Scalable Coordination of Intelligent Vehicles in Shared Markovian Dynamics

Sarah H.Q. Li

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Behçet Açıkmeşe, Chair

Mehran Mesbahi

Lillian Ratliff

Program Authorized to Offer Degree:

William E. Boeing Department of Aeronautics & Astronautics

University of Washington

Abstract

Scalable Coordination of Intelligent Vehicles in Shared Markovian Dynamics

Sarah H.Q. Li

Chair of the Supervisory Committee:
Professor Behçet Açıkmeşe
William E. Boeing Department of Aeronautics and Astronautics

Driven by the growing demand for mobility and connectivity, future aerospace-based transportation systems necessitate efficient coordination of not just one, or two, but a population of intelligent vehicles that execute independent tasks in a shared operation environment. Combining techniques from game theory, optimization, and Markov decision process, the dissertation tackles three key challenges in coordinating intelligent vehicles sharing a disruption-prone environment: 1) maximizing safety in multi-vehicle trajectory planning, 2) strengthening fleet resiliency to resource disruptions, and 3) optimizing individual performance and safety when coordination is not possible. All three challenges revolve around building **a coordination framework for intelligent autonomous vehicles that prioritizes each vehicle's performance and safety.** Grounded in this goal, this dissertation combines theoretical tools with data-driven verification to provably facilitate large-scale autonomy in urban air spaces and ground transportation.

This dissertation uses Markov games and Markov decision processes to optimize decision-making in environments influenced by unpredictable external disruptions. These models help us understand how competitive route planning and unpredictable resource disruptions impact individual safety and the overall congestion level in the environment. For instance, how can multiple aircraft owned by different airlines collectively adjust their routes, so that each aircraft's collision risk is minimized despite uncertain airport delays? Modeling each aircraft's interdependent decision-making process as a coupled Markov decision process, this dissertation derives efficient algorithms for finding routes that can be simultaneously optimal for all aircraft. Furthermore, this dissertation uses these models to derive incentives that produce fleet-level trends and investigate collision minimization techniques with and without

a central coordinator.

In the centralized coordination scheme, a Markov game is explicitly formulated for coordinating individual decision-makers who must operate in a shared state-action space while executing independent tasks. In Chapter 3, the Markov decision process routing game model is expanded to atomic Markov games. The Markov game model is then applied to minimize collision risks in air traffic management and optimize warehouse path planning considering stochastic package arrival times. Multiple necessary and sufficient conditions on the player cost functions that ensure the existence of Nash equilibrium are given, as well as a first-order gradient descent method that uses iterative dynamic programming to compute the game's Nash equilibrium of the game. In Chapter 4, the Markov decision process congestion game model is used to study the effectiveness of incentives in enforcing population constraints and demonstrated on a group of ride-hail drivers in New York City. The stability of Markov games under resource disruptions and adversarial learning dynamics are analyzed in Chapters 5 and 6.

In the uncoordinated scheme, an individual decision maker who cannot explicitly coordinate with others (but nonetheless share a state-action space) is modeled by a Markov decision process with non-stationary parameter uncertainty, and the resulting non-stationary Bellman iteration is analyzed via a novel set-theoretic approach. In Chapter 7, a novel perspective on classic contraction operators used in Markov decision processes is introduced. Interaction between decision-makers is abstracted as a compact set of parameter uncertainty on an individual Markov decision process, and a set-based operator is introduced to derive convergence guarantees for dynamic programming under non-stationary parameter uncertainty.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

As the African proverb goes, it takes a village to raise a child. In my academic journey, I was fortunate to benefit from the guidance, wisdom, and care of multiple communities. I would like to use this opportunity to acknowledge as many as I can.

My interest in academia started at the University of British Columbia. I would like to thank George Bluman and Elizabeth Croft. George Bluman was the first to enthusiastically pitch academia as a viable career to me. His advice has guided me to this day. Elizabeth Croft welcomed me into the CARIS lab, let me explain my ideas with stick people figures, and co-authored my first paper. I would like to sincerely thank Mike van der Loos, Cole Shing, Camilo Quintero, and everyone else who kindly humored me as I stumbled through my first research project.

At the University of Washington, I had the incredible serendipity of meeting exceptional mentors and colleagues. I would like to thank the administrative staff in the Aeronautics and Astronautics department, Amy Sprague, Ed Connery, Paul Neubert, Betsy Winter, and Rachel Reichert, who always made sure I signed up for enough credits and was correctly funded. The most significant influence on this dissertation is my advisor Behçet Açıkmeşe. Dear Behçet, thank you for teaching me how to be optimistic in the face of bleak outlooks, for giving me incredible academic freedom, and most of all, for helping me appreciate my non-linearity in a linear world. I would also like to thank Pierre-Loïc Garoche, who significantly impacted the second half of this dissertation. Joining Pierre-Loïc at ONERA and subsequently ENAC pushed me through a crucial period in my Ph.D.. Pierre-Loïc's attention to detail, extraordinary ability to sort out scrambled thoughts into coherent ideas, and strong work ethic continue to be a source of aspiration.

This dissertation would be wildly different without Dan Calderone and Lillian Ratliff. Conversations with Dan have always been expansive yet thought-provoking. From the beginning, I was enamored by the Markov decision process routing game

model that Dan developed in his thesis. So much so that developing its extensions formed a significant portion of my thesis. Similarly, Lillian has been an invaluable mentor from the beginning. Leading by example, Lillian motivates me to strive for excellence in all aspects of research and just be the best version of myself. I look up to her for her dedicated work ethic, mathematical prowess, and razor-sharp intuition.

Many more mentors have significantly influenced this dissertation. I would like to thank Mehran Mesbahi, whose work on graph theory and multi-agent systems first piqued my interest in control theory, Assalé Adjé, whose unique expertise in computer science and control theory has been absolutely delightful to collaborate with, Kristi Morgansen, whose outstanding leadership of UW's Aeronautics and Astronautics department has greatly benefited all of us, Claus Danielson, whose strong analytical skills and breadth of knowledge in aerospace applications taught me a lot along the way, and Meeko Oishi, whose clarity of thought and ability to distill complex ideas into simple ones is a skill I hope to emulate. Many more from the University of Washington's community have selflessly assisted my academic journey: Karen Leung, Amir Taghvaei, Sam Burden, Jeff Ban, and Dana Dabiri, thank you for all the advice and hallway chats over the years. Last but not least, I would like to thank Ufuk Topçu, who taught me how to spot a good research problem and how to navigate the academic job field.

In Seattle, I am surrounded by breathtaking landscapes and an incredible group of friends and colleagues. I would like to thank the Autonomous Control Lab crew: Yue Yu, for his steadfast guidance. Mo Zhao, Dylan Janak, Miki Szmuk, Sean Rice, and Yuanqi Mao, for welcoming me into the Gug 306 with NASA stories and control folklore. Mathias Hudoba de Badyn, Bijan Barzgaran, Siavash Alemzadeh, Jingjing Bu, and Dillon Foight, for all the stimulating conversations in the RAIN lab. Satpreet Singh, for the interesting math conversations and six-foot-apart walks during COVID. Dominic McPherson-Liao, another Canadian who studied Ph.D. in the US and interned in France, for continuously allowing me to follow you on your academic path. Skye McKeowen, Chris Hayner, Dayou Li, Taewan Kim, Kazu Echigo, Natalia Pavlasek, Oliver Sheridan, Purna Elango, Abhi Samath, and Samet Uzun, for all the SpaceX internship stories. May the monitor population in Gug 306 continue to grow.

Outside of research, my last six years have been filled with countless climbing

movies and rocky moments. I would like to thank Pablo Trefftz-Posada for showing me all the cool outdoor climbing spots in Washington, Abhiram Aithal for helping me summit Mount St. Helens, Dan Tabas for bringing me to eye-burning altitudes and coming down uncomfortably fast, Momona Yamagami and Nicole Atmadja for the meditative early morning climbing sessions, Tom Wilding-Steele for leading my first multi-pitch in the French Pyrenees, Martin Cross for exploring Malta's limestones with me, and Nicole Atmadja, Kelvin Ritland, and Aaron Braunstein for some memorable trips to the Italian Dolomites and Squamish.

I cannot tell left from the right if my life depends on it. Luckily, I can always tell a kindred soul when I meet one. I want to thank Kim Namoco, Martin Cross, and Kelvin Ritland for supporting me through my wide-ranging emotional repertoire from exuberance to hysteria, and for picking up our friendship right where we left it every time I reappear after yet another deadline. Most of all, I want to thank Aaron Braunstein for bringing so much happiness and cat fur into my life, for helping me locate my phone and keys, and for being equally late to everything as me (if we are both late then we are really on time). I am beyond excited to see where our next adventure takes us.

My first and foremost teacher is my grandmother, who taught me how to dream bigger. When my eight-year-old self excitedly exclaimed that I will be an astronaut, a mathematician, and a doctor all in one breath, my grandmother did not laugh at me. Instead, she advised me to not just aim for a profession but to be the Louis Armstrong, Johann Gauss, and Norman Bethune of their fields.

Finally, I would like to thank my parents, Hong Yang, and Bailin Li, and my brother, Kevin Li, for bringing me along on their incredible journey and shaping me into the person I am today.

FUNDING

This dissertation is partially funded by

1. National Science Foundation Grants CMMI-210563, CMMI-2105502, CMMI-1613235, CNS-1736582,
2. Feanicses project, France Grant ANR17-CE25-0018,
3. University of Washington Aero&Astro Condit Fellowship,
4. University of Washington Top Scholar Fellowship,
5. Zonta International Amelia Earhart Fellowship.

Nomenclature

$[N]$ $\{1, \ldots, N\}$, $N \in \mathbb{N}$

Δ_N $\{y \in \mathbb{R}^N_+ \mid \sum_i y_i = 1\}$. A simplex of dimension N.

$\mathbb{E}[\cdot]$ The expectation of a random variable.

$\mathbb{P}(\mathcal{X})$ The set of all *non-empty compact subsets* of $\mathcal{X}$.

$\mathcal{T}$ $\{0, \ldots, T\}$. Finite time horizon.

$\|\cdot\|$ Norm of a real vector space.

$\mathbb{1}_N$ A column vector of ones: $\mathbb{1}_N = [1, \ldots, 1]^\top \in \mathbb{R}^{N \times 1}$.

$\mathbb{R}(\mathbb{R}_+)$ The (positive) real number line.

$B_\epsilon(x^\star)$ $\{x \mid \|x - x^\star\| \leq \epsilon, x \in \mathbb{R}^n\}$. The set of all elements in $\mathbb{R}^n$ that is at most $\epsilon \in \mathbb{R}_+$ away from the input $x^\star \in \mathbb{R}^n$.

$C^r(\mathbb{R}^n, \mathbb{R})$ The set of continuously differentiable functions $f : \mathbb{R}^n \mapsto \mathbb{R}$, such that $\frac{d^j f(x)}{du^j}$ exists for all $x \in \mathbb{R}^n$ and $1 \leq j \leq r$.

I_N An identity matrix of size $N \times N$.

x^{-i} When $x = (x^i, x^{-i})$, x^{-i} is the strategy space of the opponents of player i.

Chapter 1

INTRODUCTION

This dissertation uses Markov decision process, game theory, and optimization to enable *large-scale trajectory planning and coordination between self-interested vehicles in shared operation environments*. The attraction towards large-scale systems of self-interested vehicles is spurred by several technological disruptions in the aerospace industry: 1) reduced cost of aircraft and spacecraft operating in urban air and orbital spaces [1], 2) proven capability of autonomous guidance and navigation to safely operate in uncontrolled environments [79], and 3) usage of aerospace systems for on-demand transportation and telecommunication services [130, 85]. Together, these paradigm shifts have re-ignited public interest in developing aerospace-based transportation at scale, particularly for on-demand mobility and connectivity purposes. However, the untapped potential in aerospace-based solutions cannot be fully explored without first addressing the following challenge: **how can aerial and orbital spaces safely and efficiently accommodate a large number of autonomous aircraft and spacecraft**? This dissertation tackles this challenge by formulating a mathematical framework that is geared toward enabling self-interested dynamical systems to safely and efficiently share space with one another.

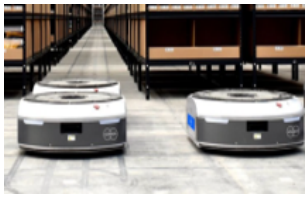
Publicly accessible air and orbital spaces. The aerospace industry has recently witnessed many disruptive technologies from the commercial sector. In January 2023, SpaceX completed its 200^{th} flight using reusable rockets, with 145 of those launched via reusable rockets [2]. Commercially available UAVs can carry loads of up to 225 kilogram for up to 45 minutes [3]. While these technologies are currently quite expensive for the average consumer, they are the prime solution for addressing the growing global demand for on-demand mobility and connectivity.

Growing on-demand mobility and connectivity markets. The global 5G core and the on-demand transportation markets are set to grow at 30.7% and 22.8% compound annual growth rates in the next decade [48, 47], respectively. As these markets grow, the supporting infrastructure must also expand to accommodate the growing consumer demand. Current mobility demands are primarily met through three transport modes: ground, air,

and marine. Among these, ground transportation is the sole means of performing door-to-door on-demand mobility. However, it is already buckling under the current usage trends in metropolitan cities around the world. Similarly, global mobile telecommunication services are predominantly conducted through cell towers. While these towers provide reliable internet service, they have limited coverage. The high start-up costs associated with cell towers further discourage telecommunication companies from providing mobile internet services to regions with low consumer density, resulting in limited internet access for many. As most services become increasingly digitized, global internet coverage is no longer a privilege, but a necessity for consumers. In summary, both current transportation and telecommunication deal with insufficient **spatial resources** that hinders global mobility and connectivity.

(a) In-orbit satellites awaiting maintenance services.

(b) Fixed wing and rotary wing UAVs sharing air space.



(c) Warehouse robots avoiding each other to retrieve packages.

(d) Robo-taxis navigating on a congested road.

Air-based transportation and orbital-based telecommunication. Urban air and orbital spaces are promising venues into which existing transportation and telecommunication infrastructure can expand into. Urban air enables three-dimension transportation, therefore greatly increasing the traffic throughput within urban regions. On the telecommunication front, orbital-based satellite networks offer the benefit of non-discriminating service to urban and rural populations alike, as well as provide greater service flexibility. As demonstrated by the recent Starlink initiative to assist Ukraine [131], satellite networks can be easily reconfigured to complement real-time demand surges and ground service outages.

In creating aerospace solutions to alleviate the growing pressure of mobility and connectivity demands on civil infrastructure, it is important to learn the lessons from building the existing transportation and telecommunication systems. The key issues in current transportation and telecommunication that reduce the efficacy of mobility and connectivity are 1) congested space usage, 2) unpredictable and potentially unsafe interactions between autonomous vehicles and other road users, and 3) global propagation of local disruptions. These challenges result in delays, collisions, and cascading service outages. When providing similar

services in urban air and orbital spaces, the following domain-specific complications have the potential to significantly amplify the delays, collisions, and service outages that result from real-time dynamic surges and unpredictable interactions.

1. **Complex aerodynamics and orbital dynamics**. Although aerospace-based telecommunication and transportation introduce greater flexibility in mobility, this freedom is accompanied by greater navigation and guidance complexity. As opposed to ground vehicles which only experience two-dimensional motion, aircraft and spacecraft are governed by three-dimensional aerodynamics and orbital dynamics. For example, fixed-winged aircraft will leave turbulent waves in its wake, which introduces temporal-spatial safety constraints to other aircraft operating in the vicinity.

2. **Task and environment uncertainty**. Aircraft and spacecraft both will experience greater uncertainty than their vehicle counterpart due to the more complex dynamics as well as the lack of sensing capabilities. For example, wind patterns in urban air spaces are difficult to predict or fully map out. Since small load-carrying UAVs can be quite sensitive to wind gusts, UAV controllers must be able to handle a wide range of operational environments. In low Earth orbits, satellites must operate autonomously for several hours while correcting their position under environmental factors including radiation exposure, debris, and atmospheric drag, which are often locally unquantifiable.

3. **Stringent resource limitations**. Resources, both in terms of fuel and space, are extremely limited in air spaces. While space limitations also exist in ground transportation, they become more stringent and complex in air spaces, where the aerodynamic mechanism that enables a UAV to stay afloat also generates predictable turbulent air flows in its proximity. These air flows can be complementary or detrimental to UAVs sharing the same air space. In orbital spaces, orbits, not satellites, tend to be the most valuable asset. When an internet-providing satellite breaks down, it must vacate its current orbit to allow an operational satellite to take its place and resume internet services.

4. **Elevated space usage control**. Traditionally, safety requirements for aircraft are more stringent than for ground vehicles due to their greater potential for catastrophe and the lack of sensing capabilities in the air. In addition to vehicle design, maintenance, and emergency response requirements, air space traffic is tightly controlled

via air traffic control, in which aircraft need to submit their flight plan typically an hour before departure. In most commercial flights, airlines petition for flight space and receive flight plans weeks in advance. In the low Earth orbit, several organizations monitor and direct spacecraft traffic to mitigate the risk of space collisions.

