2 Setup

We consider two scenarios with different types of randomly chosen but fixed agent action policies: In the deterministic case, each agent i has a fixed mapping of states and actions, i.e., an action policy $\pi_i : \mathcal{S} \to A_i$. In the stochastic case, each agent has a probability distribution for its actions for every state, i.e., an action policy $\pi_i : \mathcal{S} \to \Delta_{A_i}$.

For each of the two scenarios, we run the simulation with three different numbers of agents $n \in \{2,3,5\}$, and with a random initial state. To mitigate the risk of outliers, the data shown in the charts is calculated as the mean of 10 independent runs with identical parameters.

The cost threshold α was chosen as $\alpha := \frac{3}{2}\cdot\frac{1}{k^n}$, such that the cost associated with a uniform probability distribution (i.e., no observation) lies within the allowed margin of error.

As the set of valid states for the governance utility function, we have chosen $S_+ := \{s \in \mathcal{S} : \overline{w}(s) \vee \overline{h}(s)\}$; in other words, the windows cannot be open when the heating is on. Note that this set includes 75% of the state space.

5.3.3 Results

As can be seen in Figure 5.3 for the deterministic case and in Figure 5.4 for the stochastic case, the intervention of the governance succeeds in reducing the average cost substantially in all cases. If the governance does not act, the prior probability of being in a violating state $s \in \mathcal{S}_-$ is 25% (and the expected governance utility consequently is -0.25), which is confirmed by the unrestricted cases (dashed lines).

Moreover, both the average cost and the degree of restriction decrease over time in the governed case, indicating that the governance indeed learns to predict agent actions and fine-tune its corrective action. Notably, this learning process is independent of an estimated agent preference order: The action policies were created randomly, which implies that they most likely do not correspond to a consistent order over the environmental states.

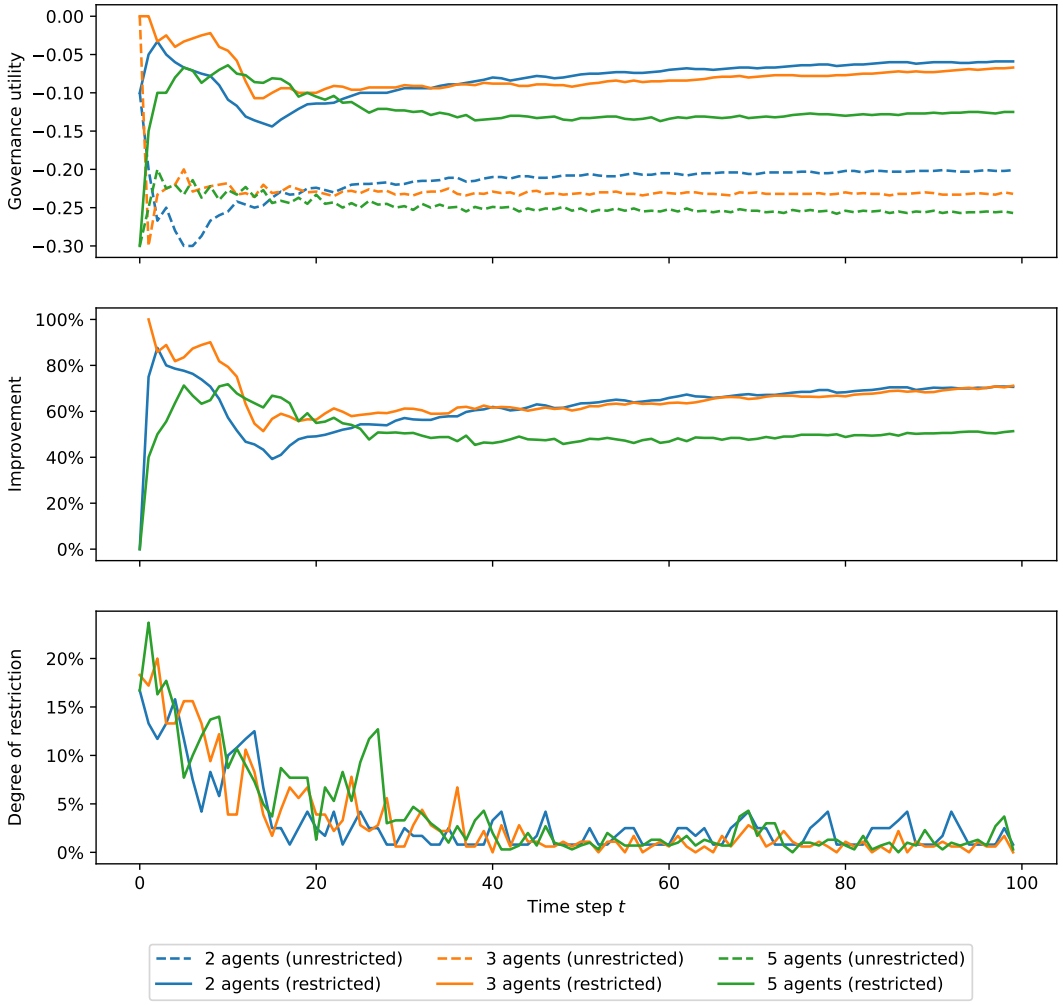


FIGURE 5.3: Results of the experiment with deterministic agent policies. **Top**: For each number of agents (i.e., color), we compare the governance utility with (solid line) and without (dashed line) restrictions. **Center and bottom**: The (relative) improvement achieved by the governance, as well as the degree of restriction required to achieve this improvement, are shown for each number of agents.

The effect of governance tends to drop with increasing number of agents. This might be due to the more widespread probability distribution, which prevents the governance from finding clear "dangerous" joint actions that it can easily forbid. Of course, this conjecture must be scrutinized with further experiments or supported by theoretical findings before conclusions are drawn.

5.4 Summary

In this chapter, we have provided a first governance learning approach for attribute-based multi-agent systems with unknown agent goals and policies. The governance is given a utility function over the environmental states and restricts the agents' action spaces in a way that increases the governance utility. Crucially, publicly observable actions and transitions are the only input the governance can use to define and optimize its restriction policy.

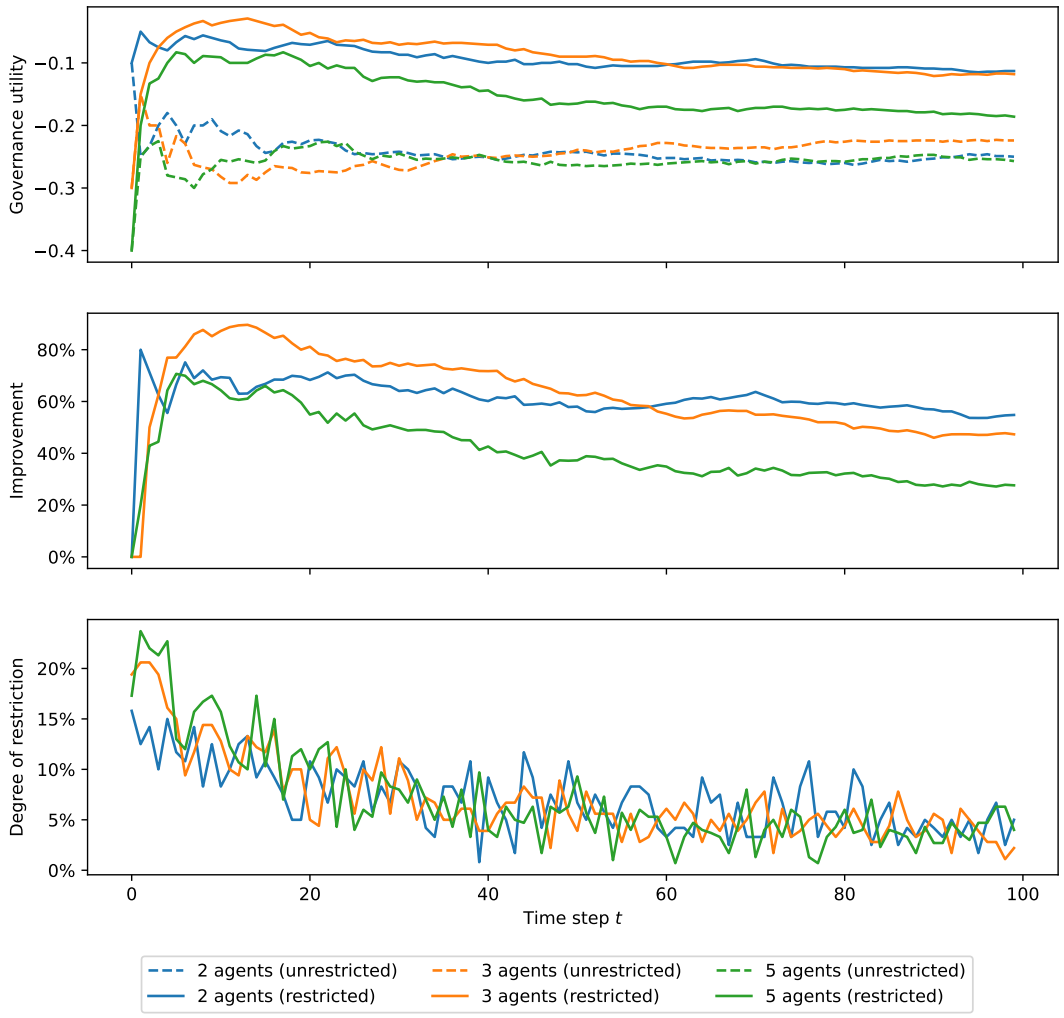


FIGURE 5.4: Results of the experiment with stochastic agent policies. The meaning of the graphs is identical to that of Figure 5.3.

We have presented, analyzed, and tested an algorithm that creates a minimal restriction for a given margin of error and thereby prevents transitions into undesirable environmental states.

The two primary success criteria—both shown to be satisfied in the experiments—are a substantial utility improvement and the fact that the degree of restriction decreases over time. The latter finding shows that the governance learns to achieve its utility improvement without being overly restrictive.

Unlike other work that assumes a transparent decision and reasoning process from its agents or even requires fixed and known agent preferences, this approach is applicable whenever actions can be observed and restricted by the governing instance.

The most evident limitation is that, while the algorithm is functional, it lacks (polynomial) scalability in terms of the number of agents and attributes, and it fully re-evaluates the minimal restriction at every step, thereby reducing the continuity of allowed actions over time. This naturally raises questions about a more efficient representation of knowledge (e.g., attribute dependencies and conditional probabilities) as well as environments with continuous attributes, irregular shapes, or more complex transitions.

Finally, the current approach applies two consecutive governance steps where the value of actions is determined before deriving a suitable restriction policy from this knowledge. In analogy to policy-optimization methods in classical RL, it seems promising to merge these two steps into an immediate policy generation from raw observations. By doing so, Assumptions 2 and 3 could be relaxed, resulting in a more general solution approach.

Chapter 6

Finding optimal restrictions via Reinforcement Learning

This chapter builds upon an idea raised in the summary of the last chapter: Instead of maintaining an explicit tabular knowledge base that contains the agents' past behavior and then deriving restrictions (heuristically) from this knowledge, we apply end-to-end Reinforcement Learning to learn an optimal restriction policy directly.

Personal Contribution. *I defined the model, built the algorithm, designed and conducted the experiments, and was the sole author of the text. The introductory example and the discussion of the results were jointly developed with Stefan Lüdtke, Christian Bartelt and Heiner Stuckenschmidt.*

6.1 Motivation

Multi-agent systems are widely used as a general model for the interaction of autonomous agents and have been applied to a vast range of real-world settings (Zhang, Yang, and Basar, 2019), for example, algorithmic trading (Abdunabi and Basir, 2014), traffic management (Padakandla, K. J., and Bhatnagar, 2020), and multi-player video games (Marín-Lora et al., 2020).

Example 6. *Consider a stock market where high-frequency trading algorithms typically generate the vast majority of orders. Agents in this setting act autonomously and in a self-interested manner in order to maximize their profit. As is known, this behavior can lead to problems like high volatility and extreme stock price behavior (McGroarty et al., 2019). It is, therefore, crucial that regulators provide both stability (i.e., ensure that extreme price movement flash crashes will not occur) and opportunity (i.e., ensure that investors can still use complex proprietary strategies to make a profit).*

In this example—as in many other application areas—the agents cannot (or should not) be fully controlled but must have a sufficient degree of freedom regarding their actions. At the same time, some level of control needs to be imposed on the agents so that a system objective can be achieved.

Our scope is thus a special class of MAS with three assumptions, inspired by the concept of Electronic Institutions as described in Chapter 4:

(a) The agents are truly autonomous entities whose goals and strategies cannot be known ("black-box agents"), but only observed through their actions,

(b) in addition to the agents' individual goals, there is a system objective that does not necessarily coincide with or relate to any of the former goals, and

(c) agent actions can be restricted by a governance that has the power to enforce such restrictions.

We propose a novel approach to governing a MAS that combines the restriction concept of an EI with dynamic rule-setting, provided by a Reinforcement Learning component. This governance observes the public information of the MAS, i.e., actions and state transitions, and learns optimal restrictions, which depend on the system state and the respective agent's observation[1].

A common method for governing agents in an EI is the use of *norms* with a focus on rewards and sanctions as the means of influencing agent behavior, while the action space itself is not affected. This makes two essential assumptions about the agents: First, "the effectiveness of these norms depends heavily on the importance of the affected social reality for the individual" (Balke et al., 2013), and second, the normative awareness needs to be comparable for all participating agents (*interpersonal utility comparison*). For unknown agents, we argue that these assumptions cannot be expected to hold, which is why we base our governance on (mandatory) restrictions of the agents' action sets. The dynamic nature of the rule-setting process (*rule synthesis*) is due to the fact that agents themselves can act strategically and are, therefore, able to exploit any static rule set.

Of course, the governance's "power to restrict" requires some sort of physical control over the MAS. This requirement is satisfied in a wide range of applications, for example, by any digital platform where agents are software components and actions are chosen by exchanging messages. Therefore, we assume the adherence to restrictions to be given.

6.2 Governance approach

The simultaneous execution and learning of a governed multi-agent system is shown in Figure 6.1. The governance is used (i.e., its restriction policy is queried) before every execution step of the MAS to determine the set of allowed and forbidden actions, whereas the learning takes place in between those execution steps.

At each learning step, the governance optimizes its restriction policy in order to maximize the system objective, given the observation of the last step. At the same time, the agents can update their own action policies, but this is not part of the ARMAS model (as mentioned above, we assume agents to be opaque).

As in the last chapter, and in accordance with ARMAS, the governance dynamically learns to optimize its restriction policy ρ during the interaction with agents and the environment.

We show in this chapter how a self-learning RL governance with the ability to restrict action spaces can add value to a MAS. This is demonstrated by comparing its performance to two natural alternatives (see also Shoham and Tennenholtz, 1995):

- *Ungoverned MAS* (UMAS), in which the agents alone decide on their actions, such that coordination or cooperation (if any) can only emerge on its own, and
- *Fully Controlled MAS* (FMAS), where the governance prescribes all agent actions, leaving no room for autonomous decisions.

To make this comparison, we conceptualize an RL governance (henceforth called *Governed MAS* (GMAS)) for the ARMAS model, analyzing the assumptions made in

[1] This form of restriction policy is described by a conditional ARMAS (see Section 3.2.4).

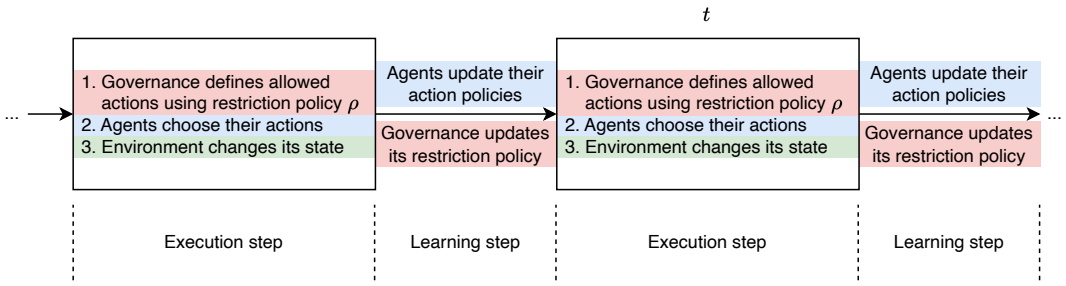


FIGURE 6.1: Sequence of execution and learning steps in a Governed Multi-Agent System

the model and describing the governance's learning behavior (Section 6.2), and we present experiments (Section 6.3) to demonstrate that this method can significantly outperform both alternatives.

The governance is a *centralized controller* insofar as it observes the entire MAS and defines restrictions in a centralized way. However, the fundamental difference to the usual notion of "centralized control" is that the governance leaves a substantial amount of autonomy to the agents. This is not enforced by its design, but—as the experiments will show—emerges naturally: The synergy between the governance's and the agents' capabilities gives a performance advantage over full control, causing the governance to allow multiple actions at most times.

6.2.1 Utility function

The system objective is given as a *reward function* (i.e., higher is better) used as the governance utility $u : \mathcal{S} \times \boldsymbol{A} \to [0, 1]$, allowing the governance to directly measure the success of its restrictions after each environment step. The normalized range of u between 0 and 1 is chosen for ease of comparability.

6.2.2 Restriction policy

We use in this chapter a *conditional* restriction policy as defined in Section 3.2.4, implying uniformity of the MAS. As a consequence, agents who make the same observation $o \in \mathcal{O}$ at a time step t are always allowed to perform the same actions $\rho(s_t, o)$. This is in line with a common-sense definition of *fairness*: The governance treats all agents the same way, independent of their identity but depending on their current situation. To achieve this, learning (i.e., a change of the governance policy) cannot take place within a time step, but only after all agents have been given their restrictions.

6.2.3 Learning

The ARMAS model does not specify any particular learning algorithm but only requires a restriction policy ρ to be available for querying at all times. This policy can be any function $\mathcal{S} \times \mathcal{O} \to 2^A$, but, of course, the goal of the governance is to find a restriction policy that maximizes the reward u, given the agents' behavior. Since the governance interacts with the ungoverned MAS in a cycle of information, reward, and action, RL is a natural way to optimize this policy: As agents update their

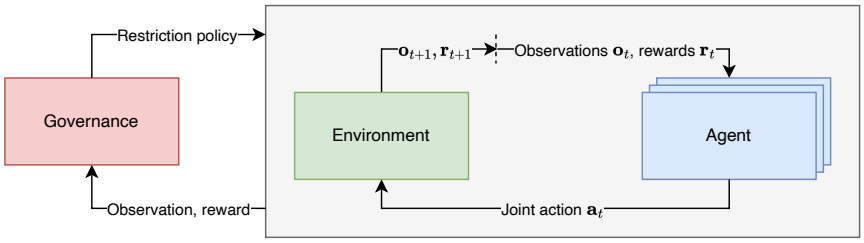


FIGURE 6.2: Illustration of the agent-environment interaction in the governance MDP, where the ungoverned MAS plays the role of the environment.

action policy while interacting with the environment, the governance updates its restriction policy according to the policy's effect on the evolution of the MAS, such that the expected cumulative governance utility is maximized.

From this perspective, the governance itself is an RL agent that acts on the entire MAS as its environment: The governance interacts with the MAS environment and the agents, but only sees how its own actions (i.e., defining sets of allowed actions for the agents) influence its reward and the environmental state. Therefore, it can be treated as a reinforcement learner with action policy ρ and reward u. Its environment has the transition function $\delta' : \mathcal{S} \times 2^A \to \Delta_{\mathcal{S}}$ with

$$\delta'(s, R) := \delta\left(s, \boldsymbol{\pi}(\boldsymbol{\sigma}(s), \boldsymbol{R})\right) \ , \tag{6.1}$$

which is a composition of observation functions σ, agent action policies $\boldsymbol{\pi}$ and MAS transition function δ.

δ' is not explicitly known to the governance, such that a model-free algorithm like A3C, DQN, or PPO must be used to learn the governance policy as the action policy of the governance agent. The governance is structurally equivalent to a multi-label classifier: Its policy outputs a subset of the (finite) fundamental action set. Thus, specialized network architectures for this type of classifier could also be applied in order to build a more effective governance policy[2].

Since agents can (and probably will) change their behavior in response to the current restriction policy, an ARMAS is inherently dynamic and therefore an *on-line* learning problem: Both sides (agents and governance) react to the other side's actions and strategies by continuously adapting their own action policies. The initial restriction policy can be a random function, or it can be set to simply allow all actions, i.e., $\rho^{(0)}(s, o) := A \ \forall s \in \mathcal{S}, o \in \mathcal{O}$. At run-time, the governance needs to learn continuously in order to keep up with changing agent behavior. Therefore, there is no distinction between training and evaluation as in classical RL, but the governance learning process continues throughout the lifecycle of the ARMAS.

6.2.4 Stationarity

It is known that, for a stationary MDP, near-optimal regret bounds can be achieved via RL (Cheung, Simchi-Levi, and Zhu, 2020). The situation is more complicated in the non-stationary case, depending on whether non-stationarity occurs in discrete steps (piece-wise stationarity) or continuously, among other criteria.

[2]We use a vanilla algorithm for our experiments, but one could, for example, adapt the network architecture of *Backpropagation for Multilabel Learning* (BP-MLL) (Zhang and Zhou, 2006) for an RL setting.

The transition function δ is assumed to be stochastic, but stationary. Therefore, the defining factor for the stationarity of an ARMAS, seen from the governance's view, is the set of agent policies π: δ' is stationary if and only if all agent policies are static, as can be seen from Equation (6.1).

While using static pre-trained models is very common for NLP, Computer Vision, and Speech Recognition (Zaib, Sheng, and Zhang, 2021), this is unusual for agent models since online learning lies at the heart of useful action selection in an unknown world. Nevertheless, safety-critical agent-based systems like fully autonomous cars will most likely require some sort of certification ensuring that they behave (exactly or approximately) in a certain way, which means that their policy should not, even when learning how to deal with unforeseen situations, be allowed to deviate too far from the approved policy.

Hence, we cannot generally assume that an ARMAS is stationary, but in some domains, there can be (quasi-)stationary agents, which means that the governance is likely to perform better than in a setting where the agents can adapt their strategies at an arbitrary rate.

6.3 Evaluation

The goal of the experimental evaluation is to investigate the effect of the governance. For this purpose, we define a game in which the agents need to agree on an action to get a reward, and then compare three types of systems: Ungoverned MAS (UMAS), which does not have a governance component at all, Fully Controlled MAS (FMAS), and our proposed approach, Governed MAS (GMAS).

6.3.1 The Dining Diplomats' Problem

Consider a MAS with agent set $I = \{1, ..., n\}$ and uniform action set $A = \{1, ..., k\}$. The agents are positioned in a circle such that each agent can only see their immediate neighbors (see Figure 6.3). At each step, the agents play a card corresponding to one of their available actions. The environmental state represents the currently played cards on the table, i.e., $\mathcal{S} = A^n$ and $\mathcal{O} = A^3$.

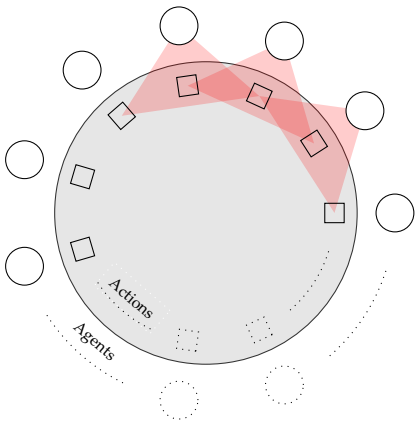


FIGURE 6.3: The dining diplomats' problem

The agents' goal is to learn to coordinate their actions in order to play the same cards at the same time. In the style of the famous *dining philosophers' problem*, we call this problem the *dining diplomats' problem*, requiring the participating agents to come to an agreement under imperfect information.

6.3.2 Reward functions

Consider two reward functions—a *state-based* reward and an *observation-based* reward:

$$r_s : \mathcal{S} \to \mathbb{R},\ r_s(s) = \begin{cases} 1 & \text{if } s_1 = \cdots = s_n \\ 0 & \text{else} \end{cases}$$

$$r_o : \mathcal{O} \to \mathbb{R},\ r_o(o) = \begin{cases} 1 & \text{if } o_1 = o_2 = o_3 \\ 0 & \text{else} \end{cases}$$

The state-based reward function only differentiates between "no coordination" and "full coordination", while the observation-based reward also shows local coordination between three agents (i.e., the observation space of one agent). The three system types, UMAS, FMAS, and GMAS, are then defined by different combinations of the reward functions r_s and r_o for agents and governance. These combinations are as follows:

	Agents	Governance
UMAS	r_o	-
FMAS	r_s	r_s
GMAS	r_o	r_s

In a UMAS, there is simply no governance. In the FMAS type, agents and governance have the same information about achieving their goals, so the governance cannot use the agents as an additional source of intelligence. In GMAS, however, the agents have access to more detailed information through r_o. Hence, the two pivotal dimensions are (a) access to low-level/high-level information and (b) dense and sparse rewards.

6.3.3 Configurations

We compare the three types of governance for four different problem sizes: Tiny ($n = 5, k = 3$), small ($n = 10, k = 5$), medium ($n = 15, k = 7$), and large ($n = 20, k = 10$). This allows us to see clearly at which complexity the different types fail to achieve coordination and, therefore, highlights the value added by GMAS.

Note that the size $|\mathcal{S}| = k^n$ of the state space grows polynomially in the number of actions, but exponentially in the number of agents: In the tiny configuration, there are $3^5 = 243$ states, while this number is $5^{10} \approx 10^7$ for the small configuration, $7^{15} \approx 4 \cdot 10^{12}$ for the medium configuration, and 10^{20} for the large configuration.

6.3.4 Frameworks and algorithms

For our experiments, we use the *RLlib* library (Liang et al., 2018) for multi-agent learning, which is based on the *Ray* distributed computing framework. Both agents and governance use a standard configuration of the PPO algorithm.

The interaction between agents, governance, and environment requires a sequential MAS execution: The governance needs to act (i.e., produce a set of allowed actions) before an agent can choose from this set. All agent actions, in turn, cause the environment to proceed to the next state. Therefore, the governance is queried n

times for each environmental step, while the agents each only act once during the same period.

All experiments were run in ten independent samples for $5 \cdot 10^6$ steps each (this number was empirically determined to ensure sufficient convergence of the action policies).

6.3.5 Results

The results of the experiments can be found in Figure 6.4. The governance utility u, as the main performance indicator, is shown on the left side, while the graphs on the right depict the corresponding degree of restriction τ (see Definition 11 in the previous chapter).

Since the governance reward at every step is either 0 or 1, we show the average reward over time, i.e., the percentage of steps where full coordination of all agent actions has been achieved.

In each graph, the mean of the ten samples (thick line) and the individual samples (thin lines) are plotted. The numbers vary strongly between samples, i.e., the mean should be seen as a general trend, not as the "average run".

Since the governance policy is initialized randomly, all governed types start with $\tau^{(0)} \approx \frac{1}{2}$. The progression of τ depends on whether the governance is able to learn a "fully controlling" way to create a high reward. If it succeeds, τ goes up to $\frac{k-1}{k}$ (i.e., allowing exactly one action) and stays there. Otherwise, the governance must utilize the agents' freedom and, therefore, allow more than one action. Notably, the degrees of restriction turn out to be roughly equal in the FMAS and GMAS types.

The detailed results are:

Tiny Configuration Both FMAS and GMAS achieve an almost perfect reward. While the FMAS solves the task by simply allowing a single action for each observation ($\tau^{(t)} \to \frac{k-1}{k} = \frac{2}{3}$), the GMAS uses a slightly lower degree of restriction. The problem is relatively easy, so the agents in the UMAS can also find a solution, albeit not a perfect one.

Small Configuration This is challenging for the UMAS, but FMAS and GMAS both achieve similar, good results. Sometimes, the GMAS uses the maximum degree of restriction, but mostly, agents are given two or three (out of five) allowed actions.

Medium Configuration The difference becomes larger: The UMAS cannot find a system state that results in a nonzero reward at all, and the FMAS performs approximately half as well as the GMAS. We can see from τ that even the FMAS governance does not use a maximally restrictive policy, since it cannot find the optimal actions for each observation.

Large Configuration Finally, both UMAS and FMAS are not able to get any rewards. In contrast, the GMAS still achieves a reward of more than 15-20% in four out of ten runs, using a degree of restriction around 50%.

The results show that the GMAS type succeeds in achieving full coordination of the agent actions in a substantial fraction of the time steps. As expected, the average reward decreases with increasing complexity of the setting, but it can handle systems where neither UMAS nor FMAS is able to get any rewards.