1.1 Thesis Contributions

To enable efficient and safe aerospace-based mobility and connectivity services, this thesis combines game theory, optimization, and Markov decision process to develop scalable planning frameworks that are resource-centric, ensure safety under uncoordinated interactions, and attenuate the impact of local disruptions on fleet-level performance. A detailed list is given below.

1. **Existence of Nash equilibria for selfish multi-vehicle routing under environmental uncertainty**. We model a group of heterogeneous players responding to stochastic demands as a congestion game under Markov decision process dynamics [71, 69]. We formulate the player-specific optimization problem, prove the equivalence between the Nash equilibrium and the solution of a potential minimization problem, and derive dynamic programming approaches to solve the Nash equilibrium with linear computation complexity in the number of players. Under the Markov decision process congestion game framework, we then study the problem of using incentives to enforce population distribution constraints on irrational players in stochastic environments. We show that myopically updating incentive results in sublinear convergence of the empirical average constraint satisfaction of irrational players with an upper bound on their irrationality [66, 74].

2. **Attenuating disturbances from fleet performance metrics**. While disturbances injected along a coordinate corresponding to any individual player's actions can always affect the overall learning dynamics, a subset of players can be disturbance decoupled. We provide the necessary and sufficient conditions to guarantee disturbance decoupling in games with quadratic cost functions [72]. Under the non-atomic Markov decision process framework, we analyze the sensitivity of the congestion game's Wardrop equilibria to perturbations in the player costs by applying implicit function analysis. The stochastic Braess paradox is defined and shown to exist in simple Wheatstone networks [70].

3. **Fixed point analysis for Bellman operator under MDP parameter uncertainty**. We study the effect of parameter uncertainty on contraction operators used in dynamic programming and reinforcement learning. For compact parameter uncertainty, we define a set-based value operator and show the existence of an invariant value function set [67, 68]. The invariant value function set is interpreted in the context of robust dynamic programming. We show that classic robust and optimistic policies correspond to bounding elements of the invariant value function set. Furthermore, we derive Hausdorff-distance convergence guarantees for diverging value iterations. We relate the existence of extrema points of the invariant value function set to properties of the MDP parameter uncertainty set. In particular, we derive a containment condition for guaranteeing the existence of the extremum points and relate it to the rectangularity condition used in robust dynamic programming.

Chapter 2

MATHEMATICAL MODELS: CONSTRAINED OPTIMIZATION, MARKOV DECISION PROCESS, AND GAME THEORY

To model fleets of intelligent vehicles subjected to shared resource disruptions and stochastic dynamics, this dissertation combines techniques from Markov decision process, game theory, and optimization. In this chapter, the key models and concepts are introduced along with the mathematical notation used throughout the rest of the dissertation.

2.1 Constrained Optimization

Optimization is widely employed in control theory for designing stable and optimal controllers. Common techniques include model predictive control (MPC) [112], state-feedback controller design [143], and linear quadratic regulator [58], among others. An optimization problem has the following general form.

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \ f(x) \\ &\text{s.t. } \ g(x) \leq 0, \ h(x) = 0, \end{aligned} \tag{2.1}$$

where x is a vector that includes both the state and control decision variables[1], $f : \mathbb{R}^n \mapsto \mathbb{R}$, $g : \mathbb{R}^n \mapsto \mathbb{R}^m$, and $h : \mathbb{R}^n \mapsto \mathbb{R}^p$ are assumed to be continuous functions. The objective function f quantifies the performance of any control input and the resulting trajectory. The functions g and h typically enforce dynamic feasibility and physical constraints on control input, safety, etc. The goal of (2.1) is to find a feasible set of controls that minimizes f while satisfying the given constraints.

Definition 2.1 (Local and global optimality) *[19, Sec.1.4] A point $x^\star \in \mathbb{R}^n$ is a* ***local***

[1] while many optimal control problems separate the state and control variables [58], we keep them together so that the optimization problem formulation (2.1) can also apply to the linear program formulation of Markov decision processes.

minimum *for the optimization problem* (2.1) *if there exists* $\epsilon > 0$*, such that*

$$f(x^\star) \leq f(x), \quad \forall\ x \in B_\epsilon(x^\star).$$

If the inequality above is strict, $f(x^\star) < f(x)$*, then* $x^\star$ *is a strict local minimum. If* $\epsilon = \infty$*, then* $x^\star$ *is* ***globally optimal****.*

Consider the unconstrained problem $\min_{x\in\mathbb{R}^n} f(x)$ where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable, a necessary condition for a candidate point $x^\star$ to be locally optimal is that the objective gradient $df(x^\star)/dx = 0 \in \mathbb{R}^n$ [25]. However, when constraints such as g and h (2.1) exist, a point $x^\star$ can be the optimal solution with a non-zero gradient vector [19]. Thus, a more general condition is needed to check whether a given point is a candidate for the optimal solution. We introduce the **Lagrangian** function of the constrained optimization problem (2.1) to facilitate the discussion of this more general condition.

$$L(x, \mu, \sigma) = f(x) + \mu^\top g(x) + \sigma^\top h(x), \quad \mu \in \mathbb{R}^m_+,\ \sigma \in \mathbb{R}^p, \tag{2.2}$$

Intuition for Lagrange multipliers. In (2.2), $\mu \in \mathbb{R}^m_+$ and $\sigma \in \mathbb{R}^p$ are referred to as the Lagrange multipliers [19, Prop.1.29]. They penalize the original objective $f(x)$ when x violates $g(x) \leq 0$ or $h(x) = 0$. If the constrained optimization problem (2.1) satisfies known constraint qualifications [96], then there exists Lagrange multipliers, $\sigma^\star \in \mathbb{R}^p$ and $\mu^\star \in \mathbb{R}^m_+$, can be tuned to specific values $(\mu^\star, \sigma^\star)$ for which the optimization problem $\min_{x\in\mathbb{R}^n} L(x, \mu^\star, \sigma^\star)$ and the constrained optimization problem (2.1) coincide in the locally or globally optimal solution. Additionally, $\min_{x\in\mathbb{R}^n} L(x, \mu^\star, \sigma^\star)$'s optimal solution $x^\star$ necessarily obeys the original constraints $g(x^\star) \leq 0$ and $h(x^\star) = 0$. This intuition led to the development of the KKT conditions.

Definition 2.2 (Kahn-Karush-Tucker (KKT) conditions) *[19] The KKT conditions for the optimization problem* (2.1) *are defined for vectors* $x \in \mathbb{R}^n$*,* $\mu \in \mathbb{R}^m$*, and* $\sigma \in \mathbb{R}^p$*, and given by*

$$\begin{aligned} \frac{df}{dx}(x) + \mu^\top \frac{dg}{dx}(x) + \sigma^\top \frac{dh}{dx}(x) &= 0 \\ g(x) &\leq 0 \\ h(x) &= 0 \\ \mu &\geq 0 \\ \mu_i g_i(x) &= 0,\ \forall\ i \in [m]. \end{aligned} \tag{2.3}$$

Necessary conditions for constrained optimality. When the optimization objective and constraint functions are continuously differentiable, $f, g, h \in C^2(\mathbb{R}^n, \mathbb{R})$, and the optimization problem (2.1) satisfies certain constraint qualifications, Lagrange multipliers $\mu^\star$ and $\sigma^\star$ that satisfy the KKT conditions must exist at locally optimal and globally optimal solutions [19, Prop.1.29]. Of the many constraint qualifications that have been defined [105], the following ones address the optimization formulations found in this dissertation.

1. **Affine constraints** The constraints g and h are affine functions of x [14, Thm.10.5] .

2. **Convexity** The constraint function g and the objective f are convex in x, and the equality constraint function h is affine in x [22, Chp.5.5]. In particular, the KKT conditions are necessary and sufficient for optimality: there exist Lagrange multipliers $(\mu^\star, \lambda^\star)$ and a feasible vector $x^\star \in \mathbb{R}^n$ that satisfy the KKT conditions (2.3) if and only if $x^\star$ is the globally optimal solution.

Under these constraint qualifications, Lagrange multipliers can be found via the following min-max formulation.

$$\max_{\mu \in \mathbb{R}^m_+, \sigma \in \mathbb{R}^p} \min_{x \in \mathbb{R}^n} L(x, \mu, \sigma) \tag{2.4}$$

The inner minimization problem is a function of the Lagrange multipliers μ and σ, and is referred to as the dual function, $d : \mathbb{R}^m_+ \times \mathbb{R}^p \mapsto \mathbb{R}$, given by

$$d(\mu, \sigma) = \min_{x \in \mathbb{R}^n} L(x, \mu, \sigma), \;\; \mu \in \mathbb{R}^m_+, \sigma \in \mathbb{R}^p.$$

When the dual function d can be evaluated analytically, the dual form of constrained optimization problem (2.1) is given by

$$\max_{\mu \in \mathbb{R}^m_+, \sigma \in \mathbb{R}^p} d(\mu, \sigma). \tag{2.5}$$

In Section 2.2, we use the KKT conditions to derive the dual formulation of the Markov decision process, a framework for decision-making in stochastic environments. KKT conditions are also used in Markov decision processes to derive dynamic programming, where the value function corresponds to the Lagrange multipliers introduced in KKT.

2.1.1 Frank-Wolfe

The Frank-Wolfe algorithm is an iterative first-order gradient descent algorithm for optimization problems [62] with convex constraints. At each iteration, the algorithm minimizes the gradient of the linearized objective function over the feasible set to obtain the search direction. The next iterate is then obtained by moving along the search direction.

$$\begin{aligned} y^{k+1} &\in \underset{y\in\mathbb{R}^n,\ g(y)\leq 0,\ h(y)=0}{\operatorname{argmin}} \nabla f(x^k)^\top y \\ x^{k+1} &= (1-\alpha^k)x^k + \alpha^k y^{k+1} \end{aligned} \tag{2.6}$$

Frank-Wolfe can be leveraged to solve optimal control problems with complex objective functions. We note that the first step in (2.6) is equivalent to solving the optimization problem (2.1) with the objective replaced by its first-order approximation at x^k, $f(x) = f(x^k) + \nabla f(x^k)^\top (x - x^k)$. In subsequent sections, we apply this to game settings in which players follow Markov decision process dynamics. We leverage existing dynamic programming algorithms to solve Markov decision processes with coupled player costs. Frank-Wolfe under a *diminishing step size* has a sublinear convergence rate [62].

Proposition 2.1 *[55, Thm.1] When the objective function f in the optimization problem (2.1) is convex and continuously differentiable, constraint set $\mathcal{X} = \{x \in \mathbb{R}^n \mid g(x) \leq 0, h(x) = 0\}$ has diameter R[2], and the objective gradient ∇f is β-Lipschitz continuous, Frank-Wolfe (2.6) with a step size $\alpha^k = \frac{2}{k+2}$ converges to f's optimal solution, $x^\star$, as*

$$f(x^k) - f(x^\star) \leq \frac{2\beta R^2}{k+1},\ k \in \mathbb{N}.$$

2.2 Markov Decision Process

Markov decision processes (MDPs) [107] are widely used in artificial intelligence [135] and control theory [107, 109], due to their flexibility and ability to model a wide range of decision-making problems in uncertain and dynamic environments. In a discrete-time MDP, a decision maker selects an action from the current state, which then determines the transition probability distribution for the next state. This probabilistic transition model is known and only depends on the current state and action [107, Sec.2.1.3]. The decision maker's goal

[2]The diameter of set $\mathcal{X}$ is given by $\max_{x,y\in\mathcal{X}} \|x - y\|$.

is to minimize the expected cumulative cost obtained over a finite or infinite time horizon. Equivalently, the decision maker aims to find a policy, which maps states to actions, that minimizes the expected cumulative costs. When the cost and transition parameters of an MDP are unknown, the MDP becomes a reinforcement learning problem [126].

We focus on the discretized state-action MDP given by $([S], [A], P, C, \gamma, \mathcal{T})$, where $\gamma \in (0, 1)$ is the discount factor, $[S] = \{1, \ldots, S\}$ is the **finite set of states** and $[A] = \{1, \ldots, A\}$ is the **finite set of actions**, and $\mathcal{T} = \{0, \ldots, T\}$, where $T \in \mathbb{N} \cup \{\infty\}$, is the time horizon. Without loss of generality, assume that every action is admissible from each state $s \in [S]$.

1. **Costs**: The cost of taking an action a from state s at time step t is a realization of the random variable $c_t(s, a)$ that takes on values in real numbers. We assume that the random variable $c_t(s, a)$ has a well-defined expected value $\mathbb{E}[c_t(s, a)] = C_{tsa} \in \mathbb{R}$, which represents the average cost over all possible realizations of $c_t(s, a)$.

2. **Transition dynamics**. At each time step t in state s and while taking action a, the decision maker's next state is determined by a random variable $p_t(s, a)$ that takes on a value in the state space $[S]$. The probability distribution associated with $p_t(s_t, a_t)$ only depends on the current state and action, (s_t, a_t). We denote the probability of transitioning to state s' at $t+1$ by $P_{ts'sa} \in \mathbb{R}_+$. The transition dynamics satisfy simplex constraints at each time step and for each MDP state-action pair, given by

$$\sum_{s' \in [S]} P_{ts'sa} = 1, \; P_{t\hat{s}sa} \geq 0, \; \forall (t, \hat{s}, s, a) \in \{0, \ldots, T-1\} \times [S] \times [S] \times [A]. \tag{2.7}$$

3. **Policy**. The decision maker selects actions at each state via a policy: a state-dependent random variable $\pi(s)$ that takes on values in the action space. The action selected at time t at state s is given by $\pi_t(s)$ that takes on values from the action set $[A]$. We denote the conditional probability of choosing action a at (s, t) by $\pi_{tsa} \in \mathbb{R}_+$ and the probability distribution associated with the random variable $\pi_t(s)$ by $\pi_{ts} \in \Delta_A$. A policy is **deterministic** if at every state and time step (s, t), $\pi_{tsa} = 1$ for exactly one action and 0 for all other actions. A non-deterministic policy is a **stochastic** policy. A time-sequenced policy is given by $\pi := [\pi_0, \ldots, \pi_T]$.

4. **MDP Objective**. The decision maker's goal is to minimize the expected cost incurred over a time horizon $\mathcal{T}$. When initialized at state s and following the policy π, the

decision maker incurs an expected cumulative cost at time t,

$$V_{ts}(\pi) := \mathbb{E}\Big[\sum_{k=t}^{T}\gamma^k c_k\Big(s_k,\pi_k(s_k)\Big) \;\Big|\; s_0 = s, s_{k+1}\sim p_k\Big(s_k,\pi_k(s_k)\Big),\ \forall k = t,\dots,T\Big], \tag{2.8}$$

which we call the **value function** $V_{ts}(\pi)\in\mathbb{R}$.

When $T=\infty$, the infinite horizon value function is denoted as $V^{\infty}(\pi)\in\mathbb{R}^S$. The value function $V_s^{\infty}(\pi)$ exists if $\gamma<1$ and the cost, transition, and policies do not vary with time. In addition, we assume that the Markov chains generated by policy π are ergodic [107]—i.e., by following policy π, the decision maker asymptotically converges to a unique probability distribution over the state space $[S]$ and visits all states an infinite number of times, regardless of where the decision maker started.

2.2.1 Linear Program Formulation

An MDP can be formulated as a constrained optimization problem [42, Sec. 2.3], given by

$$\min_{\pi_t\in\Pi}\mathbb{E}\Big[\sum_{k=0}^{T}\gamma^k c_k\Big(s_k,\pi_k(s_k)\Big) \mid s_0\sim z, s_k\sim p_{k-1}\Big(s_{k-1},\pi_{k-1}(s_{k-1})\Big),\ \forall k\in[T]\Big],\ \Pi=\Delta_A^S. \tag{2.9}$$

If the decision maker's initial state probability distribution is given by $z\in\Delta_S$, the objective of the constrained optimization problem (2.9) is equivalent to $\sum_{s\in[S]} z_s V_{0s}(\pi)$ (2.8) when it exists. Using the Bellman principle [107, Sec.4.3], the value function $V_{ts}(\pi)$ can be recursively computed from the time step T as

$$\begin{aligned} V_{Ts}(\pi) &= \sum_{a\in[A]}\pi_{Tsa}C_{Tsa},\ \forall s\in[S],\\ V_{(t-1)s}(\pi) &= \sum_{a\in[A]}\pi_{(t-1)sa}\Big(C_{(t-1)sa}+\gamma\sum_{s'\in[S]}P_{(t-1)s'sa}V_{ts'}(\pi)\Big),\ \forall (t,s)\in[T]\times[S]. \end{aligned} \tag{2.10}$$

We can then reformulate the constrained optimization (2.9) as a linear program with value function as its decision variable.

$$\begin{aligned} \max_{V\in\mathbb{R}^{(T+1)S}}\ & \sum_{s\in[S]} z_{0s}V_{0s}\\ \text{s.t. }\ & V_{Ts}\le C_{Tsa},\ \forall s,a\in[S]\times[A]\\ & V_{(t-1)s}\le C_{(t-1)sa}+\gamma\sum_{s'\in[S]}P_{(t-1)s'sa}V_{ts'},\ \forall(t,s,a)\in[T]\times[S]\times[A], \end{aligned} \tag{2.11}$$

where $\pi = \pi_0, \ldots, \pi_T$ the policy is implicitly defined through the value function V as

$$\pi_{tsa} = \begin{cases} 1/|\underset{a' \in [A]}{\operatorname{argmin}} \; C_{tsa'} + \gamma \sum_{s'} P_{ts'sa'} V_{(t+1)s'}| & a \in \underset{a' \in [A]}{\operatorname{argmin}} \; C_{tsa'} + \gamma \sum_{s'} P_{ts'sa'} V_{(t+1)s'} \\ 0 & \text{o.w.} \end{cases}.$$

Remark 2.1 (Policy and value uniqueness) *The optimal objective value in the linear program formulation of the MDP (2.11) is unique while the corresponding optimal policy is not. The set of optimal policies always includes at least one deterministic stationary policy in the unconstrained setting [107, Thm 6.2.11]. If there are constraints on the policy and state space, deterministic policies may become infeasible [38].*

2.2.2 Dual Program Formulation

The linear programming formulation of MDP (2.11) can be interpreted as minimizing the expected cost incurred by a state and time-dependent policy, with the policy being the primary decision variable. However, as decision-makers share an operation space, knowing each other's position in addition to their policy is crucial to avoid collisions. Towards the goal of incorporating knowledge about each decision maker's position, we derive the dual formulation of the linear program in (2.5).

State-action probability distribution/occupancy measure. We consider an alternative to the policy as the primary decision variable: the state-action probability distribution, also known as the occupancy measure in MDP literature [8]. The state-action distribution $x \in \Delta_A^{(T+1)S}$ is defined such that each element x_{tsa} denotes the probability that the decision maker will be in state s and taking action a at time t.

The state-action probability distribution embeds more information than the policy. Given a policy π, let $z^\pi \in \Delta_S^{(T+1)}$ be the resulting probability distribution in the state space, i.e., z^π_{ts} is the probability that the decision maker is in state s at time t under the policy π. The corresponding state-action probability distribution x is given by

$$x_{tsa} = z^\pi_{ts} \pi_{tsa}, \forall \; (t, s, a) \in \mathcal{T} \times [S] \times [A].$$

If z^π is unique for a given policy π, then x is unique for π. Conversely, we can also compute the policy π that produces a state-action probability distribution x. Given a state-action

probability distribution x, one possible policy is given by

$$\pi_{tsa} = \begin{cases} \frac{x_{tsa}}{\sum_{a' \in [A]} x_{tsa'}} & \sum_{a' \in [A]} x_{tsa'} > 0 \\ \frac{1}{A} & \text{otherwise} \end{cases}, \quad \forall (t, s, a) \in \mathcal{T} \times [S] \times [A]. \tag{2.12}$$

Dual formulation of MDP. The state-action probability distribution combines both the probability of being in a state and the probability of taking an action from a given state. The set of all feasible state-action probability distributions under transition probabilities P (2.7) and initial probability distribution $z \in \Delta_S$ is given by

$$\mathcal{X}(P, z) := \left\{ x \in \Delta_A^{(T+1)S} \;\middle|\; \sum_{a \in [A]} x_{(t+1)sa} = \gamma \sum_{s' \in [S]} \sum_{a \in [A]} P_{ts'sa} x_{ts'a}, \; \sum_a x_{0sa} = z_s, \; \forall \; (t, s) \in [T] \times [S] \right\}. \tag{2.13}$$

We can show that the linear programming formulation of an MDP (2.11) [107, Sec.8.9] is in fact the dual formulation of the following optimization problem with the state-action density as the primary decision variable and constrained to $\mathcal{X}(P, z)$ (2.13).

$$\begin{aligned} \min_{x \in \mathbb{R}_+^{(T+1)SA}} & \sum_{t \in \mathcal{T}} \sum_{s \in [S]} \sum_{a \in [A]} x_{tsa} C_{tsa} \\ \text{s.t. } & x \in \mathcal{X}(P, z_0). \end{aligned} \tag{2.14}$$

Proposition 2.2 *When $T < \infty$, the linear program (2.5) produces the dual linear program (2.11).*

Proof: We first write out the Lagrangian function (2.2) of the linear program (2.14) by assigning the Lagrange multiplier ν_{ts} for the constraints $\sum_a x_{tsa} - \gamma \sum_{s',a} P_{tss'a} x_{(t-1)s'a}$, ν_{0s} for the constraints $\sum_a x_{0sa} - z_{0s}$, and μ_{tsa} for the constraints $x_{tsa} \geq 0$. The resulting Lagrangian is

$$\begin{aligned} L(x, \nu, \mu) = \sum_{t \in \mathcal{T}} \sum_s \sum_a x_{tsa} C_{tsa} + \sum_{t=1}^{T} \sum_s \nu_{ts} (\gamma \sum_{s'a} P_{tss'a} x_{(t-1)s'a} - \sum_a x_{tsa}) \\ + \sum_s \nu_{0s} (z_{0s} - \sum_a x_{0sa}) - \sum_{t \in \mathcal{T}} \sum_s \sum_a \mu_{tsa} x_{tsa}. \end{aligned} \tag{2.15}$$

Since the optimization problem (2.11) is a linear program over a compact set of solution

variable V, it has an optimal solution, and that solution must satisfy the KKT conditions. We write out the KKT conditions applied to the Lagrangian (2.15).

$$\frac{\partial L}{\partial x_{0sa}} = C_{0sa} + \gamma \sum_{s'} P_{0s'sa}\nu_{1s'} - \nu_{0s} - \mu_{0sa} = 0, \quad \forall\ (s,a) \in [S] \times [A], \tag{2.16a}$$

$$\frac{\partial L}{\partial x_{tsa}} = C_{tsa} + \gamma \sum_{s'} P_{ts'sa}\nu_{(t+1)s'} - \nu_{ts} - \mu_{tsa} = 0, \quad \forall (t,s,a) \in \{1,\dots,T-1\} \times [S] \times [A], \tag{2.16b}$$

$$\frac{\partial L}{\partial x_{Tsa}} = C_{Tsa} - \nu_{Ts} - \mu_{Tsa} = 0, \quad \forall (s,a) \in [S] \times [A], \tag{2.16c}$$

$$\sum_{a} x_{tsa} = \gamma \sum_{s',a} P_{tss'a} x_{(t-1)s'a}, \quad \forall (t,s) \in \{0,\dots,T-1\} \times [S], \tag{2.16d}$$

$$x_{tsa} \geq 0, \quad \forall (t,s,a) \in \mathcal{T} \times [S] \times [A], \tag{2.16e}$$

$$\mu_{tsa} \geq 0, \quad \forall (t,s,a) \in \mathcal{T} \times [S] \times [A], \tag{2.16f}$$

$$\mu_{tsa} x_{tsa} = 0, \quad \forall (t,s,a) \in \mathcal{T} \times [S] \times [A]. \tag{2.16g}$$

From complementary slackness properties of μ, we can derive the following recursive relationship on ν for $t = 0, \dots T-1$,

$$\begin{cases} \nu_{ts} = C_{tsa} + \gamma \sum_{s'} P_{ts'sa}\nu_{(t+1)s'} & x_{tsa} > 0 \\ \nu_{ts} < C_{tsa} + \gamma \sum_{s'} P_{ts'sa}\nu_{(t+1)s'} & x_{tsa} = 0 \end{cases}, \quad \forall t = 0, \dots, T-1, \tag{2.17}$$

$$\begin{cases} \nu_{Ts} = C_{Tsa} & x_{Tsa} > 0 \\ \nu_{Ts} < C_{Tsa} & x_{Tsa} = 0 \end{cases}, \quad t = T. \tag{2.18}$$

From here, we can derive the dual formulation with primal variable $V = \nu$ and by adding $\sum_{t \in \mathcal{T} s \in [S], a \in [A]} x_{tsa} \frac{\partial L}{\partial x_{tsa}}$ back into the Lagrangian (2.15), exactly as in (2.11) which we recall below.

$$\begin{aligned} \max_{V \in \mathbb{R}^{(T+1)S}} & \sum_{s \in [S]} z_{0s} V_{0s} \\ \text{s.t. } & V_{Ts} \leq C_{Tsa}, \ \forall s, a \in [S] \times [A] \\ & V_{(t-1)s} \leq C_{(t-1)sa} + \gamma \sum_{s' \in [S]} P_{(t-1)s'sa} V_{ts'}, \ \forall (t,s,a) \in [T] \times [S] \times [A]. \end{aligned}$$

■

In the infinite time horizon, $T = \infty$, a joint state-action probability distribution is feasible if only if it is *stationary* for the policy-induced Markov chain [107] and does not depend on the initial distribution under the ergodic assumption. We denote this set of feasible state-action distributions as $\mathcal{X}_\infty(P)$.

$$\mathcal{X}_\infty(P) = \Big\{ x \in \Delta_A^S \;\Big|\; \sum_{a\in[A]} x_{sa} = \gamma \sum_{s'\in[S]} \sum_{a\in[A]} P_{ss'a} x_{s'a},\ \forall\ s \in [S] \Big\}. \tag{2.19}$$

2.2.3 Dynamic Programming

Among MDP solution methods, value iteration is a dynamic programming algorithm that is commonly used.

Finite horizon value iteration. Value iteration in the finite horizon requires evaluating the *Q-value* [107] of individual state-actions.

$$\begin{aligned} Q_{Tsa} &:= C_{Tsa},\ \forall\ (s,a) \in [S]\times[A] \\ Q_{(t-1)sa} &:= C_{(t-1)sa} + \textstyle\sum_{s'} P_{(t-1)s'sa} \min_{a'} Q_{ts'a'},\ \forall\ (t,s,a) \in [T]\times[S]\times[A]. \end{aligned} \tag{2.20}$$

Value iteration in the finite horizon setting can be derived from the KKT conditions and converges to the optimal value functions [107, Thm.4.3.2]—i.e., $V_{ts}^\star = \min_{a\in[A]} Q_{tsa}$ for all $(t,s) \in \mathcal{T}\times[S]$.

Infinite horizon value iteration. For discounted infinite horizon MDPs with stationary costs and dynamics, the optimal value function for (2.11) is the fixed point of the Bellman operator, a value function vector $V^\star \in \mathbb{R}^S$ such that

$$V_s^\star = \min_{a\in[A]}\ C_{sa} + \gamma \sum_{s'\in[S]} P_{s'sa} V_{s'}^\star, \forall s \in [S]. \tag{2.21}$$

Proposition 2.3 *[107, Thm.6.2.5] For discounted infinite horizon MDPs, where $S < \infty$, and $C_{sa} < \infty$ for all $(s,a) \in [S]\times[A]$,*

1. *There exists $V^\star \in \mathbb{R}^S$, such that (2.21) holds.*
2. *$V^\star \in \mathbb{R}^S$ is the optimal value function of the optimization problem (2.11) under constraints $\mathcal{X}_\infty(P)$ (2.19).*

2.3 Game Theory

Game theory is a branch of mathematics that studies strategic interactions among multiple decision-makers. It provides a powerful framework for analyzing decision-making processes where the outcomes of one decision-maker's actions depend on the actions of others. In the context of this thesis, game theory is used to analyze the interaction between mobile autonomous agents in a resource-sharing environment. Specifically, we use game-theoretical notions to model agent interactions and analyze the interaction outcome. In this section, we summarize the non-cooperative game under a complete information framework.

2.3.1 Atomic Game with Continuous Action Spaces

In an atomic finite action game [111], a finite number of players $[N]$ each optimizes their own utility functions via their own actions, yet every player's utility function depends on all players' actions. Player i can select from a set of actions, denoted as $\mathcal{A}_i$, and the joint action space is given by $\mathcal{A} = \Pi_{i\in[N]}\mathcal{A}_i$. We assume that each player is selfish and only interested in minimizing his or her own utility function $f_i : \mathcal{A} \mapsto \mathbb{R}$. Each player's coupled optimization problem is given by

$$\min_{a_i\in\mathcal{A}_i} \; f_i(a_i, a_{-i}), \; \forall i \in [N]. \tag{2.22}$$

Since each player's utility depends on the joint action, the simplest preferable outcome for all players is to find a joint action in which everyone has minimized their individual utility under the assumption that the rest of the players will not change their current actions. This joint action is the *Nash equilibrium* of the game. A Nash equilibrium extends the concept of optimality for (2.1) in the sense that one joint strategy now simultaneously optimizes N different utility functions (2.22).

Definition 2.3 (Nash equilibrium) *A joint action $a^\star \in \mathcal{A}$ is a **local Nash equilibrium** if for every player $i \in [N]$, there exists an open set $\mathcal{M}_i \subseteq \mathcal{A}_i$ such that each $a_i^\star \in \mathcal{M}_i$, and*

$$f_i(a_i^\star, a_{-i}^\star) \leq f_i(a_i, a_{-i}^\star), \quad \forall \; a_i \in \mathcal{M}_i. \tag{2.23}$$

*Moreover, if $\mathcal{M}_i = \mathcal{A}_i$ for all $i \in [N]$, then $a^\star$ is a **global Nash equilibrium**.*

At Nash equilibrium, no player has the incentive to deviate from their current strategy if all other players do not change their strategies.

Continuous games We consider a class of games in which each player's pure strategy is an element of a finite-dimensional space, $\mathcal{A}_i \subset \mathbb{R}^{m_i}, \forall i \in [N]$, and each player's utility function is twice differentiable over the joint action space, $f_i \in C^2(\mathbb{R}^m, \mathbb{R})$. Equivalently, players have a continuum and an infinite number of actions to select from. The joint action space is denoted as $\mathcal{A} = \mathbb{R}^m = \mathbb{R}^{m_1} \times \ldots \times \mathbb{R}^{m_N}$.

For continuous games, the set of Nash equilibria is equivalent to the set of joint actions whose player cost gradients and Hessians satisfy the following necessary and sufficient conditions.

Proposition 2.4 (Necessary and sufficient conditions for Nash equilibria) *[110] If the joint strategy $x = (x_1, \ldots, x_N)$ is a local Nash equilibrium, then*

$$\omega(x) = \Big[\partial f_1(x)/\partial x_1, \ldots, \partial f_N(x)/\partial x_N\Big] = 0, \;\; \partial f_i^2(x)/\partial x_i^2 \succ 0, \;\; \forall i \in [N].$$

Continuous game models are used in Chapter 5 to understand how disturbances in players' gradients affect the stability of other players in gradient-based learning dynamics.

2.4 Non-atomic MDP Congestion Games

In non-atomic MDP congestion games, a continuum of players competes for congested resources under discrete state-action MDP dynamics [27].

In contrast to atomic games in which the player set is finite and each player's dynamics and objectives are unique, non-atomic games feature players who are identical in their transition dynamics and costs. Instead of having a joint action, the player population action profile within non-atomic games is characterized by a continuous distribution where an individual player's action contributes to the zero-measure subset of the joint action distribution (see [59, Sec.2] for details). Non-atomic games are particularly useful for computing and analyzing the trends of a large group of decision makers. In addition to the MDP congestion game that we describe below, non-atomic games encompass routing games [114], continuous stochastic games [125], and mean field games [64].

Consider a continuous population of players, each with identical MDP dynamics and congestion costs over a state-action set $[S] \times [A]$ for $(T+1)$ time steps. The players have a total population mass M and an initial population distribution $p_0 \in \Delta_S$ over the state space. All players follow MDP transition dynamics given by P (2.7). The set of feasible population

distributions is given by

$$\mathcal{Y}(P, p_0) = \left\{ y \in \mathbb{R}_+^{(T+1)SA} \middle| \sum_a y_{tsa} = \sum_{s',a} P_{(t-1)ss'a} y_{(t-1)s'a}, \sum_a y_{0sa} = p_{0s}, \ \forall (t,s) \in [T] \times [S] \right\}, \tag{2.24}$$

where y_{tsa} is the portion of the playing population that takes action a from state s at time t. We emphasize that y is a vector in $\mathbb{R}_+^{(T+1)SA}$ whose coordinates are ordered as

$$y = \begin{bmatrix} y_{000} & \cdots & y_{010} & \cdots & y_{100} & \cdots & y_{T(S-1)(A-1)} \end{bmatrix}^\top. \tag{2.25}$$

We allow $T = \infty$. Similar to infinite horizon MDPs, the set of feasible population distributions is only those that are stationary for the resulting Markov chain.

$$\mathcal{Y}_\infty(P) = \left\{ y \in \mathbb{R}_+^{SA} \middle| \textstyle\sum_a y_{sa} = \sum_{s',a} P_{ss'a} y_{s'a} \right\}, \tag{2.26}$$

Player costs. At the time t, each player incurs a cost as a function of the population distribution y, given by $\ell_{tsa} : \mathbb{R}^{(T+1)SA} \to \mathbb{R}$. We collect ℓ_{tsa} into a cost vector $\ell(y) \in \mathbb{R}^{(T+1)SA}$ under the same ordering as y in (2.25).