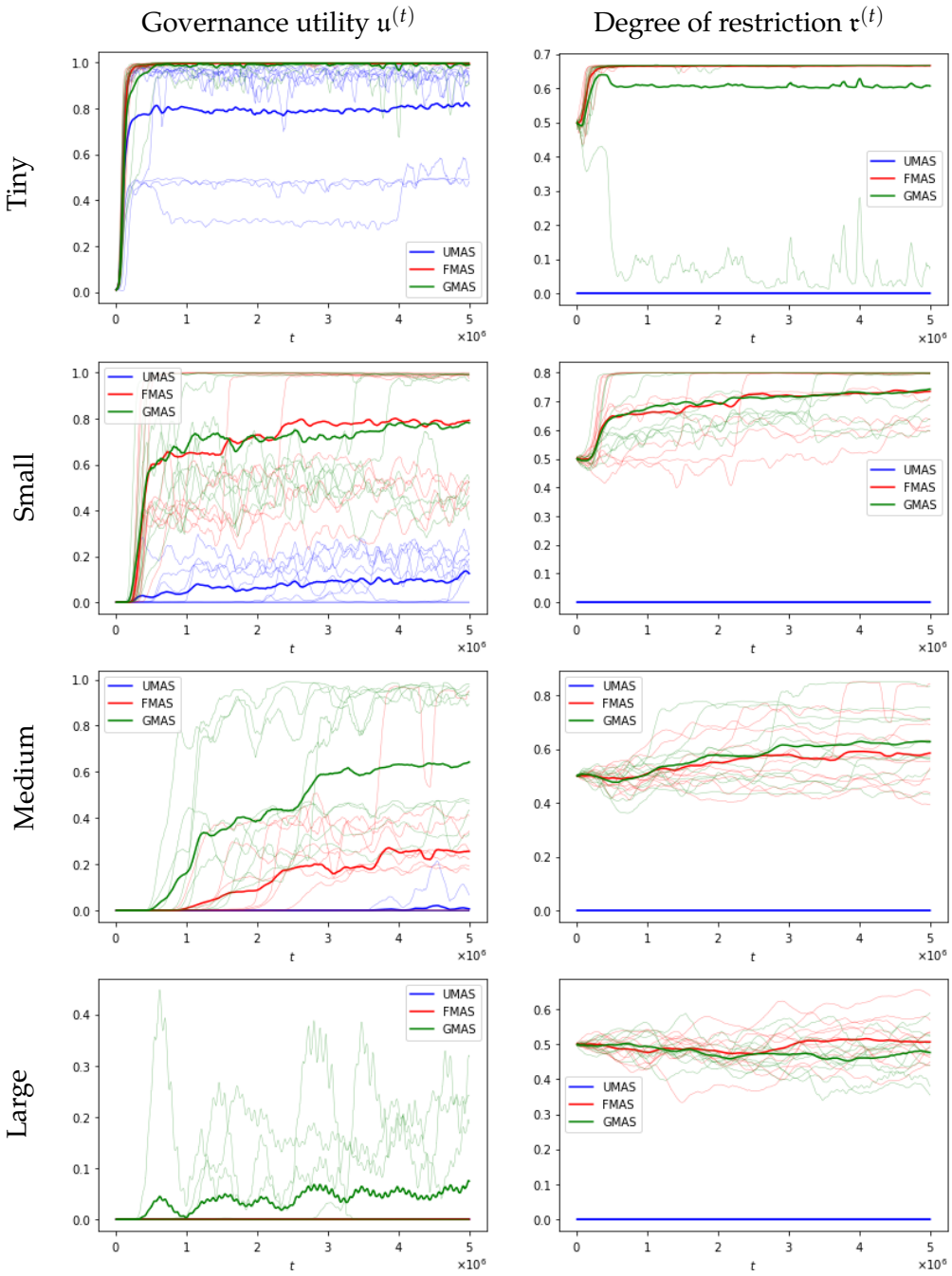


FIGURE 6.4: Experimental results. Thick lines show the mean of $u^{(t)}$ and $r^{(t)}$ over ten independent samples, while thin lines are the results of the individual samples.

6.3.6 Discussion

Qualitatively, we make the following observations for the solution capabilities of the three types of governance:

	Tiny	Small	Medium	Large
UMAS	✓	✓		
FMAS	✓	✓	✓	
GMAS	✓	✓	✓	✓

The hypothesis indeed holds that the synergy of agents and governance significantly outperforms the conventional approaches of ungoverned agents and centralized control. Notably, in all three cases, the agents simply apply their own selfish strategies, have no normative awareness, and their rewards are not influenced by the governance.

In this section, we give an interpretation of the observed results:

System objective and degree of restriction

The governance in the GMAS type does have the power to fully control the MAS—it could simply allow only one action for any state and observation. Therefore, the crucial observation in the experiments is that the degree of restriction does *not* generally converge to $\frac{k-1}{k}$ for $t \to \infty$.

Instead, the right side of Figure 6.4 clearly shows that the governance leaves a substantial amount of freedom to the agents and that this freedom causes the governance reward to be much higher than using full control (i.e., the FMAS type).

The balance between governance control and agent freedom is constantly changing, depending on how well the system objective (as measured by the governance reward function) is achieved. It is an essential feature of our approach that the optimal balance is determined via RL and not defined in advance.

Micro-level and macro-level knowledge

There are different types of knowledge in a GMAS: The governance can see the entire environmental state and knows which states are most desirable but does not know effective actions to get there since its reward function only indicates whether the system objective has been fully achieved or not. The agents, on the other hand, lack a view of the big picture but have a better grasp of how to act on a lower level since their reward function tells them when they are locally coordinated.

In the UMAS, the overall state is not available to the agents at all, not even through the governance. This prevents the agents from finding a globally coordinated solution, even though they can coordinate locally. In the FMAS, the governance sees the big picture but cannot figure out the necessary actions for the agents to move in the right direction, and does not get support from the locally coordinated agents.

The combination of these two levels allows the GMAS to reach global coordination—without ever being instructed on how to combine agent and governance knowledge. This setting was chosen since it represents a common pattern in MAS: Individual agents are situated at a specific location in the environment and are only able to perceive their surroundings, i.e., a small part of the environment. At the

same time, this small part is where their actions have the biggest impact. The system designer or operator, in contrast, sees the environment as a whole but does not have micro-level knowledge about optimal or even useful agent actions. Therefore, the goal is clear, but the way to get there is unknown.

Incentives for autonomy and restriction

The governance can freely choose the restrictions without being penalized for high degrees of restriction. Consequently, there is no real incentive for the governance to allow multiple actions: The chosen degree of restriction directly reflects the highest expected reward. In the small scenarios, we observe that allowing only one action per observation is a feasible strategy that leads to high rewards. As the scenarios get more complex, however, the governance policy is not maximally restrictive anymore: The governance learns that the autonomous decisions of the agents are more helpful than centralized control. Still, by selectively forbidding actions, the governance can support the agents' action policies.

Penalties for restrictions

A reasonable goal for the governance is to use the least amount of restrictions to achieve its objective and, therefore, strive to reduce the degree of restriction whenever this does not counteract the system objective (see our discussion in Section 1.2.5). To this end, we also experimented with giving the governance a penalty in proportion to the current degree of restriction by redefining its utility function as $u' := u - \omega \cdot r$ with a constant weighting hyperparameter ω. This resulted in a much lower utility (even when looking at the utility without penalty), making the governance drop nearly all restrictions early in the training before it then defined new, more effective restrictions. However, the penalty often prevented the governance from sufficiently exploring the possible restrictions, so there were many samples where there was never any reward, even in small scenarios.

6.4 Summary

This chapter re-defined governance within the ARMAS approach of Chapter 3 as an RL agent acting on a complete MAS, including agents and environment. By acting through action space restrictions, this governance agent can use a standard RL approach to find an optimal restriction policy, as measured by its governance utility function.

The main claim, supported by the experiments, is that such a governance outperforms the two extreme cases on the scale from ungoverned MAS to fully centralized control. We have demonstrated that full control as well as ungoverned learning agents fail to achieve their goals even in simple scenarios; a challenge solved considerably better by a self-learning restriction-based governance.

An obvious limitation of this approach is that the governance's action space consists of all possible restrictions of a discrete action space. As such, there is the well-known curse of dimensionality when dealing with larger agent action spaces and more agents. Additionally, we see the following open questions and directions for future work:

- In the experiments presented here, the objectives of agents and governance were strongly correlated. How can the approach be applied to an arbitrary

combination of goals, and how do conflicts in the objective functions influence governance learning?

- What would an extension of the restriction policy to continuous action spaces look like?
- How do action space restrictions compare (empirically and theoretically) to other forms of governance, e.g., norms or inter-agent communication?

Chapter 7

Finding optimal restrictions via exhaustive search

The approach proposed in the previous chapter is, as mentioned in the summary (Section 6.4), only feasible for discrete action spaces. In this case, the governance can be modeled as an RL agent whose actions are subsets of the agents' action space, defining precisely their allowed actions. For continuous action spaces, it is not clear how to canonically represent subsets and, therefore, how to build and train a governance to find an optimal restriction policy.

In this chapter, we thus investigate continuous agent action spaces, proposing a third governance approach: Our governance applies a breadth-first search over possible restrictions, using an oracle function for equilibrium strategies under a specific restriction.

There are a few additional assumptions on the general ARMAS model for this investigation: First, we only consider stateless systems, particularly NFGs (see Section 2.1.3). Second, the continuous action spaces are assumed to be one-dimensional. Finally, the same restriction is applied for all agents; in other words, restrictions are *non-discriminatory* as defined in the uniform subclass of ARMAS (Section 3.2.3).

Personal Contribution. *I defined the model, built the algorithm, designed and conducted the experiments, and was the primary author of the text. The definition of the research questions, the focus on continuous action spaces, and the final version of the paper were jointly developed with Guni Sharon.*

7.1 Motivation

Consider a multi-agent game with an additional governance utility function over the joint actions. Assuming that the agents are self-interested and learn independently, they might converge to joint actions ("user equilibria") which are sub-optimal, both from their own perspective (e.g., with respect to Pareto efficiency) and from the viewpoint of social welfare (Cigler and Faltings, 2011). This can be demonstrated in minimal setups (see Examples 7 and 8 in Section 7.2.4), but it is also common in real-world settings (Ding and Song, 2012; Acemoglu et al., 2016; Memarzadeh, Moura, and Horvath, 2020).

While the challenge of reconciling selfish optimization and overall social utility in multi-agent settings has long been known (Roughgarden and Tardos, 2002; Andelman, Feldman, and Mansour, 2009), it has become increasingly relevant with the rise of ubiquitous autonomous agents and automated decision-making in recent years. Advancements in deep reinforcement learning have enabled agents to learn very effective (but still selfish) policies not only in well-defined games but also

in multi-agent systems with large, complex, and unknown environments (Du and Ding, 2021; Gronauer and Diepold, 2021).

A common solution method for this problem involves *reward shaping*, where the agents' utility functions are altered by giving them additional positive rewards for socially desirable behavior and negative rewards (i.e., sanctions) for undesirable behavior. Normative Systems (Andrighetto et al., 2013) derive such rewards and sanctions from norms, while Vickrey–Clarke–Groves (VCG) mechanisms (Nisan and Ronen, 2004) attribute to each agent the marginal social cost of its actions.

Reward-shaping methods generally make two assumptions which limit their applicability:

1. *Rewards can be changed at will, and agents simply accept the new reward function.* This assumption is feasible in stylized settings but involves an arbitrary amount of additional incentives (in other words, money) when applied in real-world settings.

2. *It is both possible and ethically justifiable to discriminate between agents by shaping their reward functions differently.* On top of ethical issues, this approach might not be applicable whenever agents are not identifiable or distinguishable.

As in Chapters 5 and 6, we want to close the gap between user equilibrium and governance optimum, based on shaping the action space available to the agents at any given time (as commonly done by regulating governmental entities). Therefore, agents continue to optimize their own objective function over the restricted action space. This motivates the problem of finding an *optimal non-discriminatory restriction* of the agents' action space, i.e., a restriction that is identical for all agents and maximizes the governance utility of a stable joint action.

In this chapter, we analyze the problem of finding socially optimal restrictions for normal-form games with continuous action spaces. We define the concept of a Restricted Game (RG), which is a subclass of ARMAS, and present a novel algorithm denoted *Action-Space Restrictor for Optimal Governance Utility* (AROGU) which finds optimal restrictions via an exhaustive Breadth-First Search (BFS) over the restriction space, assuming that (a) there is always a Nash Equilibrium, and (b) there is an oracle function which provides such a Nash Equilibrium for a given restriction. We then demonstrate the algorithm's performance using two well-known game-theoretic problems—Braess' Paradox and the Cournot Game. Our experiments show that applying AROGU can find favorable outcomes even when we relax the assumptions. Finally, we outline how the approach developed for (stateless) multi-agent Normal-Form Games is also applicable to Stochastic Games with state transitions. This extension, however, is far from trivial due to combinatorial explosion and generalization over the state space (see also Section 10.2.1).

7.2 Restricted NFGs over continuous action spaces

Restrictions of finite discrete action spaces have a canonical representation, given by a list of the allowed actions; since there is only a finite number $2^{|A|}$ of potential restrictions, it is possible (at least in theory) to list them all and select the optimal one. For continuous action spaces, such a representation does not exist, which means that finding and even defining an optimal restriction requires a more elaborate approach.

7.2.1 Restrictions

Here, we limit our discussion to real-valued interval action spaces A. By doing so, we can consider restrictions that are finite unions of half-open intervals in A. Note that it is not clear how to efficiently represent subsets of a multi-dimensional action space; moreover, action spaces with more than one dimension require a different approach for defining tentative restrictions (see Section 7.3), which respects the space's topology and possible correlations between dimensions.

Assumption 4 (Interval-Union Restrictions)**.** *We assume in this chapter that the uniform (i.e., equivalent for all agents) action space, A, is a one-dimensional interval $[a, b)$ (using $\pm\infty$ for unbounded spaces), and that the governance can define restrictions of A which are finite unions of half-open intervals:*

$$R = \bigcup_i [l_i, u_i) \tag{7.1}$$

with interval bounds $l_i, u_i \in A \; \forall i \in I$[1]*.*

A unique representation of such a restriction can be achieved by additionally demanding that $u_{i-1} < l_i < u_i < l_{i+1}$ for all i.

A joint action $\boldsymbol{a} \in \boldsymbol{A}$ is *allowed* if all components of $\boldsymbol{a}$ are in R (i.e., $a_i \in R \; \forall i \in I$), or equivalently, if $\boldsymbol{a} \in \boldsymbol{R} = R^I$, since the restriction R applies equally to all agents.

It is important to note that adding or removing an interval $[l, u) \subseteq A$ to or from R—which is what the AROGU algorithm will do—does not violate Equation (7.1) since this family of restrictions is closed under finite unions and set differences. We call a restriction R' *more constrained* than R if $R' \subset R$. Finally, for a restriction R of form (7.1), let $|R| := \sum_i (u_i - l_i)$ denote the *size* of R.

7.2.2 Restricted normal-form games

A stateless, uniform ARMAS (see Section 3.2.1) can be written as the tuple $(I, A, \rho, \boldsymbol{r}, \mathrm{u})$, with the (trivial) environmental state and observation functions omitted. For a fixed restriction $R = \rho(A) \subseteq A$, we have a new, restricted normal-form game with optimal strategies and equilibria that can deviate arbitrarily from the "original" game with action space A.

Therefore, it makes sense to consider a function that maps a restriction to the Nash equilibria under the restriction. We first define a *restricted game*:

Definition 12. *Let $G = (I, A, \boldsymbol{r}, \mathrm{u})$ be a uniform normal-form game together with a governance utility function*[2]*. For a restriction $R \subseteq A$, we define the* Restricted Game *(RG) $G|_R = (I, R, \boldsymbol{r}, \mathrm{u})$ such that the agents are only allowed to use actions in R instead of the full action space A. The domain of the utility functions is hence restricted to $\boldsymbol{R} := R^I$.*

The definitions of best responses and Nash equilibria (Definitions 6 and 7) can be applied to restricted games, and are denoted as $\mathcal{B}_i|_R$ and $\mathcal{N}|_R$, respectively. It is noteworthy that they can, in general, change arbitrarily (for better or worse) by restricting a game.

[1] Technically, the upper bounds of intervals only need to lie in the *closure* of A, not in A itself.

[2] We call this tuple a *governed game*.

7.2.3 Governance utility of equilibria

The Nash Equilibria of a MAS can yield different governance utility values. Arguably, the governance has the goal of achieving a *guaranteed* high utility $\mathfrak{u}$, so we need to consider the minimum governance utility of a NE.

Definition 13. *A* minimum Nash Equilibrium

$$\mathcal{N}^- := \arg\min_{\boldsymbol{a}\in\mathcal{N}} \mathfrak{u}(\boldsymbol{a})$$

is an equilibrium with the lowest governance utility (i.e., the worst NE from the governance's perspective)[3].

We focus on the minimum NE in this definition, since the governance cannot decide which one of the equilibria the agents converge to in a restricted game $G|_R$. The governance utility of the minimum Nash equilibrium is therefore an important measure for the evaluation of a restriction algorithm. Since we assume the governance utility to be the social welfare in this chapter, we define this measure as follows:

Definition 14 (Minimum Equilibrium Social Utility)**.** *Let R be a restriction of A. Then*

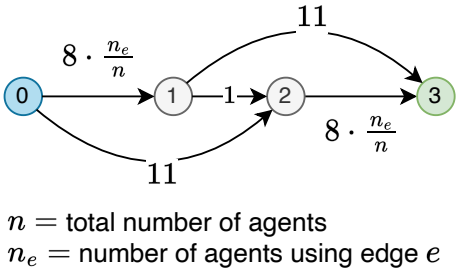
$$\mathcal{S}(R) := \min_{\boldsymbol{a}\in\mathcal{N}|_R} \mathfrak{u}(\boldsymbol{a}) = \mathfrak{u}(\mathcal{N}^-|_R)$$

denotes the Minimum Equilibrium Social Utility *(MESU) of the restricted game $G|_R$ as a function of the restriction R.*

7.2.4 Examples: Braess' Paradox and the Cournot Game

To illustrate our governance approach, we start with the discrete case of Braess' Paradox (Braess, 1968, see also Section 4.3). Nonetheless, our main contribution applies to the more general case of continuous action spaces.

Example 7 (Braess' Paradox)**.** *Braess' Paradox can be translated from its original domain of traffic routing into a two-agent matrix game as shown in Figure 7.1, where the row action is controlled by agent* 1*, and the column action by agent* 2*. By convention, both agents want to maximize their respective payoffs.*



	a	b	c
a	$0,0$	$0,5$	$2,2$
b	$5,0$	$1,1$	$5,0$
c	$2,2$	$0,5$	$0,0$

FIGURE 7.1: Braess' Paradox as a routing problem with n agents (left) and an equivalent two-agent Matrix Game (right).

The best response for both agents is always b. Selfish agents will converge to the user equilibrium (b, b) and, therefore, end up with a payoff of 1*. Let us now forbid action b, i.e.,*

[3] Like the set $\mathcal{N}$ of Nash equilibria, $\mathcal{N}^-$ is obviously a property of a specific game G. When this is unclear, we will write $\mathcal{N}_G$ and $\mathcal{N}_G^-$, but otherwise omit the subscript.

restrict the action space to $\{a, c\}$*. The user equilibria become* (a, c) *and* (c, a) *with a payoff of* 2 *each.*

There is a vast amount of theoretical and applied work on Braess' Paradox (see Section 4.3), showing that Braess-like scenarios are not only technical cases but appear often in random networks (Valiant and Roughgarden, 2006; Chung and Young, 2010).

Apart from illustrating the efficacy of a restriction-based governance approach, this example also shows the "meta challenge" of restrictions: if we allowed for *individual* restrictions of the agents' action spaces, it would be straight-forward for the governance to achieve any possible outcome (i.e., combination of actions) by allowing each agent to use exactly one action. This procedure reduces the (multi-agent) game to a (single-agent) optimization problem, where the governance computes the socially optimal matrix cell $\max_{\boldsymbol{a} \in \boldsymbol{A}} \mathfrak{u}(\boldsymbol{a})$ with $\boldsymbol{A} = \prod_{i \in I} A_i$, and then simply assigns the respective actions to the agents.

Things become more challenging when only considering *non-discriminatory* restrictions, as we have done in Example 7. This also satisfies an extremely desirable property for any form of governance: All agents are treated fairly by having the same space of allowed actions. In the example, the governance could enforce the (socially optimal) solutions (a, b), (b, a), (b, c), or (c, b) with governance utility 5 by using *individual* restrictions. This is not possible with uniform restrictions, but we can still improve the game's MESU from 2 to 4.

Let us now consider a game with a *continuous* action space where rewards are given as individual utility functions over the joint action space: The *Cournot Game* (Cournot, 1838) is a classical example of a NFG with one-dimensional continuous action spaces and one of the fundamental economic models for establishing produced quantities and prices on a market.

Example 8 (Cournot Game)**.** *Let two agents decide on the produced quantities* $\boldsymbol{q} = (q_1, q_2) \in \mathbb{R}^2$ *of a good whose price is defined as* $p(\boldsymbol{q}) = \max(p_{max} - q_1 - q_2, 0)$ *with* $p_{max} > 0$*. Both agents produce at a constant cost of* $c \geq 0$ *per unit. The agents' rewards (i.e., their profits) are therefore given as* $r_i(\boldsymbol{q}) = q_i \cdot (p(\boldsymbol{q}) - c)$*.*

Choosing $p_{max} = 120$ *and* $c = 12$*, the BR of agent* i *to action* q_j *is* $\mathcal{B}_i(q_j) = 54 - \frac{q_j}{2}$*, which leads to a unique NE of* $\boldsymbol{q}^* = (36, 36)$ *and a payoff of* $r_1(\boldsymbol{q}^*) = r_2(\boldsymbol{q}^*) = 1296$*. By restricting the quantities produced by each agent to the range* $q_i \leq 27$*, it would be possible to improve the equilibrium payoff to* 1458 *per agent.*

In these examples, we have the particular situation that the restriction improves the rewards of *all agents*, which makes a very strong case for using such restrictions. In general, it is not the case that all agents will be better off, so the governance's goal is simply to maximize the governance utility $\mathfrak{u}$[4].

We revisit the examples in the experiments, using our AROGU algorithm (see Section 7.3) to find optimal restrictions.

7.3 Governance approach

In this section, we present the *Action-Space Restrictor for Optimal Governance Utility* (AROGU) algorithm for continuous-action games with a finite (i.e., bounded) action space A. AROGU defines a search tree of increasingly constrained restrictions by

[4] As argued above, any other considerations like fairness are assumed to be baked into this utility function.

identifying and testing reasonable subsets of existing restrictions, starting from the unrestricted action space. The theoretical results and conclusions in this section hold for an arbitrary governance utility function u.

7.3.1 Restricting the action space

For a given joint action, $\boldsymbol{a}$, we say that a restriction R *invalidates* $\boldsymbol{a}$ if $\boldsymbol{a} \notin \boldsymbol{R}$, i.e., at least one individual action is not allowed by R. In general, a restriction R that invalidates an existing NE does not simply cause a new NE to appear at the boundary of R (i.e., as close to the old NE as allowed by R)—instead, a new NE might appear anywhere else in the joint action space, or the restricted game might not have an NE at all. However, a restriction that does not invalidate any existing NE (we call such a restriction *irrelevant*) leaves the existence of those NEs unchanged. More formally:

Proposition 1. *Given some $x \in \mathbb{R}$, let $\mathcal{U}_\varepsilon(x) := [x - \varepsilon, x + \varepsilon)$ denote the half-open ε-neighborhood of x, and for a vector $\boldsymbol{x} \in \mathbb{R}^n$, let $\mathcal{U}_\varepsilon(\boldsymbol{x}) := \cup_{i=1}^n \mathcal{U}_\varepsilon(x_i) \subseteq \mathbb{R}$[5]. Assume that $\boldsymbol{a} \in \boldsymbol{A}$ is a joint action such that $\mathcal{N} \subseteq \boldsymbol{R}$ with $R := A \setminus \mathcal{U}_\varepsilon(\boldsymbol{a})$. Then*

$$\mathcal{N} \subseteq \mathcal{N}|_R \,,$$

which means that invalidating actions within the ε-neighborhood of $\boldsymbol{a}$ does not remove any of the Nash equilibria from G.

Proof. Let $\boldsymbol{x} \in \mathcal{N}$ be a NE over the action space A, and let R be defined as in the statement of the proposition. Then

$$\begin{aligned}
& x_i \in \mathcal{B}_i(x_{-i}) \;\forall i \in I \\
\xRightarrow{\text{Def. 6}} \quad & u_i(\boldsymbol{x}) \geq u_i(a', \boldsymbol{x}_{-i}) \;\forall a' \in A \;\forall i \in I \\
\xRightarrow{R \subseteq A} \quad & u_i(\boldsymbol{x}) \geq u_i(a', \boldsymbol{x}_{-i}) \;\forall a' \in R \;\forall i \in I \\
\xRightarrow{\boldsymbol{x} \in \boldsymbol{R}} \quad & x_i \in \mathcal{B}_i|_R(x_{-i}) \forall i \in I \quad \Longrightarrow \quad \boldsymbol{x} \in \mathcal{N}|_R \,.
\end{aligned}$$

□

As a direct consequence, any restriction that improves the MESU of a game must invalidate all existing minimum Nash Equilibria.

7.3.2 The AROGU algorithm

Given an action space, the broad idea of AROGU is to define successively more constrained restrictions and then search for the best of those restrictions in terms of their MESU (see Algorithm 2). Basically, we can check every possible restriction of the form (7.1) (see Assumption 4), starting from A and ending with maximally constrained restrictions. Of course, this brute-force method is not practical since it requires computing the MESU of infinitely many restrictions.

We propose the following improvement for a current restriction R at any step of the process: Calculate the minimum NE, $\boldsymbol{a}^* := \mathcal{N}^-|_R$, and derive all *relevant actions*, i.e., the set $\Omega := \bigcup_{i \in I} a_i^*$ of all (individual) actions that are part of $\mathcal{N}^-|_R$. For each $\omega \in \Omega$, define a new restriction by removing an ε-neighborhood $\mathcal{U}_\varepsilon(\omega)$ from R (see Figure 7.2).

[5] Note that, by definition, this neighborhood is still one-dimensional!

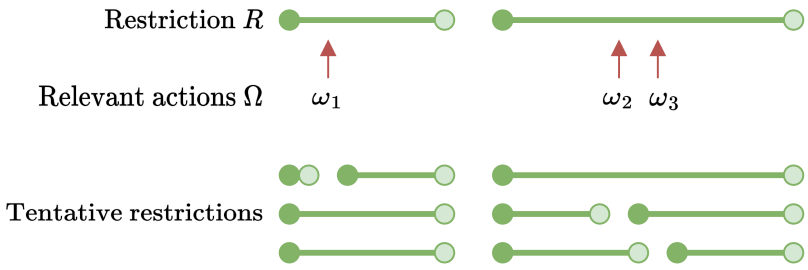


FIGURE 7.2: Tentative restrictions for a set Ω of relevant actions. Each tentative restriction is defined by removing an ε-neighborhood of one of the relevant actions $\omega \in \Omega$.

Let us consider a graphical example: It follows from Proposition 1 that, in the setting of Figure 7.2, any restriction $R' \subset R$ which includes ω_1, ω_2 and ω_3, would not eliminate $\mathcal{N}^-|_R$, and therefore cannot have a higher MESU than R. Hence, it is not necessary to check those restrictions, effectively pruning them from the search tree. As we show experimentally later, this can lead to a significant reduction in the number of NE calculations required compared to uniformly checking all restrictions.

For each of the tentative restrictions, we repeat the process of computing the NE and relevant actions, subsequently restricting them further until the action space is empty. Of all those restrictions, we then select the one that gives the highest MESU, resulting in a (pruned) breadth-first search over the restriction space. To ensure that restrictions are not considered multiple times, we keep a set (i.e., a closed/duplicate list) of already explored restrictions. Moreover, the state space size can be controlled via the hyperparameter ε (the *resolution* of AROGU), which defines the size of the interval around a relevant action that is removed for tentative restrictions.