Non-stationary Q-value iteration. Similar to MDP literature, a player's expected cost-to-go at (t, s, a) is its Q-value function, recursively defined as

$$Q_{tsa}(y) = \begin{cases} \ell_{tsa}(y) & t = T \\ \ell_{(t-1)sa}(y) + \gamma \sum_{s' \in [S]} P_{(t-1)s'sa} \min_{a' \in [A]} Q_{ts'a'}(y) & t \in [T] \end{cases}, \ \forall s, a \in [S] \times [A] \tag{2.27}$$

When $T = \infty$, the Q-value functions are vector-valued functions that satisfy the following condition.

$$Q_{sa}(y) = \ell_{sa}(y) + \gamma \sum_{s' \in [S]} P_{s'sa} \min_{a' \in [A]} Q_{s'a'}(y), \ \forall s, a \in [S] \times [A]. \tag{2.28}$$

Non-atomic games have historically been used in traffic assignment research for modeling competitive behavior in transportation systems [104], where the equilibrium is typically referred to as a Wardrop equilibrium. In the case of non-atomic MDP congestion games, while the Wardrop equilibrium and the Nash equilibrium evaluate as the same, we adopt the convention used in traffic assignment literature and refer to the population distribution at Nash equilibrium as Wardrop equilibrium.

Definition 2.4 (MDP Wardrop Equilibrium [27]) *A population distribution $y^\star \in \mathcal{Y}(P, p_0)$ (2.24), is an* MDP Wardrop equilibrium *for the MDP congestion game if its Q-value function (2.27) satisfies*

$$y^\star_{tsa} > 0 \Rightarrow Q_{tsa}(y^\star) = \min_{a' \in [A]} Q_{tsa'}(y^\star), \ \forall (t, s, a) \in \mathcal{T} \times [S] \times [A]. \tag{2.29}$$

When $T = \infty$, $y^\star \in \mathcal{Y}(P)$ (2.26) is an MDP Wardrop equilibrium if its Q-value function (2.28) satisfies

$$y^\star_{sa} > 0 \Rightarrow Q_{sa}(y^\star) = \min_{a' \in [A]} Q_{sa'}(y^\star), \ \forall (s, a) \in [S] \times [A]. \tag{2.30}$$

At MDP Wardrop equilibrium, every positive portion of the decision-making population exclusively chooses actions that incur the minimum expected cost-to-go.

An MDP congestion game has a unique MDP Wardrop equilibrium for the class of strictly increasing congestion cost functions.

Assumption 2.1 *$\ell_{tsa} : \mathbb{R} \mapsto \mathbb{R}$ is a strictly increasing function of y_{tsa} for all $(t, s, a) \in \mathcal{T} \times [S] \times [A]$.*

In agreement with the colloquial definition of congestion, the cost in each time-state-action triplet increases as more decision makers choose the triplet through their policy. When the player cost functions satisfy Assumption 2.1, the MDP congestion game can be characterized as a *potential game*. In Chapter 3, we show that other classes of congestion costs can also generate a potential game.

Definition 2.5 (Potential game [117, 27]) *We say that the MDP congestion game is a* potential game *if there exists a continuously differentiable function $F : \mathbb{R}^{(T+1)\times S\times A} \mapsto \mathbb{R}$ such that $\partial F(y)/\partial y_{tsa} = \ell_{tsa}(y)$ for all $(t, s, a) \in \mathcal{T} \times [S] \times [A]$ and $y \in \mathcal{Y}(P, p_0)$ (2.24).*

While a potential function may not always exist for an arbitrary congestion cost, multi-agent systems in which the players compete for a shared resource (space, demand) are likely to have a potential function. We show in Chapter 3 and Chapter 4 scenarios in air traffic management, warehouse logistics, and transportation in which we can explicitly derive a potential function. Furthermore, we show that *nonconvex* potential games can be used to model competitive air traffic routing.

Connection to stochastic games. MDP congestion games and stochastic games are both multi-player extensions of MDP via its linear program formulation (2.11). The coupling

quantities between players, the *population distribution* in MDP congestion games, and the *joint policy* in stochastic games [122], are the primal and dual variables of the MDP [107, Eqn 6.9.2], respectively. In stochastic games, each player's cost function depends on the joint policy π. On the other hand, MDP congestion games model congestion effects more accurately by defining the congestion cost as a function of the population distribution y. This key difference allows MDP congestion games to avoid superficially inflating congestion costs, which can happen in stochastic games when a state-action pair (s, a) is assigned a high cost despite the population having a low probability of being in state s. In MDP congestion games, high congestion costs are assigned only when both the portion of players in state s and the likelihood of these players taking action a are high. This leads to more precise and realistic modeling of congestion effects in air traffic and ground transportation.

Chapter 3

MARKOV GAMES WITH INDEPENDENT TRANSITION DYNAMICS

As autonomous path planning algorithms become widely adapted in aerospace and robotics [144, 99], the standard assumption that the operating environment is stationary is no longer sufficient. More likely, autonomous vehicles *share* the operating environment with other vehicles that may have conflicting objectives. For single-vehicle planning without inter-vehicle coordination, the possibility of collision has led to a greater emphasis on the algorithm's obstacle avoidance and robustness properties. However, assuming that a centralized communication framework exists for autonomous vehicles, the overarching goal should be to coordinate each vehicle path to achieve simultaneous optimality. We are motivated by applications such as air traffic management [20] and warehouse package retrieval [61].

We focus on settings in which a group of heterogeneous players plan paths in a shared environment and are collectively influenced by the same exogenous random variables. For example, fleets of robo-taxis fulfilling ride demands while avoiding congestion in traffic [137], warehouse robots retrieving packages that arrive at uncertain times [61, 76], and commercial aircraft attempting to stick to scheduled paths subjected to weather and passenger loading uncertainty. The common feature in these applications is that each vehicle has an independent goal subjected to some exogenous random variable, yet competes for space in a shared operation environment with other vehicles. We assume that the desired outcome is a competitive equilibrium: every player has an optimal path for completing their independent task in the shared operation space.

In this chapter, these path coordination problems are analyzed under the Markov game framework. We extend the existing MDP congestion game framework to the atomic setting for a finite number of players and non-symmetric game Jacobians. For the potential game setting, we show that both convex and nonconvex potential functions arise in air traffic and warehouse routing, and derive equivalence between KKT points and stationary points of coupled Q-value iteration.

Contribution summary. We propose a Markov game with finite players and independent

player probability distributions. We define dynamic programming-type optimality conditions for the Nash equilibrium and provide necessary and sufficient conditions for its existence for different subsets of game functions. For the multi-player path coordination problem, we provide examples of player cost functions that result in jointly optimal paths that minimize the probability of collision between players. Finally, we provide a distributed algorithm that converges to the Nash equilibrium and give its convergence rate. We demonstrate our model and algorithm on a two-dimensional autonomous warehouse navigation problem where robots retrieve and deliver packages with stochastic arrival times while sharing a common navigation space, and an air traffic management example in which each flight must minimize its deviation from pre-scheduled flight plan while avoiding collision in shared air spaces.

3.1 Atomic Markov Game with Independent State Transitions

In this section, we formulate an atomic Markov game that extends individual MDPs from Section 2.2, prove necessary and sufficient conditions for quantifying its Nash equilibrium via non-stationary Q-value iteration, solve for the Nash equilibrium via Frank-Wolfe algorithm, and provide application examples in air traffic management and warehouse logistics.

We consider N players each solving an MDP $([S], [A], \mathcal{T}, P^i, \ell^i, p_0^i)$ for $i \in [N]$. Each player controls their own location situated within the common state space $[S]$, accesses a common set of actions $[A]$ at each state, and solves an MDP with time horizon $\mathcal{T} = \{0, \dots, T\}$. Player i's state transitions are controlled by player i's actions and are denoted by $P^i \in \Delta_S^{TSA}$. Player i's state-action costs are coupled with the opponent state-action distributions via ℓ^i, described in detail below.

Joint state-action distribution. We collectively denote all players' state-action distributions as

$$x = (x^1, \dots, x^N) \in \Delta_A^{N(T+1)S}. \tag{3.1}$$

Each player i's state-action distribution is defined by (2.13). We assume that x is fully observable and may denote it as $x = (x^i, x^{-i})$ where $x^{-i} = (x^j)_{j \in [N]/\{i\}}$ to emphasize the strategies of player i's opponents. Player i's state-action distribution can be extracted from player i's policy $\pi^i \in \Delta_A^{(T+1)S}$ via (2.12).

Player costs. Also referred to as state-action costs, the player costs are defined for each time-state-action triplet as continuously differentiable *functions* of the joint state-action

distribution x, given in vector form as

$$\ell^i : \Delta_A^{N(T+1)S} \mapsto \mathbb{R}^{(T+1)SA}, \ i = 1, \ldots, N. \tag{3.2}$$

Player i's MDP cost function at time $t \in \mathcal{T}$, state $s \in [S]$, and action $a \in [A]$ is specifically referred to as $\ell^i_{tsa} : \Delta_A^{N(T+1)S} \mapsto \mathbb{R}$, and all players' state-action costs as a vectored function $\xi : \Delta_A^{N(T+1)S} \mapsto \mathbb{R}^{N(T+1)SA}$, sequentially ordered by increasing action, state, time, and player number, in that order, as

$$\xi(x) = [\ell^1_{011}(x), \ell^1_{012}(x), \ldots, \ell^N_{TSA}(x)] \in \mathbb{R}_+^{N(T+1)SA}. \tag{3.3}$$

We assume that each player's state-action cost ℓ^i is the gradient of a continuously differentiable function.

Assumption 3.1 (Objective existence) *For each player $i \in [N]$, there exists a continuously differentiable function $F^i : \Delta_A^{N(T+1)S} \mapsto \mathbb{R}$, such that $\partial F^i(x)/\partial x^i_{tsa} = \ell^i_{tsa}(x)$, for all $(t, s, a) \in \mathcal{T} \times [S] \times [A]$ and $x \in \Delta_A^{N(T+1)S}$.*

Remark 3.1 *Assumption 3.1 is analogous to assuming objective differentiability in games with continuous action spaces [128]. If each x^i is treated as a continuous playing population, then Assumption 3.1 is identical to the existence of a potential function for each population [32].*

Individual MDP. Under Assumption 3.1, players solve MDPs with coupled objectives and independent transitions, denoted as

$$\min_{x^i} F^i(x^i, x^{-i}) \text{ s.t. } x^i \in \mathcal{X}(P^i, p_0^i), \ \forall i \in [N]. \tag{3.4}$$

Remark 3.2 *We make a distinction between the player's MDP objective, F^i, and the player's MDP state-action costs, ℓ^i. The objective is the overall function being minimized by player i, while the state-action costs are the stage-wise incurred costs for being at a state-action pair. Under Assumption 3.1, $\partial F^i/\partial x^i = \ell^i$. In a single-player finite horizon MDP, the objective is given by $\sum_{t,s,a} C_{tsa} x_{tsa}$ and the state-action costs are C_{tsa} at each $(t, s, a) \in \mathcal{T} \times [S] \times [A]$. The optimization problem is explicitly formulated in (2.14).*

Best response. When player i's opponents select strategies that result in the state-action distribution x^{-i}, player i's *best response* state-action distribution is given by the set $\operatorname{argmin}_{u^i \in \Delta_A^{(T+1)S}} F^i(u^i, x^{-i})$. If all players simultaneously select the best response, the corresponding joint state-action distribution is a *Nash equilibrium* (Definition 2.3). We recall Nash equilibrium from Definition 3.1 for Markov games below.

Definition 3.1 (Nash equilibrium) *The joint state-action distribution $x^\star \in \prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$ is a **local Nash equilibrium** if there exists an open set $\mathcal{X}_i \subseteq \mathcal{X}(P^i, p_0^i)$ such that $x^{i\star} \in \mathcal{X}_i$, and*

$$F^i(x^{i\star}, x^{(-i)\star}) \leq F^i(x^i, x^{(-i)\star}), \quad \forall\ x^i \in \mathcal{X}^i \subseteq \Delta_A^{(T+1)SA},\ i = 1, \ldots, N. \tag{3.5}$$

*If $\mathcal{X}^i = \mathcal{X}(P^i, p_0^i)$ for all $i \in [N]$, then $x^\star$ is a **global Nash equilibrium**.*

Definition 3.1 adapts the standard Nash optimality framework from Definition 2.3 to a Markov environment via the dual formulation of MDPs (2.14).

Coupled Q-value functions. In MDP congestion games, we can evaluate whether the playing population taking optimal strategies by checking the associated Q-value functions $Q \in \mathbb{R}^{(T+1)SA}$ (2.20) against the optimality condition (2.29). Similarly, given a joint state-action distribution $x = (x^1, \ldots, x^N)$, we want to be able to qualify it as a Nash equilibrium using the corresponding Q-value functions $Q^i(x) \in \mathbb{R}^{(T+1)SA}$. For Markov games of the form (3.4), each player i's Q-value at joint state-action x is given by

$$\begin{aligned} Q^i_{Tsa}(x) &:= \ell^i_{Tsa}(x),\ \forall\ (i,s,a) \in [N]\times[S]\times[A] \\ Q^i_{(t-1)sa}(x) &:= \ell^i_{(t-1)sa}(x) + \textstyle\sum_{s'} P^i_{(t-1)s'sa} \min_{a'} Q^i_{t,s'a'}(x),\ \forall\ (i,t,s,a) \in [N]\times[T]\times[S]\times[A]. \end{aligned} \tag{3.6}$$

In (3.6), player i's Q-value changes both when its opponents change their policies and when player i changes its policy. This is a significant departure from the non-atomic setting in which the playing population is represented via a continuous distribution in Section 2.4.

Q-value optimality. For a continuous population distribution y and corresponding Q-values $Q(y)$ (2.28), recall that y is the MDP Wardrop equilibrium (Definition 2.4) if and only if $Q(y)$ satisfies (2.23) . The analogous Q-value optimality condition for atomic players is given by

$$x^i_{tsa} > 0 \Rightarrow a \in \operatorname*{argmin}_{a'\in[A]} Q^i_{tsa'}(x),\ \forall (i,t,s,a) \in [N]\times\mathcal{T}\times[S]\times[A]. \tag{3.7}$$

Unlike the non-atomic Markov games, having optimal Q-values is not equivalent to achieving

Nash equilibrium. In the following section, we will show that for specific player cost structures, the equivalence will hold. However, for all player costs that satisfy Assumption 3.1, Q-value optimality (3.7) remains a necessary condition for achieving Nash equilibrium.

Proposition 3.1 *When the player costs ξ (3.3) satisfy Assumption 3.1, $Q^i(x)$ satisfies Q-value optimality for $i = 1, \dots, N$ if and only if the corresponding the joint state-action distribution $x = (x^1, \dots, x^N)$ (3.1) satisfies the KKT conditions (2.3) of player i's MDP (3.4) for $i = 1, \dots, N$ with the appropriate Lagrange multipliers.*

Proof: Under Assumption 3.1, $\partial F^i(x)/\partial x^i = \ell^i(x)$ for all $i \in [N]$. Assign the dual variables $\mu^i \in \mathbb{R}_+^{(T+1)SA}$ for $x^i \geq 0$ and $\nu^i \in \mathbb{R}^{(T+1)SA}$ for the equality constraints in $\mathcal{X}(P^i, p_0^i)$ (2.13). The Lagrangian of (3.4) for player i is given by

$$L(x^i, \nu^i, \mu^i) = F^i(x^i, x^{-i}) - \sum_{t,s,a} \mu^i_{tsa} x^i_{tsa} + \sum_s \nu^i_{0s}(x^i_{0s} - \sum_a x^i_{0sa}) + \sum_{s,t} \nu^i_{ts}(\sum_{s'a} P^i_{(t-1)ss'a} x^i_{(t-1)sa} - \sum_a x^i_{tsa}).$$

The KKT conditions are 1) primal feasibility $x^i \in \mathcal{X}(P^i, p_0^i)$, 2) dual feasibility $\mu^i \geq 0$, 3) complementary slackness $\mu^i_{tsa} x^i_{tsa} = 0$, $\forall (t, s, a) \in \mathcal{T} \times [S] \times [A]$, and 4) stationarity condition (Definition 2.2). In particular, the stationary condition can be derived for all $(t, s, a) \in \mathcal{T} \times [S] \times [A]$ as

$$\begin{cases} \ell^i_{tsa}(x^i, x^{-i}) + \sum_{s'} P^i_{ts'sa} \nu^i_{(t+1)s'} = \nu^i_{ts} + \mu^i_{tsa} & t \neq T, \\ \ell^i_{Tsa}(x^i, x^{-i}) = \nu^i_{Ts} + \mu^i_{Tsa} & t = T. \end{cases} \tag{3.8}$$

We show that the KKT conditions (3.8) are equivalent to the Q-value definition (3.6) by showing that one implies the other. To simplify notation, we use Q^i_{tsa} to denote $Q^i_{tsa}(x)$.

($\Rightarrow$): suppose (x^i, ν^i, μ^i) satisfy the KKT conditions. When $x^i_{tsa} > 0$, ν^i_{ts} represents the value function and $\nu^i_{ts} + \mu^i_{tsa}$ represents Q-value. When $x^i_{tsa} = 0$, we *shift* (ν^i, μ^i) to $(\hat{\nu}^i, \hat{\mu}^i)$ to generate the optimal Q-values. To this end, define $\lambda^i \in \mathbb{R}^{(T+1)SA}$, $\Delta^i \in \mathbb{R}^{(T+2)S}$, $\hat{\mu}^i \in \mathbb{R}^{(T+1)SA}$, $\hat{\nu}^i \in \mathbb{R}^{(T+1)S}$ recursively backwards in time from $t = T$. At $T = t$, let $\Delta^i_{(T+1)s'} = 0 \ \forall s' \in [S]$. All other variables are recursively defined as

$$\lambda^i_{tsa} = \mu^i_{tsa} + \textstyle\sum_{s'} P^i_{ts'sa} \Delta^i_{(t+1)s'}, \quad \Delta^i_{ts} = \min_{a'} \lambda^i_{tsa'}, \qquad \hat{\mu}^i_{tsa} = \lambda^i_{tsa} - \Delta^i_{ts}, \quad \hat{\nu}^i_{ts} = \nu^i_{ts} + \Delta^i_{ts}. \tag{3.9}$$

The variables $\lambda^i, \Delta^i, \hat{\mu}^i, \hat{\nu}^i$ have the following recursive property: if $x^i_{tsa} > 0$ implies $\lambda^i_{tsa} = 0$ at time t for all $(s, a) \in [S] \times [A]$, then $x^i_{(t-1)sa} > 0$ implies that $\lambda^i_{(t-1)sa} = 0$ for all $(s, a) \in$

$[S] \times [A]$. To see this: from complementary slackness, $x^i_{(t-1)sa} > 0$ implies $\mu^i_{(t-1)sa} = 0$. Subsequently, $\lambda^i_{(t-1)sa} = 0$ (3.9) if $P^i_{(t-1)s'sa}\Delta^i_{ts'} = 0\ \forall s' \in [S]$. At $s' \in [S]$, we can show that either $P^i_{(t-1)s'sa} = 0$ or $\Delta^i_{ts'} = 0$. Suppose $P^i_{(t-1)s'sa} > 0$, under the original assumption that $x^i_{(t-1)sa} > 0$, $P^i_{(t-1)s'sa}x^i_{(t-1)sa} > 0$. Since $\sum_{a' \in [A]} x^i_{ts'a'} = \sum_{a,s} P^i_{(t-1)s'sa}x^i_{(t-1)sa} > 0$, there exists $a' \in [A]$ such that $x^i_{ts'a'} > 0$. From the induction condition, $x^i_{ts'a'} > 0$ implies that $\lambda^i_{ts'a'} = 0$. Then, from its definition (3.9), $\Delta^i_{ts'} \leq 0$. If $\Delta^i_{(t+1)s'} \geq 0$, then $\lambda^i_{ts'a} \geq 0$, therefore, $\Delta^i_{ts'}$ cannot be negative and $\Delta^i_{ts'} = 0$. This shows that $P^i_{(t-1)s'sa}\Delta^i_{ts'} = 0\ \forall s' \in [S]$.

At $t = T$, $x^i_{Tsa} > 0$ implies $\mu^i_{Tsa} = 0$ and $\lambda^i_{Tsa} = 0$. Therefore, the recursive property that $x^i_{tsa} > 0$ implies $\lambda^i_{tsa} = 0$ holds for all $t \in \mathcal{T}$.
We add $\sum_{s'} P^i_{(t+1)s'sa}\Delta^i_{(t+1)s'}$ to (3.8), simplify it via (3.9), and obtain

$$\begin{cases} \ell^i_{tsa}(x) + \sum_{s'} P^i_{ts'sa}\hat{\nu}^i_{(t+1)s} = \hat{\nu}^i_{ts} + \hat{\mu}^i_{tsa} & t \in [T] \\ \ell^i_{Tsa}(x) = \hat{\nu}^i_{Ts} + \hat{\mu}^i_{Tsa} & t = T. \end{cases} \tag{3.10}$$

We define $Q^i_{tsa} = \hat{\nu}^i_{ts} + \hat{\mu}^i_{tsa}$. From (3.9), $\hat{\mu}^i_{tsa}$ is always non-negative, and $\hat{\mu}^i_{tsa'} = 0$ for some $a' \in [A]$. Therefore $\min_{a'} Q^i_{tsa'} = \hat{\nu}^i_{ts}$, and Q^i substituted in (3.10) is equivalent to the Q-value definition (3.6).

($\Leftarrow$): given Q-values $Q^i(x)$ that satisfy (3.7) for $i = 1, \ldots, N$, we can construct the dual multipliers $\{\nu^i, \mu^i\}_{i \in [N]}$ that satisfy the KKT conditions. Let $\nu^i_{ts} = \min_a Q^i_{tsa}$ for all $(i, t, s) \in [N] \times \mathcal{T} \times [S]$, and $\mu^i_{tsa} = \nu^i_{ts} - Q^i_{tsa}$ for all $(i, t, s, a) \in [N] \times \mathcal{T} \times [S] \times [A]$. We can show that (x^i, ν^i, μ^i) satisfy primal feasibility $x^i \in \mathcal{X}(P^i, p^i_0)$, 2) dual feasibility $\mu^i \geq 0$, and 3) complementary slackness $\mu^i_{tsa}x^i_{tsa} = 0,\ \forall (t, s, a) \in \mathcal{T} \times [S] \times [A]$. The stationarity condition (3.8) is equivalent to the Q-value definition (3.6). ■

KKT conditions characterize regular locally optimal solutions to optimization problems. We show here that they are also necessary conditions for local Nash equilibria.

Lemma 3.1 (Necessary condition for Nash equilibrium) *When Assumption 3.1 is satisfied and each player solves a coupled MDP problem (3.4), the Markov game's local Nash equilibrium $x^\star$ (2.23) generates Q-value functions $Q^i(x^\star)$ (3.6) that satisfies the Q-value optimality condition (3.7) for $i = 1, \ldots, N$.*

Proof: From Proposition 3.1, $(x^{1\star}, \ldots, x^{N\star})$ satisfy the KKT conditions of the coupled MDPs (3.4) for $i = 1, \ldots, N$ if and only if $Q^i(x^\star)$ satisfies (3.7) with respect to $x^{i\star}$ for $i = 1, \ldots, N$. Each optimization problem (3.4) optimizes a continuously differentiable objective

over affine constraints $\mathcal{X}(P^i, p_0^i)$ (2.13). As such, the linear constraint qualification [105] applies to (3.4), and the KKT conditions are necessary for local optimality. ■

Proposition 3.1 shows that Q-value optimality (3.7) is necessary for a joint state-action distribution x to be a Nash equilibrium, but whether or not it's sufficient is yet to be determined. To show that Q-value optimality is indeed not enough for Nash equilibrium, we give an example of when the Q-value optimality conditions are satisfied but the corresponding state-action distribution is not optimal below in Example 3.1. This is a significant departure from non-atomic MDP congestion games, in which the MDP Wardrop equilibrium is necessarily and sufficiently characterized by the continuous player Q-value optimality condition (2.29).

Remark 3.3 *In non-atomic MDP congestion games, individual players cannot change the state-action costs unilaterally, whereas, in atomic Markov games, the player-specific state-action costs will change when individual players unilaterally change their policy.*

Example 3.1 (Self-inflicted state-action congestion) *We give an example of a state-action distribution that is optimal with respect to the Q-values (3.7) but is not optimal for the player-specific optimization problem (3.4). Consider a one-time step, single-state, two-action MDP, such that $T = S = 1$ and $A = 2$. We assume that a single player occupies this MDP state-action space, and denote the joint state-action distribution by $x = (x_{011}, x_{012}) \in \Delta_2$, where x_{011} is the probability of the player taking action a_1 and x_{012} is the probability of the player taking action a_2. The single player MDP (3.4) can be written as*

$$\min_x 2x_{011} + x_{012}(3 - 2x_{012}) \ \ s.t. \ \ x_{011} + x_{012} = 1, \ \ x_{011}, x_{012} \geq 0. \tag{3.11}$$

The two state-action cost functions are given by $\ell_{011}(x) = 2$, $\ell_{012}(x) = 3 - 2x_{012}$. At the joint state-action distribution $x = (1, 0)$, the corresponding Q-values are given by

$$Q^1_{011}\big((1,0)\big) = 2, \ \ Q^1_{012}\big((1,0)\big) = 3. \tag{3.12}$$

We can verify that at the state-action distribution $x = (1, 0)$, the Q-values (3.12) satisfy the optimality condition (3.7), but $x = (1, 0)$ is not an optimal solution to the linear program (3.11) . In fact, any $x_\epsilon = (\epsilon, 1 - \epsilon)$ will incur a smaller cost if $\epsilon \in (0, 1)$.

For the set of player cost functions that satisfy Assumption 3.1, Q-value optimality (3.7) is necessary but insufficient for establishing simultaneous optimality (3.5). However, we can

classify sets of player costs for which Q-value optimality (3.7) is both necessary and sufficient for characterizing the corresponding joint state-action distribution x as a Nash equilibrium.

Lemma 3.2 (Sufficiency via convexity) *When player cost functions $(\ell^1, \ldots, \ell^N)$ satisfy Assumption 3.1,*

1. *if $\partial \ell^i(x^{i\star}, x^{-i\star})/\partial x^i \succ 0$, for all $i \in [N]$ at a joint state-action distribution $x^\star \in \prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$, and $Q^i(x^\star)$ is Q-value optimal (3.7) for all $i \in [N]$, then $x^\star$ is a **local Nash equilibrium**;*
2. *if $\partial \ell^i(x)/\partial x^i \succeq 0$ for all $x \in \prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$ and $i \in [N]$, and $x^\star$ is Q-value optimal (3.7) for all $i \in [N]$, then $x^\star$ is a **global Nash equilibrium**.*

Proof: Under Assumption 3.1, player i solves (3.4) with objective F^i for $i = 1, \ldots, N$. Since F^i is continuously differentiable, its Hessian $\partial^2 F(x^i, x^{-i})/(\partial x^i)^2$ exists and is equivalent to the Jacobian of the cost player cost function, $\partial \ell^i(x)/\partial x^i$, which is positive definite by lemma assumption. From the second-order sufficiency condition [96, Thm.12.6], x is a strict local solution of the optimization problem (3.4) for player i. Since this holds for all $i \in [N]$, $x^\star$ is a local Nash equilibrium.

If $\partial \ell^i(x)/\partial x^i \succeq 0$ for all $x^i \in \mathcal{X}(P^i, p_0^i)$ and $i = 1, \ldots, N$, then every player's optimization problem (3.4) is convex in x^i, and their KKT conditions are necessary and sufficient for global optimality. From Proposition 3.1, $x^{i\star}$ satisfies the KKT conditions if and only if $Q^i(x^{i\star}, x^{-i\star})$ satisfies Q-value optimality (3.7) for $i = 1, \ldots, N$. ■

In Lemma 3.2, the first result on local Nash effectively results from a second-order sufficient condition due to strict convexity, while the second result on global Nash results from convex optimization.

Remark 3.4 (Existence of Nash equilibria under convexity) *In Lemma 3.2, the condition $\partial \ell^i(x)/\partial x^i \succeq 0$ for all $x^i \in \mathcal{X}(P^i, p_0^i)$ implies that $F^i(x^i, x^{-i})$ is convex in x^i for $i = 1, \ldots, N$. This ensures that at least one Nash equilibrium will exist [113, Thm.1].*

Remark 3.5 (Set of Nash equilibria under monotonicity) *In Lemma 3.2, the condition that $\partial \ell^i(x)/\partial x^i \succeq 0$ for all $x^i \in \mathcal{X}(P^i, p_0^i)$ also implies that $\xi(x)$ (3.3) is a monotone mapping, i.e., $(x-y)^\top \big(\xi(x) - \xi(y)\big) \geq 0 \; \forall x, y \in \prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$, then convexity and uniqueness properties for the set of Nash equilibria can be established via [120, Thm.3].*

3.1.1 MDP Congestion Games

Markov games of the form (3.4) are equivalent to N MDPs that are coupled in their cost functions. We briefly discussed convexity and monotonicity restrictions in Remarks 3.4 and 3.5, where we restricted the structures of $\partial \ell^i(x^i, x^{-i})/\partial x^i$ for $i = 1, \ldots, N$. In this section, we restrict the structure of the gradient of ℓ^i with respect to x^j for all $j \neq i$ and $i \in [N]$, and show that under a specific symmetry condition, the resulting Markov game can be reformulated as a single optimization problem via the existence of a universal potential function.

Definition 3.2 (Universal Potential Function) *The function $F : \mathbb{R}^{N(T+1)SA} \mapsto \mathbb{R}$ is a universal potential function if it is continuously differentiable and satisfies*

$$\frac{\partial F(x^1, \ldots, x^N)}{\partial x^i} = \ell^i(x^1, \ldots, x^N), \ \forall (x^1, \ldots, x^N) \in \prod_{i \in [N]} \mathcal{X}(P^i, p_0^i), \ \forall i \in [N]. \tag{3.13}$$

A universal potential function can only exist if Assumption 3.1 is satisfied. In Assumption 3.1 individual MDPs (3.4) are assumed to possess a unique objective function $F^i : \mathbb{R}^{N(T+1)SA} \mapsto \mathbb{R}$. If $F^1 = F^2 = \ldots = F^N$, then each F^i is a universal potential function as defined in Definition 3.2. However, note that having identical optimization objectives is not equivalent to players having identical MDP costs.

Proposition 3.2 (External symmetry condition) *[117] [63, Thm.3.2,Thm.3.4] A universal potential function $F : \mathbb{R}^{N(T+1)SA} \mapsto \mathbb{R}$ that satisfies (3.13) exists if and only if the player costs $\ell^1, \ldots, \ell^N$ satisfy*

$$\frac{\partial \ell^i_{tsa}(x)}{\partial x^{i'}_{t's'a'}} = \frac{\partial \ell^{i'}_{t's'a'}(x)}{\partial x^i_{tsa}}, \ \forall (i,t,s,a), (i',t',s',a') \in [N] \times \mathcal{T} \times [S] \times [A], \ \forall x \in \prod_{i \in [N]} \mathcal{X}(P^i, p_0^i). \tag{3.14}$$

Remark 3.6 *Proposition 3.2 provides intuition for when a universal potential function may exist. When players are coupled to each other through sharing resources, Proposition 3.2 holds if congestion in the shared resource impacts all players equally. Symmetric congestion effects occur in multi-agent pick-up and delivery [116], air traffic management [124], and ground vehicle collision avoidance [56].*

The symmetry condition (3.14) is related to the Hessian assumptions made in Lemma 3.2: in addition to requiring that $\partial \ell^i(x)/\partial x^i$ is symmetric, (3.14) further restricts $\ell^i(x)$'s gradient

with respect to all of $x^1, \ldots, x^N$. On the other hand, the existence of a universal potential function F (3.13) does not imply that the individual MDP given by (3.4) is convex.

Finding the Nash equilibria of an MDP congestion game with the potential function F is equivalent to solving the following optimization problem,

$$\min_{x^1,\ldots,x^N} F(x^i, x^{-i}) \text{ s.t. } x^i \in \mathcal{X}(P^i, p_0^i), \ \forall i \in [N]. \tag{3.15}$$

Proposition 3.3 *For a Markov game in which the player cost functions satisfy the symmetry condition (3.14), the following conditions are equivalent.*

1. $x^\star \in \prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$ *is a KKT point of* (3.15),
2. $x^\star = (x^{1\star}, \ldots, x^{N\star}) \in \prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$ *has Q-value functions* $Q^i(x)$ *that satisfy* (3.7) *for* $i = 1, \ldots, N$.

Proof: $1 \Rightarrow 2$: if $x^\star \in \prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$ is a KKT point of (3.15), then $x^{i^\star}$ is a KKT point of (3.4) with objective function $F^i = F$ for $i = 1, \ldots N$. From Proposition 3.1, x^i satisfies (3.7) for $i = 1, \ldots N$.

$2 \Rightarrow 1$: from the symmetry condition (3.14), there exists a universal potential function F (3.13) that satisfies Assumption 3.1 by taking $F^i = F$ for $i = 1, \ldots, N$. From Proposition 3.1, $x^{i\star}$ satisfying (3.7) implies that $x^{i\star}$ is a KKT point of $\min_{x^i \in \mathcal{X}(P^i, p_0^i)} F(x^i, x^{-i})$. If this holds for all $i \in [N]$, then $x^1, \ldots, x^N$ will be the KKT point of (3.15), which is simply minimizing F over all inputs rather than one individual input x^i. ■

Remark 3.7 *While Proposition 3.3 is very similar to Proposition 3.1, Proposition 3.3 states that under the existence of a universal potential function,* $Q^1(x), \ldots, Q^N(x)$ *being Q-value optimal* (3.7) *is equivalent to satisfying the KKT condition of a* single *optimization problem.*

Corollary 3.1 (Existence and uniqueness of Nash equilibrium) *When a universal potential function* (3.13) *exists under Assumption 3.1, at least one global Nash equilibrium (Definition 3.1) exists.*

If $\xi(x) \succ 0$ *for all* $x \in \prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$*, then there is a unique global Nash equilibrium.*

Proof: When a universal potential function (3.13) exists, (3.15) is an optimization problem with a continuously differentiable objective $F(x)$ and a non-empty set of compact constraints $\prod_{i\in[N]} \mathcal{X}(P^i, p_0^i)$. From the Weierstrasse extreme value theorem, there exists a

global solution to (3.15), which we denote by $x^\star = (x^{1\star}, \ldots, x^{N\star})$. Since $x^\star$ is the global minimum solution, $x^{i\star}$ is a directional minimum solution within $\mathcal{X}(P^i, p_0^i)$, which implies that it is the global minimum solution to the individual coupled MDP problem (3.4) under opponent strategy $x^{(-i)\star}$ for $i = 1, \ldots, N$.