Algorithm 2: Action-Space Restrictor for Optimal Governance Utility (AROGU)

Data: Governed Game $G = (I, A, \boldsymbol{r}, \mathfrak{u})$, equilibrium oracle μ, resolution ε
Result: Optimal restriction $\hat{R}^* \subseteq A$

$(\hat{R}^*, \hat{u}^*) \leftarrow (A, \mathfrak{u}(\mu(A)))$
$Q \leftarrow$ Queue with content $\{A\}$
while *Q is not empty* **do**
  $R \leftarrow Q.dequeue()$
  `// Loop through relevant actions`
  for $\omega \in \Omega(\mu(R))$ **do**
    $R' \leftarrow R.remove(\mathcal{U}_\varepsilon(\omega))$ `// Tentative restriction`
    if *R' is not empty and has not been explored before* **then**
      $Q.enqueue(R')$
      if $\mathfrak{u}(\mu(R')) > \hat{u}^*$ **then**
        $(\hat{R}^*, \hat{u}^*) \leftarrow (R', \mathfrak{u}(\mu(R')))$
      end
    end
  end
end
return $\hat{R}^*$

7.3.3 Equilibrium oracle

To add restrictions purposefully, we need to know where the current equilibria are. For this work, we assume that there is an oracle function μ which, for a given RG $G|_R$, returns a joint action $\boldsymbol{a} \in \boldsymbol{R}$ which is an equilibrium of $G|_R$ with minimum governance utility. In Appendix A.1, we show how to implement such an oracle for quadratic utility functions[6].

7.3.4 Complexity and correctness

Proposition 2. *Let $A = [a, b)$ and $\varepsilon > 0$. Then, any restriction chain*

$$A = R_0 \sqsupset R_1 \sqsupset ... \sqsupset R_x = \varnothing$$

consists of at most $\lceil \frac{b-a}{\varepsilon} \rceil$ elements, where $R \sqsupset R'$ means that R' is a tentative restriction over R as created by Algorithm 2. In other words, the depth of the search tree is bounded by $\lceil \frac{b-a}{\varepsilon} \rceil$.

Proof. For any subsequent pair $R_i \sqsupset R_{i+1}$ in a restriction chain, let us denote by ω_i the action whose ε-neighborhood was removed at this step. We see that $|\omega_i - \omega_j| \geq \varepsilon \ \forall\, i < j$ (otherwise, ω_j would have been already forbidden before its removal). There cannot be more than $\lceil \frac{b-a}{\varepsilon} \rceil$ points with pairwise distance $\geq \varepsilon$ on the interval A which has length $(b - a)$. □

As a result of Proposition 2, we can bound the runtime of AROGU by $|\Omega_{max}|^d$, where $d = \lceil \frac{b-a}{\varepsilon} \rceil$, and Ω_{max} is the largest set $\Omega(\mu(R))$ we encounter in the **for** loop (line 5) of Algorithm 2.

Definition 15. *A restriction R^* is called* optimal *for a game G if $\mathcal{S}_G(R^*) \geq \mathcal{S}_G(R)\ \forall\, R \subseteq A$.*

Assumption 5. *We assume that there is always a Nash Equilibrium for a restricted game, i.e., $\Omega(R) \neq \varnothing\ \forall R \subseteq A$.*

Proposition 3. *Let R^* be an optimal restriction for a game G. Then, under Assumption 5,*

$$R^* \subset R \ \Rightarrow\ \exists \omega \in \Omega(R) : \omega \notin R^* .$$

Proof. Assume that $\forall \omega \in \Omega(R) : \ \omega \in R^*$. Then $\mathcal{N}^-|_R \in \boldsymbol{R^*}$, and since $R^* \subset R$, $\mathcal{N}^-|_R \subseteq \mathcal{N}|_{R^*}$ according to Proposition 1. Therefore, $\mathcal{S}(R^*) \leq \mathcal{S}(R)$, which, together with $R^* \subset R$, contradicts the optimality of R^*. □

Proposition 4. *Throughout Algorithm 2, (at least) one of the following two conditions holds:*

(i) The restriction queue Q contains a restriction R which is a superset of an optimal restriction R^*

(ii) $\hat{R}^*$ is already set to an optimal restriction

Proof. After the initialization step, condition (i) holds since any optimal restriction R^* is a subset of A, which is in Q.

[6] In general, finding a Nash Equilibrium of a given NFG is a hard problem by itself (Daskalakis, Goldberg, and Papadimitriou, 2009).

From Definition 15, we see immediately that the update step

$$(\hat{R}^*, \hat{u}^*) \leftarrow (R', \mathfrak{u}(\mu(R')))$$

in line 10 satisfies two properties: A non-optimal restriction never replaces an optimal one, and an optimal restriction always replaces a non-optimal one. Thus, once condition (ii) is satisfied, it stays satisfied until AROGU terminates. Let us, therefore, assume that (ii) does not hold yet.

Whenever a restriction R is dequeued from Q, condition (i) either still holds (this is the case if there is another such restriction still in Q), or R is a superset of an optimal restriction R^*. Since (ii) is not satisfied, we know that R itself is not optimal. Proposition 3 asserts that there is a relevant action $\omega \in \Omega(R)$ which is not in R^*. Hence, at the respective pass of the **for** loop, we will have $R' := R \setminus \mathcal{U}_\varepsilon(\omega)$, and, for a sufficiently small ε, $R' \supseteq R^*$.

If R' has been explored before, it was enqueued then, meaning that (i) still holds. Otherwise, R' is enqueued now. If $R' \supset R^*$, (i) holds, and if not, R' is optimal, such that (ii) becomes true. □

Theorem 2. *Let $G = (I, A, \boldsymbol{r}, \mathfrak{u})$ be a governed game. If Assumption 5 holds, and for a sufficiently small $\varepsilon > 0$, Algorithm 2 finds an optimal restriction R^*.*

Proof. AROGU terminates after finitely many steps: Any tentative restriction R' produced by a reduction of some $R \in Q$ continues a chain of increasingly constrained restrictions, as in Proposition 2, and the length of such a chain is bounded by $\lceil \frac{b-a}{\varepsilon} \rceil$.

At the point of termination, Q is empty. Condition (i) in Proposition 4 does not hold anymore, which means that $\hat{R}^*$ is indeed an optimal restriction. □

7.4 Evaluation

We have shown that the AROGU algorithm finds an optimal restriction for a given NFG under some assumptions. However, these assumptions are not always satisfied in real-world settings. Our experimental study is thus set to address the following open questions:

Q1 If Assumption 5 is not guaranteed to hold, does AROGU still find (close to) optimal restrictions?

Q2 Does the state-space pruning technique used by AROGU allow for reasonable run-times, despite the fact that the size of the search tree is exponential in $\frac{b-a}{\varepsilon}$?

To answer these questions, we examine parameterized continuous-action versions of the Cournot Game (CG) and Braess' Paradox (BP). First, we use domain knowledge about both games to establish theoretical results for their governance optimum and optimal restriction (see Appendix A). Afterward, we compare these findings with the results of AROGU for a range of parameters to obtain insights into AROGU's scaling behavior. The values of ε were empirically chosen to provide a good balance between run-time and accuracy, but the results are actually reasonably insensitive to this choice: Varying ε by a factor of 5 causes the MESU to change by less than 1% in both games.

7.4.1 Quadratic utility functions

Let us start by observing that many interesting problems, including the continuous Braess Paradox (see Definition 18), the Cournot Game, and the continuous version of any 2x2 matrix game, can be represented as NFGs with *quadratic reward functions*. They have the convenient property of being convex or concave (or both, i.e., linear) in each variable x_i, depending on the sign of the coefficient of x_i^2. They allow for efficient computation of best responses and Nash Equilibria and, therefore, lend themselves well to the examination of RGs and the optimization of restrictions.

Definition 16. *A reward function $r : \boldsymbol{A} \to \mathbb{R}$ is called* quadratic *if it is polynomial in the agents' actions a_i and has a maximum degree of 2. This means that, for n agents,*

$$r(\boldsymbol{a}) = \sum_{\alpha \in \mathbb{N}^n} c_\alpha \cdot a_1^{\alpha_1} \cdots a_n^{\alpha_n}$$

with $c_\alpha \in \mathbb{R}$ and $\max_{c_\alpha \neq 0} \left(\sum_{i=1}^n \alpha_i\right) \leq 2$.

For two agents, quadratic reward functions have the form

$$r(a_1, a_2) = c_1 a_1^2 + c_2 a_2^2 + c_3 a_1 a_2 + c_4 a_1 + c_5 a_2 + c_6$$

with coefficients $c_1, ..., c_6 \in \mathbb{R}$.

Quadratic reward functions allow us to construct an equilibrium oracle μ for AROGU without any specific knowledge about the game (see Appendix A.1).

7.4.2 Definition of parameterized games

Definition 17 (Cournot Game)**.** *A parameterized Cournot Game (CG) with parameter $\lambda := p_{max} - c$ is defined by $I = \{1, 2\}$, $A = [0, \lambda]$, $r_1(a_1, a_2) = -a_1^2 - a_1 a_2 + \lambda a_1$ and $r_2(a_1, a_2) = -a_2^2 - a_1 a_2 + \lambda a_2$.*

In the continuous version of Braess' Paradox, agents do not choose one of the available routes but decide which fraction of their flow they send through each route (see Appendix A.4 for the derivation of the reward functions):

Definition 18 (Continuous Braess Paradox)**.** *A parameterized continuous Braess Paradox (BP) with parameter $b \geq 0$ is defined by $I = \{1, 2\}$, $A = [0, 1]$, $r_1(\boldsymbol{a}) = -4a_1^2 + (b-5)a_1 - 4a_2 + 17$ and $r_2(\boldsymbol{a}) = -4a_2^2 - 4a_1 + (b-5)a_2 + 17$.*

Varying b changes the attractiveness of taking the "cooperative" routes, compared to the "selfish" route. This degree of freedom is sufficient to change the structure of the game and its equilibria.

7.4.3 Metrics

To measure the performance of AROGU, we use the following metrics:

Definition 19. *For an action space A and a restriction $R \subseteq A$, the* degree of restriction *is defined as $\mathfrak{r}(R) := 1 - \frac{|R|}{|A|}$, where $|R|$ is the size of R as defined in Section 7.2.1.*

Definition 20. *The* relative improvement *of a restriction R is*

$$\Delta(R) := \frac{\min_{a \in \mathcal{N}|_R} \mathfrak{u}(a) - \min_{a \in \mathcal{N}} \mathfrak{u}(a)}{|\min_{a \in \mathcal{N}} \mathfrak{u}(a)|} .$$

Moreover, we measure the number of oracle calls in AROGU as a proxy for the cost of finding an optimal restriction, implying that μ is assumed to have constant run-time[7].

7.4.4 Theoretical expectation

Cournot Game

The optimal restriction R^* for the CG with parameter λ is $R^* = [0, \frac{\lambda}{4}) \cup [\frac{\lambda}{2}, \lambda)$ with a constant degree of restriction $\mathfrak{r}(R^*) = 25\%$ (see Appendix A.2 for details). We expect the result of AROGU to fluctuate around these values, depending on the size of ε. The value of λ does not change the structure of the game but scales the action space size, the equilibria, and the restrictions, thereby providing insights into the scaling behavior of AROGU.

Braess' Paradox

The unique unrestricted NE (user equilibrium) is $\left(\frac{b-5}{8}, \frac{b-5}{8}\right)$, while the governance optimum is $\left(\frac{b-9}{8}, \frac{b-9}{8}\right)$. This means that for $b \notin [5, 17]$, both joint actions coincide as the action space is $A = [0, 1]$, and restricting the action space cannot improve the MESU. Within the interval $[5, 17]$, however, the agents' actions need to be pushed down (toward 0) to match the governance optimum, giving the optimal restriction $R^* = A \setminus \left[\frac{b-9}{8}, \frac{b-5}{4}\right]$ with a degree of restriction of $\mathfrak{r}(R) = \frac{b-5}{4}$ on $b \in [5, 9]$ and $\mathfrak{r}(R) = \frac{17-b}{8}$ on $b \in [9, 17]$. The formal analysis is given in Appendix A.5.

7.4.5 Experimental results

Cournot Game

Figure 7.3 shows the results of AROGU for $\lambda \in \{10, 11, ..., 200\}$ with $\varepsilon = 0.1$. The MESU of the restrictions found by AROGU is consistently $\approx 12.5\%$ larger than the unrestricted MESU, which matches the theoretical prediction. Together with a degree of restriction of $\approx 25\%$, this answers Q1 affirmatively for this setting. The number of oracle calls (i.e., tentative restrictions) increases quadratically in $|A|$ (see Appendix A.3), as opposed to the exponential bound shown above. Regarding Q2, this indicates that the pruning technique eliminates a large part of the possible restrictions.

Braess' Paradox

Figure 7.4 shows the results of AROGU for $b \in [4, 18]$ in steps of 0.1 with $\varepsilon = 0.001$. Let us have a look at $b \in [5, 9]$ first: While the user equilibrium decreases when b exceeds 5 (agents find it increasingly advantageous to take the center route, causing more and more congestion), this effect can be completely eliminated using restrictions (as we see, the restricted MESU stays at 34). For $b > 9$, the optimal restriction stops pushing the agents to choose action 0 but allows an interval of $[0, \frac{b-9}{8}]$. Hence, both governance optimum and user equilibrium have increasing governance utility, eventually joining at $b = 17$. Again, the degree of restriction and the restricted MESU approximately match the theoretical optimum (Q1). Since the action space

[7] Nemirovsky and Yudin, 1983 have defined the *oracle complexity* of an algorithm to capture scenarios like this.

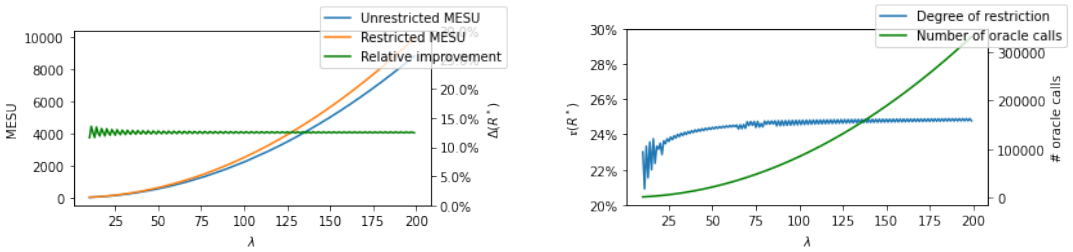


FIGURE 7.3: Unrestricted and restricted MESU, relative improvement, degree of restriction and number of oracle calls for the Cournot Game.

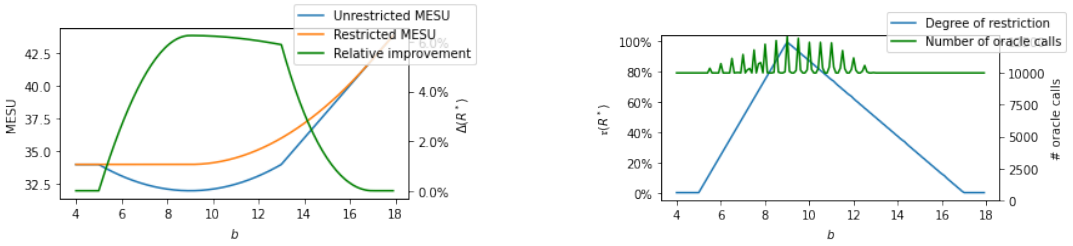


FIGURE 7.4: Unrestricted and restricted MESU, relative improvement, degree of restriction, and number of oracle calls for the Braess Paradox.

has a constant size, the number of oracle calls is asymptotically constant, only impacted by the required degree of restriction and the subsequent pruning (Q2).

7.5 Summary

In this chapter, we have addressed the governance learning problem for continuous action spaces. Specifically, we had to deal with the challenges of representing continuous restrictions and searching a potentially unlimited number of restrictions to find an optimal one.

The AROGU algorithm can significantly improve a game's minimum equilibrium governance utility by aligning user equilibrium and governance optimum with non-discriminatory restrictions. While its theoretical complexity is (as expected from the problem specification) exponential in the size of the action space, we have (a) solved the combinatorial explosion with respect to the number of agents by only using non-discriminatory restrictions, and (b) shown empirically that our breadth-first search approach manages with a much lower than exponential number of oracle calls in practice. This makes AROGU applicable to real-world settings like the Cournot Game, even though the theoretical results do not guarantee its efficiency.

We conjecture that restriction-based mechanism design approaches (ultimately, the vision is that of optimally restricted general Stochastic Games) are a crucial step to building powerful governance entities for an emergent multi-agent society. For this vision, however, it is necessary to develop computational frameworks that support this governance paradigm from both the agent and the environmental perspective. In other words, commonly used multi-agent frameworks need to acquire the ability to act as electronic institutions, coordinating not only the interplay between agents and environment but also the information flow and learning behavior with respect to restrictions and restricted actions.

Chapter 8

Implementing dynamic restrictions in MARL frameworks

Both the execution of an ARMAS and the governance learning problem require multi-agent systems to communicate action space restrictions between governance and agents. In contrast to a standard MAS, this requirement manifests as a rather complex dispatching algorithm that queries the governance before each agent step while correctly assigning rewards and handling terminations.

Existing MARL implementations do not explicitly offer support for such a governance, but it turns out that their structure is sufficiently generic to be enhanced with a governance wrapper that provides the above-mentioned functionality.

In this chapter, we motivate, implement, and analyze a governance wrapper for *PettingZoo* (Terry et al., 2021), a state-of-the-art MARL framework.

Personal Contribution. *I defined the architecture of the code package, developed two of the three use cases, and was the primary author of the text. The implementation was jointly written with Tim Grams, who also provided the navigation use case (Section 8.3.2).*

8.1 Motivation

Since its first release in 2017, OpenAI's *Gym*[1] environment specification (Brockman et al., 2016) has become the standard for Reinforcement Learning environments represented as MDPs or, more generally, as POMDPs. Gym's minimalistic design offers enough freedom and flexibility to allow users to create and train RL agents in their own environments. Consequently, popular RL frameworks like Keras RL (Plappert, 2016), Tensorforce (Kuhnle, Schaarschmidt, and Fricke, 2017), Coach (Caspi et al., 2017), Acme (Hoffman et al., 2020), Stable Baselines (Raffin et al., 2021), and CleanRL (Huang et al., 2022) have adopted Gym environments as their default environment class.

However, Gym is designed for single-agent learning only, employing a loop between the agent `act()` and environment `step()` until the episode is `done` (see Figure 8.1). To enable multi-agent settings, approaches based on SGs or POSGs have been proposed, but, as Terry et al., 2021 have pointed out, implementing them in code raises several unsolved challenges[2]. To overcome these limitations, they introduce the *Agent Environment Cycle* (AEC) model and the corresponding PettingZoo library, which has gained widespread adoption and works seamlessly with RL

[1] By now, Gym is maintained by the Farama Foundation (`https://farama.org/`) under the name *gymnasium*.

[2] Specifically, they criticize dummy actions for turn-based games and fixed number of agents for the POSG implementation of RLlib (Liang et al., 2018), and the lack of intermediate rewards and continuous action spaces for the Extensive-Form Games of OpenSpiel (Lanctot et al., 2020).

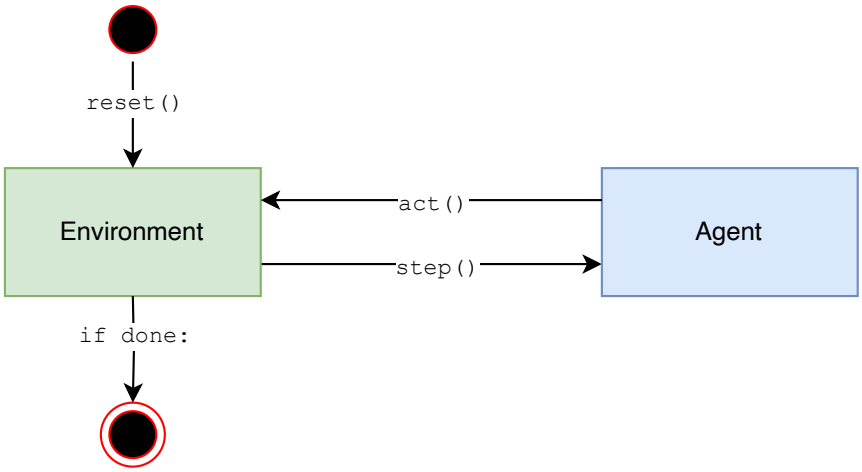


FIGURE 8.1: Agent-Environment loop for a single-agent setting (as implemented in Gym).

frameworks such as The Autonomous Learning Library (Nota, 2020), AI-Traineree (Laszuk, 2020), PyMARL (Samvelyan et al., 2019), RLlib (Liang et al., 2018), Stable Baselines (Hill et al., 2018; Raffin et al., 2021), CleanRL (Huang et al., 2022; Terry, Black, and Hari, 2020)), and Tianshou (Weng et al., 2022).

In Gym and PettingZoo, agents typically have access to the same set of actions throughout an episode, either as a discrete set or a box-shaped continuous space[3]. Recent versions of PettingZoo seem to allow changing observation and action spaces at run-time, but instructions are inconsistent[4], and compatibility issues arise with RL algorithms that expect invariant input-output shapes (which is the case for most common deep learning algorithms).

A commonly used solution for dynamic action spaces is *invalid action masking* (Vinyals et al., 2017; Huang and Ontañón, 2022). However, this method, which involves providing a Boolean vector of valid and invalid actions as part of the observation, is limited to discrete spaces and can be inefficient. For instance, in Dota 2, where the action space comprises 1,837,080 actions (Berner et al., 2019), the masking approach becomes burdensome with respect to the storage of observation batches.

Dynamic restrictions of the action spaces, as imposed in many real-world scenarios by physical, legal, or other constraints (Mandel et al., 2017; Boutilier et al., 2018; Chandak et al., 2020), can therefore not be represented by existing RL frameworks in a principled way. To address this limitation, we propose an extension with the following components:

1. The action space, referred to as the *base space*, remains static.
2. Agents receive a *restriction* as part of their observation, representing an arbitrary subset of the base space.
3. Restrictions, represented by `gym.spaces`, efficiently capture arbitrary sets of valid actions.

[3]More complex spaces can be obtained by defining tuples or dictionaries of basic spaces, but these are internally flattened for processing and, therefore, need to have fixed shapes as well.

[4]The documentation contains an example with the comment "If your spaces change over time, remove this line (disable caching)", but also says "This space should never change for a particular agent ID.". The docstring of `AECEnv.action_space()` even states that the function "MUST return the same value for the same agent name".

4. The internal representation of valid actions in a restriction is opaque, while compatibility with RL models is ensured through fixed-length flattening.
5. Restrictions can be defined by the environment or provided by a *restrictor* agent which produces restrictions as actions.
6. A restrictor agent can be treated like any other agent and may be an RL agent or a static function.
7. The interplay between the environment, restrictor, and agents is managed by a *restriction wrapper*.

Our proposed *Dynamically Restricted Action Spaces for Multi-Agent Systems* (DRAMA) implementation is based on the ARMAS model and notation introduced in Chapters 2 and 3. In this chapter, we explain our reference implementation (Section 8.2) and present three use cases in Section 8.3.

8.2 Implementation

In the standard AEC, three entities are of importance: `gym.Spaces` are passed back and forth between an `AECEnv` and one or more `Agents`, as shown in the minimal execution loop (see Algorithm 3).

Algorithm 3: Agent-environment cycle (AEC).

```
env.reset()
for agent in env.agent_iter():
    observation, _* = env.last()
    action = agents[agent].act(observation)
    env.step(action)
```

DRAMA directly builds upon this setup, using the exact same loop. We define three more classes with their respective base classes, each corresponding to one of the above entities:

`class Restriction(gym.Space)` represents any subset of an action space.

`class Restrictor(Agent)` is an agent whose actions are `Restrictions`.

`class RestrictionWrapper(AECEnv)` manages the order of agent and restrictor actions, as well as the enhancement of agent observations with the respective restrictions.

This reflects one of the fundamental design decisions of DRAMA: The agent policies π and the restriction policies ρ are defined in the same way, such that both kinds of policies can be learned within the training process implemented by PettingZoo-compatible MARL frameworks. Hence, any restriction needs to be a valid `gym.Space` which can be batched for training and evaluation workflows, and restriction policies are queried (and potentially trained) like an agent policy in the AEC. Moreover, DRAMA is designed to be extensible by sub-classing any of its components (e.g., to define more complex restrictions), even though the reference implementation already contains the necessary classes for a range of applications.

8.2.1 Restriction

Discrete restriction For discrete spaces `gym.Discrete(n, start=s)` with the action set $\{s, s+1, ..., s+n-1\}$, restrictions have traditionally been implemented as action masks. These masks are Boolean vectors of length n, where each entry indicates whether an action is allowed (`True`) or forbidden (`False`). While this approach is suitable for small n, it becomes inefficient when n is large and only a small fraction of actions is allowed at each step. To address this issue, we introduce two implementations of `DiscreteRestriction`: `DiscreteSetRestriction` stores a set of allowed actions, while `DiscreteVectorRestriction` follows the conventional vector representation, but as a subclass of `gym.Space`.

Continuous restriction We provide two restriction classes for one-dimensional continuous spaces as used in Chapter 7: The `IntervalUnionRestriction` class represents a union of closed allowed intervals, and `BucketSpaceRestriction` represents a Boolean vector of equally sized allowed and forbidden buckets. For multi-dimensional spaces, these classes can be combined as long as the dimensions are independent, such as in `gym.Box` spaces. For more complex dependencies between dimensions (e.g., the "circle restriction" $A = [0,1]^2$, $R = \{a \in A : ||a||_2 \leq 1\}$), we offer the generic `PredicateRestriction` class, which supports the definition of an arbitrary predicate function, but is not flattenable (see also "Agent observations" in Section 8.2.3).

8.2.2 Restrictor

The `Restrictor` class is designed as a regular agent but with `Restrictions` as actions. Consequently, the AEC of PettingZoo can handle DRAMA natively without any special considerations for restrictors.

Observation space The observation structure of a restrictor is not predefined but comprises the entire `env.state()` by default. As such, it includes any information available in the environment, such as the identifier of the next agent or the agents' latest rewards. In particular, it is not necessarily linked to the observation functions σ of the agents. Optionally, a custom preprocessing function can be applied before calling `Restrictor.act()` (see Section 8.2.3).

Action space Usually, the action space of a restrictor comprises all possible restrictions of a given agent action space A (for a more flexible mapping, see "Action Post-Processing" in Section 8.2.3). To represent this space, which is mathematically equivalent to the power set of A, we provide the base class `RestrictorActionSpace`. Initialized with the base space A, it enables the restrictor to generate any `Restriction` compatible with A.

Reward function The restrictor's reward function[5] can be constructed using any information available in the environment. By default, we use the *social welfare* $r = \sum_{i \in I} r_i$, which sums over all agent rewards.

[5] In ARMAS terminology, this is the governance utility function.

8.2.3 Restriction wrapper

The `RestrictionWrapper` is responsible for managing the interaction between the environment, agents, and restrictor(s). Prior to querying an agent, the wrapper requests a restriction from the corresponding restrictor and then passes this restriction to the agent as part of its observation.

Agent-restrictor mapping In the simplest case of DRAMA, a single restrictor is utilized for all agents. However, multiple restrictors can be defined to accommodate, for instance, agents with different action spaces. By establishing a mapping between the set of agents and the set of restrictors, the wrapper can obtain the appropriate restrictions for each agent from the corresponding restrictor.

Observation pre-processing As mentioned above, the default observation for a restrictor is set to `env.state()`. Optionally, a pre-processing function can be specified for each restrictor, which the wrapper applies in analogy to the agent observation functions σ.

Action post-processing In situations where a restrictor functions as a learning agent, its restriction space might not align with the action space of the (ordinary) agents. For instance, if the restrictor selects from a predetermined set of restrictions, it is advisable to define its action space as `Discrete`, and subsequently map the chosen action to a corresponding restriction. To accommodate these scenarios, a restrictor-generated restriction can undergo post-processing before being passed to the respective agent. This allows for seamless integration and compatibility between the restrictor's actions and the agent's expected input.

Agent observations The observation received from the environment is passed to the agents as part of a two-key dictionary: `{"observation": ..., "restriction": ...}`. The keys of this dictionary can be customized. For example, seamless compatibility with Tianshou's built-in agents requires using `DiscreteVectorRestriction` in conjunction with the restriction key `action_mask`.

By default, the wrapper flattens all observation and action spaces, including restrictions, into fixed-shape `gym.Box` spaces (with possible padding or overflow) to ensure compatibility with existing libraries. To anticipate the emergence of algorithms in the future that can natively handle a wider range of spaces (i.e., any class adhering to the `gym.Space` specification), we offer three more options: (a) Flattening into variable-shape `gym.Sequence` spaces, (b) applying a custom flattening function, and (c) using the original spaces without flattening.

Restriction violations The consequences of violating a restriction can be arbitrarily defined and may be individual per agent. By default, an invalid action causes the wrapper to throw a custom `RestrictionViolationException`. An invalid action can also be replaced by sampling uniformly from the allowed set or projected to the nearest action, and custom methods can be added by specifying a function for each restrictor.