If $\xi(x) \succ 0$ for all feasible joint state-action distributions x, then there exists a unique joint state-action distribution $x^\star$ that satisfies the KKT conditions of (3.15) and it must also be the global minimum solution [22]. ■

Remark 3.8 *We note the existence of a global Nash equilibrium in these Markov games does not rely on the convexity of player costs as in Remark 3.4. However, strict convexity of the universal potential function F is a sufficient condition for the uniqueness of the global Nash equilibrium.*

3.1.2 Games with Interaction-induced Congestion

In certain scenarios where multiple decision makers interact in a shared operation environment, individual decision makers' state-action cost $\ell^i(x)$ is explicitly independent of its state-action probability distribution, x^i. We refer to this property as having interaction-induced congestion and is captured by the following assumption.

Assumption 3.2 (Interaction-induced congestion) *Player i experiences interaction-induced congestion if changes in its state-action distribution $x^i \in \mathcal{X}(P^i, p_0^i)$ does not change its state-action cost $\ell^i_{tsa}(x^i, x^{-i})$ for all $(t, s, a) \in \mathcal{T} \times [S] \times [A]$—i.e.,*

$$\frac{\partial \ell^i_{tsa}(x^i, x^{-i})}{\partial x^i_{tsa}} = 0, \ \forall (t, s, a) \in \mathcal{T} \times [S] \times [A], \ \forall (x^i, x^{-i}) \in \prod_{i \in [N]} \mathcal{X}(P^i, p_0^i). \tag{3.16}$$

Under Assumption 3.2, the resulting Markov game is an atomic MDP congestion game with a multi-linear potential function given by

$$F(x) = \sum_{i,t,s,a} x^i_{tsa} \ell^i_{tsa}(x). \tag{3.17}$$

We prove this more formally below.

Corollary 3.2 (Sufficient conditions for NE) *When player i only experiences interaction-induced congestion (3.16), for $i = 1, \ldots, N$, the resulting Markov game has a universal poten-*

tial function F given by (3.17), and all joint state-action distributions $x \in \prod_{i\in[N]} \mathcal{X}^i(P^i, p_0)$ that satisfy the Q-value optimality condition (3.7) are global Nash equilibria.

Proof: Under Assumption 3.2, we first verify that $\partial F(x^1, \ldots, x^N)/\partial x^i = \ell^i(x)$ for $i = 1, \ldots, N$ holds for F in (3.17). As F is written, $\partial F(x^1, \ldots, x^N)/\partial x^i_{tsa} = \ell^i_{tsa}(x) + x^i_{tsa} \frac{\partial \ell^i_{tsa}(x)}{\partial x^i_{tsa}}$ for all $(i, t, s, a) \in [N] \times \mathcal{T} \times [S] \times [A]$. Under Assumption 3.2, $\frac{\partial \ell^i_{tsa}(x)}{\partial x^i_{tsa}} = 0$. We next consider the individual coupled MDP problem (3.4) for player i. Under Assumption 3.2, we note that (3.4) is a linear program with solution variable x^i, as such, all feasible state-action distributions x^i that satisfy the KKT conditions of (3.4) are optimal. From Proposition 3.3, a joint state-action generates $Q^1(x), \ldots, Q^N(x)$ that satisfies the Q-value optimality condition (3.7) if and only if x^i is a KKT point for player i's coupled MDP problem (3.4). Therefore, a joint state-action distribution x is a Nash equilibrium if and only if the corresponding Q-values $Q^1(x), \ldots, Q^N(x)$ satisfy the Q-value optimality (3.7). ■

Remark 3.9 *Interestingly, the universal potential function F (3.17) produces a Hessian matrix $d^2F(x)/dx^2$ that has a diagonal of zero blocks, making it a hollow matrix and the corresponding potential game (3.15) a nonconvex optimization problem. Furthermore, Corollary 3.2 shows that any KKT point of this game is a global Nash equilibrium.*

Under the interaction-induced congestion assumption, player i solves a standard MDP when its opponents do not change their policies. Player i's Q-value iteration (3.6) is *stationary* to changes in player i's state-action distribution x^i, and non-stationary to changes in the opponent's state-action distribution x^{-i}.

3.2 Modeling Spatial Conflicts in Multi-player Non-cooperative Path Planning

When multiple autonomous vehicles owned by different organizations share an operation space either in the air or on the ground, individual vehicles are non-cooperative yet non-adversarial: any two vehicles owned by different organizations would compete for road space to minimize their task completion time, yet both will avoid colliding with the other in the close encounter situations. We assume that all vehicles have independently assigned tasks, and no vehicle will change its actions to optimize its opponents' paths. On the other hand, if multiple vehicles collide due to conflicting paths, all vehicles will incur a penalty cost.

3.2.1 Probability of Co-occupying the Same Operation Space

To evaluate all possible path conflicts between vehicles under MDP dynamics (3.4), we take the probability of two or more autonomous vehicles co-occupying a common state (state-action) as a proxy for collision.

Consider the finite time horizon finite state-action Markov game for N players introduced in Section 3. Player i's state-action distribution is given by $x^i \in \Delta_A^{(T+1)S}$. We can directly compute the probability of co-occupying a state or state-action with another player.

Lemma 3.3 *Under the joint state-action distribution $x = (x^1, \ldots, x^N)$ (3.1), player i's probability of being in state s and state-action (s, a) at time t with at least one other player are respectively denoted by $D_{ts}^i(x)$ and $G_{tsa}^i(x)$ and computed as*

$$D_{ts}^i(x) = 1 - \prod_{j \neq i}(1 - \sum_{a'} x_{tsa'}^j),\ G_{tsa}^i(x) = 1 - \prod_{j \neq i}(1 - x_{tsa}^j)\ \forall\ i, t, s, a \in [N] \times \mathcal{T} \times [S] \times [A]. \tag{3.18}$$

Proof: The probability of player j taking state-action (s, a) at time t is x_{tsa}^j. The probability that player j does *not* take state-action (s, a) at time t is $1 - x_{tsa}^j$. The probability that *none* of the players $j \neq i$ take state-action (s, a) at time t is $\prod_{j \neq i}(1 - x_{tsa}^j)$. The probability of *at least one* other player $j \neq i$ taking state-action (s, a) at time t is given by $G_{tsa}^i(x)$ in (3.18). To derive $D_{ts}^i(x)$ (3.18), we apply similar arguments to the probability of player j being in state s at time t, given by $\sum_a x_{tsa}^j$. ■

Task completion under non-cooperative spatial conflicts. To model path conflicts in a shared operation space, we consider the following state-action cost.

$$\ell_{tsa}^i(x) = C_{tsa}^i + \kappa\Big(D_{ts}^i(x) + G_{tsa}^i(x)\Big),\ \forall (i, t, s, a) \in [N] \times \mathcal{T} \times [S] \times [A], \tag{3.19}$$

where $C^i \in \mathbb{R}^{(T+1)SA}$ models player i's individual task completion cost, $D^i : \Delta_A^{N(T+1)S} \mapsto \mathbb{R}^{(T+1)S}$ and $G^i : \Delta_A^{N(T+1)S} \mapsto \mathbb{R}^{(T+1)S}$ are the co-occupation probabilities given in (3.18), and $\kappa \in \mathbb{R}_+$ is a user-defined parameter that models the player i's willingness to risk co-occupying any state-action with an opponent. Risk-ignorant players can be modeled by $\kappa = 0$, and risk-averse players can be modeled by $\kappa \to \infty$.

Remark 3.10 *As shown in Section 3.5, D^i and G^i can be used to penalize flight separation constraint violations in air traffic management, where the resources are air spaces and they cannot be occupied by two aircraft at the same time.*

The resulting state-action cost $\ell^i(x)$ (3.19) is independent of player i's state-action distribution x^i. Therefore, player i only experiences interaction-induced congestion and satisfies Assumption 3.2 for $i = 1, \ldots, N$. The corresponding potential function F (3.17) is given by

$$F(x^1, \ldots, x^N) = \sum_{i,t,s,a} x^i_{tsa} C^i_{tsa} + \sum_{t,s} k\Big(\sum_{i,a} 2x^i_{tsa} + \prod_{i\in[N]}(1 - \sum_a x^i_{tsa}) + \sum_a \prod_{i\in[N]}(1 - x^i_{tsa}) \Big). \tag{3.20}$$

Under Assumption 3.2, the Markov game under state-action costs modeled by (3.19) has at least one global Nash equilibrium and any joint state-action distribution whose corresponding Q-value functions $Q^1(x), \ldots Q^N(x)$ satisfy the Q-value optimality condition (3.7) for $i = 1, \ldots, N$ is a global Nash equilibrium. Since the corresponding potential function, F (3.20) is multi-linear, there can be multiple Nash equilibria.

3.2.2 Resource Usage Approximation via Congestion Function

In Section 3.2.1, an underlying assumption is that the shared spatial resources are only accessible by one player at a time. Next, we consider the setting in which each resource is simultaneously accessible by multiple players, but its efficiency is reduced with greater player access.

Extending the concept of co-occupation probabilities (3.18), it is technically feasible to evaluate the resource efficiency under m players by computing the probability of at least m players co-occupying any state-action for $m = 2, \ldots, N$. However, the associated computation complexity will grow in m. To remain computationally feasible for an N player game, we leverage the weighted congestion distribution.

Definition 3.3 (Weighted congestion distribution) *Given a finite number of players $[N]$ in a Markov game, the weighted congestion distribution is the weighted sum of individual state-action distributions, given by*

$$y := \sum_{i\in[N]} \alpha_i x^i \in \mathbb{R}^{(T+1)SA}, \ \alpha_i > 0, \forall i \in [N], \ \sum_{i\in[N]} \alpha_i = 1. \tag{3.21}$$

In Definition 3.3, a higher α_i denotes a higher congestion impact. If all players contribute to congestion equally, $\alpha_i = \frac{1}{N}\ \forall i \in [N]$.

Task completion under congested spatial resources. To model shared resources with

reduced efficiency under collective usage, we consider the following state-action cost.

$$\ell^i_{tsa}(y, x^i) = \alpha_i f_{ts}\big(\textstyle\sum_{a'} y_{tsa'}\big) + \alpha_i g_{tsa}\big(y_{tsa}\big) + h^i_{tsa}(x^i_{tsa}),\ (i,t,s,a) \in [N]\times\mathcal{T}\times[S]\times[A], \quad (3.22)$$

where $\alpha_i \in \mathbb{R}$ is the same as in (3.21), $f_{ts} : \mathbb{R} \mapsto \mathbb{R}$ is the state-dependent congestion and takes the congestion level of (t, s) as input, $g_{tsa} : \mathbb{R} \mapsto \mathbb{R}$ is the state-action-dependent congestion and takes the congestion level of (t, s, a) as input, and $h^i_{tsa} : \mathbb{R} \mapsto \mathbb{R}$ is the player-specific objective and takes player i's probability of being in (t, s, a) as input. Player-specific objectives such as obstacle avoidance and target reachability can be incorporated as constant offsets in h^i.

Remark 3.11 *As shown in Section 3.4, f and g can be interpreted as increased traversal time in multi-robot warehouse path planning, in which low-level conflict resolution of multiple robots attempting to cross a common work floor space will cause their traversal time to decrease.*

Effect of α_i The impact factor α_i scales player i's relative impact on the total congestion and the total congestion's impact on player i. When $\alpha_i < \alpha_j$, player i impacts congestion less and cares about the congestion less than player j. When $\alpha_i > \alpha_j$, player i impacts congestion more and cares about the congestion more than player j.
Road-sharing vehicles often do not view congestion symmetrically. We give the following example to illustrate how the impact factor α_i from (3.22) can be used to reflect asymmetry in congestion effects.

Example 3.2 (Heterogeneous congestion perceptions in road-sharing vehicles) *Consider a sedan (player 1, $\alpha_1 = 0.1$) and a trailer (player 2, $\alpha_2 = 0.9$) sharing a road. Player i wants to reach state $s_i \in [S]$. The player-specific objective is $h^i_{tsa}(x^i_{tsa}) = -\mathbb{1}[s = s_i] + \epsilon_i x^i_{tsa}$, where $\mathbb{1}[w]$ is 1 when w is true and 0 otherwise. The term $\epsilon_i x^i_{tsa}$ where $\epsilon_i > 0$ encourages player i to randomize its policy over all optimal actions. Players experience state-based congestion as $f_{ts}(w) = \exp(w)$. The player cost (3.22) is $\ell^i_{tsa}(y, x^i) = \alpha_i \exp(\sum_{a'} y_{tsa'}) + \epsilon_i x^i_{tsa} - \mathbb{1}[s = s_i]$.*

If the congestion functions f, g, and h^i, $\forall i \in [N]$ are strictly increasing functions, the resulting optimization problem is strongly convex and there exists a unique Nash equilibrium.

Proposition 3.4 *If $h^i_{tsa}(\cdot)$ is strictly increasing and $f_{ts}(\cdot)$, $g_{tsa}(\cdot)$ are non-decreasing $\forall(i,t,s,a) \in [N]\times\mathcal{T}\times[S]\times[A]$, then 1) the corresponding potential optimization formulation 3.15 is strongly convex and 2) there exists a unique Nash equilibrium.*

Proof: Let I_Z be an identity matrix of size $Z \times Z$, $\mathbb{1}_Z$ be an one's vector of size $Z \times 1$, $\vec{\alpha} = [\alpha_1, \ldots, \alpha_N] \in \mathbb{R}^{N \times 1}$, $h(x) = [h^1(x), \ldots, h^N(x)] \in \mathbb{R}^{N(T+1)SA}$, and $\otimes$ be a Kronecker product. We define the matrices $M = \vec{\alpha} \otimes I_{(T+1)SA}$ and $J = (I_{(T+1)S} \otimes \mathbb{1}_A^\top)M$, and verify that for any joint state-action distribution x (3.1), 1)$Mx = y$, where y is the congestion distribution (3.21), 2) $[Jx]_{ts} = \sum_{a'} y_{tsa'}\ \forall (t,s) \in \mathcal{T} \times [S]$, and 3) $\xi(x) = J^\top f(Jx) + M^\top g(Mx) + h(x)$.

Let $w = Jx$, we can take ξ's (3.3) Jacobian as $\nabla \xi(x) = J^\top \nabla f(w) J + M^\top \nabla g(y) M + \nabla h(x)$. Under Proposition assumptions, $\nabla f(w)$ and $\nabla g(y)$ are non-negative diagonal matrices, and $\nabla h(x)$ is a strictly positive diagonal matrix. Therefore, $\nabla^2 F(x) = \nabla \xi(x) \succ 0$, and F is strongly convex. Since (3.15) is a strongly convex optimization problem over linear and feasible constraints, there exists a unique KKT point. From Lemma 3.2, there is a unique Nash equilibrium. ■

Proposition 3.4 implies that a strictly increasing h^i is crucial to ensuring a unique Nash equilibrium. Therefore, h^i can be interpreted as a regularization term.

The resulting state-action cost (3.22) satisfies the external symmetry condition (3.14), and the associated universal potential function is given by

$$F(x) = \sum_{t,s} \int_0^{\sum_{a'} y_{tsa'}} f_{ts}(u)\partial u + \sum_{t,s,a} \int_0^{y_{tsa}} g_{tsa}(u)\partial u + \sum_{i,t,s,a} \int_0^{x^i_{tsa}} h^i_{tsa}(u)\partial u. \tag{3.23}$$

As opposed to the spatial conflict state-action cost (3.19), the resource congestion state-action cost (3.22) implies that players can induce congestion by themselves. However, Assumption 3.1 is still satisfied as well as the cost symmetry condition (3.14). Therefore, there is at least one global Nash equilibrium that satisfies the Q-value optimality conditions (3.7) for $Q^1(x), \ldots, Q^N(x)$. If the player state-action costs additionally satisfy $\partial \ell^i(x)/\partial x^i \succeq 0$ for $i = 1, \ldots N$ and for all $x^i \in \mathcal{X}(P^i, p_0^i)$, then there will be at least one global Nash equilibrium.

3.3 Solving for Nash Equilibrium via Frank-Wolfe Learning Dynamics

When the state-action cost functions satisfy the external symmetry condition (3.14) and there exists a universal potential function for the Markov game, we can compute a Nash equilibrium by leveraging single-agent dynamic programming.