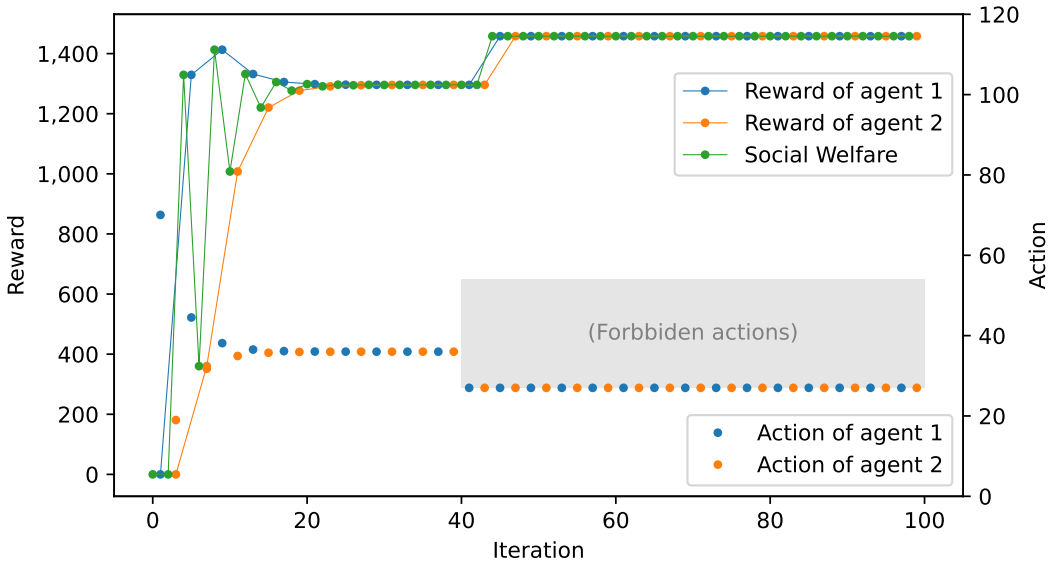


FIGURE 8.2: Reward and action curves for the Parameterized Cournot Game with a learning restrictor. At iteration 40, when the restriction is defined, the reward of both agents undergoes an abrupt improvement, showing that the restriction is boosting social welfare. This plot explicitly shows the alternation between restrictor and agent actions.

8.3 Use cases

In this section, we provide several illustrations of how DRAMA can be applied to a variety of multi-agent scenarios. It is important to note that these use cases are intentionally designed to be simple[6]. The primary emphasis is placed on exploring the interaction and learning dynamics between the agents and restrictor(s) facilitated by the restriction wrapper.

8.3.1 Learning optimal restrictions in a continuous-action game

To demonstrate learned restrictions in a continuous action space, let us consider the Parameterized Cournot Game as defined in Chapter 7.

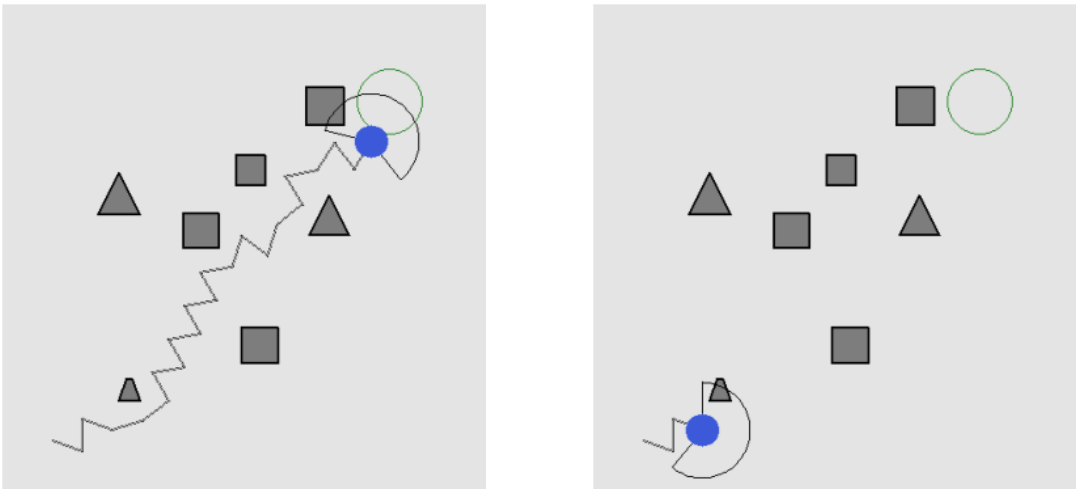
We aim to learn an optimal restriction by observing the actions of the agents. The restrictor observes the agents' behavior until their strategies converge, and then formulates an optimal restriction based on its estimation of the game's parameters, namely p_{max} and c. In response to the restriction, the agents adjust their actions, resulting in a stable outcome that differs from the original equilibrium. The governance reward (i.e., social welfare), as depicted in Figure 8.2, exhibits a noticeable increase at the point where the restrictor comes into action. While this learning behavior is simplistic, it effectively demonstrates the dynamic interaction between the restrictor and the agents.

[6]Two of the three multi-agent systems are taken from previous chapters, while the third one was first defined in Grams, 2023.

8.3.2 Training an RL agent with dynamic restrictions in an obstacle avoidance scenario

Using DRAMA, we train an RL agent to navigate a dynamic environment with complex action spaces. In this scenario, restrictions are not necessarily tied to the agent's observation but serve as an additional source of information. Consider a navigation task where an agent aims to reach a goal on a two-dimensional map. The environment can contain temporary obstacles, such as other agents or objects, which are not directly sensed by the agent. An external entity can, therefore, suggest restrictions on the agent's action space to avoid collisions, such that the agent can select actions that maximize the expected return over varying subsets of the action space.

The environment, as shown in Figure 8.3, is a 15×15 field where the agent (blue circle) has a location $l_t \in \mathbb{R}^2$, perspective $p_t \in [0, 360]$, and starting position $p_0 = (2, 2)$. At each time step t, the agent observes l_t and p_t, as well as the distance and angle to the goal $g = (12, 12)$. It then chooses as its action an angle $a_t \in [-110, 110]$ to determine the subsequent step's direction (with a step length of 1). The goal is reached when the distance $d(p_t, g)$ is ≤ 1. Seven obstacles of various shapes, defined by their location and radius, are randomly generated at each episode's start and are not observed by the agent.



(A) Projection (B) Unrestricted

FIGURE 8.3: Examplary agent trajectories for the navigation task. While projection allows surrounding obstacles by bringing actions into the feasible set, an unrestricted agent is stuck with repeated forbidden movements.

To handle the dynamic restrictions, we employ the `IntervalUnionRestriction` class to represent the union of open intervals that correspond to actions leading to collisions. The valid action space is also computed based on these intervals. Since the number of intervals can vary, the boundaries cannot be treated as static model inputs. We train a *Twin Delayed DDPG* (TD3) algorithm (Fujimoto, Hoof, and Meger, 2018) to find the shortest path to the goal, comparing two cases: First, the agent learns without knowledge of restrictions and may collide with obstacles. Second, we provide dynamic restrictions to the agent using DRAMA, allowing the agent to choose feasible actions that are closest to its preferred actions.

The experiment demonstrates that DRAMA improves learning in scenarios with dynamic action spaces: The fraction of evaluation environments where the agent succeeds significantly increases when handling restrictions, as depicted in Figure 8.4

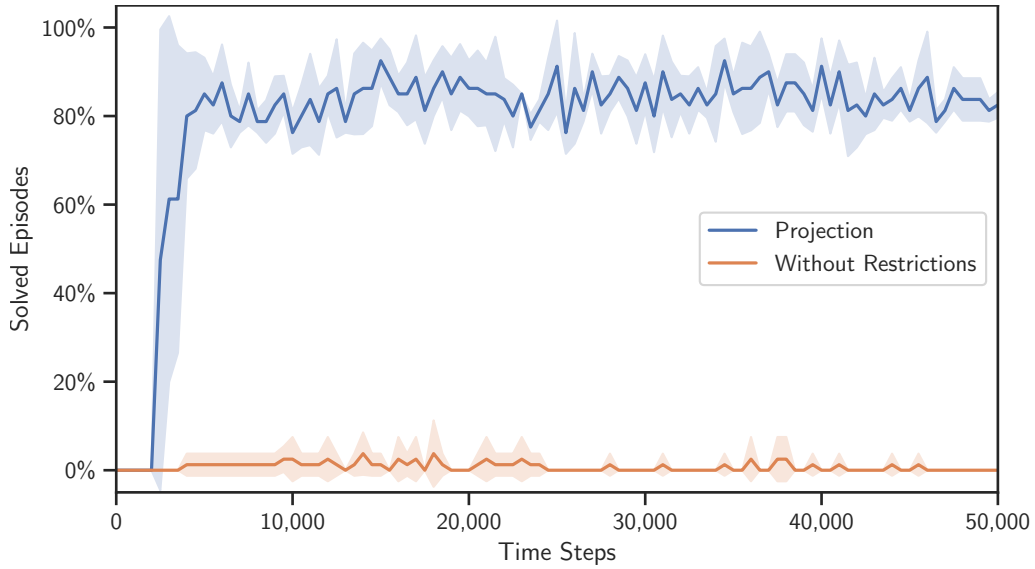


FIGURE 8.4: Mean and standard deviation for the fraction of solved evaluation episodes, measured every 500 steps.

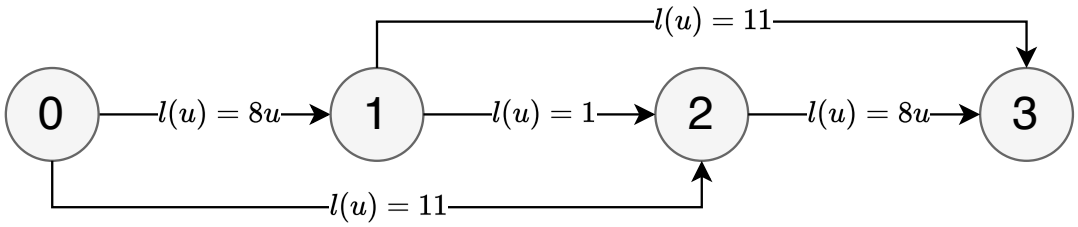


FIGURE 8.5: Traffic network with dynamic restrictions. The restrictor learns which roads (edges) to close in order to maximize throughput, measured as the negative mean of all agents' travel times.

for multiple model runs. With projection, most obstacles can be smoothly navigated, while unrestricted agents frequently encounter obstacles, as can be seen from the trajectories in Figure 8.3. We note, however, that the average number of steps remains relatively high compared to the shortest path, even when restrictions are used and the success rate is high. This highlights the need for agents to make more informed decisions when dealing with variable action spaces. Notably, Grams, 2023 have recently explored RL architectures for dynamic restrictions and conducted experiments in this environment.

8.3.3 Training an RL restrictor for a discrete action space

Consider a traffic network where agents $i \in I$ are tasked with selecting a shortest route from their starting points s_i to their respective destinations d_i. The travel time along each road segment is influenced by its utilization, i.e., the relative number of agents using it. This is described by a latency model $l_e(u) = a + bu^c$, where individual parameters (a, b, c) are assigned to each edge (see Maerivoet and Moor, 2005).

In this context, we consider once more a variation of Braess' paradox (see also Section 7.2.4), where closing roads in the network can actually lead to an increase in overall throughput under specific conditions. Motivated by this paradox, we train a restrictor to determine the optimal configuration of open and closed roads. This

restrictor operates within a network of self-interested agents who aim to minimize their individual travel times. The restrictor serves as a governance mechanism with the objective of minimizing the total travel time of all agents.

For the sake of simplicity, we utilize the graph network depicted in Figure 8.5. All simple paths in this network are enumerated and used as the discrete action space for the agents. At each step, each agent i selects the shortest route from s_i to d_i based on the current edge latencies. Without any governance in place, all agents traveling from $s = 0$ to $t = 3$ naturally choose the route $0 \to 1 \to 2 \to 3$, resulting in an average travel time of ≈ 17 (as shown by the red line in Figure 8.6).

To improve traffic flow, we introduce a governance mechanism that can selectively close individual roads (i.e., remove edges). The action space for the governance is represented by `MultiBinary(5)`[7]. While the agents use a fixed strategy to determine the shortest routes given the restrictions, the governance, acting as an RL agent, learns the optimal set of restrictions by observing the agents and the environment. In our approach, we use an off-the-shelf DQN algorithm (Mnih et al., 2013), where the current edge latencies serve as the observation.

In our setting, the governance learns to close the edge $1 \to 2$ (as illustrated in Figure 8.7), leading the agents to distribute themselves across the routes $0 \to 1 \to 3$ and $0 \to 2 \to 3$, with each route having an approximate utilization of 50%. As a result, the average travel time decreases to ≈ 15 (indicated by the green line in Figure 8.6), which indeed represents the optimal configuration for the given network structure.

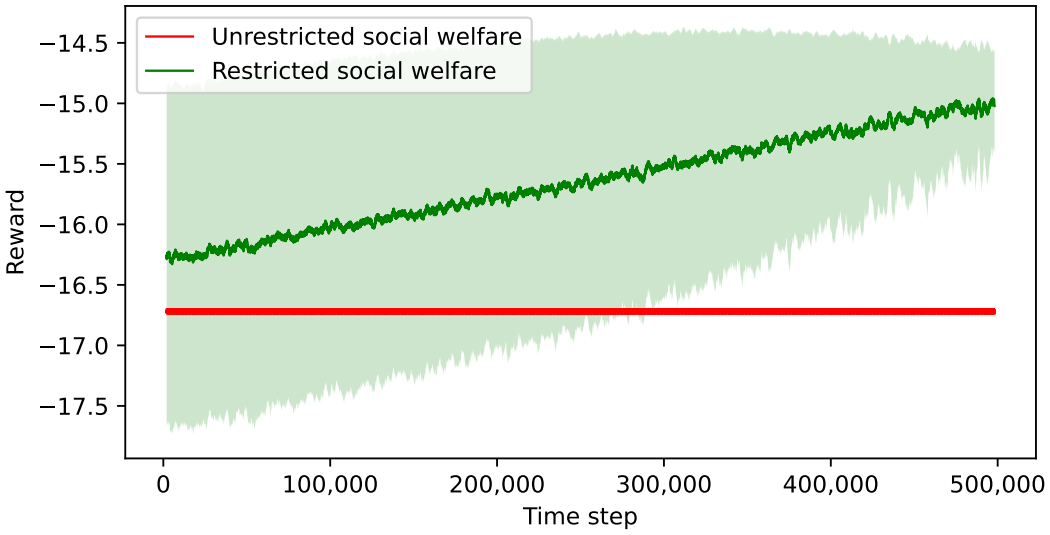


FIGURE 8.6: Social welfare (sliding average over 5,000 steps, mean and standard deviation over 5 runs) of traffic network during training (green), compared to unrestricted traffic (red).

8.3.4 Blue-sky idea: Text-based governance of LLM debate games

The governance of human communities primarily relies on the formulation and implementation of laws conveyed through natural language. These laws encompass

[7] In our implementation, however, we ensure that there is always at least one open path available for agents to use.

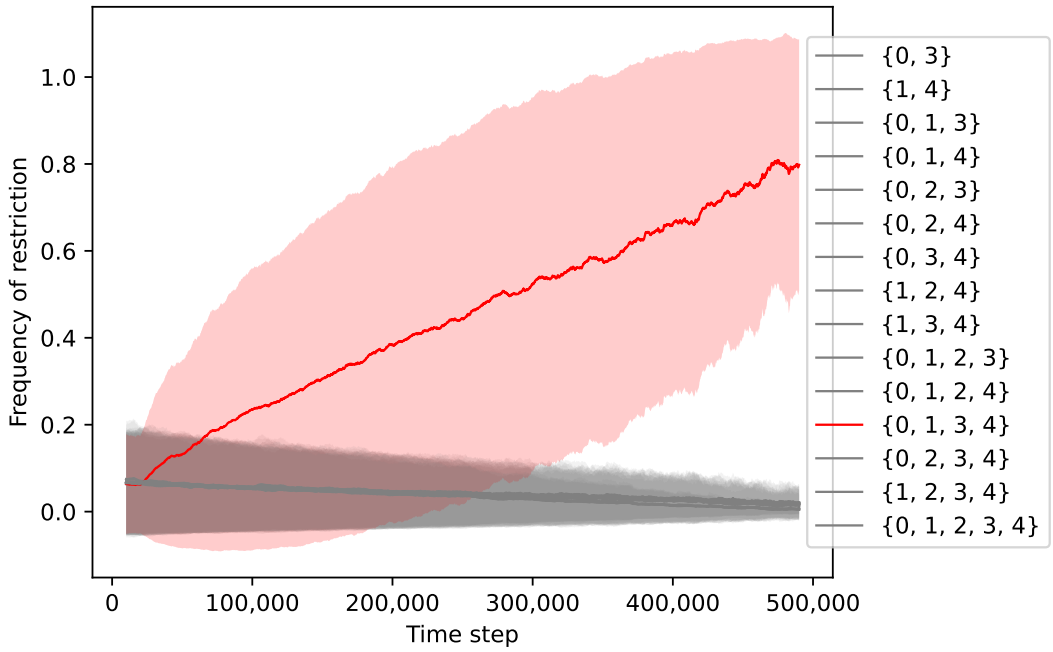


FIGURE 8.7: Frequency of the restrictions chosen by the restrictor during training (sliding average over $1,000$ steps, mean and standard deviation over 5 runs). The dominant restriction $\{0,1,3,4\}$ corresponds to closing the edge $(1,2)$ in the network.

explicit and implicit guidelines delineating permissible actions and behavioral constraints (i.e., action space restrictions), as well as information about penalties (i.e., reward shaping definitions) for breaking the restrictions. It therefore appears plausible that LLMs possess the capacity to learn and optimize text-based rules in alignment with the behavioral tendencies exhibited by the (human and/or artificial) agents subject to these rules. The recently published *ChatArena* library (Wu et al., 2023) offers a testbed for such environments, while Park et al., 2023 explore language-based *Generative Agents* which can exhibit emergent social behaviors, albeit without governance. Combining these approaches with DRAMA could bridge the gap between RL and language models, thereby facilitating the automated generation of intricate rule sets guided by specific objectives. Ultimately, this might even hold the potential to revolutionize the process of law creation for human and artificial communities.

8.4 Summary

In this chapter, we have described an extension of the Agent Environment Cycle for MARL with a component that has thus far been handled in a limited, "hacky" way: Complex dynamic action space restrictions.

Many practical scenarios involve nuanced action constraints, such as physical, legal, or safety considerations, which require intelligent agents to navigate through a complex decision-making space while adhering to restrictions. By explicitly integrating and modeling dynamic action space restrictions, we provide a more realistic and comprehensive framework for the development of intelligent agents (and self-learning restrictors!) capable of effectively operating within the confines of real-world constraints.

We deliberately chose to seamlessly place the DRAMA extension within a simple and widely used MARL framework instead of defining a new platform from scratch. Our experience has shown that there is a myriad of highly specific libraries available, many of which lack widespread adaptation, and almost none can keep up with the integration of novel, state-of-the-art algorithms. As a consequence, we want to encourage the research and development of restriction-aware RL agents (and restriction classes) to pave the way for practical applications in domains where compliance with explicit rules and regulations is paramount.

Chapter 9

Evaluating efficacy and fairness of restriction-based governance

The previous chapters have investigated solutions to the governance learning problem for various classes of multi-agent systems and proposed a practical approach for integrating restriction-based governance mechanisms in existing MARL frameworks.

In this chapter, corresponding to the last publication, we use the domain of traffic management to evaluate the ARMAS approach with respect to its efficacy (i.e., by how much does restricting action spaces improve the governance utility?) and fairness (which, of course, needs to be motivated and defined before it can be quantified). In Chapters 6 and 7, we argued that our restriction-based governance is fair because it treats all agents the same way. As opposed to this notion ("fairness of treatment"), we consider here the notion of "fairness of outcome", which takes into account that equal treatment might affect different agents in different ways.

An empirical comparison with a commonly used reward-shaping approach leads us back to the motivation of this thesis: Restriction-based governance, as proposed by the ARMAS model, constitutes a valuable complement to reward shaping for achieving system-level goals in multi-agent systems.

Personal Contribution. *I defined the model and the research question, developed the experimental setup, and was the primary author of the text. The experiments were jointly conducted with Tim Grams, especially the experiment on the $G_{n,p}$ graph. The selection of the application domain and the discussion of the results were joint work with Christian Bartelt and Heiner Stuckenschmidt.*

9.1 Motivation

In competitive multi-agent systems, the selfish strategies of the participating agents (i.e., strategies that maximize the agent's utility) often deviate from the socially optimal solution, which maximizes the social welfare. This discrepancy is a defining trait of the class of *Social Dilemmas*, or *Collective Action Problems* (Olson, 1965; Van Lange et al., 2013). It emerges across diverse application areas, with traffic flow optimization (Beckmann, McGuire, and Winsten, 1955) being a notable example: Agents leveraging heuristics or machine learning to identify the shortest routes on a directed weighted graph might inadvertently reduce the social welfare significantly below the optimal value (Joshi, Joshi, and Lamb, 2005). For affine latency functions on graph edges, this *price of anarchy* can be as high as 33% (Roughgarden and Tardos, 2002) and can rise indefinitely for non-linear latency functions (Lin et al., 2011).

However, social welfare is not the only benchmark for optimality in a MAS. Other objectives, like fairness or even goals unrelated to agents, can influence the

design and functioning of such a system. Within the traffic context, this objective might manifest as a utility function for traffic authorities or state administration, aiming to curtail road erosion, decrease noise, or increase toll revenue, for instance. To emphasize the universality of such objectives, we use the term *Governance Utility* instead of social utility. A MAS then becomes a *Governance Dilemma* when a stable, joint strategy fails to attain the maximum governance utility. Contrasting with a social dilemma, this broader definition captures scenarios where the governance objective is not a straightforward (i.e., equally weighted linear) function of agent objectives.

Let us now take a closer look at fairness: The prevalent definition of social welfare ($\mathfrak{u} = \sum u_i$) does not distinguish between two agents achieving equal utility ($u_1 = u_2 = x$) and one agent achieving $u_1 = 2x$ while the other gets $u_2 = 0$. To prevent such disparities, the governance utility could include a metric that diminishes with increased inequality between agents (e.g., variance, entropy, or Gini coefficient). Illustratively, consider a traffic junction where each car's utility inversely corresponds to its waiting time. From a social welfare viewpoint, a scenario where a hundred cars each wait five seconds is identical to one where a single car waits 500 seconds while all others proceed immediately. However, the latter scenario, being inherently inequitable, would be deemed suboptimal. The term *equity* formalizes this intuition as "the absence of unfair, avoidable or remediable differences among groups of people" (World Health Organization, 2023); we use this as a well-defined operationalization of the subjective word "fairness" (see Section 9.3.4 for our concrete measure of equity).

In this chapter, we investigate two governance paradigms—action-space shaping and reward shaping—in terms of efficacy (i.e., improvement in social welfare) and equity (i.e., equitable treatment). Specifically, we compare dynamic action-space restrictions with dynamic marginal tolling (Sharon et al., 2017b). The traffic management domain serves as a well-suited application area to explore and compare the effect of both governance schemes.

We will, in contrast to Chapters 5 to 7, not focus on finding optimal restrictions for a given traffic network. There exist numerous complexity and inapproximability results for such systems in the traffic domain, one being Roughgarden, 2006's proof that barring $\mathcal{P} = \mathcal{NP}$, no polynomial-time algorithm can achieve an approximation ratio $< \frac{n}{2}$ to find optimal edge constraints in a congested network with n nodes[1]. We will, therefore, use existing knowledge about suitable restrictions and concentrate on evaluating their impact relative to an established reward-shaping technique.

9.2 The effect of restrictions

9.2.1 Static restrictions

The most simple and extensively studied social dilemmas are two-player, two-action matrix games such as the Prisoner's Dilemma (Rapoport and Chammah, 1965), Stag Hunt (Skyrms, 2003) and the Chicken Game (Rapoport and Chammah, 1966). Clearly, restriction-based governance is inappropriate in such cases: Forbidding even a single action would result in fully prescribed strategies, simplifying agent behavior to the point of triviality.

[1] Note that this hardness result is due to the fact that we are now considering general POSGs without any simplifications or additional assumptions.

		Agent 2		
		a	*b*	*c*
Agent 1	*a*	0, 0	0, 1	0, 0
	b	1, 0	2, 2	0, 3
	c	0, 0	3, 0	1, 1

(A) Restricting action *c* increases the MESU.

		Agent 2		
		a	*b*	*c*
Agent 1	*a*	1, 1	1, 0	0, 0
	b	0, 1	2, 2	3, 6
	c	0, 0	6, 3	4, 4

(B) Restricting actions never increases the MESU.

FIGURE 9.1: Payoff matrices for exemplary matrix games where action-space restrictions have different effects. Following conventional notation, the first number in each cell is the utility of agent 1, and the second one of agent 2.

Let us, therefore, consider a two-player, three-action matrix game with symmetric payoffs and assume the governance utility $\mathfrak{u}$ to be the social welfare. The following examples offer some initial insight into the possible impacts of restrictions:

Example 9. *Given the game payoffs in Figure 9.1a, where both agents have the action space* $A = \{a, b, c\}$*, the unique (pure) NE in an unrestricted scenario is the joint strategy* (c, c) *with agent utilities* $u_1(c, c) = u_2(c, c) = 1$ *and* $\mathfrak{u}(c, c) = 2$*. The unique social optimum is* (b, b) *with* $\mathfrak{u}(b, b) = 4$*.*

By excluding action c for both agents, we can align the unique NE with the SO at (b, b)*. As a result, the action space restriction increases the MESU from 2 to 4.*

Example 10. *Conversely, if the payoffs are given by Figure 9.1b, the unique unrestricted NE is* (c, c)*, where* $u_1(c, c) = u_2(c, c) = 4$ *and* $\mathfrak{u}(c, c) = 8$*. The SOs are* (b, c) *and* (c, b)*, both with* $\mathfrak{u}(b, c) = \mathfrak{u}(c, b) = 9$*.*

If we were to eliminate action c, both the NE and the SO become unattainable. The new NE evolves to (b, b)*, with utilities* $u_1(c, c) = u_2(c, c) = 2$ *and* $\mathfrak{u}(c, c) = 4$*. Even though this NE now matches the* new *SO, its governance utility is less than its original value.*

Example 11. *Revisiting the payoffs from Figure 9.1a, but this time restricting action a, we observe no governance effect since this limitation does not impact either the SO* (b, b) *or the NE* (c, c)*.*

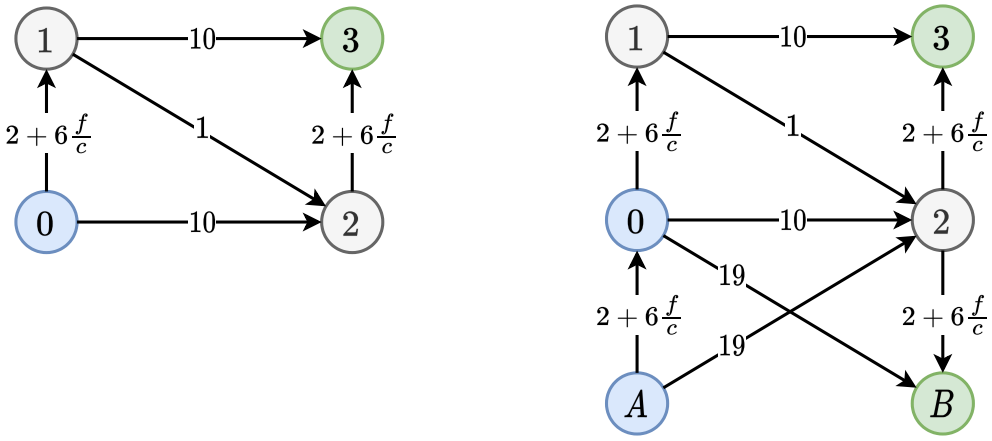
9.2.2 Dynamic restrictions

Braess' Paradox, depicted in Figure 9.2a, is a frequently referenced illustration of the effectiveness of restrictions in stateful MAS[2]. In the present work, we use edge latency functions of the form

$$l(f) = a + b\left(\frac{f}{c}\right)^d \tag{9.1}$$

with parameters $a > 0$, $b \geq 0$, and $c, d \geq 1$. This model is based on a proposal of the Bureau of Public Roads (BPR) (Public Roads, 1964) and is a common choice in the literature. The latency functions originally suggested by Braess, $l(f) = 0$ and

[2]The formal definition of graph and latency functions is provided in Section 9.3.



(A) Original Braess Paradox

(B) Double Braess Paradox

FIGURE 9.2: Each edge of these traffic networks has a latency function $l_e(f)$, indicating the length (i.e., travel time) of the edge for a given flow $f \in \mathbb{N}_0$ (c is a capacity parameter). For agents traveling from node 0 to 3, social welfare is increased from -17 to -15 by closing the road between nodes 1 and 2. If two Braess Paradoxes are combined (as shown on the right), different road closures result in optimal social welfare, depending on the dominating demand.

TABLE 9.1: Travel times at equilibrium for the Double Braess Paradox (optimal values underlined for each demand pattern).

Demand	$(0,2),(1,2)$	$(0,2),\cancel{(1,2)}$	$(0,2),\cancel{(1,2)}$	$\cancel{(0,2)},\cancel{(1,2)}$
$(0,3)$	17	<u>15</u>	17	18
(A,B)	25	26	25	<u>24</u>

$l(f) = f$, do not fit this model since their free-flow time $l(0)$ is zero. Hence, we use a slightly modified version of the paradox that retains its essential characteristics but ensures $l_e(0) > 0 \; \forall e \in E$.

The original network has a well-known static solution for the problem of finding the best restriction: Closing the edge $(1,2)$ is socially optimal. Thus, a one-off analysis might suggest a permanent road closure, deeming dynamic restrictions unnecessary. However, when this network is superposed with a second, similar structure, the restriction's effect becomes demand-dependent:

Example 12. *Consider the traffic network shown in Figure 9.2b. When all (or most) agents travel from node 0 to node 3, the optimal restriction is to close only road $(1,2)$; however, when demand between A and B dominates, closing both $(0,2)$ and $(1,2)$ is optimal*[3]*. It is, therefore, not possible to find the optimal restriction without taking into account the real-time behavior of the agents and adapting the governance policy when the behavior changes.*

Example 12 illustrates the importance of dynamic restrictions, which necessarily rely on real-time MAS observations. This perspective contrasts with much of the existing Braess Paradox research and related problems (see Section 4.3), where a static flow is the basis for determining the optimal edge subset.

[3]See Appendix B.2 in the supplementary material for more details about this setup.

9.2.3 Limitations

It is evident that action space restrictions can never improve the social optimum, as the maximum is taken from a strictly smaller joint strategy space. As for stable strategies, our results of Chapter 7 show that the MESU can only rise if the relevant NE is eliminated, i.e., if at least one action present in the lowest-welfare equilibrium is restricted.

Applying restrictions to individual agents can achieve the SO by dictating specific actions for all agents, thereby removing their decision-making autonomy. However, transforming a MAS into a single-agent optimization problem by centralizing all decisions is typically not feasible, for reasons ranging from ethical and legal concerns to issues of resilience and scalability.

Viewing restrictions through a fairness lens (as discussed in Section 9.4.2), it becomes apparent that restrictions often need to be uniform. This ensures all agents are treated equitably and have the same actions available at each time step. However, the uniformity of restrictions might impact the achievable governance utility, as observed in Example 10.

9.3 Evaluation

In this section, we use a microscopic multi-step traffic model (see Appendix B.1.1) to simulate agent behavior across a number of network structures, both with and without governance. We analyze the impact of two distinct governance mechanisms on agent behavior and overall system outcomes by varying parameters such as traffic rate, latency functions, demand, and value of money (see Section 9.3.3 for the definition of the latter concept).

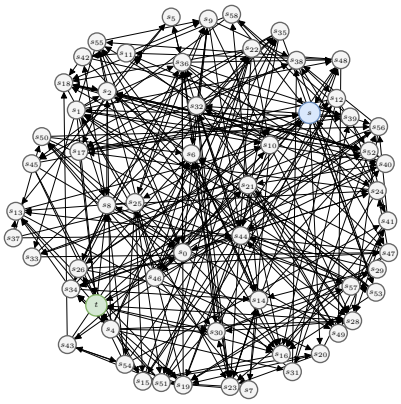
Remark 7. *As our focus is on the observed interaction between agents, their environment, and the governance, there is only one equilibrium: The joint strategy which is achieved experimentally after a sufficient number of steps. This simplifies the MESU concept, which is based on a larger set of NEs. Our experiments indicate sufficient convergence for us to consider joint strategies as stable after a few thousand steps.*

9.3.1 Traffic model

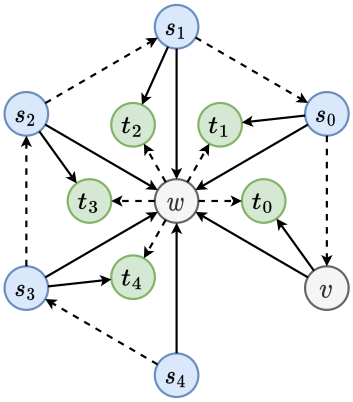
Let $G = (V, E, \boldsymbol{l})$ be a directed graph with a BPR latency function $l_e : \mathbb{N}_0 \to \mathbb{R}^+$ as in Equation (9.1) for each edge $e \in E$. These functions map a flow value (i.e., the number of agents currently using the edge) to a latency value, which indicates the number of time steps required to traverse the edge (see Figure 9.2). Each agent i has a starting node s_i, a current position $p_i \in [0, 1]$ along an edge $e_i \in E$ and a designated destination node d_i[4]. At any time step, the flow f_e of an edge e is defined as $f_e = |\{i \in I : e_i = e\}|$, and the corresponding latency on e is $l_e(f_e)$. The graph, together with the tuple (p_i, e_i, d_i) for all agents, represents the system's current state.

An agent i can only decide its next move (i.e., select its next edge) upon reaching a node, specifically when $p_i = 1$. Therefore, the agent needs to observe only its current node v_i, destination node d_i, and the current latency values of all edges. As described in the definition of the ARMAS model (Definition 10), the set $R \in 2^E$ of currently permissible actions is also provided as an input.

[4]In this context, each source/sink pair that is used by an agent as starting and destination nodes is called a *commodity*.



(A) $G_{n,p}$ graph with Braess' Paradox. For this graph, we have chosen $n = 59$ and $p = 0.07$, in accordance with Valiant and Roughgarden, 2010's assumption that $p = \Omega(n^{-1/2+\varepsilon})$ for some $\varepsilon > 0$.



(B) Generalized Braess graph B_4. Solid lines denote constant-latency edges, while dashed lines are edges with affine latency functions (details and the construction schema for B_n are provided in Appendix B.4).

FIGURE 9.3: Network structures used for the experiments. Start nodes are marked blue, while target nodes are green. For each graph, we measure travel times, improvement, and fairness for unrestricted traffic, edge restrictions, and Δ- tolling.

9.3.2 Δ-tolling

Δ-tolling, introduced by Sharon et al., 2017b, is a dynamic reward-shaping strategy for congested networks. It updates per-edge tolls based on the difference between free-flow time and actual latency. This method has been proven to be equivalent to marginal-cost tolling for BPR latency functions, ensuring optimal system performance among the set of tolling schemes. Its adaptability, scalability, and straightforward implementation make it a suitable benchmark, representing the reward-shaping paradigm of multi-agent governance.

The toll update in Δ-tolling is expressed as

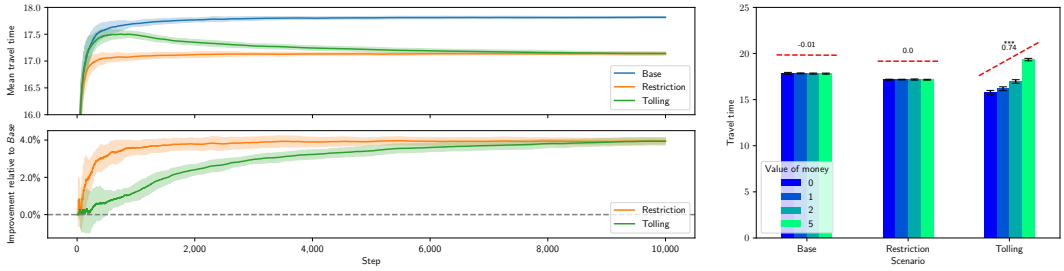
$$\tau_e^{(t)} := R\left[d_e \cdot (l_e(f_e) - l_e(0))\right] + (1 - R)\tau_e^{(t-1)}\,, \tag{9.2}$$

where d is the exponent of the latency function (see Equation (9.1)), and R is a responsiveness parameter. Intuitively, Δ-tolling assigns a toll to each edge in proportion to its congestion, while using exponential smoothing to prevent abrupt updates.

9.3.3 Setup

We evaluate the travel time of agents in unrestricted traffic ("Base"), edge restrictions ("Restriction"), and edge tolls ("Tolling") across two different network types, as depicted in Figure 9.3:

Random Erdős-Rényi ($G_{n,p}$) graphs Braess' Paradox was shown to occur with high probability on random graphs as the number of nodes tends to infinity (Valiant and Roughgarden, 2010), but this result comes with some caveats; most notably, the edge likelihood and the traffic rate need to be chosen carefully in order to generate the paradox. We have thus selected a graph (Figure 9.3a)



(A) Travel time (i.e., social welfare) and improvement (B) Fairness

FIGURE 9.4: Results of the $G_{n,p}$ graph experiment. First, we measure travel times and improvement for the *Base*, *Restriction*, and *Tolling* scenarios as described in Section 9.3.3. In addition, we investigate the dependency of the travel time on the value that an agent assigns to money compared to time.

with $n = 59$ via random search where restricting a single edge improves social welfare[5]. More details and graphs are shown in Appendix B.3, along with the results of the analysis.

Generalized Braess graphs The graph family $B_n, n \in \mathbb{N}^+$ (see Figure 9.3b) generalizes the original network and the "Double Braess" structure from Figure 9.2b. It allows the routing of n different commodities (i.e., source/sink pairs) on the graph B_n, each commodity with a different optimal restriction.

For the Braess-based graphs, the optimal edges to close for improved latency are already known; as previously mentioned in Section 9.1, our aim is not to provide new results on the Braess Paradox's occurrence or detection.

Agents in the MAS employ a simple shortest-path algorithm to select the optimal edge upon reaching an intersection. This approach is well-defined for the unrestricted and restricted scenarios. However, for the Δ-tolling scenario, a relative weighting between travel time and tolls is necessary: Each agent $i \in I$ has a *value of money* $v_i \in \mathbb{R}_0^+$, which defines the time-equivalent worth of one unit of tolls. In other words, the agent minimizes $\sum_{e \in P} \left(l_e(f_e) + v_i t_e \right)$ over all paths P from its current position to its target node and then selects one of the optimal paths.

Remark 8. *As highlighted by Pas and Principio, 1997 and Roughgarden, 2006, traffic rate plays a pivotal role in the occurrence and severity of Braess' Paradox. For this parameter, our choices are:*

$G_{n,p}$ *Using random search, we found a traffic rate of $f = 56$ to be adequate for the selected graph.*

Braess *A rather generic traffic rate of $f = \frac{5}{2}c$ for edge capacity c is sufficient*[6].

9.3.4 Performance metrics

The *mean travel time* of all agents (representing social welfare) is presented as the main performance indicator for the respective governance methods. As a measure

[5] However, efficacy compared to Δ-tolling was not considered in the selection process, nor was the fairness metric defined in Section 9.3.4.

[6] Intuitively, this can be explained by the fact that the routes on these graphs consist of either 3 or 4 edges, two of which are dominating in latency. Therefore, cars are randomly distributed along a route whose length is $\approx \frac{5}{2}$ times the latency of a "long" edge.

for equity, we examine the *correlation between travel time and value of money* for all agents. Specifically, we assess the slope of the closest linear regression between these two variables[7]. This evaluation is vital as it probes the governance mechanism's *fairness*, i.e., the treatment equality towards agents from different groups.

9.3.5 Results

The experiments are fully reproducible, with the seeds used for the experiments' randomized components listed in Appendix B.5.

Random Erdős-Rényi ($G_{n,p}$) graphs

Figure 9.4 shows the performance metrics for ten independent runs on the graph from Figure 9.3a (the mean is drawn as a solid line, while the shaded area denotes the standard deviation of the results). With respect to travel times, both *Restriction* and *Tolling* outperform *Base* by approximately 3%. The fairness metric, however, shows a substantial difference between the two governance paradigms: While agents with different values of money are treated largely equally in the *Base* and *Restriction* scenarios, their travel times differ significantly ($p \leq 0.01$) for *Tolling*, such that agents with higher value of money v_i have longer travel times.

Generalized Braess Graphs B_n

The performance metrics for the graph family B_n with the single commodity (s_n, t_n) are shown in Figure 9.5 for $n \in \{0, 1, ..., 20\}$ and five independent runs. Similarly to the results on the $G_{n,p}$ graph, both *Restriction* and *Tolling* improve the *Base* case, and this time *Restriction* (using optimal road closures) outperforms *Tolling* by a few percentage points. Regarding fairness, Figure 9.5b displays the regression lines' slopes (see Figure 9.4b for a single graph) in an aggregated way for all graphs B_n. This fairness metric shows that *Base* and *Restriction* are nearly unbiased across all commodities. For *Tolling*, the correlation between the value of money and travel time in this particular setup shifts from disadvantaging high values of money for small values of n to disadvantaging low values of money for larger n.

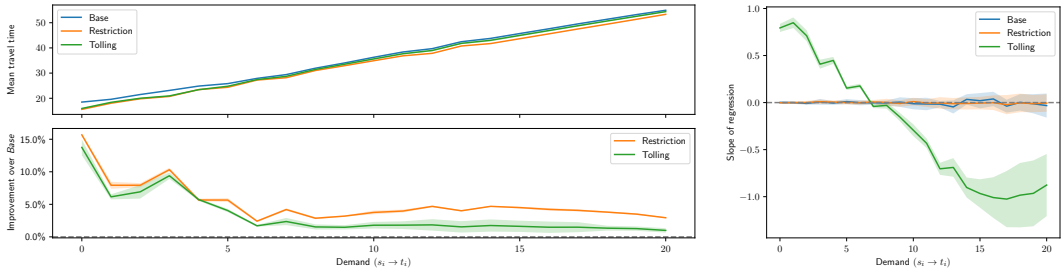
9.4 Discussion

9.4.1 Efficacy

The declared objective of the governance is maximizing social welfare, i.e., minimizing the travel time of all agents. To this end, both approaches succeed in improving the status quo (the unrestricted *Base* scenario), and the improvements are comparable in magnitude.

The results in Figures 9.4 and 9.5 show the mean travel time but not the mean *total cost*, i.e., the weighted combination of travel time and tolls which reveal the additional reward that the toll-based governance has to invest. As can be seen in Figure 9.6 for the generalized Braess graphs, including the tolls results in a much lower efficacy of the reward-shaping scheme. This can be an additional argument for restrictions, where no monetary rewards are involved.

[7] A correlation coefficient like Pearson or Spearman is not suitable since it only measures the strength of the connection, but not its direction; in particular, if all agents have the same travel time (i.e., the mechanism is perfectly fair), these coefficients are not zero, but one.



(A) Social welfare (i.e., mean travel time) (B) Fairness

FIGURE 9.5: Results of the generalized Braess graph experiment, showing the performance metrics in relation to the demand pattern. Both *Restriction* and *Tolling* improve the *Base* case, but the fairness measure is very different: In *Tolling*, the value of money has a major influence on the travel time.

Remark 9. *Marginal-cost tolling, by definition, targets edges where heightened demand results in latency spikes. However, in situations such as Braess' Paradox, it is the "constant-low-latency edges" that need to be closed to attain optimal flow. Such edges, by definition of the tolling scheme, can never be tolled since $l_e(f_e) = l_e(0)$ for any flow $f_e > 0$. As a result, the tolling strategy must reduce demand for these roads by imposing higher tolls on connecting roads. This culminates in a "proxy-tolling" outcome where certain high-demand edges remain toll-free, while others bear the brunt with exorbitant tolls.*

9.4.2 Fairness

Reward-shaping strategies inherently differentiate between agents based on how additional rewards influence them. In essence, an agent with a minimal value of money might remain largely unaffected by rewards or penalties, while others might drastically alter their behavior. Given that rewards and penalties often manifest as monetary values, this can inadvertently compromise fairness, especially in scenarios where an agent's wealth should not dictate their actions. Our experiments reveal that while Δ-tolling effectively reduces average travel time, it simultaneously introduces significant variance between agent groups. Notably, the relationship between the value of money and travel time is not straightforward but varies with the network structure. Restriction-based governance, in contrast, offers comparable travel-time efficiency but ensures a distribution of travel times that is close to equity.

9.4.3 Resource restrictions

To conclude the discussion of restriction-based governance and reward shaping, we outline *resource restriction* as a hybrid governance paradigm, combining elements of restriction-based and reward-based governance:

Restrictions, as they have been defined so far, directly limit the action spaces of the agents, thereby generating new equilibria. Another way restrictions can be used in multi-agent systems is to restrict *resources* in order to change incentives. The restrictions, therefore, serve to *indirectly* shape the rewards by encouraging or discouraging actions that relate to the restricted resources. In Section 10.3.1, we outline a parking management scenario based on existing work for dynamic pricing (Kappenberger, Theil, and Stuckenschmidt, 2022) and show that closing some of the parking

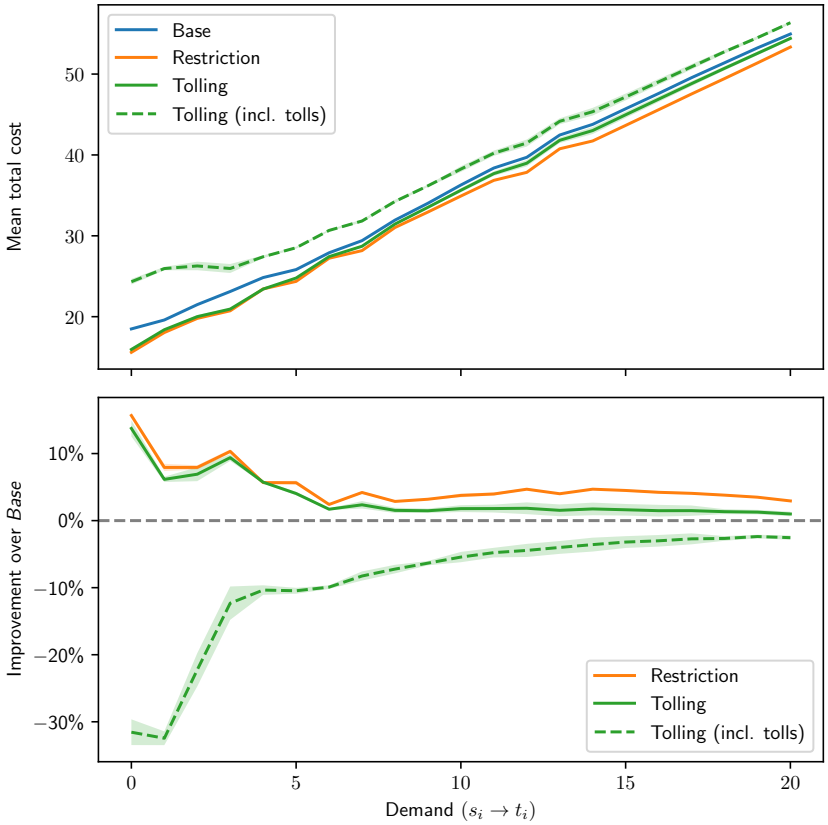


FIGURE 9.6: Total cost of travel (including tolls) for the generalized Braess graphs.

spaces can increase social welfare. Without forbidding any agent actions, the restriction of resources (in this case, parking spaces) affects how the agents valuate their options, which, in turn, steers their behavior in the desired direction.

In contrast to "pure" reward shaping approaches, this method avoids monetary incentives and, therefore, maintains some of the mentioned advantages of action-space shaping. On the other hand, it does not lend itself to the direct calculation of equilibrium strategies without explicit knowledge of the connection between resource restriction and corresponding changes in agent utility.

9.5 Summary

Action-space restrictions seem to be inferior to reward shaping at first glance, as they only allow a binary distinction between allowed and forbidden actions (similar to a reward of 0 and $-\infty$ for choosing an action, respectively). However, as we have shown in the present work, the actual comparison is more complex (see, e.g., Remark 9), and restrictions come with a considerable advantage regarding fairness.

The study of action-space restrictions as a means of governing multi-agent systems is far from exhausted: Only recently have common multi-agent learning environments like PettingZoo (Terry et al., 2021) been equipped with governance capabilities (Oesterle and Grams, 2024), and there is still scarce consideration of restrictions for Reinforcement Learning algorithms (first steps are described in Grams, 2023). It has been shown that finding optimal restrictions for dynamic systems can

be hard, but the dependency of their effect on the (a priori unknown) behavior of the agents makes real-time adaptation and optimization necessary.

Despite these challenges, its unique way of interacting with agents and environment makes action-space shaping a valuable tool for governance entities, both in abstract game-theoretic settings and in real-world systems. We want to emphasize that the acceptance of governance mechanisms, be it reward shaping or restrictions, crucially depends on their (perceived and objective) fairness. With respect to this condition, restriction-based governance, together with equity considerations, has the potential to substantially push the applicability of governance schemes for systems consisting of human or artificial agents.

Chapter 10

Discussion

The preceding chapters have introduced and investigated the general idea of governing multi-agent systems using dynamic action-space restrictions. After defining the ARMAS model and looking at the governance learning problem for various subclasses, an implementation proposal, and an evaluation of ARMAS against reward shaping, we now round our thesis off by discussing our findings in a wider context.

10.1 Solution approaches for governance learning

10.1.1 Classification

We have shown solutions to the governance learning problem for three subclasses of ARMAS. These solutions, albeit not covering the whole spectrum of multi-agent systems, represent the three broad categories of methods through which any optimization problem can be approached: Exact methods, heuristics, and learning methods. Naturally, all three classes come with their respective advantages and drawbacks.

Exact methods The AROGU algorithm of Chapter 7 is an exact method in the sense that it finds an optimal restriction if its assumptions are fully satisfied (see Theorem 2). As such, its disadvantages are two-fold: First, it strictly relies on these assumptions and on exact knowledge of the agents' reward functions to provide a guaranteed optimum, and can only handle restrictions of a specific form. And second, its worst-case run-time is exponential in the size of the action space, since an exhaustive search might be necessary. The main advantage of such a method, of course, is that it finds an optimal solution.

Heuristics The tabular approach of Chapter 5 is a heuristic that draws upon the implicit assumption that past joint actions are a good predictor for future joint actions. Using this assumption, we then derive minimal restrictions by removing undesired joint actions in a greedy manner. The advantage of this algorithm is that it can leverage the observed agent behavior without needing to explicitly model the agents' inner workings, i.e., their preferences and utilities. However, distribution shifts can disturb its prediction mechanism and, therefore, impair performance. The use of a table of all joint actions and their respective probabilities requires an exponential amount of memory with respect to the number of agents and the size of their action spaces.

Learning methods The RL approach used in Chapter 6 is a classical learning approach. The governance is provided with state and reward values and learns an optimal restriction policy by mapping states to restrictions that result in maximum expected rewards. Drawbacks of using the RL paradigm are that (a) the governance's action space needs to conform with what the algorithm (in

this case, PPO) supports, (b) the resulting restriction policy is not easily interpretable, and (c) the restriction policy can change unpredictably and does not provide any stability. On the other hand, the approach can make use of state-of-the-art learning algorithms, can be integrated into existing frameworks, and only requires minimal assumptions since it simply states the governance learning problem as an RL problem.

10.1.2 Comparison with reward shaping

Action-space restrictions can be seen as a special form of reward shaping in which the assigned reward is either 0 (for allowed actions) or $-\infty$ (for forbidden actions). While this perspective makes the approach look less powerful in terms of governance influence, it avoids the problem that agents, in general, react differently to the same reward. In real-world scenarios, this is due to the fact that agents have different valuations of the currency of the governance's reward. In Chapter 9, we have borrowed and adapted the concept of *value of time* from transport economics, which quantifies the opportunity cost of the time that an agent spends traveling. In other words, the value of time is the amount that a traveler would be willing to pay in exchange for saved time or the amount they would accept as compensation for lost time.

Monetary rewards as a means of governance also have other implications that complicate their use in large-scale, real-world applications: Strong incentives are only created by sufficient amounts of money, and real money needs to come from somewhere in order to be spent on incentives. Unless the governance entity of a system can create unlimited amounts of monetary resources[1], this leads to the problem of weighing efficacy against expenditure.

10.2 Challenges for governance learning

Tackling the governance learning problem for a restriction-based governance scheme in Chapters 5 to 7, we have come across several limitations and challenges. Some of these apply to any governance approach, while others are specific to the restriction-based approach we have chosen. Accordingly, using a different governance paradigm might solve the latter kind of challenges, but can also introduce new ones, and be faced with the same general problems that are inherent in the task.

10.2.1 Paradigm-independent challenges

Hardness

The effect of governance action on agent behavior is, in general, not easily predictable, and it does not need to be continuous: By targeting a single action (e.g., by forbidding it or applying a negative reward to it), the resulting equilibrium can remain unchanged, or the intervention can have minimal consequences for the chosen actions, or it can completely change how agents act. This lack of continuity prevents the use of common optimization techniques like gradient descent or bisection.

Finding an optimal governance policy, therefore, essentially requires searching the entire space of interventions. Any intervention potentially depends on the current environmental state, meaning that the governance policy is a mapping from the

[1] In general, the presence of extremely large or infinite amounts of money tend to make money worthless, so this is not a sustainable solution even when it is possible.

state space to the intervention space[2]. A governance policy therefore (naïvely) suffers from a combinatorial explosion with respect to at least three scaling factors: The number of agents, the size of the action spaces, and the number of environmental states.

Non-stationarity

All governance approaches for multi-agent systems (not just restriction-based approaches) have in common that they operate in a heavily non-stationary environment: Not only do the agents act in unpredictable ways and systematically change their action policies over time; they typically adapt their policies *adversarially* to circumvent any governance interventions which keep them from reaching their own objectives.

Moreover, any change in the governance policy causes a distribution change in the environment as seen from an agent's perspective, leading to strategy adaptations and the gradual settling into a new equilibrium where all agents keep their policies approximately constant. This means that an update of the governance policy might first result in chaotic behavior (this can be positive or negative from the governance's viewpoint), and its *real* effect, i.e., the change in the system equilibrium developing under the new policy, can only be seen after a phase of convergence[3].

Sample efficiency

For the reasons described in the previous section, the governance can reliably learn the effect of an action (i.e., an intervention) only in very large intervals. The number of samples from which a governance policy can be learned is, therefore, lower than required by most machine learning approaches, and particularly Reinforcement Learning approaches, to converge to an optimum.

Practicality dictates that a governance for unknown real-world agents cannot be trained for arbitrary amounts of training data but needs to provide a sensible and, therefore, acceptable policy from the very beginning of its operation.

Interpretability

Except in very specific circumstances, governance is always a matter of acceptance by the agents that participate in a governed system. Therefore, a governance policy needs not only to be effective but also *reasonable* in the sense that its interventions can be explained, understood, and accepted. For explicit rules, this can be straightforward (although a table of rewards and sanctions still might need an aggregated description in order to be interpretable), but policies that are encoded by, e.g., a neural network, do not innately have this property. As usual in machine learning, lack of interpretability and verifiability are crucial challenges when it comes to the application of a governance scheme in systems with real-world relevance.

Fairness

The fairness of a governance scheme is an inherently moral question, requiring a measure that is independent of performance. Generally defined as "impartial and

[2]We intentionally use the vague term *intervention space* since this thought applies equally to rewards, restrictions, or any other form of governance intervention.

[3]This does not even include cases of oscillating behavior without any convergence.

just treatment without favoritism or discrimination", fairness in multi-agent systems can be interpreted in a number of different ways. Section 4.2.6 lists a few proposals for fairness in ML, but ultimately, fair treatment of agents is highly domain-specific.

A relatively unquestionable objective could be that the governance should only enforce outcomes that are Pareto efficient: If it is possible to give all agents simultaneously a higher reward[4], then the governance should always do this.

Which outcome on the *Pareto front* should be preferred, however, is mostly a consideration in which there is no right or wrong. In particular, the maxim that the outcomes (i.e., rewards) of all agents should be equal, requires that these outcomes can be compared and do not depend on easily manipulable statistics like self-reporting of satisfaction. In general, Goodhart's law[5] suggests that this is not an easy problem.

Compliance

Whatever means a governance uses to influence the interaction of agents and environment, it always needs to ensure that these interventions are effective, that is, that the agents comply with the governance's actions. Eventually, the only means of enforcing compliance is physical power.