In Algorithm 3.1, players access an *oracle* that evaluates the player costs. In line 5, $\pi^i \in \Delta_A^{(T+1)S}$ is any optimal policy for the finite time MDP with cost C^{ik}, transition probability

Algorithm 3.1 Frank-Wolfe with dynamic programming

Input: $\{\ell^i\}_{i\in[N]}$, $\{P^i\}_{i\in[N]}$, $\{p_0^i\}_{i\in[N]}$, N, $[S]$, $[A]$, $\mathcal{T}$.
Output: $\{\hat{x}^i_{tsa}\}_{t\in\mathcal{T},s\in[S],a\in[A]}$.
1: $x^{i0} \in \mathcal{X}(P^i, p_0^i) \in \mathbb{R}^{(T+1)SA}, \quad \forall\, i \in [N]$.
2: **for** $k = 1, 2, \ldots,$ **do**
3: **for** $i = 1, \ldots, N$ **do**
4: $C^{ik} = \ell^i([x^{1k}, \ldots, x^{Nk}])$
5: $\pi^i = \mathrm{MDP}(C^{ik},\, P^i,\, [S],\, [A],\, T,\, p_0^i)$
6: $b^{ik} = \textsc{RetrieveDensity}(P,\, p_0^i,\, \pi^i)$ ▷ Alg. 3.2
7: $x^{i(k+1)} = \big(1 - \frac{2}{k+1}\big)x^{ik} + \frac{2}{k+1}b^{ik}$
8: **end for**
9: **end for**

P^i, and initial distribution p_0^i. We use value iteration to recursively find π^i as

$$\begin{aligned} V^i_{Ts} &= \min_a C^{ik}_{Tsa},\ \ \pi^i_{Ts} \in \operatorname{argmin}_a C^{ik}_{Tsa}, \\ V^i_{(t-1)s} &= \min_a C^{ik}_{(t-1)sa} + \textstyle\sum_{s'} P^i_{ts'sa} V^i_{ts'},\ \forall t \in [T] \\ \pi^i_{(t-1)s} &\in \operatorname{argmin}_a C^{ik}_{(t-1)sa} + \textstyle\sum_{s'} P^i_{ts'sa} V^i_{ts'},\ \forall t \in [T] \end{aligned} \tag{3.24}$$

Algorithm 3.1 then retrieves the corresponding state-action density b^{ik} via Algorithm 3.2 and combines it with the current state-action density x^{ik} to derive the next joint state-action density. All steps within lines 4 to 7 are parallelizable.

Algorithm 3.2 Retrieving state-action distribution from π

Input: P, z, π.
Output: $\{d_{tsa}\}_{t\in\mathcal{T},s\in[S],a\in[A]}$
1: $d_{0s\pi_{0s}} = z_s,\ \forall s \in [S]$
2: **for** $t = 1, \ldots, T$ **do**
3: $d_{ts(\pi_{ts})} = \sum_a \sum_{s'} P_{tss'a} d_{(t-1)s'a},\ \forall\ s \in [S]$
4: **end for**

Theorem 3.1 *When a universal potential function F (3.13) exists, if the player state-action costs satisfy $\partial\ell^i(x)/\partial x^i \succeq 0$ for $i = 1, \ldots, N$, then Algorithm 3.1 converges towards the Nash equilibrium $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^N)$ as*

$$\tfrac{\alpha}{2} \textstyle\sum_{i\in[N]} \left\|x^{ik} - \hat{x}^i\right\|_2^2 \le \frac{2C_F}{k+2} \tag{3.25}$$

where C_F *is the* curvature constant *of the potential function* F (3.13)*, defined as*

$$C_F := \sup_{\substack{x^i, s^i \in \mathcal{X}(P^i, z_0^i) \\ \gamma \in [0,1] \\ w^i = x^i + \gamma(s^i - x^i)}} \frac{2}{\gamma^2} \Big(F(s) - F(x) - \textstyle\sum_{i \in [N]} (x^i - w^i)^\top \ell^i(x) \Big).$$

Proof: Algorithm 3.1 is a straightforward implementation of [55, Alg.2] for optimization problem 3.15, which optimizes the universal potential function F (3.13). Continuously differentiable potential functions are locally Lipschitz. Therefore, (3.25) then follows directly from [55, Thm.1] and Algorithm 3.1 converges to the nearest KKT point. If $\partial \ell^i / \partial x^i \succeq 0$ for all $i \in [N]$, then the Markov game is convex, and therefore all KKT points are Nash equilibria under Lemma 3.2. ■

Remark 3.12 (Scalability) *Algorithm 3.1 has linear complexity in the number of players.*

Algorithm 3.1 has the following interpretation in repeated game play: each player executes a fixed strategy determined at the start of each game without feedback and receives the state-dependent costs based on the observed y^k (3.21) at the end of game k. Algorithm 3.1 models the players incrementally updating their policies each iteration using the best response of that iteration. Their resulting state-action trajectory is then computed using Algorithm 3.2. Under diminishing step sizes, Algorithm 3.1 will converge to the Nash equilibrium state-action distribution.

3.4 Warehouse Path Coordination under Stochastic Package Arrival Times

We apply our game model to a multi-player pick up and delivery scenario with stochastic package arrival times. As shown in Figure 3.1, N players navigate a 2D space. Each player's goal is to transport packages from the pick up chutes to the drop off chutes while avoiding collision with others. The code for the simulation is available at `https://github.com/lisarah/mdp_path_coordination`.

Players operate in a two-dimensional grid world with 5 rows and 10 columns. In addition to capturing location, each state also dictates whether the robot is in pick up or delivery mode. The state space is given by

$$[S] = \Big\{ (v, w, m) \mid 1 \leq v \leq 5,\ 1 \leq w \leq 10,\ m \in \{1, 2\} \Big\}.$$

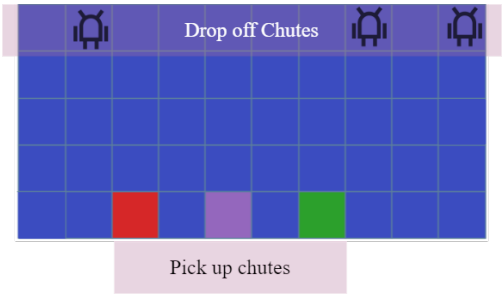


Figure 3.1: Operation environment for multi-robot warehouse scenario.

At each state, available actions are $[A] = \{u, d, r, l, s\}$, corresponding to up, down, right, left, stay. Player transition dynamics and rewards are *stationary* in time. The transition probability of each state (v, w, m) extends the location-based transition probabilities P^0.

Location-based transition. Let $u = (v, w)$ denote the location component of the state. At each location, each action either points to a feasible target $u_{targ}(a)$ or is infeasible. The set of all feasible targets from u is $\mathcal{N}(u)$. When a target exists, players have $1 > q > 0$ chance of reaching it and $1 - q$ chance of reaching other states in $\mathcal{N}(u)$.

$$P^0_{u'ua} = \begin{cases} q & u' = u_{targ}(a), \\ \frac{1-q}{|\mathcal{N}(u)|} & u' \in \mathcal{N}(u)/\{u_{targ}(a)\}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.26}$$

When the target location is infeasible, the player transitions into a neighboring state $u' \in \mathcal{N}(u)$ at random.

$$P^0_{u'ua} = \begin{cases} \frac{1}{|\mathcal{N}(u)|} & u' \in \mathcal{N}(u), \\ 0 & \text{otherwise.} \end{cases} \tag{3.27}$$

Full transition dynamics. Within the same mode, players transition between locations via dynamics P^0. Player modes transition at pick up chutes $\mathcal{P}$ and drop off chutes $\mathcal{D}$, both are subsets of the MDP state space, as illustrated in Figure 3.1.

1. When player i is in mode 1 (pick up) and about to transition into $p^i \in \mathcal{P}$, player i's

mode has r^i probability of switching to mode 2 (drop off).

$$\begin{cases} P^i_{t(p^i,2)sa} = r^i P^0_{tp^iua}, \\ P^i_{t(p^i,1)sa} = (1-r^i) P^0_{tp^iua}, \end{cases} \qquad \forall s = (u,1),\ s \in [S].$$

2. When player i is in mode 2 and about to transition into $d^i \in \mathcal{D}$, player i switches to mode 1 with probability 1.

$$\begin{cases} P^i_{t(d^i,1)sa} = P^0_{td^iua}, \\ P^i_{t(p^i,2)sa} = 0, \end{cases} \qquad \forall s = (u,2),\ s \in [S].$$

Here, $r^i \in \mathbb{R}$ denotes the probability of package arrival when player i is in p^i. Modeled as an independent Poisson process with rate λ_i and interval $\Delta t = 1s$, $r^i = \exp(-\lambda_i \Delta_t)$.

Warehouse player costs For all $(t, s, a) \in \mathcal{T} \times [S] \times [A]$ and congestion distribution y (3.21), player i's cost is given by

$$\ell^i_{tsa}(y, x^i) = \epsilon x^i_{tsa} - c^i_{tsa} + \alpha_i f_{ts}(y).$$

The player-specific task completion objective c^i_{tsa} is defined as

$$c^i_{t(v,w,m)a} = \begin{cases} 1 & (v,w) = p^i,\ m = 1, \\ 1 & (v,w) = d^i,\ m = 2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.28}$$

The congestion function is strictly state-based and is an exponential function given by

$$f_{t(v,w,m)}(y) = -\beta \exp\Big(\beta(\sum_{m' \in \{1,2\}} \sum_{a' \in [A]} y_{t(v,w,m')a'} - 1)\Big), \tag{3.29}$$

where $\alpha_i > 0$ for all $(t, s, a) \in \mathcal{T} \times [S] \times [A]$. As opposed to c^i (3.28), $f_{t(v,w,m)}(y)$ (3.29) computes the congestion in $(v, w, \cdot)$ using both $(v, w, 1)$'s and $(v, w, 2)$'s congestion levels.

Simulation hyper-parameters. We simulate the path coordination game using parameters from Table 3.1. Player i's pick up locations is the i^{th} element of $\mathcal{P} = \{(4, w^i) \mid w^i \in [8, 7, 2]\}$, and its drop-off location is the i^{th} element of $\mathcal{D} = \{(0, w^i) \mid w^i \in [4, 5, 8]\}$. At $t = 0$,

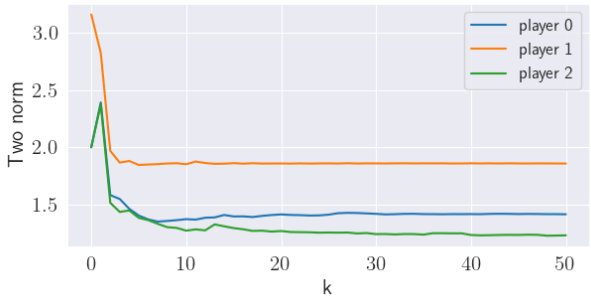


Figure 3.2: $\|\cdot\|_2$ of player i's state-action distribution over Algorithm 3.1 iterations.

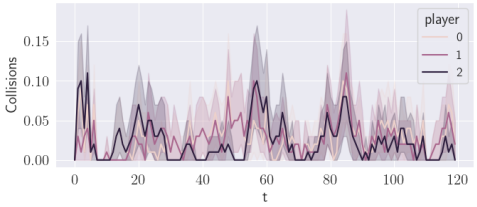


Figure 3.3: Collisions per player as a function of MDP time step t.

players are initialized at their drop off location.

N	q	γ_i	λ_i	α_i	Δt	T	ϵ	β
3	0.98	0.99	0.5	{0.5, 1, 1.5}	1s	120s	1e-3	40

Table 3.1: Parameters for the simulation environment.

Results and discussion. We run Algorithm 3.1 for 100 iterations, where line 5 is solved via value iteration (3.24). The two norm of x^i is shown in Figure 3.2 as a function of the algorithm iterations, where the state-action densities stabilize in about 20 steps. Performance is evaluated by: 1) the expected number of collisions, 2) the expected packages delivery time, 3)worst package delivery time. The results for 100 random trials are visualized in Figures 3.3 and 3.4.

We compare the *jointly optimal congestion-free wait time* computed using Algorithm 3.1 to the shortest wait time available in the absence of opponents. Each path is the number

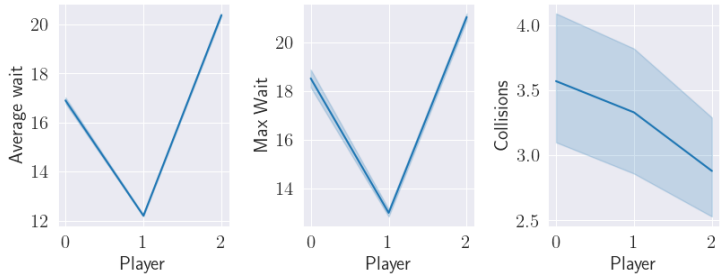


Figure 3.4: Average waiting time per package, worst case waiting time per package, and average number of collisions in T for each player.

of steps to complete the drop off-pick up-drop off cycle. Based on the pick-up and drop-off locations, each player's shortest wait time without opponents is 16, 12, 20 respectively. This matches well with the average wait time shown in Figure 3.4.

We set the player impact factors as $\{0.5, 1, 1.5\}$ as in Table 3.1. From Figure 3.4, the impact factors directly correlate with the rate of collision players experience. Player 0 impacts congestion the least and is the least sensitive to congestion. As a result, it encountered the most collisions. Player 2 impacts congestion the most and is the most sensitive to congestion. As a result, it encountered the least collisions. The collision rate is spread out evenly over $\mathcal{T}$ (Figure 3.3).

3.5 Collision Risk Reduction in Air Traffic via Multi-linear MDP Congestion Game

Air traffic management operates under high operational uncertainty and strict collision risk requirements [124]. Presently, air traffic authorities centrally plan deterministic trajectories and rely on human controllers to resolve local collision risks. Using the co-occupation probabilities (3.18), we formulate a nonconvex MDP congestion game to embed real-time operation uncertainty into path planning and find global collision risk-free trajectories

Individual aircraft MDP. We use an MDP to model the probability distribution of an aircraft following a deterministic flight plan under operational uncertainty. Aircraft i's flight plan is $\{(w_t^i, f_t^i) \ | t \in \mathcal{T}^i\}$, where w_t^i are discrete waypoints used by the European Union Aviation Safety Agency (EASA), f_t^i are discrete flight levels from 0 (sea level) to 450 (45000 feet) in increments of 50, and $\mathcal{T}^i$ are timestamps of the waypoints and flight-levels.

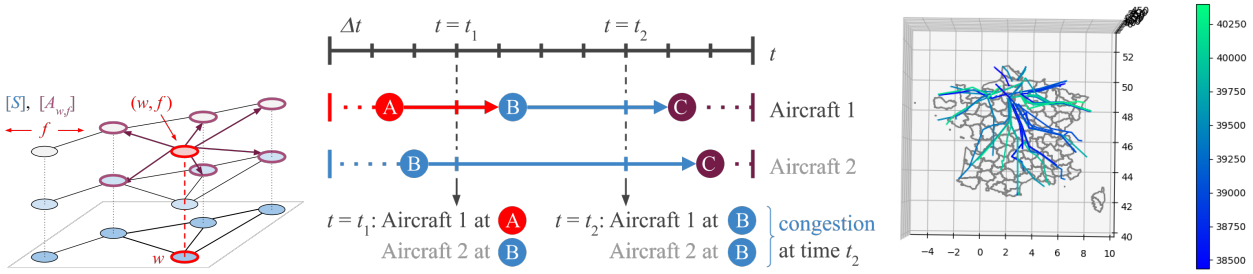


Figure 3.5: Left: Airspace state-action definitions. Center: Interval-based congestion computation. Right: Expected aircraft trajectories without congestion costs D^i (3.18). Colors correspond to time.

Time horizon. Aircraft i's time horizon is given by $\mathcal{T}^i \cup \{t_L + b\Delta t_{int} \,|0 \leq b \leq B\}$, where $\mathcal{T}^i$ is from the flight plan, t_L is the planned landing time, and $\Delta t_{int}, B \in \mathbb{N}$ are user-defined parameters.

States. Each state $(w, f) \in [S]$ consists of a waypoint w and a flight level f, as shown in Figure 3.5.

Actions. At state (w, f), actions correspond to reaching one of (w, f)'s neighbors in the next time step. The set of neighbors is given by $\mathcal{N}(w, f) = \big\{(w', f') \mid w' \in \mathcal{N}(w),\ f' \in \{f - 50, f, f + 50\},\ 0 \leq f' \leq 450\big\}$, where $\mathcal{N}(w)$ is the set of reachable waypoints from w. Aircraft cannot loiter at (w, f). The action of going to (w', f') is $a_{w',f'}$, such that $[A_{w,f}] = \{a_{w',f'} \mid (w', f') \in \mathcal{N}(w, f)\}$.

Transition Dynamics. Under action $a_{w',f'}$ from (w, f), an aircraft has β probability of reaching (w', f') and $1 - \beta$ probability of diverting to another state in $\mathcal{N}(w, f)$.