The most common way to deal with this challenge in theoretical models of multi-agent systems is simply to postulate that agents, by assumption, do not violate the governance rules. Of course, this only shifts the problem to the next level: What happens if agents *do* act against these rules? Therefore, any real-world governance scheme needs to account for the possibility of non-compliance and define an escalation process that eventually leads to a physical restriction of agents to allowed behavior.

The requirement of practical control over the MAS can be satisfied in a wide range of relevant applications, for example, by any digital platform where agents are software components, actions are chosen by exchanging messages, and the governance is managed and enforced by the platform operator. In physical systems, however, this is much harder to achieve.

10.2.2 Challenges of restriction-based governance

Trade-off between restriction and autonomy

For a governance that restricts the freedom of action of the agents, a reasonable maxim is that *fewer restrictions are better*, all else being equal (see Section 1.2.5). This target, like the ones described in Section 10.2.1, is of a moral nature and introduces a natural trade-off between the degree of restriction (see Definition 11) and efficacy (e.g., relative improvement of governance utility). It is easy to recognize this trade-off when we look at the extremes, as shown in Figure 10.1: If the degree of restriction is always zero, the MAS is ungoverned, and the relative improvement is also zero. The highest improvement, on the other hand, is achieved with the maximum degree of restriction when the governance optimum is prescribed by allowing exactly one action for each agent[6]. Although the relation between the degree of restriction and relative improvement of governance utility is not necessarily monotonic, there is a sweet spot for any weighting between these two measures. Again, the "right"

[4] This refers, of course, to the expected cumulative reward which is the base of the agents' optimization.

[5] "When a measure becomes a target, it ceases to be a good measure."

[6] Assuming that the governance possesses the capabilities necessary to identify the actions that lead to the SO.

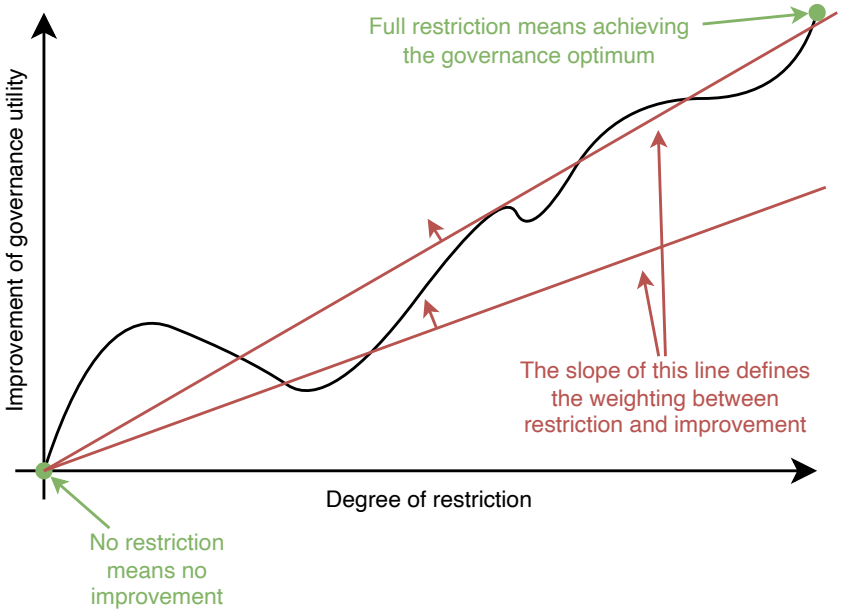


FIGURE 10.1: Trade-off between degree of restriction and relative improvement of governance utility. No restriction always means no improvement, while full restriction results in maximum improvement by enforcing the governance optimum. Consequently, a weighting between these two targets is required to determine the optimal governance action.

weighting is a matter of performance and ethical considerations and cannot be universally determined: Overly restrictive governance control may stifle the innovative and adaptive capabilities of autonomous agents, while too little control might lead to undesirable or harmful outcomes.

Restrictions for contribution games

A common classification of multi-agent systems distinguishes between systems where cooperation is achieved by all agents choosing the same actions or strategies and games where cooperation requires the choice of different or complementary actions or strategies. The former category is called *coordination games*, while the latter type of systems are *contribution games*. The difference between these two categories is best illustrated in the traffic domain:

- Route selection is a contribution game. Effective routing in a traffic network relies on an even distribution of agents over the available routes (i.e., different agents should take different actions); if all agents take the same route, it will become congested, while the capacity of other routes is unused. Of course, this requires that agents either know about the other agents' choices, or that an equilibrium develops over time through (asynchronous) strategy adjustments.

- Agent behavior on a specific congested road is a coordination game. If every agent keeps a steady speed without tailgating or constantly switching lanes (i.e. if everyone takes the same actions), traffic jams often clear much faster. However, it is a dominant strategy to "exploit the weakness" of the other cars who follow this cooperative policy.

Non-discriminatory restrictions can, by their very definition, only solve coordination games, since it is not possible to prescribe different actions to different agents; any action that is used by at least one agent in a socially optimal solution needs to be allowed for *all* agents. To address contribution games, it is thus necessary to use individual restrictions, thereby introducing new challenges regarding scalability and fairness (see Section 10.2.1).

It is, however, conceivable to devise more complex rules to enforce a solution of a contribution game in a fair way. Two exemplary approaches are *conditional* and *randomized* restrictions, illustrated as follows for the route selection scenario:

Conditional restrictions Instead of telling each agent which route to take, we can define a condition for taking one or the other route. For example, you can only take the freeway if you stay on it for more than 20 miles, otherwise you have to take the highway.

Randomization The governance defines different roles and randomly assigns agents to roles. By the law of large numbers, the proportion of agents using a certain strategy will be close to the probability of the corresponding role. For example, the entrance to a freeway is only open 30% of the time (and the decision is made individually for each car); if it is currently closed, you need to take an alternative road.

Both approaches share the property that fairness is only implemented *in expectation*: For every individual restriction decision, some agents might have a greater degree of freedom than others, and the associated rewards might differ. However, the expected restrictions do not differ between agents, since the conditions and randomization are agent-agnostic.

Representation of restrictions and policies

The scalability issue described in Section 10.2.1 applies to both individual restrictions and the restriction policy as a whole: The number of possible restrictions for discrete action spaces $\boldsymbol{A} = (A_i)_{i \in I}$ is $2^{\prod_{i \in I} |A_i|}$[7]. An explicit mapping from the state space to this restriction space is usually not efficient, while an approximation comes with its own limitations. For continuous action spaces, the situation is even worse since there are infinitely many possible restrictions[8].

10.2.3 Challenges of reward-based governance

Reward as part of the environment

The standard Reinforcement Learning loop posits that the environment provides both state and reward and the agent chooses an action based on these two values. The implicit assumption is that the agent will just act in a way that maximizes the expected (discounted) future reward as given by $R = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$. In this term, only the discount rate γ is chosen by the agent, while r_t is the reward as defined by the environment. The agent can decide to be more or less myopic, but the general valuation of states and actions comes from an outside entity.

[7] For non-discriminatory restrictions over the action space A, this reduces to $2^{|A|}$.

[8] By Cantor's theorem, the number of restrictions over any infinite action space is uncountably infinite, such that looping over these restrictions in a suitable order with a stopping condition is not an option.

In contrast, humans tend to optimize a reward that they *themselves* assign to the environmental state after observing the state. From an outside perspective, this assignment is not known and does not even have to be consistent (i.e., a well-defined function), since the same environmental state can seem more or less favorable to an agent, depending on its internal state (which, again, is not known to the environment or other agents, if applicable)[9].

Not having access to the agents' reward values can be a severe challenge for a reward-shaping governance: As described in Chapter 5, the direct observation of actions needs to be translated back into agent preferences and utilities, such that the governance can act on this estimated utility when optimizing social welfare. Obviously, this translation (or reverse engineering) step introduces the possibility of error and manipulation on the part of an agent.

Comparability of rewards

Rewards as a means of changing agent behavior implicitly assume that additional rewards and sanctions indeed affect agents in a way that can be predicted and exploited. This means that agents actually act rationally with respect to rewards (i.e., they try to maximize their future expected reward) and, therefore, avoid actions that lead to unnecessarily low rewards. For a governance that has no access to the agents' actual decision-making process, the process of finding an optimal reward-shaping policy consists of tentatively rewarding and sanctioning certain behaviors and observing the effects. From these observations, the rewards can then be adjusted until the desired behavior is achieved.

When agents have highly different valuations of additional rewards—for instance, if one agent gets rewards between 0 and 10, while another agent gets rewards between 100 and 1000—it can be very hard to find a governance policy that strikes the right balance for all agents between over-governing and not having any influence at all.

10.3 New ideas for multi-agent governance

10.3.1 Hybrid governance approaches

In Section 2.4, we have treated governance interventions at the reward level and the action level as two separate approaches. In practice, though, these intervention points are not always clear-cut.

In many normative approaches, researchers intend to provide restrictions but concede that, given the lack of enforceability in real-world systems, sanctions are required as the "back-up measure" in case the restrictions are violated by the agents. Negative consequences of any kind (which, ultimately, can always be expressed in terms of negative rewards) serve both as threats to deter future perpetrators and to punish those who have already acted in undesired ways.

Let us show an exemplary case of a hybrid approach that uses restrictions as a means of changing the reward structure of a system, again in the domain of traffic:

In Kappenberger, Theil, and Stuckenschmidt, 2022's dynamic pricing scenario (see Figure 10.2), parking spaces are available at a number of different places within a city district. A number of cars (agents) with individual targets on the map navigate

[9]For human agents, behavioral psychology has identified a myriad of biases and other features which prevent actual preferences from being represented as a fixed utility function.

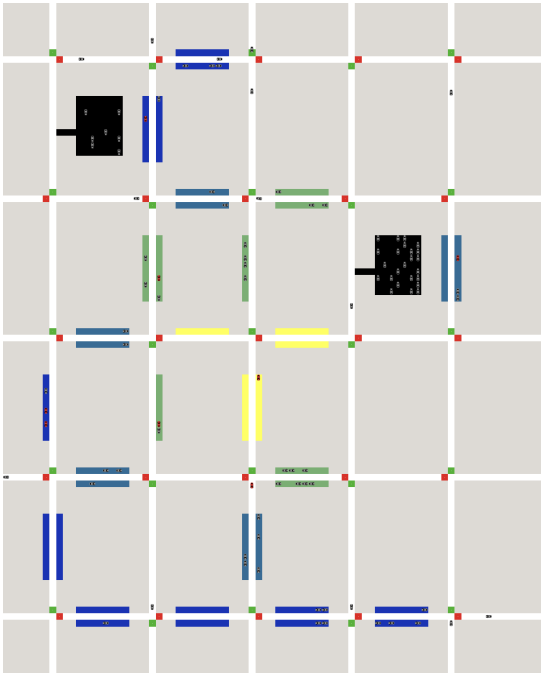


FIGURE 10.2: Grid network for dynamic pricing of parking spaces as proposed by Kappenberger, Theil, and Stuckenschmidt, 2022.

the road network in order to find a parking space. For each agent, the utility of finding a parking space depends on various factors such as the travel time, the distance from the parking space to the target, or the agent's preference to park roadside or in a garage[10].

We formalize this scenario in the following simplified way: Using a directed graph with latency functions as in the experiments of Chapter 9, we now assign two-dimensional positions to all nodes. In this geography, parking lots are located at certain locations on the map and connected to nodes, such that an agent can only use a parking space when it has reached the respective node. Cars have uniformly random entry and exit points (nodes) and normally distributed targets ((x, y)-coordinates) within the limits of the grid. The utility of an agent is a weighted sum (where the weights correspond to the preference conversion factor between driving and walking) of travel time between start node and parking space and distance between parking space and target.

To illustrate resource restrictions, let us use the simple grid shown in Figure 10.3, and assume that there are two parking lots which are accessible from nodes 1 and 4, respectively. All agents enter the graph at node 0 and leave at node 5. Crucially, the center of the agents' target distribution is $(0.5, 0.7)$, meaning that lot 1 is slightly more attractive than lot 4.

As the parking management authority, we can only set the number of available parking spaces at lot 1 to a number k between 0 and 10; at lot 4 there are always 10 spaces. Similarly to Braess' Paradox, closing some of the parking spaces can increase social welfare: Simulating the scenario for all values $k \in \{0, 1, ..., 10\}$ gives the social welfare graph in Figure 10.4. We can see that social welfare takes its maximum at $k = 6$, while both extreme values of k are detrimental.

This phenomenon can be explained as follows: When there are 10 parking spaces in lot 1, most cars decide to go there, resulting in over-utilization and congestion of

[10] As in Chapter 9, agents can have individual weightings for the relative importance of these factors.

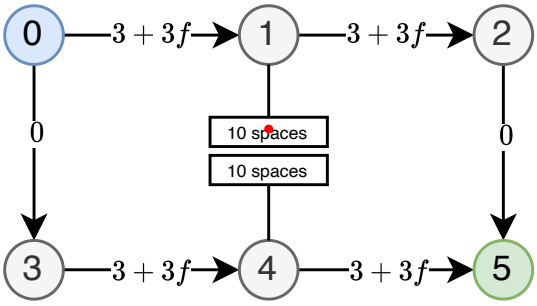


FIGURE 10.3: Parking environment with two parking lots which can be reached from nodes 1 and 4, respectively. The red dot denotes the center of the normal distribution of targets, meaning that lot 1 is more attractive on average.

edge $(0, 1)$. When there are fewer parking spaces available, the lot becomes gradually less attractive, until the incentives are sufficiently balanced to allow for even traffic on both edges $(0, 1)$ and $(3, 4)$.

While we do have restrictions in this scenario (i.e., the use of certain parking spaces can be allowed or forbidden), they are not aimed at removing specific actions from the agents' action spaces. Instead, the agents can still choose the same actions (i.e., use the same edges and nodes), but the utility of doing so—in RL terms, their action-value functions—are affected by the restrictions.

10.3.2 Emergent governance as a meta-strategy

Open, self-describing systems like human language allow agents to reason about and define governance schemes as part of their interaction with the environment. Simple examples that are commonly expressed in natural language but cannot easily be represented in the prevailing mathematical models are contracts, elections, or enforcement of rules by a monopoly on violence. While agents in a POSG usually have very well-structured, unalterable action spaces, real-world agents can use communication to build meta-actions that open up new strategies and thereby change the entire system to allow for new, better equilibria. The emerging field of *Cooperative AI* (Dafoe et al., 2020) aims to translate this ability, which humans have used for a long time to shape their environment and interactions, to artificial agents. At best, this will allow us to transcend the rigid rules usually assumed in multi-agent systems, and create governance forms that are not constrained by the intervention points described in Section 2.4 for the POSG framework.

10.3.3 Language-based governance

As an example of the meta-strategies outlined in the previous section, assume that agents act upon an environment by exchanging natural-language messages. In such a "debate arena" (see also Section 8.3.4), agent goals can be achieved by giving the best arguments and thereby winning debates, while the governance takes the role of mediating the discussion and facilitating a constructive process. Of course, it is utterly infeasible to explicitly describe the transition function or the governance utility function of such a system in terms of token-based actions; for example, the

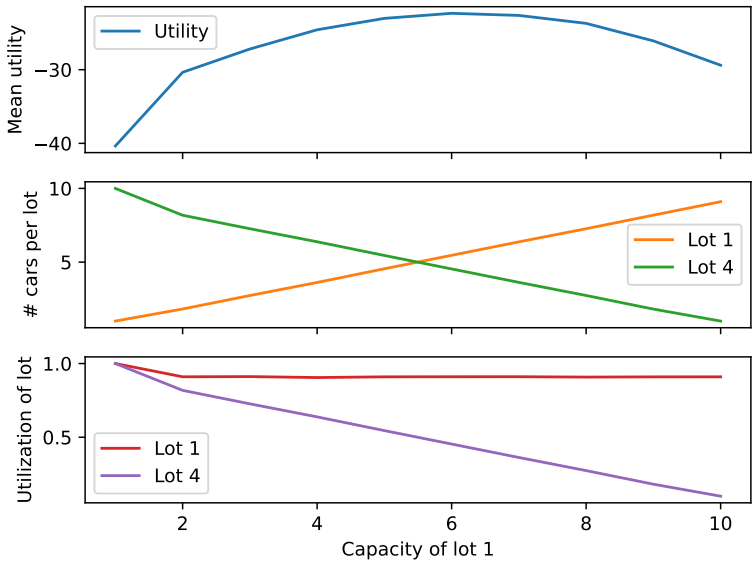


FIGURE 10.4: Social welfare of restricted parking for different numbers of parking spaces at lot 1. When all 10 spaces are open, route $0 \to 1 \to 2 \to 5$ is congestion-prone since lot 1 is the better option for most agents. Demand can be balanced by closing some of the spaces at lot 1, leading to optimal social welfare when 4 spaces are closed.

action space of an agent is the set of all token tuples that have a meaning in the language under consideration[11].

This setting is much less clearly defined than any commonly used POSG model, but it is a lot closer to how intelligent agents would interact: The openness of the action spaces allows for an unbounded wealth of creative strategies, and ever more complex patterns of cooperation can be learned over time. As humans have, within the gigantic multi-agent system that is society, developed their own "self-contained multi-agent systems" with rules, norms, rewards, and enforcement mechanisms, debating agents can be imagined to build similar structures if given a sufficiently expressive language and enough time.

[11] Of course, we can simply set the action space as $A := T^*$, but the effect of an action $a \in A$ depends on its *meaning*, which itself is a complex (and poorly understood) function of the sequence of tokens.

Chapter 11

Conclusion and outlook

In this thesis, we have introduced the formal model of an action-restricted multi-agent system and examined its feasibility as a dynamic governance mechanism. For the two settings of discrete action spaces over a Stochastic Game and continuous action spaces over a Normal-Form Game, we have proposed and analyzed solution approaches to find an optimal governance policy. Additionally, we have proposed an implementation of ARMAS using the agent-environment cycle of PettingZoo, and we have evaluated the restriction-based governance approach with respect to efficacy and fairness.

Drawing upon the analogy of laws as restrictions of the action spaces of humans in any kind of community, we see great potential in this approach for designing socially optimal interactions among artificial agents in a future where such agents are highly evolved and ubiquitous.

Naturally, there are many open questions regarding scalability, convergence, safety guarantees, manipulability, and other concerns that need to be addressed to facilitate adoption in real-world settings. In this chapter, we outline a few limitations of our current approach and propose directions for further research. The last section of this work will take another step back and look at the potential long-term impact of multi-agent governance on society—ultimately, the goal of our work is to develop a theoretical understanding of how autonomous decision-makers of all sorts can be supported by self-learning governance structures to collectively make the world a better place.

11.1 Limitations

11.1.1 Scalability

The space of possible governance schemes for multi-agent systems is fundamentally unbounded and has no inherent structure that can easily be exploited. By limiting ourselves to restriction-based governance and defining the governance restriction policy as a function from the environmental state to subsets of action spaces, we have already heavily narrowed down the search space. Still, finding an optimal restriction policy for a general multi-agent system of any meaningful size is an open challenge, and keeping the policy optimal over time in the face of adaptive agent strategies is even harder.

11.1.2 Definition of real-world MAS

Many systems of practical relevance are not well-defined and, therefore, do not allow for a good translation into an accurate mathematical model. For example, the actions that humans perform in their environment could be precisely expressed as

movements in space; after all, even spoken words are nothing but sound waves caused by moving certain parts of the body. However, this perspective is not at all helpful for describing what happens when humans talk to each other. High-level actions like "go to the supermarket" or "convince your spouse to buy a house together", on the other hand, are far too broad to allow for new creative combinations that add new expressivity to the system. Again, language might be the key to an interaction protocol at the right level of detail, accuracy, and conciseness.

11.1.3 Ethical implications of restrictions

Depending on the nature of the agents in a multi-agent system, restricting action spaces can have far-reaching physical consequences or fundamentally touch the freedom and privacy of agents. This is especially serious when human agents are involved, but also artificial agents ultimately act on behalf of human stakeholders[1]. Therefore, ethical considerations, which usually do not play any discernible role in a technical treatment of multi-agent systems and their governance, must be taken into account when defining the objective and impact of a governance scheme. Naturally, our assumption is that the governance objective embodies the "greater good" for which it is acceptable to restrict the freedom of the agents. This perspective, however, implicitly presumes that the governance does not abuse its power and that the greater good is agreed upon by all involved agents.

To take the step from a technically sound and well-performing governance approach to a safely deployable component for real-world MAS, the following considerations are essential: (a) Governance systems might have to make complex ethical decisions, particularly in scenarios where agents' actions have significant moral or societal implications. The principles guiding these decisions need to be carefully considered and ethically sound. To allow for an independent assessment of these principles, it is essential to ensure that the workings of the governance system are transparent, such that agents and other stakeholders understand why certain actions are restricted. (b) The criteria used to restrict actions must be fair and unbiased. There is a risk that the governance system might inadvertently discriminate against certain agents or behaviors, for example, if the system is built or trained on biased data. (c) Determining who is responsible for the actions of an autonomous agent becomes more complex under a governed system. If something goes wrong, it can be challenging to ascertain whether the fault lies with the agent, the governance rules, or the designers of the system. (d) While the primary intent of governance might be to ensure safety and security, overly restrictive or poorly designed controls could lead to vulnerabilities, especially if the governance is not able to adapt to unforeseen situations in due time.

11.2 Future work

The above-mentioned limitations immediately give rise to exciting future work. Some of these research directions seem fairly straightforward (as a challenge, though not as a solution!), while others require a whole set of new ideas and frameworks in order to be operationalized into technical work.

[1] Should, at some point in the future, artificial agents be able and allowed to act as separate entities, this development will most likely *exacerbate* the issue instead of solving it: Fully autonomous artificial agents should enjoy the same rights as humans and therefore be protected from unethical governance influences the same way humans are today.

11.2.1 Rigorous classification of governance schemes

We have seen in Section 2.4 that the POSG model for multi-agent systems naturally exhibits a few intervention points where a governance can hook into the system. At the same time, Section 10.3.1 has shown that these points can be combined to make governance more complex and potentially more powerful. While this investigation was merely based on examples, an exhaustive treatment of potential governance schemes (first for the POSG model, but eventually for more general interaction models) might uncover a wealth of previously unimagined approaches.

Alternatively, there might be a general representation of *any* governance scheme in the same form, relieving us from manually devising a classification of possible governance methods. To use an analogy: Just as neural networks, by virtue of their structure and number of parameters, allow for a much wider range of functions to be represented than, say, linear or quadratic regression, we can imagine a parameterized governance scheme that is much more general than just defining action-space restrictions or governance rewards. By tuning the parameters of such a "universal" governance, we can then find a truly optimal policy, not just within the realms of directly addressing action spaces and rewards.

11.2.2 Generalization to Partially Observable Stochastic Games

In the present work, we have proposed a general model which can represent a broad spectrum of systems, at the expense of a unified solution algorithm. Instead, we have presented governance learning algorithms for smaller problem classes by making additional assumptions about the systems under consideration.

Connecting these dots to create an efficient (e.g., polynomially computable or polynomial-time converging) algorithm to find an optimal restriction policy for general POSGs (and therefore, ARMASs) would be a breakthrough for the significance of restriction-based governance schemes.

The development of Reinforcement Learning, where scalability experienced a quantum leap several decades after the birth of the field through the application of deep learning, gives hope that known challenges, previously unsolvable due to their computational complexity, will eventually be solved as technology advances.

11.2.3 Language-based restrictions

One of the major limitations of our current approach is that most multi-agent systems, especially when humans are involved, are not well-defined in terms of the interaction frameworks of Chapter 2. This limitation, briefly described in Section 11.1, means that in these systems, *there just is no definitive set of actions that could meaningfully be restricted*. In contrast, human language is precisely what we use to describe the world around us, reason about what we want to do, and even carry out a lot of our actions—those with which we communicate with other people.

Solving the governance learning problem with state-of-the-art language processing systems would immediately result in restrictions expressed in natural language, therefore allowing for infinite expressivity combined with high accuracy and inherent human interpretability.

11.3 Societal relevance

A governance system that can automatically generate optimal rule sets and laws could have a significant impact on the functioning of large societies, such as countries or even the global community. Such a system has the potential to greatly improve the efficiency and fairness of legal and regulatory frameworks, leading to better outcomes for individuals and society as a whole.

One major benefit of an automated governance system is that it can rapidly adapt to changing circumstances and emerging issues. By analyzing data in real time, the system can identify areas of concern and generate new rules and regulations to address them. This could be particularly important in fields such as public health and environmental protection, where timely action is crucial and accurate forecasts are difficult.

Another potential benefit is the reduction of bias and discrimination in decision-making. By basing decisions on objective data and algorithms, an automated governance system could reduce the impact of personal biases and prejudices that exist in virtually all human decision-making. This could lead to fairer outcomes for all individuals, regardless of their background or circumstances.

Furthermore, an automated governance system could increase transparency and accountability in decision-making processes. All decisions and rule changes would be made traceable and auditable, allowing for greater scrutiny and oversight by the public and other stakeholders.

However, it is important to note that an automated governance system is not a panacea and may raise concerns about privacy, security, and the potential for unintended consequences. Therefore, careful consideration and regulation would be needed to ensure that such a system is developed and used responsibly and ethically.

Ultimately, even the physical universe can be seen as a multi-agent system that is governed by the laws of physics, prescribing any decision-making entity which actions it can and cannot perform in any given situation. After all, it is hardly conceivable that the world could produce intricate structures and purposefully crafted sub-systems, if not for these laws.

Multi-agent systems consisting of human agents most often apply a multi-faceted approach to governance: There are static action constraints, conditional rules, monetary and other incentives, but also moral maxims, appeals to people's sense of fairness and solidarity, and many other forms of normative influence on behavior. We can see that for at least two thousand years, philosophers and politicians have struggled to find an optimal method to govern communities, and the systems that have emerged have been shown to be brittle, prone to manipulation, and in perpetual need of adjustment. At the same time, there is a broad consensus that certain elements like individual freedom, voting rights, and corrigibility should be included in any feasible form of governance, both for ethical and performance reasons[2].

It would be exciting to see whether a self-learning governance equipped with superior data collection, reasoning, and forecasting capabilities would come to the same conclusions and devise similar governance schemes.

[2]One could argue, though, that the continued worldwide occurrence of non-democratic forms of government means that this consensus is less pronounced than commonly assumed from our own viewpoint.

Appendix A

Supplementary material for Chapter 7

A.1 Equilibrium oracle for quadratic utilities

For a quadratic reward function $r : \mathbb{R}^2 \to \mathbb{R}$ and a restriction $R \subset \mathbb{R}$, the best response function $\mathcal{B}_1|_R(x_2)$ can be found by a straight-forward case analysis: Let r be defined as

$$r(x_1, x_2) = ax_1^2 + bx_2^2 + cx_1x_2 + dx_1 + ex_2 + f \ ,$$

and define five "candidate points"

$$\begin{aligned} x_l &:= \min_{x \in R} x \ , \\ x_u &:= \max_{x \in R} x \ , \\ x^* &:= \frac{cx_2 + d}{-2a} \ , \\ x_- &:= \max_{x \in R, x < x^*} x \ , \text{ and} \\ x_+ &:= \min_{x \in R, x > x^*} x \ . \end{aligned}$$

With these points,

- if r is constant in x_1 (i.e., $a = 0$ and $cx_2 + d = 0$), then $\mathcal{B}_1(x_2) = R$,
- if r is linear in x_1 with positive slope (i.e., $a = 0$ and $cx_2 + d > 0$), then $\mathcal{B}_1(x_2) = \{x_u\}$,
- if r is linear in x_1 with negative slope (i.e., $a = 0$ and $cx_2 + d < 0$), then $\mathcal{B}_1(x_2) = \{x_l\}$,
- if r is convex in x_1 (i.e., $a > 0$), then $\mathcal{B}_1(x_2) = \arg\max_{x \in \{x_l, x_u\}} r(x)$,
- if r is concave in x_1 (i.e., $a < 0$) and $x^* \in R$, then $\mathcal{B}_1(x_2) = \{x^*\}$, and
- if r is concave in x_1 (i.e., $a < 0$) and $x^* \notin R$, then $\mathcal{B}_1(x_2) = \arg\max_{x \in \{x_-, x_+\}} r(x)$.

Note that $\mathcal{B}_1(x_2)$ is not necessarily unique (or even a finite set). To find the NE, observe that the unrestricted best response functions $\mathcal{B}_1(x_2) = -\frac{c_1x_2 + d_1}{2a_1}$ and $\mathcal{B}_2(x_1) = -\frac{c_2x_1 + e_2}{2b_2}$ lead to the unique unrestricted NE

$$\boldsymbol{x}^* = (x_1^*, x_2^*) = \left(\frac{c_1e_2 - 2d_1b_2}{4a_1b_2 - c_1c_2}, \frac{c_2d_1 - 2e_2a_1}{4a_1b_2 - c_1c_2} \right) \ .$$

If this point exists and is allowed by R, i.e., $4a_1b_2 - c_1c_2 \neq 0$ and $\boldsymbol{x}^* \in \boldsymbol{R}$, then $\mathcal{N}|_R = \{\boldsymbol{x}^*\}$. Otherwise, we use fictitious play (i.e., successive mutual best responses) to find the fixed points, repeatedly calling the restricted best response functions while maintaining a list of candidate solutions.

A.2 Expected results for the Cournot Game

Let the function

$$D : \boldsymbol{A} \to \mathbb{R}, D(\boldsymbol{a}) = \sum_{i \in I} \min_{x \in \mathcal{B}_i(\boldsymbol{a}_{-i})} (a_i - x)^2 \tag{A.1}$$

measure the deviation of the joint action $\boldsymbol{a}$ from a Nash Equilibrium; by definition, its roots are exactly the Nash Equilibria of the respective best response functions. If there are no roots (i.e., $\min_{\boldsymbol{a} \in \boldsymbol{A}} D(\boldsymbol{a}) > 0$), there is no (pure) Nash Equilibrium.

For the (unrestricted) CG, the unique best responses are $\mathcal{B}_1(q_2) = \left\{\frac{\lambda - q_2}{2}\right\}$ and $\mathcal{B}_2(q_1) = \left\{\frac{\lambda - q_1}{2}\right\}$. Therefore, we get

$$\begin{aligned} D(\boldsymbol{q}) &= \left(q_1 - \frac{\lambda - q_2}{2}\right)^2 + \left(q_2 - \frac{\lambda - q_1}{2}\right)^2 \\ &= \frac{5}{4}(q_1^2 + q_2^2) + 2q_1q_2 - \frac{3}{2}\lambda(q_1 + q_2) + \frac{1}{2}\lambda^2 \end{aligned}$$

which has a unique global minimum $\boldsymbol{q}^* = (\frac{\lambda}{3}, \frac{\lambda}{3})$ with $D(\boldsymbol{q}^*) = 0$.

If we allow interval union restrictions for the CG, best responses are not unique anymore, but still follow a simple pattern: If the unrestricted best response q^* is not part of an allowed interval, the restricted best responses are the closest allowed actions on either one or both sides of q^*.

More formally: Let $q^* := \frac{\lambda}{3}$ be the unrestricted optimal quantity, and define, for a given restriction $R \subseteq [0, \lambda]$, the two closest allowed quantities $q^+ := \min_{q \in R}(\{q > q^*\})$ and $q^- := \max_{q \in R}(\{q < q^*\})$. Setting $\Delta^+ := q^+ - q^*$ and $\Delta^- := q^* - q^-$, the Nash Equilibria $\mathcal{N}|_R$ of the restricted CG are:

$$\mathcal{N}|_R = \begin{cases} \{(q^+, q^+)\} & \text{if } \Delta^+ < \frac{1}{2}\Delta^- \\ \{(q^+, q^+), (q^+, q^-), (q^-, q^+)\} & \text{if } \Delta^+ = \frac{1}{2}\Delta^- \\ \{(q^+, q^-), (q^-, q^+)\} & \text{if } \frac{1}{2}\Delta^- < \Delta^+ < 2\Delta^- \\ \{(q^-, q^-), (q^+, q^-), (q^-, q^+)\} & \text{if } \Delta^+ = 2\Delta^- \\ \{(q^-, q^-)\} & \text{if } \Delta^+ > 2\Delta^- \end{cases}.$$

This suggests the following sequence of successive restrictions for the AROGU algorithm:

- Identify $\frac{\lambda}{3}$ as the unique relevant action of the unrestricted game and therefore exclude $\overline{R} := \left[\frac{\lambda}{3} - \epsilon, \frac{\lambda}{3} + \epsilon\right)$ from the action space
- Identify both boundary actions as relevant and exclude one of them, increasing the excluded region $\overline{R}$ around $\frac{\lambda}{3}$
- Whenever $\overline{R}$ becomes imbalanced by a factor of > 2 around $\frac{\lambda}{3}$, a symmetric equilibrium appears at one end of it
- Finally, $\overline{R}$ is large enough to produce the symmetric equilibrium $(\frac{\lambda}{4}, \frac{\lambda}{4})$

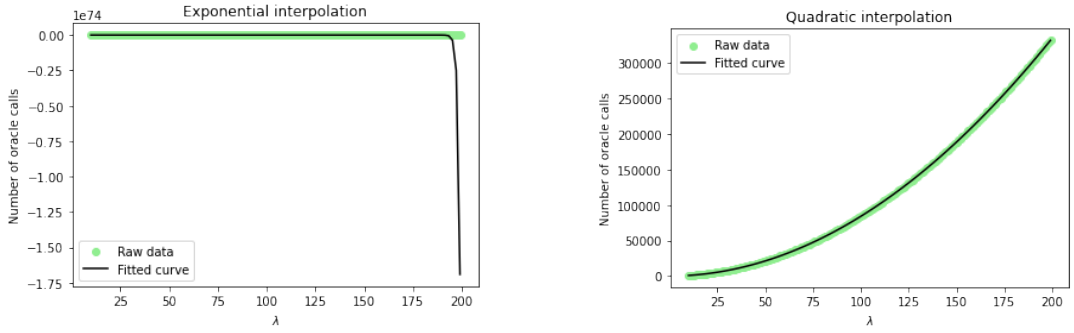


FIGURE A.1: Exponential and quadratic interpolation of the number of oracle calls in the Cournot Game

- This occurs when $\overline{R} = [\frac{\lambda}{4}, \frac{\lambda}{2})$, and therefore $R = [0, \frac{\lambda}{4}) \cup [\frac{\lambda}{2}, \lambda)$
- The algorithm goes on to enlarge $\overline{R}$ until the set of allowed actions becomes empty
- Since no further restriction produces a socially better stable solution, the largest (i.e., least restrictive) R with $(\frac{\lambda}{4}, \frac{\lambda}{4}) \in \mathcal{N}|_R$ is finally returned as the optimal restriction R^*
- The resulting degree of restriction is $\mathfrak{r}(R^*) = 25\%$

The optimal restriction R^* has the unique equilibrium $(\frac{\lambda}{4}, \frac{\lambda}{4})$ which gives the MESU $\mathcal{S}(R^*) = \mathfrak{u}(\frac{\lambda}{4}, \frac{\lambda}{4}) = \frac{1}{4}\lambda^2$. In contrast, the unrestricted game produces a unique equilibrium of $(\frac{\lambda}{3}, \frac{\lambda}{3})$, such that $\mathcal{S}(A) = \mathfrak{u}(\frac{\lambda}{3}, \frac{\lambda}{3}) = \frac{2}{9}\lambda^2$. The resulting relative improvement is $\Delta_{rel} = \frac{1}{8}$.

A.3 Number of oracle calls in the Cournot Game

To show that the number of oracle calls for AROGU's solution of the Cournot Game grows quadratically rather than exponentially, let us fit the two curves $f_1(\lambda) = ae^{b\lambda} + c$ and $f_2(\lambda) = a\lambda^2 + b\lambda + c$ to the data and check their deviation. Recall that the experimental data is $f(10) = 912$, $f(11) = 1095$, $f(12) = 1294$, $f(13) = 1513$, and so on (the full data set can be reproduced using the supplementary material).

As can be seen from Figure A.1, the quadratic interpolation polynomial f_2 gives a close-to-perfect fit with parameters $a = 8.33$, $b = 8.00$, and $c = -1.45$. In contrast, the exponential fit with f_1 produces the degenerate parameter values $a = 0.00$, $b = 1.00$, and $c = 1.12 \cdot 10^{62}$.

A.4 Continuous Braess Paradox

In the original (discrete) version of Braess' Paradox (see Example 7), each agent has three route options, of which they must choose exactly one. The travel time from node 0 to node 3 is then used as their cost function (i.e., it is to be minimized).

When transforming this into a one-dimensional continuous NFG, we have to address two points: (a) There has to be a continuum of actions, and (b) we need utility functions instead of cost functions. Therefore, we define the action space as $A = [0, 1]$ and give it the following meaning: Agent 1 routes a flow of x_1 through route $0 \to 1 \to 2 \to 3$, and the remaining flow of $(1 - x_1)$ through route $0 \to$

$2 \to 3$. Similarly, agent 2 routes a flow of x_2 through route $0 \to 1 \to 2 \to 3$, and the remaining flow of $(1 - x_2)$ through route $0 \to 1 \to 3$. This means that for both agents, 0 is the "cooperative" action, while 1 is the "competitive" action. The edge weights are adjusted such that full utilization (which is now a flow of 2 along an edge) gives the same travel time as utilization of 1 in the original setting (see Figure A.2).

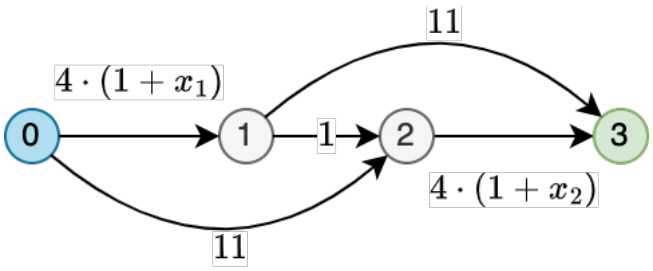


FIGURE A.2: Continuous version of Braess' Paradox

We calculate the expected travel time $c_i(\boldsymbol{x})$ for both agents and subtract them from a virtual baseline of 32 in order to get the reward functions $r_i(\boldsymbol{x})$. The expected travel time along a route is the flow on the route, multiplied by the sum of the edge latencies $l_{v,w}(\boldsymbol{x})$, given this flow[1]. For agent 1, this calculation is:

$$\begin{aligned} c_1(\boldsymbol{x}) &= x_1(l_{0,1} + l_{1,2} + l_{2,3}) + (1 - x_1)(l_{0,2} + l_{2,3}) \\ &= x_1(l_{01} + l_{1,2}) + (1 - x_1) \cdot l_{02} + l_{2,3} \\ &= x_1(4(1 + x_1) + 1) + 11(1 - x_1) + 4(1 + x_2) \\ &= 4x_1^2 - 6x_1 + 4x_2 + 15 \,, \end{aligned}$$

and the corresponding reward function is

$$r_1(x_1, x_2) = -4x_1^2 + 6x_1 - 4x_2 + 17 \,.$$

In the same way, we get

$$r_2(x_1, x_2) = -4x_2^2 - 4x_1 + 6x_2 + 17 \,.$$

To generalize this setting, let us assume the affine latency functions $l_{0,2}(\boldsymbol{x}) = l_{1,3}(\boldsymbol{x}) = a(x_1 + x_2) + b$ and $l_{0,1}(\boldsymbol{x}) = l_{2,3}(\boldsymbol{x}) = c(x_1 + x_2) + d$, while leaving the constant latency $l_{1,2}(\boldsymbol{x}) = 1$ unchanged. This gives the parameterized reward functions

$$r_1(\boldsymbol{x}) = -(a + c)x_1^2 + (2a + b - c - 1)x_1 - cx_2 + (4c + d + 1)$$

and

$$r_2(\boldsymbol{x}) = -(a + c)x_2^2 - cx_1 + (2a + b - c - 1)x_2 + (4c + d + 1) \,.$$

To obtain a one-dimensional range of experiments, we fix $a = 0$, $c = 4$ and $d = 0$ and vary b (intuitively, we vary the attractiveness of taking the cooperative routes, compared to the selfish route). The parameterized reward functions $r_i(\boldsymbol{x})$ are therefore

$$r_1(\boldsymbol{x}) = -4x_1^2 + (b - 5)x_1 - 4x_2 + 17$$

[1]Here, $l_{v,w}$ denotes the latency function of the edge from node v to node w.

and

$$r_2(\boldsymbol{x}) = -4x_2^2 - 4x_1 + (b-5)x_2 + 17\,.$$

A.5 Expected results for the Braess Paradox

From r_1 and r_2 as defined in Section 7.4.4, we can immediately derive the best response functions $\mathcal{B}_i(x_j) = \left\{\frac{b-5}{8}\right\}$ for all i, resulting in $\mathcal{N} = \left\{\left(\frac{b-5}{8}, \frac{b-5}{8}\right)\right\}$. Moreover, since $\mathfrak{u}(\boldsymbol{x}) = r_1(\boldsymbol{x}) + r_2(\boldsymbol{x}) = -4x_1^2 - 4x_2^2 + (b-9)x_1 + (b-9)x_2 + 34$, we get the social optimum $\boldsymbol{x}^* = \left(\frac{b-9}{8}, \frac{b-9}{8}\right)$.

Finally, we conclude from $A = [0,1]$ that, for $b \leq 5$ and $b \geq 17$, $\mathcal{N} = \{\boldsymbol{x}^*\}$ such that the unrestricted and the restricted MESU are equal. For $b \in (5,17)$, however, the two values differ, such that restricting A can improve the MESU.

Let us first assume that $b \in (5,9]$. To make $\frac{b-9}{8}$ a best response for an agent, we have to exclude any action from A that this agent would prefer over $\frac{b-9}{8}$. It is easy to see that the range of actions that needs to be excluded is $(0, \frac{b-5}{4})$, giving the unique optimal restriction $R^* = \{0\} \cup [\frac{b-5}{4}, 1]$.

For $b \in [9,17)$, a similar analysis yields that $(\frac{b-9}{8}, 1)$ needs to be excluded, and therefore $R^* = [0, \frac{b-9}{8}]$.

From the optimal restriction R^*, we can calculate the unrestricted and restricted MESU as well as the degree of restriction:

$$\mathcal{S}(A) = \begin{cases} 34 & \text{for } b \leq 5 \\ \frac{1}{8}\left(b^2 - 18b + 337\right) & \text{for } b \in [5,13] \\ 2b + 8 & \text{for } b \geq 13 \end{cases},$$

$$\mathcal{S}(R^*) = \begin{cases} 34 & \text{for } b \leq 9 \\ \frac{1}{8}(b-9)^2 + 34 & \text{for } b \in [9,17] \\ 2b + 8 & \text{for } b \geq 17 \end{cases},$$

and

$$\mathfrak{r}(R^*) = \begin{cases} 0 & \text{for } b \leq 5 \\ \frac{b-5}{4} & \text{for } b \in [5,9] \\ \frac{17-b}{8} & \text{for } b \in [9,17] \\ 0 & \text{for } b \geq 17 \end{cases}.$$

Appendix B

Supplementary material for Chapter 9

B.1 Traffic models

B.1.1 Multi-step microscopic model

The pseudo-code of the environmental model used for the experiments is listed in Algorithm 4. Some of the design choices are noteworthy:

- When an agent chooses its next edge, it does so using the *anticipated latency* of the network, i.e., it calculates the latency of an edge e as $l_e(f_e + 1)$, since the utilization will be incremented as soon as the car enters the road. There are two ways to use the anticipated latency for finding a shortest path: (a) Only use the anticipated latency for the first edge, or use it for all edges along the route. Moreover, to account for the total cost, each agent uses its value-of-time and value-of-money properties and chooses a shortest path with respect to the weighted sum of anticipated travel time and anticipated tolls.
- The speed of an agent along an edge (i.e., the number of steps it takes until it reaches the end of the edge) is set when the agent enters the edge, and is not changed anymore, even if the latency of the edge changes in subsequent steps when the agent is still traversing the edge. This gives a more realistic behavior than having incoming agents change the speed of agents which are further along the edge.
- At each step, the agents move along their current edge according to their speed. As soon as the end of the edge is reached, the agent stops, and at the next step makes its choice about the subsequent edge. This means that in this model, it always takes at least one step to traverse an edge, even if the latency is less than 1. Accordingly, the latency values need to be large enough to allow for differentiated travel times under this discretization.
- When an agent reaches its destination, the position is reset to the source. This step is executed prior to deciding on the subsequent edges and progressing traffic. The approach facilitates a more even flow with a fixed number of agents on the graph.

B.1.2 Single-step microscopic model

For the examples in Section 9.2, we use a model in which each agent is still a distinct entity; however, agents do not move on the network, but instead choose their

Algorithm 4: Multi-step traffic simulation

```
   input: Network, cars, number of steps
 1 foreach step do
 2 |   Compute flow for each edge in network based on car positions and
   |    update network attributes accordingly;
 3 |   foreach car c do
 4 |   |   Move c based on speed;
 5 |   |   if c has reached the end of its edge e then
 6 |   |   |   Decrement flow on e;
 7 |   |   end
 8 |   end
 9 |   foreach car c do
10 |   |   if c has reached its target then
11 |   |   |   Reset c;
12 |   |   end
13 |   end
14 |   foreach car c (in random order) do
15 |   |   if c is at a node then
16 |   |   |   Determine next edge e for c;
17 |   |   |   Set c's speed based on latency of e ;
18 |   |   |   Increment flow on e;
19 |   |   end
20 |   end
21 end
```

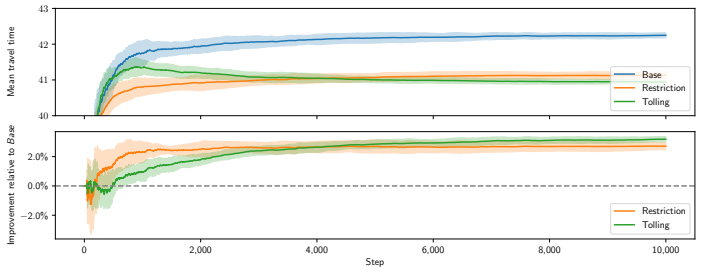
entire route at once, whereupon the travel times and total costs are computed in a single step. To avoid spikes and oscillations, we still let the agents choose their routes sequentially in random order (observing what the previous agents have chosen at the same step). From a game-theoretic viewpoint, this setup corresponds to an Extensive-Form Game, while the multi-step setup in Appendix B.1.1 is a Stochastic Game. Its results are "cleaner" than in the multi-step model, since an agent can be thought of as a continuous flow which is, at the same time, located at each edge of its route.

B.2 Double Braess Paradox

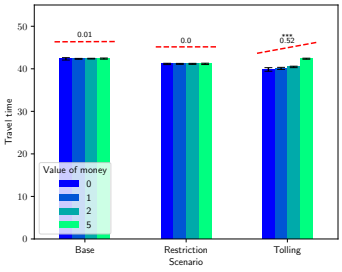
To construct this "Double Braess Paradox", we first observe that in the original network (Figure 9.2a), there are three types of latency functions: (a) *High constant latencies* like edge $(0,2)$, (b) *affine latencies* like $(0,1)$ and (c) *low constant latencies* like $(1,2)$. The latter type of edge is removed to resolve the paradox.

Therefore, we re-use the high-constant-latency edge $(0,2)$ as the *low-constant-latency* edge in another, superimposed paradox, such that its removal improves the flow on this second structure. Accordingly, we add two nodes A and B (taking the roles of 0 and 3 in the original setting), and connect them to the existing nodes 0 and 2. The latency functions of the new edges need to be chosen such that the unique optimal route from A to B is $A \to 0 \to 2 \to B$, regardless of the traffic rate.

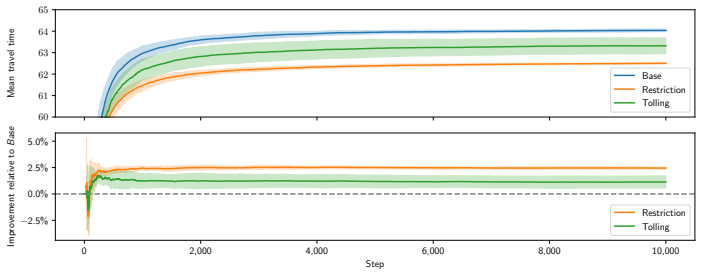
Note that the edge $(0,2)$ now has a double role: To resolve the paradox on the subgraph $\{0,1,2,3\}$, it needs to be present; to resolve the paradox on the extended



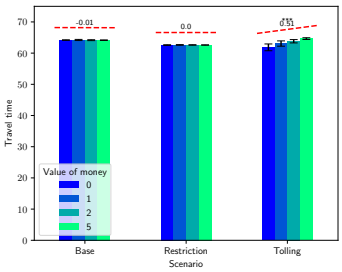
(A) $G_{50,0.07}$: Social welfare (i.e., mean travel time)



(B) $G_{50,0.07}$: Fairness



(C) $G_{34,0.06}$: Social welfare (i.e., mean travel time)



(D) $G_{34,0.06}$: Fairness

FIGURE B.1: Additional results of the $G_{n,p}$ graph experiment. Similar to the results shown in Figure 9.4, both *Restriction* and *Tolling* improve the mean travel time compared to *Base*. However, value of money significantly influences travel time with *Tolling* ($p \leq 0.01$).

graph $\{A, 0, 2, B\}$, it needs to be removed. Hence, we now have demand-dependent optimal restrictions as intended.

B.3 $G_{n,p}$ graphs

Our experiments follow the formal notation of Valiant and Roughgarden, 2010 in which each graph G is drawn from the standard Erdős-Rényi model $G_{n,p}$. Each edge is given an independent affine latency function $l_e(n) = a + b(\frac{n}{c})$ where a, b and c are integers drawn independently from three fixed uniform distributions $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. We search for Braess' Paradox by repeatedly sampling graphs, traffic rates, source nodes s, and destination nodes d. Hereby, agents are initialized on random edges of the simple paths between s and d. The restriction effect is measured by independently removing each edge e with a high equilibrium flow from G. We find Braess' Paradoxes when generating traffic scenarios with the following parameters and seed 46:

	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	f	s	d	e
$G_{50,0.07}$	$\mathcal{U}_{\{3,4\}}$	$\mathcal{U}_{\{2,3\}}$	$\mathcal{U}_{\{1,3\}}$	38	23	33	(23, 30)
$G_{34,0.06}$	$\mathcal{U}_{\{3,4\}}$	$\mathcal{U}_{\{2,4\}}$	$\mathcal{U}_{\{1,4\}}$	38	6	20	(25, 9)
$G_{59,0.08}$	$\mathcal{U}_{\{2,4\}}$	$\mathcal{U}_{\{2,3\}}$	$\{3\}$	31	3	27	(20, 41)

The travel time and fairness metrics for $G_{50,0.07}$ and $G_{34,0.06}$ are shown in Figure B.1.

B.4 Generalized Braess graphs

The same reasoning as in Appendix B.2 applies to the generalized Braess graphs: In every iteration i, the high-constant-latency edge (s_{i-1}, w) becomes a low-constant-latency edge in the subgraph $\{s_i, s_{i-1}, w, t_i\}$.

Formally, the graph $B_n = (V_n, E_n)$ is defined as:

$$\begin{aligned} V_n &:= \{s_0, ..., s_n, v, w, t_0, ..., t_n\} \\ E_n &:= \{(s_0, v), (v, w), (v, t_0)\} \cup \{(s_1, s_0), ..., (s_n, s_{n-1})\} \cup \\ &\quad \{(s_0, w), ..., (s_n, w)\} \cup \{(s_0, t_1), ..., (s_{n-1}, t_n)\} \cup \\ &\quad \{(w, t_0), ..., (w, t_n)\} \end{aligned}$$

with

$$l_e(f) := \begin{cases} 1 & \text{if } e = (v, w) \\ 2 + 6 \cdot \frac{f}{c} & \text{if } e = (s_i, s_{i-1}) \text{ or } (s_0, v) \text{ or } (w, t_i) \\ 11 + 2i & \text{if } e = (s_i, w) \text{ or } (s_i, t_{i+1}) \end{cases} .$$

It is clear that edges involving nodes s_j or t_j with $j > i$ are irrelevant for traffic flowing from s_i to t_i since there are no directed paths going through these nodes. For commodity (s_i, t_i), the optimal restriction is closing edges $\{(s_k, w), 0 \leq k \leq i - 1\}$.

B.5 Reproducing the experiments

Using the code provided in the accompanying repository and the seeds listed below, the results reported in Chapter 9 can be fully reproduced. Different seeds, of course, can give slightly different results, but will confirm that all claims are robust with respect to randomization.

We have used the following seeds for our experiments:

Experiment	Seeds
$G_{n,p}$	41, 42, 43, 44, 45, 46, 47, 48, 49, 50
Braess	42, 43, 44, 45, 46

Bibliography

Abdunabi, T. and Otman Basir (2014). "Holonic Intelligent Multi-Agent Algorithmic Trading System (HIMAATS)". In: *International Journal of Computers and their Applications*.

Acemoglu, Daron et al. (2016). "Informational Braess' Paradox: The Effect of Information on Traffic Congestion". In: *Operations Research* 66.

Achiam, Joshua et al. (2017). "Constrained Policy Optimization". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. JMLR.org, pp. 22–31.

Aires, João Paulo and Felipe Meneguzzi (2017). "Norm conflict identification using deep learning". In: *AAMAS workshops*.

Aldewereld, H. et al., eds. (2016). *Social coordination frameworks for social technical systems*. Springer.

Alur, Rajeev, Thomas A. Henzinger, and Orna Kupferman (2002). "Alternating-Time Temporal Logic". In: *J. ACM* 49.5.

Andelman, Nir, Michal Feldman, and Yishay Mansour (2009). "Strong Price of Anarchy". In: *Games and Economic Behavior* 65.2.

Andrighetto, Giulia et al., eds. (2013). *Normative Multi-Agent Systems*. Dagstuhl Follow-Ups.

Arcos, Josep Lluís, Juan A. Rodríguez-Aguilar, and Bruno Rosell (2008). "Engineering autonomic electronic institutions". In: *Engineering environment-mediated multi-agent systems: International workshop, EEMMAS 2007*. Springer-Verlag.

Asadi, Mehran and Manfred Huber (2004). *State Space Reduction For Hierarchical Reinforcement Learning*.

Axelrod, Robert (1984). *The Evolution of Cooperation*. New York: Basic.

Bade, Sophie (2005). "Nash Equilibrium in Games with Incomplete Preferences". In: *Economic Theory* 26.2. Publisher: Springer, pp. 309–332. (Visited on 07/30/2020).

Balke, Tina et al. (2013). *Norms in MAS: Definitions and Related Concepts*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Barbuceanu, M. (1997). "Coordinating agents by role based social constraints and conversation plans". In: *AAAI/IAAI*.

Barocas, Solon, Moritz Hardt, and Arvind Narayanan (2019). *Fairness and Machine Learning: Limitations and Opportunities*. http://www.fairmlbook.org.

Barrio, Eustasio del, Paula Gordaliza, and Jean-Michel Loubes (2020). "Review of Mathematical frameworks for Fairness in Machine Learning". In: *ArXiv* abs/2005.13755.

Bazzan, Ana L.C. and Franziska Klügl (2005). "Case studies on the Braess Paradox: Simulating route recommendation and learning in abstract and microscopic models". In: *Transportation Research Part C: Emerging Technologies* 13.4, pp. 299–319.

Beckmann, Martin J., C. B. McGuire, and C. B. Winsten (1955). *Studies in the Economics of Transportation*. Santa Monica, CA: RAND Corporation.

Berner, Christopher et al. (2019). *Dota 2 with Large Scale Deep Reinforcement Learning*. arXiv: 1912.06680 [cs.LG].

Bittihn, Stefan and Andreas Schadschneider (2021). "The effect of modern traffic information on Braess' paradox". In: *Physica A: Statistical Mechanics and its Applications* 571.

Boella, Guido, Leendert van der Torre, and Harko Verhagen (2006). "Introduction to normative multiagent systems". In: *Computational & Mathematical Organization Theory* 12.

— (2008). "Introduction to the special issue on normative multiagent systems". In: *Autonomous Agents and Multi-Agent Systems* 17.1.

Bou, Eva, Maite López-Sánchez, and Juan Antonio Rodríguez-Aguilar (2007). "Towards Self-configuration in Autonomic Electronic Institutions". In: *Coordination, Organizations, Institutions, and Norms in Agent Systems II*. Ed. by Pablo Noriega et al. Springer Berlin Heidelberg.

Boutilier, Craig et al. (2018). "Planning and Learning with Stochastic Action Sets". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization.

Braess, Dietrich (1968). "Über ein Paradoxon aus der Verkehrsplanung". In: *Unternehmensforschung* 12.1.

Brockman, Greg et al. (2016). "Openai gym". In: *arXiv preprint arXiv:1606.01540*.

Brustoloni, José Carlos (1991). "Autonomous Agents: Characterization and Requirements". In.

Bulling, Nils and Mehdi Dastani (2016). "Norm-based Mechanism Design". In: *Artif. Intell.* 239.C, pp. 97–142.

Cacciamani, Federico et al. (2021). "Multi-agent coordination in adversarial environments through signal mediated strategies". In: *Proceedings of the 20th international conference on autonomous agents and MultiAgent systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Cai, Qingpeng et al. (2018). "Reinforcement Mechanism Design for E-commerce". In: *Proceedings of the 2018 World Wide Web Conference*. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.

Canese, Lorenzo et al. (2021). "Multi-Agent Reinforcement Learning: A Review of Challenges and Applications". In: *Applied Sciences* 11, p. 4948.

Caspi, Itai et al. (2017). *Reinforcement Learning Coach*. URL: https://doi.org/10.5281/zenodo.1134899.

Chandak, Yash et al. (2020). "Lifelong Learning with a Changing Action Set". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.

Cheung, Wang Chi, David Simchi-Levi, and Ruihao Zhu (2020). "Reinforcement Learning for Non-Stationary Markov Decision Processes: The Blessing of (More) Optimism." In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*.

Chung, Fan and Stephen J. Young (2010). "Braess's Paradox in Large Sparse Graphs". In: *Internet and Network Economics*. Ed. by Amin Saberi. Springer Berlin Heidelberg.

Cigler, Ludek and Boi Faltings (2011). *Reaching Correlated Equilibria through Multi-Agent Learning*. Vol. 1. International Foundation for Autonomous Agents and Multiagent Systems.

Claus, Caroline and Craig Boutilier (1998). "The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems". In: *AAAI/IAAI*.

Conitzer, Vincent and Tuomas Sandholm (2003). "AWESOME: A General Multiagent Learning Algorithm that Converges in Self-Play and Learns a Best Response Against Stationary Opponents". In: *Machine Learning* 67.

Conte, Rosaria, Rino Falcone, and Giovanni Sartor (1999). "Introduction: Agents and Norms: How to fill the gap?" In: *Artificial Intelligence and Law* 7.1.

Corbett-Davies, Sam and Sharad Goel (2018). "The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning". In: *ArXiv* abs/1808.00023.

Cournot, Antoine Augustin (1838). *Recherches sur les principes mathématiques de la théorie des richesses*. Goldsmiths'-Kress library of economic literature 1450-1850. L. Hachette.

Cramton, Peter (2006). "Combinatorial Auctions". In: *European Economic Review*. MIT Press.

Dafoe, Allan et al. (2020). *Open Problems in Cooperative AI*. arXiv: 2012.08630 [cs.AI].

Daskalakis, Constantinos, Paul W. Goldberg, and Christos H. Papadimitriou (2009). "The Complexity of Computing a Nash Equilibrium". In: *SIAM Journal on Computing* 39.1, pp. 195–259.

Dean, Thomas, Robert Givan, and Sonia Leach (1997). "Model reduction techniques for computing approximately optimal solutions for markov decision processes". In: *Proceedings of the thirteenth conference on uncertainty in artificial intelligence*. UAI'97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 124–131.

Dell'Anna, Davide, Mehdi Dastani, and Fabiano Dalpiaz (2019). "Runtime Revision of Norms and Sanctions Based on Agent Preferences". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '19. event-place: Montreal QC, Canada. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 1609–1617.

Dietterich, Thomas G. (2000). "Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition". In: *J. Artif. Int. Res.* 13.1.

Ding, Chengri and Shunfeng Song (2012). "Traffic Paradoxes and Economic Solutions". In: *Journal of Urban Management* 1.1.

Doan, Thinh T., Siva Theja Maguluri, and Justin K. Romberg (2019). "Convergence Rates of Distributed TD(0) with Linear Function Approximation for Multi-Agent Reinforcement Learning". In: *arXiv: Optimization and Control*.

Du, Wei and Shifei Ding (2021). "A Survey on Multi-Agent Deep Reinforcement Learning: From the Perspective of Challenges and Applications". In: *Artificial Intelligence Review* 54.5.

Dulac-Arnold, Gabriel et al. (2016). *Deep Reinforcement Learning in Large Discrete Action Spaces*. arXiv: 1512.07679 [cs.AI].

Durugkar, Ishan, Elad Liebman, and Peter Stone (2020). "Balancing Individual Preferences and Shared Objectives in Multiagent Reinforcement Learning". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Christian Bessiere. International Joint Conferences on Artificial Intelligence Organization, pp. 2505–2511.

Eickmeyer, Kord and Ken-ichi Kawarabayashi (2013). "Approximating Multi Commodity Network Design on Graphs of Bounded Pathwidth and Bounded Degree". In: *Algorithmic Game Theory*. Ed. by Berthold Vöcking. Berlin, Heidelberg: Springer Berlin Heidelberg, 134–145.

EMERSON, E. Allen (1990). "CHAPTER 16 - Temporal and Modal Logic". In: *Formal Models and Semantics*. Ed. by JAN VAN LEEUWEN. Handbook of Theoretical Computer Science. Amsterdam: Elsevier, pp. 995–1072.

Esteva, Marc et al. (2001). "On the Formal Specification of Electronic Institutions". In: *Agent Mediated Electronic Commerce*. Vol. 1991. Springer.

Esteva, Marc et al. (2008). "Electronic institutions development environment". In: *AAMAS Demo Proceedings*. Vol. 3. International Foundation for Autonomous Agents and Multiagent Systems.

Fan, Zhou et al. (2019). "Hybrid Actor-Critic Reinforcement Learning in Parameterized Action Space". In: *International Joint Conference on Artificial Intelligence*.

Fitoussi, David and Moshe Tennenholtz (2000). "Choosing social laws for multi-agent systems: Minimality and simplicity". In: *Artificial Intelligence* 119.1, pp. 61 –101.

Foerster, Jakob N. et al. (2018). "Counterfactual Multi-Agent Policy Gradients". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press.

Franklin, Stan and Arthur C. Graesser (1996). "Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents". In: *ATAL*.

Frantz, Christopher and Gabriella Pigozzi (2018). "Modelling norm dynamics in multi-agent systems". In: *Journal of Applied Logic* 5.

Friedman, E.J. (2004). "Genericity and congestion control in selfish routing". In: *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. Vol. 5. Nassau, Bahamas: IEEE, pp. 4667–4672.

Fujimoto, Scott, Herke van Hoof, and David Meger (2018). "Addressing Function Approximation Error in Actor-Critic Methods". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1587–1596.

García-Camino, Andrés et al. (2006). "A rule-based approach to norm-oriented programming of electronic institutions". In: *SIGecom Exchanges* 5.

Gomez-Sanz, Jorge J. and Ruben Fuentes Fernandez (2016). "Ingenias". In: *Social Coordination Frameworks for Social Technical Systems*. Springer.

Grams, Tim (2023). "Dynamic interval restrictions on action spaces in deep reinforcement learning for obstacle avoidance". In: *Master's Thesis*. URL: https://arxiv.org/abs/2306.08008.

Greenwald, Amy and Keith Hall (2003). "Correlated-q learning". In: *Proceedings of the twentieth international conference on international conference on machine learning*. ICML'03. AAAI Press.

Grgic-Hlaca, Nina et al. (2016). "The Case for Process Fairness in Learning: Feature Selection for Fair Decision Making". In: *Proceedings of the Symposium on Machine Learning and the Law at the 29th Conference on Neural Information Processing Systems (NIPS 2016)*. Barcelona, Spain: NIPS.

Gronauer, Sven and Klaus Diepold (2021). "Multi-Agent Deep Reinforcement Learning: A Survey". In: *Artificial Intelligence Review*.

Gutierrez, Julian, Giuseppe Perelli, and Michael Wooldridge (2018). "Imperfect information in reactive modules games". In: *Information and Computation* 261, pp. 650 –675.

Hanna, Josiah P. et al. (2019). "Selecting Compliant Agents for Opt-in Micro-Tolling". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01.

Hausknecht, Matthew and Peter Stone (2016). "Deep Reinforcement Learning in Parameterized Action Space". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. San Juan, Puerto Rico.

Hernandez-Leal, Pablo, Bilal Kartal, and Matthew Taylor (2019). "A survey and critique of multiagent deep reinforcement learning". In: *Autonomous Agents and Multi-Agent Systems*.

Hill, Ashley et al. (2018). *Stable Baselines*. https://github.com/hill-a/stable-baselines.

Hoek, Wiebe and Michael Wooldridge (2012). "Logics for Multiagent Systems". In: *AI Magazine* 33, pp. 92–105.

Hoen, Pieter Jan 't et al. (2006). "An Overview of Cooperative and Competitive Multiagent Learning". In: *Learning and Adaption in Multi-Agent Systems*. Ed. by Karl Tuyls et al. Springer.

Hoffman, Matthew W. et al. (2020). "Acme: A Research Framework for Distributed Reinforcement Learning". In: URL: https://arxiv.org/abs/2006.00979.

Huang, Shengyi and Santiago Ontañón (2022). "A Closer Look at Invalid Action Masking in Policy Gradient Algorithms". In: *The International FLAIRS Conference Proceedings* 35.

Huang, Shengyi et al. (2022). "CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms". In: *Journal of Machine Learning Research* 23.274.

Hurwicz, Leonid and Stanley Reiter (2006). *Designing Economic Mechanisms*. Cambridge University Press.

Hwang, K., W. Jiang, and Y. Chen (2015). "Model Learning and Knowledge Sharing for a Multiagent System With Dyna-Q Learning". In: *IEEE Transactions on Cybernetics* 45.5.

Joseph, Matthew et al. (2016). "Rawlsian Fairness for Machine Learning". In: *ArXiv* abs/1610.09559.

Joshi, Mary Sissons, Vijay Joshi, and Roger Lamb (2005). "The Prisoners' Dilemma and City-Centre Traffic". In: *Oxford Economic Papers* 57.1, pp. 70–89.

Kalweit, G. et al. (2021). "Q-Learning with Long-term Action-space Shaping to Model Complex Behavior for Autonomous Lane Changes". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Kanervisto, Anssi, Christian Scheller, and Ville Hautamäki (2020). *Action Space Shaping in Deep Reinforcement Learning*. IEEE.

Kappenberger, Jakob, Kilian Theil, and Heiner Stuckenschmidt (2022). "Evaluating The Impact Of AI-Based Priced Parking With Social Simulation". In: *Social Informatics: 13th International Conference, SocInfo 2022, Glasgow, UK, October 19–21, 2022, Proceedings*. Berlin, Heidelberg: Springer-Verlag, pp. 54–75.

Kim, Seongmoon, M. E. Lewis, and C. C. White (2005). "State space reduction for nonstationary stochastic shortest path problems with real-time traffic information". In: *IEEE Transactions on Intelligent Transportation Systems* 6.3, pp. 273–284.

Konda, Vijay and John Tsitsiklis (1999). "Actor-Critic Algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press.

Koriche, Frédéric and Bruno Zanuttini (2010). "Learning conditional preference networks". In: *Artificial Intelligence* 174.11, pp. 685–703.

Krasowski, Hanna et al. (2023). *Provably Safe Reinforcement Learning: A Theoretical and Experimental Comparison*. arXiv: 2205.06750 [cs.LG].

Kuhnle, Alexander, Michael Schaarschmidt, and Kai Fricke (2017). *Tensorforce: a TensorFlow library for applied reinforcement learning*. Web page. URL: https://github.com/tensorforce/tensorforce.

Lanctot, Marc et al. (2020). *OpenSpiel: A Framework for Reinforcement Learning in Games*. arXiv: 1908.09453 [cs.LG].

Laszuk, Dawid (2020). *AI Traineree: Reinforcement learning toolset*. https://github.com/laszukdawid/ai-traineree.

Lecarpentier, Erwan and Emmanuel Rachelson (2019). *Non-Stationary Markov Decision Processes a Worst-Case Approach using Model-Based Reinforcement Learning*.

Lee, Ken Ming, Sriram Ganapathi Subramanian, and Mark Crowley (2021). *Investigation of Independent Reinforcement Learning Algorithms in Multi-Agent Environments*. arXiv: 2111.01100 [cs.MA].

Liang, Eric et al. (2018). "RLlib: Abstractions for Distributed Reinforcement Learning". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR.

Lillicrap, Timothy P. et al. (2016). "Continuous control with deep reinforcement learning". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Lin, Henry et al. (2011). "Stronger Bounds on Braess's Paradox and the Maximum Latency of Selfish Routing". In: *SIAM J. Discrete Math.* 25, pp. 1667–1686.

Littman, Michael L. (1994). "Markov games as a framework for multi-agent reinforcement learning". In: *Proceedings of the eleventh international conference on international conference on machine learning*. ICML'94. Morgan Kaufmann Publishers Inc.

Liu, L., A. Chattopadhyay, and U. Mitra (2019). "On Solving MDPs With Large State Space: Exploitation of Policy Structures and Spectral Properties". In: *IEEE Transactions on Communications* 67.6, pp. 4151–4165.

Lopes Cardoso, Henrique et al. (2016). "ANTE: A Framework Integrating Negotiation, Norms and Trust". In: *Social Coordination Frameworks for Social Technical Systems*. Vol. 30. Springer.

Lowe, Ryan et al. (2017). "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 6382–6393.

Lüdtke, Stefan et al. (2018). "State-space abstractions for probabilistic inference: A systematic review". In: *J. Artif. Int. Res.* 63.1, pp. 789–848.

Maerivoet, Sven and Bart De Moor (2005). *Transportation Planning and Traffic Flow Models*. arXiv: physics/0507127 [physics.soc-ph].

Maes, Pattie (1995). "Artificial Life Meets Entertainment: Lifelike Autonomous Agents". In: *Commun. ACM* 38.11.

Majeed, Sultan Javed and Marcus Hutter (2020). "Exact reduction of huge action spaces in general reinforcement learning". In.

Makhlouf, Karima, Sami Zhioua, and Catuscia Palamidessi (2020). "Survey on Causal-based Machine Learning Fairness Notions". In: *ArXiv* abs/2010.09553.

Mandel, Travis et al. (2017). "Where to Add Actions in Human-in-the-Loop Reinforcement Learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press.

Marín-Lora, Carlos et al. (2020). "A game engine to make games as multi-agent systems". In: *Advances in Engineering Software* 140.

Mataric, Maja J (1994). "Reward Functions for Accelerated Learning". In: *Machine Learning Proceedings 1994*. Ed. by William W. Cohen and Haym Hirsh. San Francisco (CA): Morgan Kaufmann.

Mazumdar, Eric V. and Lillian J. Ratliff (2018). "On the Convergence of Competitive, Multi-Agent Gradient-Based Learning". In: *ArXiv* abs/1804.05464.

McGroarty, Frank et al. (2019). "High frequency trading strategies, market fragility and price spikes: an agent based model perspective". In: *Annals of Operations Research* 282.1.

Mehrabi, Ninareh et al. (2021). "A Survey on Bias and Fairness in Machine Learning". In: *ACM Comput. Surv.* 54.6, pp. 1–35.

Mellema, René, Maarten Jensen, and Frank Dignum (2021). "Social Rules for Agent Systems". In: *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII*. Ed. by Andrea Aler Tubella et al. Springer International Publishing.

Memarzadeh, Milad, Scott Moura, and Arpad Horvath (2020). "Multi-Agent Management of Integrated Food-Energy-Water Systems Using Stochastic Games: From Nash Equilibrium to the Social Optimum". In: *Environmental Research Letters* 15.9.

Mittelmann, Munyque et al. (2022). "Automated Synthesis of Mechanisms". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by Lud De Raedt. International Joint Conferences on Artificial Intelligence Organization.

Mnih, Volodymyr et al. (2013). *Playing Atari with Deep Reinforcement Learning*. arXiv: 1312.5602 [cs.LG].

Mnih, Volodymyr et al. (2015). "Human-level control through deep reinforcement learning". In: *Nature* 518, pp. 529–33.

Mnih, Volodymyr et al. (2016). "Asynchronous methods for deep reinforcement learning". In: *Proceedings of the 33rd international conference on international conference on machine learning - volume 48*. ICML'16. JMLR.org, pp. 1928–1937.

Morales, J. (2016). "On-line norm synthesis for open Multi-Agent systems". PhD thesis. Universitat de Barcelona.

Morales, Javier et al. (2013). "Automated Synthesis of Normative Systems". In: vol. 1. International Foundation for Autonomous Agents and Multiagent Systems.

Morris-Martin, Andreasa, Marina De Vos, and Julian Padget (2019). "Norm emergence in multiagent systems: a viewpoint paper". In: *Autonomous Agents and Multi-Agent Systems* 33.6, pp. 706–749.

Morris-Martin, Andreasa, Marina De Vos, and Julian Padget (2021). "A Norm Emergence Framework for Normative MAS – Position Paper". In: *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII*. Ed. by Andrea Aler Tubella et al. Springer International Publishing.

Nemirovsky, Arkadii Semenovich and David Borisovich Yudin (1983). *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons.

Neufeld, Emery et al. (2021). "A Normative Supervisor for Reinforcement Learning Agents". In: *Automated Deduction – CADE 28*. Ed. by André Platzer and Geoff Sutcliffe. Springer International Publishing.

Neunert, Michael et al. (2020). "Continuous-Discrete Reinforcement Learning for Hybrid Control in Robotics". In: *CoRR* abs/2001.00449. arXiv: 2001.00449.

Nisan, Noam and Amir Ronen (2004). "Computationally Feasible VCG Mechanisms". In: *Journal of Artificial Intelligence Research* 29.

Noriega, Pablo (1997). "Agent-mediated auctions: The fishmarket metaphor". PhD thesis. Universitat Autonoma de Barcelona.

Noriega, Pablo and Dave Jonge (2016). "Electronic Institutions: The EI/EIDE Framework". In: *Social Coordination Frameworks for Socio-Technical Systems*. Vol. 30. Springer.

Nota, Chris (2020). *The Autonomous Learning Library*. https://github.com/cpnota/autonomous-learning-library.

Nowé, Ann, Peter Vrancx, and Yann-Michaël De Hauwere (2012). "Game Theory and Multi-agent Reinforcement Learning". In: *Reinforcement Learning: State-of-the-Art*. Ed. by Marco Wiering and Martijn van Otterlo. Berlin, Heidelberg: Springer Berlin Heidelberg.

Oesterle, Michael and Tim Grams (2024). "DRAMA at the PettingZoo: Dynamically Restricted Action Spaces for Multi-Agent Reinforcement Learning Frameworks". To appear in: Proceedings of the 57th Hawaii International Conference on System Sciences (HICSS).

Olson, Mancur (1965). *The logic of collective action: public goods and the theory of groups*. en-US. Harvard economic studies 124. Cambridge, Mass.: Harvard Univ. Press.

Oneto, Luca and Silvia Chiappa (2020). "Fairness in Machine Learning". In: *Recent Trends in Learning From Data*. Springer International Publishing, pp. 155–196.

Padakandla, Sindhu, Prabuchandran K. J., and Shalabh Bhatnagar (2020). "Reinforcement learning algorithm for non-stationary environments". In: *Applied Intelligence* 50.11.

Pala, Marco et al. (2012). "A new transport phenomenon in nanostructures: A mesoscopic analog of the Braess paradox encountered in road networks". In: *Nanoscale research letters* 7, pp. 472–472.

Park, Joon Sung et al. (2023). *Generative Agents: Interactive Simulacra of Human Behavior*. arXiv: 2304.03442 [cs.HC].

Pas, Eric I. and Shari L. Principio (1997). "Braess' paradox: Some new insights". In: *Transportation Research Part B: Methodological* 31.3, pp. 265–276.

Penchina, Claude M. (1997). "Braess paradox: Maximum penalty in a minimal critical network". In: *Transportation Research Part A: Policy and Practice* 31.5, pp. 379–388.

Perelli, Giuseppe (2019). "Enforcing Equilibria in Multi-Agent Systems". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '19. International Foundation for Autonomous Agents and Multiagent Systems.

Pernpeintner, Michael, Christian Bartelt, and Heiner Stuckenschmidt (2021). "Governing Black-Box Agents in Competitive Multi-Agent Systems". In: *Multi-Agent Systems - 18th European Conference, EUMAS 2021, Revised Selected Papers*. Ed. by Ariel Rosenfeld and Nimrod Talmon. Vol. 12802. Lecture Notes in Computer Science. Springer.

Plappert, Matthias (2016). *keras-rl*. https://github.com/keras-rl/keras-rl.

Pnueli, Amir (1977). "The temporal logic of programs". In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pp. 46–57.

Public Roads, United States. Bureau of (1964). *Traffic Assignment Manual for Application with a Large, High Speed Computer*. Washington, D.C.: U.S. Department of Commerce, Bureau of Public Roads, Office of Planning, Urban Planning Division.

Raffin, Antonin et al. (2021). "Stable-Baselines3: Reliable Reinforcement Learning Implementations". In: *Journal of Machine Learning Research* 22.268.

Rajkomar, Alvin et al. (2018). "Ensuring Fairness in Machine Learning to Advance Health Equity". In: *Annals of Internal Medicine* 169, pp. 866–872.

Rapoport, A. and A.M. Chammah (1965). *Prisoner's Dilemma: A Study in Conflict and Cooperation*. Ann Arbor paperbacks. Michigan: University of Michigan Press.

Rapoport, Anatol and Albert M. Chammah (1966). "The Game of Chicken". In: *American Behavioral Scientist* 10.3, pp. 10–28.

Relund Nielsen, Lars, Erik Jørgensen, and Søren Højsgaard (2011). "Embedding a state space model into a Markov decision process". In: *Annals of Operations Research* 190.1, pp. 289–309.

Riad, Maha and Fatemeh Golpayegani (2021). "Run-time Norms Synthesis in Multi-Objective Multi-Agent Systems".

Rizk, Yara, Mariette Awad, and E. Tunstel (2018). "Decision Making in Multi-Agent Systems: A Survey". In: *IEEE Transactions on Cognitive and Developmental Systems* PP.

Rotolo, Antonino (2011). "Norm compliance of rule-based cognitive agents." In: IJCAI International Joint Conference on Artificial Intelligence, pp. 2716–2721.

Rotolo, Antonino and Leendert van der Torre (2011). "Rules, Agents and Norms: Guidelines for Rule-Based Normative Multi-Agent Systems". In: *Rule-Based Reasoning, Programming, and Applications*. Ed. by Nick Bassiliades, Guido Governatori, and Adrian Paschke. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 52–66.

Roughgarden, Tim (2006). "On the severity of Braess's Paradox: Designing networks for selfish users is hard". In: *J. Comput. Syst. Sci.* 72, 922–953.

Roughgarden, Tim and Éva Tardos (2002). "How Bad Is Selfish Routing?" In: *Journal of The Acm* 49.2.

Rummery, G. and Mahesan Niranjan (1994). "On-Line Q-Learning Using Connectionist Systems". In: *Technical Report CUED/F-INFENG/TR 166*.

Rychetnik, L et al. (2002). "Criteria for evaluating evidence on public health interventions". In: *Journal of epidemiology and community health* 56, pp. 119–127.

Samvelyan, Mikayel et al. (2019). "The StarCraft Multi-Agent Challenge". In: *CoRR* abs/1902.04043.

Schulman, John et al. (2017). "Proximal Policy Optimization Algorithms". arXiv.

Sharon, Guni et al. (2017a). "Network-Wide Adaptive Tolling for Connected and Automated Vehicles". In: *Transportation Research Part C: Emerging Technologies* 84.

Sharon, Guni et al. (2017b). "Real-Time Adaptive Tolling Scheme for Optimized Social Welfare in Traffic Networks". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '17. International Foundation for Autonomous Agents and Multiagent Systems.

Sharon, Guni et al. (2018). "Traffic Optimization for a Mixture of Self-Interested and Compliant Agents". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1.

Sharon, Guni et al. (2019). "Marginal Cost Pricing with a Fixed Error Factor in Traffic Networks". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '19. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1539–1546.

Shoham, Yoav and Kevin Leyton-Brown (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge, UK: Cambridge University Press.

Shoham, Yoav and Moshe Tennenholtz (1995). "On social laws for artificial agent societies: off-line design". In: *Artificial Intelligence* 73.1.

Sinclair, Sean et al. (2020). "Adaptive Discretization for Model-Based Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc.

Skyrms, Brian (2003). *The Stag Hunt and the Evolution of Social Structure*. Cambridge: Cambridge University Press.

Steinberg, Richard and Willard I. Zangwill (1983). "The Prevalence of Braess' Paradox". In: *Transportation Science* 17.3, pp. 301–318.

Stirling, Wynn C. and Teppo Felin (2013). "Game theory, conditional preferences, and social influence". In: *PLOS ONE* 8.2, pp. 1–11.

Sunehag, Peter et al. (2018). "Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2085–2087.

Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book.

Tan, Ming (1997). "Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents". In: *International Conference on Machine Learning*.

Tang, Pingzhong (2017). "Reinforcement Mechanism Design". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*.

Tang, Yunhao and Shipra Agrawal (2020). *Discretizing Continuous Action Space for On-Policy Optimization*. arXiv: 1901.10500 [cs.LG].

Terry, Jordan, Benjamin Black, and Ananth Hari (2020). "SuperSuit: Simple Microwrappers for Reinforcement Learning Environments". In: *arXiv preprint arXiv:2008.08932*.

Terry, Jordan et al. (2021). "Pettingzoo: Gym for multi-agent reinforcement learning". In: *Advances in Neural Information Processing Systems* 34.

Turvey, Ralph (1969). "Marginal Cost". In: *The Economic Journal* 79.314.

Uther, William T. B. and Manuela M. Veloso (1998). "Tree Based Discretization for Continuous State Space Reinforcement Learning". In: *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*. AAAI '98/IAAI '98. American Association for Artificial Intelligence.

Valiant, Greg and Tim Roughgarden (2006). "Braess's Paradox in Large Random Graphs". In: *Proceedings of the 7th ACM Conference on Electronic Commerce*. EC '06. Association for Computing Machinery, pp. 296–305.

Valiant, Gregory and Tim Roughgarden (2010). "Braess's Paradox in large random graphs." In: *Random Struct. Algorithms* 37, pp. 495–515.

Van Lange, Paul A.M. et al. (2013). "The psychology of social dilemmas: A review". In: *Organizational Behavior and Human Decision Processes* 120.2, pp. 125–141.

Vinyals, Oriol et al. (2017). *StarCraft II: A New Challenge for Reinforcement Learning*. arXiv: 1708.04782 [cs.LG].

von Neumann, J. and O. Morgenstern (1947). *Theory of games and economic behavior*. Princeton University Press.

Wai, Hoi-To et al. (2018). "Multi-Agent Reinforcement Learning via Double Averaging Primal-Dual Optimization". In: *ArXiv* abs/1806.00877.

Wang, Xiaofeng and Tuomas Sandholm (2002). "Reinforcement learning to play an optimal nash equilibrium in team markov games". In: *NIPS*.

Watkins, Christopher (1989). "Learning From Delayed Rewards". In.

Weng, Jiayi et al. (2022). "Tianshou: A Highly Modularized Deep Reinforcement Learning Library". In: *Journal of Machine Learning Research* 23.267.

Weyns, Danny, Sven Brückner, and Yves Demazeau (2007). *Engineering Environment-Mediated Multi-Agent Systems*. Springer.

Williams, Ronald J. (1992). "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning". In: *Mach. Learn.* 8.3–4, 229–256.

World Health Organization (2023). *Health equity*. URL: https://www.who.int/health-topics/health-equity.

Wu, Yuxiang et al. (2023). *ChatArena: Multi-Agent Language Game Environments for Large Language Models*. https://github.com/chatarena/chatarena. Version 0.1.

Xiong, Jiechao et al. (2018). *Parametrized Deep Q-Networks Learning: Reinforcement Learning with Discrete-Continuous Hybrid Action Space*.

Zahavy, Tom et al. (2018). "Learn What Not to Learn: Action Elimination with Deep Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31.

Zaib, Munazza, Quan Z. Sheng, and Wei Emma Zhang (2021). *A short survey of pre-trained language models for conversational AI-A NewAge in NLP*.

Zhang, Kaiqing, Zhuoran Yang, and Tamer Basar (2019). "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms". In: *Handbook of Reinforcement Learning and Control*. Springer.

Zhang, Min-Ling and Zhi-Hua Zhou (2006). "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization". In: *IEEE Transactions on Knowledge and Data Engineering* 18.10, pp. 1338–1351.

Zinkevich, Martin, Amy Greenwald, and Michael L. Littman (2005). "Cyclic equilibria in markov games". In: *Proceedings of the 18th international conference on neural information processing systems*. NIPS'05. Cambridge, MA, USA: MIT Press.

Zverovich, Vadim E. and Erel Avineri (2012). "Braess' Paradox in a Generalised Traffic Network". In: *ArXiv* abs/1207.3251, pp. 114–138.

Åström, K.J (1965). "Optimal control of Markov processes with incomplete state information". In: *Journal of Mathematical Analysis and Applications* 10.1.