Cost: Each state-action pair $(w, f, a_{w',f'})$ has a flight-dependent **deviation cost**, given by

$$C^i_{t,w,f,a} = d(w^i_t, w) + \alpha_f |f - f^i_t| + L(t, w, f),\ \forall (w, f) \in [S], a \in [A_{w,f}], \tag{3.30}$$

where (w^i_t, f^i_t) is the aircraft i's planned location at t, $d(v, w) \in \mathbb{N}$ is the number of edges between v and w, $\alpha_f \in \mathbb{R}$ is a user defined parameter, and $L : \mathcal{T}^i \times [S] \mapsto \mathbb{R}$ is a tardiness cost. If the aircraft plans to land at (w_T, f_T, T), then $L(t, w, f) = 0$ if $(w, f) = (w_T, f_T)$ or $t \leq T$, else $L(t, w, f) = c_{tardy}(t - T)$. The expected cost under the flight plan is zero and strictly positive otherwise. Therefore, aircraft are inclined to follow the flight plan in the absence of congestion.

Based on the individual aircraft MDP model, we build an MDP congestion game for the

air traffic plan over France on July 3^{rd}, 2017. Between timestamps 39000 and 41000, 75 planes left the Paris airports CDG and ORY to various destinations as shown in Figure 3.5. The collision risks D^i and G^i (3.18) can be interpreted as standard aircraft radial/vertical separation and longitudinal separation [24], respectively. In our simulations, only D^i increases congestion costs.

Interval-based collision risk computation. Since each aircraft's time stamp is unique, we compute the congestion for time intervals. As shown in Figure 3.5, aircraft whose time stamp fall into the interval $[t_k, t_k + \Delta t_{cong})$ will contribute to the congestion in time interval k.

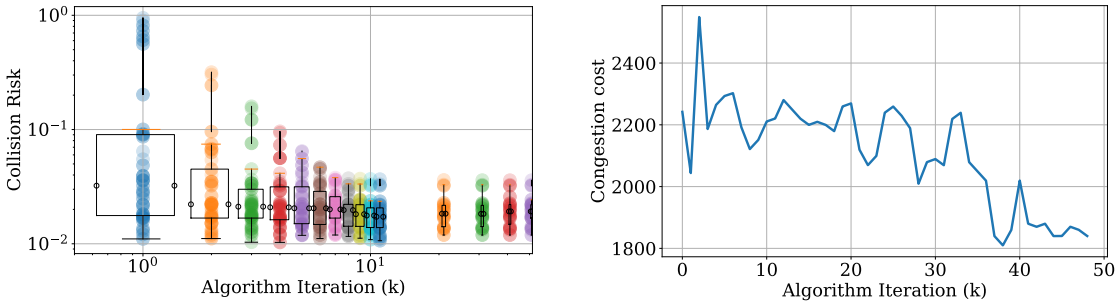


Figure 3.6: Left: Collision risk as a function of Frank-Wolfe algorithm iteration. Right: Congestion cost $\sum_{i,t,s,a} kD^i_{tsa}(x)$ (3.19) as a function of Frank-Wolfe algorithm iteration.

Results and discussion. We build the individual MDP and the interval-based congestion costs with the following user-defined parameter values: $\Delta t_{int} = 300$, $B = 3$, $\beta = 0.95$, $\alpha_f = 10$, $c_{tardy} = 2$, $\Delta t_{cong} = 19$, and $k = 10$. First, we verify that when solved without congestion cost D^i, all individual MDPs result in expected trajectories that match the original flight plan. The results are shown in Figure 3.5. We then define collision risk as $\sum_a x^i_{tsa} D^i_{tsa}(x)$, and found that for multiple flights, the maximum collision risk at any time was greater than 10%. The overall spread of collision risks for the original flight plan is shown on the $x = 10^0$ line in Figure 3.6 left. We then augment individual costs with congestion cost D^i and solve for the Nash equilibrium via the Frank Wolfe algorithm from [69, Alg.1]. The resulting collision risks and objective values are shown in Figure 3.6. In the right figure, we see that the objective value decreases from 2200 to 1900 within the first 50 iterations. Accompanying this, we observe that the maximum collision risks drop from 94% to around 3% within the first 10 iterations of the Algorithm. Therefore, we conclude that our model was effective in

reducing uncertainty-induced collision risks.

Chapter 4

INCENTIVE DESIGN IN NON-ATOMIC MARKOV DECISION PROCESS CONGESTION GAMES

In Chapter 3, we introduced the atomic Markov game model and used it to perform scalable trajectory planning for completing individual tasks in a shared operation space. In this chapter, we use Markov games, specifically the MDP congestion game model [28], to design desirable user behavior for competitive users operating in large-scale network systems such as autonomous swarms [33], urban transportation [77], and competitive electricity markets [87]. Specifically using incentives, we show that the Markov game model can help system operators enforce user population constraints under resource uncertainty and user irrationality. For example, the Department of Transportation tolls fossil fuel vehicles on freeways to reduce travel-related carbon emissions [77]. In electricity markets, power auctions often create power grid usage violations. A power system operator tolls grid users to minimize the operational cost of addressing such violations [87].

A crucial assumption we make in this chapter is that the players are non-atomic: all the players are equipped with identical congestion costs and transition probabilities in a finite state-action space. Instead of computing individual trajectory plans, we focus on deriving population behavior trends from the group probability distribution. Using network-level trends, we study the feasibility and computation of system-level incentives for population constraints.

Contributions. In Section 4.2, we formulate the constraint satisfaction problem as a constrained optimization problem and prove that for non-atomic MDP congestion games, it is possible to design finite state-action-based incentives that result in constraint-satisfying Nash equilibria. We also show that the *minimum toll value* corresponds to the greatest lower bound of the set of constraint-enforcing incentives and that it is equivalent to the optimal dual multiplier value of the Lagrangian of the MDP congestion game.

In Section 4.4, we analyze the incentive design model with two additional assumptions: *player irrationality* and *private player costs*. These assumptions are motivated by large-scale network applications in transportation and electric grids, where human decision-makers with

unknown objectives participate in the game. We derive an adaptive incentive design scheme in which the empirical averages of the toll value, population distribution, and constraint satisfaction are all proven to converge to the true minimum toll value, optimal constrained population distribution, and zero, respectively, up to an error directly correlated with the sub-optimality of the players. We apply our algorithm to adaptively enforce constraints on a fictitious group of irrational ride-hail drivers in Manhattan, New York City using historical neighborhood-based ride-demand data from December 2019.

4.1 Non-atomic MDP Congestion Game with Irrational Players

The non-atomic MDP congestion game was first introduced in [28]. Similar to the atomic MDP congestion game model from Chapter 3, the non-atomic game analyzes competitive resource-sharing players following MDP dynamics. The crucial difference is that the players' joint state-action distribution becomes a *continuous population distribution*, where each player has identical transition dynamics, cost functions, and impacts on the overall congestion (see [59, Sec.2] for details). Presently, we do not analyze the policies of individual players and only study the resulting continuous population distribution. We quickly summarize some key concepts from Section 2.4 that are used in this chapter.

Population state-action distribution. A continuous population of players with total mass $M \in \mathbb{R}_+$ follows identical MDP dynamics (2.13) on a finite state-action space, denoted as $[S] \times [A]$. We denote this population distribution by $y \in \mathbb{R}_+^{(T+1)SA}$ (2.24).

MDP dynamics. The transition dynamics are given by $P \in \mathbb{R}^{T\times S\times S\times A}$ and explained in detail in Section 2.2. The set of feasible population distributions, $\mathcal{Y}(P,p)$, is given by

$$\mathcal{Y}(P,p) = \left\{ y \in \mathbb{R}_+^{(T+1)SA} \,\middle|\, \textstyle\sum_a y_{tsa} = \sum_{s',a} P_{(t-1)ss'a} y_{(t-1)s'a},\ \sum_a y_{0sa} = p_s, s \in [S], t \in [T] \right\}. \tag{4.1}$$

Player costs and Q-value iteration. At time t, each player incurs a cost as a function of y, $\ell_{tsa} : \mathbb{R}^{(T+1)SA} \to \mathbb{R}$. A player's expected cost-to-go at (t,s,a) is its Q-value function (2.27), recursively defined as

$$Q_{tsa}(y) = \begin{cases} \ell_{tsa}(y) & t = T \\ \ell_{tsa}(y) + \sum_{s'} P_{ts'sa} \min_{a' \in [A]} Q_{t+1,s'a'}(y) & t \in [T] \end{cases} \tag{4.2}$$

Identical to the atomic scenario, each player aims to minimize their own Q-value function by

choosing optimal actions at each state. We recall the definition of MDP Wardrop equilibrium from Definition 2.4.

Definition 4.1 (MDP Wardrop Equilibrium [27]) *A population distribution $y^\star \in \mathcal{Y}(P,p)$ (4.1) is an MDP Wardrop equilibrium if*

$$y^\star_{tsa} > 0 \Rightarrow a \in \operatorname*{argmin}_{a' \in [A]} Q_{tsa'}(y^\star), \quad \forall (t,s,a) \in \mathcal{T} \times [S] \times [A] \tag{4.3}$$

The set of MDP Wardrop equilibria is denoted by $\mathcal{W}(\ell)$.

If ℓ is a continuous vector-valued function and there exists an explicit potential function F satisfying $\nabla F(y) = \ell(y)$, then the MDP congestion game is a *potential game* [90].

Proposition 4.1 *[27, Thm.1.3] Given a continuous population of players following identical MDP transition dynamics $P \in \Delta_S^{TSA}$ with initial population distribution $p \in \Delta_S$ and game cost vector ℓ, if a potential function F satisfies*

$$\nabla F(y) = \ell(y), \quad F : \mathbb{R}^{(T+1)\times[S]\times[A]} \mapsto \mathbb{R}, \tag{4.4}$$

then the MDP Wardrop equilibrium of the MDP congestion game with player transition dynamics $P \in \Delta_S^{TSA}$ and initial population distribution $p \in \Delta_S$ is given by the optimal argument of

$$\min_y F(y), \ \ s.t. \ \ y \in \mathcal{Y}(P,p). \tag{4.5}$$

We can characterize the suboptimality of any feasible population distribution within $\mathcal{Y}(P,p)$ by the difference in its potential function value from the potential value achieved by any MDP Wardrop equilibria.

Definition 4.2 (ϵ-MDP Wardrop equilibrium) *For a game with the cost vector ℓ, the potential function F (4.4), MDP Wardrop equilibrium $y^\star$ (4.3), and $\epsilon > 0$, the set of ϵ-MDP Wardrop equilibria is given by*

$$\mathcal{W}(\ell,\epsilon) := \{\hat{y}(\epsilon) \in \mathcal{Y}(P,p) \mid F\big(\hat{y}(\epsilon)\big) \leq F(y^\star) + \epsilon\}. \tag{4.6}$$

Among cost vectors ℓ that have explicit potential functions, we focus on those that are strongly convex [18, Eqn B.6].

Assumption 4.1 *The cost vector ℓ has an explicit potential function F (4.4) that is α-strongly convex for all $y \in \mathcal{Y}(P,p)$.*

$$\nabla_y \ell(y) \succeq \alpha I_{M\times M} \in \mathbb{R}^{M\times M},\ M = (T+1)SA,\ \alpha > 0.$$

Assumption 4.1 implies that congestion occurs in *all* state-action costs. To model games in which *some* state-action costs are constant, we can approximate the constant costs by increasing functions with infinitesimal growth rates.

Remark 4.1 *If at each $(t,s,a) \in \mathcal{T}\times[S]\times[A]$, $\ell_{tsa} : \mathbb{R}_+ \mapsto \mathbb{R}\ \forall\ (t,s,a) \in [T+1]\times[S]\times[A]$ is scalar functions of a single input y_{tsa}, then Assumption 4.1 implies that each ℓ_{tsa} strictly increases and satisfies $\alpha|y_{tsa} - y'_{tsa}| \leq |\ell_{tsa}(y_{tsa}) - \ell_{tsa}(y'_{tsa})|$, and that its potential function is given by*

$$F_0(y) = \textstyle\sum_{t,s,a} \int_0^{y_{tsa}} \ell_{tsa}(u)du. \tag{4.7}$$

For an ϵ-MDP Wardrop equilibria $\hat{y}(\epsilon)$, Assumption 4.1 implies $\|\hat{y}(\epsilon) - y^\star\|_2^2 \leq \frac{2\epsilon}{\alpha}$.

4.2 Constraining Rational Players with Known Congestion Costs

In this section, we formulate system-level, affine population constraints and relate the inexact oracle of the tolled MDP congestion game to an ϵ-MDP Wardrop equilibrium. Affine constraints cover many design requirements for large-scale networks. For example, meeting carbon emission goals in transportation and minimizing generator initialization costs in power grids are affine constraints on the fossil fuel vehicle population and local grid voltages, respectively.

Suppose we have an MDP congestion game defined with transition dynamics, initial population distribution, and player costs, such that the potential form is given by (4.5). The social planner may want to shift the equilibrium population distribution to satisfy the constraints of the form

$$g^i(y) \geq 0,\ g^i : \mathbb{R}^{(T+1)SA} \mapsto \mathbb{R}, y \in \mathcal{Y}(P,p) \quad \forall\ i \in \mathcal{C} \tag{4.8}$$

where g^i are continuously differentiable concave functions in the player population distribution (4.1).

The social planner cannot explicitly constrain players' behavior, but rather seeks to add incentive functions $\{f^i_{tsa}\}_{i\in\mathcal{I}}$ to the cost functions $\ell(y)$ to shift the equilibrium to be within

the constrained set defined by (4.8). The modified cost functions have the form

$$\hat{\ell}_{tsa}(y) = \ell_{tsa}(y) + \sum_{i \in \mathcal{C}} f^i_{tsa}(y) \tag{4.9}$$

When the game potential function $F : \mathbb{R}^{(T+1)SA} \mapsto \mathbb{R}$ (4.4), player transition dynamics $P \in \Delta_S^{TSA}$ (4.1), and initial state distribution $p \in \Delta_S$ (4.1) are known, the social planner first solves the following constrained optimization problem to determine the incentive functions.

$$\begin{aligned}
&\min_{y} \; F(y) \\
&\text{s.t.} \sum_{a \in [A]} y_{sa} = \sum_{s' \in [S]} \sum_{a \in [A]} P_{(t-1)ss'a} y_{(t-1)s'a}, \; \forall \; t \in [T], \\
&\quad \sum_{a \in \mathcal{A}} y_{0sa} = p_s, \; \forall \; s \in [S], \\
&\quad y_{tsa} \geq 0, \; \forall \; s \in [S], \; a \in [A], \; t \in \mathcal{T}, \\
&\quad g_i(y) \leq 0, \; \forall \; i \in \mathcal{C},
\end{aligned} \tag{4.10a}$$

The social planner can then compute the incentive functions $f^i_{tsa} : \mathbb{R}^{(T+1)SA} \mapsto \mathbb{R}$ as

$$f^i_{tsa}(y) = \tau_i^\star \frac{\partial g^i}{\partial y_{tsa}}(y), \quad \forall (t, s, a, i) \in \mathcal{T} \times [S] \times [A] \times \mathcal{C}, \tag{4.11}$$

where $\{\tau_i^\star \in \mathbb{R}_+\}_{i \in \mathcal{C}}$ are the optimal Lagrange multipliers associated with the additional constraints (4.10a).

The following theorem shows that the Wardrop equilibrium of the MDP congestion game with the tolled player costs in (4.9) satisfies the new constraints in (4.10a).

Theorem 4.1 *Consider a MDP congestion game (4.5) with costs $\ell(y)$ with a potential function F (4.4) that is strictly convex. If $y^\star$ is a MDP Wardrop equilibrium for the tolled MDP congestion game with cost functions $\hat{\ell}_{tsa}(y) = \ell_{tsa}(y) + \sum_{i \in \mathcal{C}} \tau_i^\star \frac{\partial g^i}{\partial y_{tsa}}(y)$, then $y^\star$ also solves (4.10) and thus satisfies the additional constraints (4.8).*

Proof: The Lagrangian of (4.10) is given by

$$L(y,\mu,V,\tau) = F(y) - \sum_{tsa}\mu_{tsa}y_{tsa} + \sum_i \tau^i g^i(y) + \sum_{t=0}^{T-1}\sum_s \Big(\sum_{as'} P_{t,ss'a}y_{t,s'a} - \sum_a y_{t+1,sa}\Big)V_{t+1,s}$$
$$+ \sum_s \Big(p_s - \sum_a y_{1sa}\Big)V_{1s} \tag{4.12}$$

and note that by strict convexity, $\min\limits_{y\geq 0}\max\limits_{\mu\geq 0, V,\tau\geq 0} L(y,\mu,V,\tau)$ has unique solution, which we denote by $(y^\star,\mu^\star,V^\star,\tau^\star)$. We then note that

$$\bar{F}(y) = F(y) + \sum_i (\tau^i)^\star g^i(y) \tag{4.13}$$

is a potential function for the MDP congestion game with desired modified rewards. Since $F(y)$ is strictly concave, $g^i(y)$ is concave, and $(\tau^i)^\star$ is positive, $\bar{F}(y)$ is strictly concave. The equilibrium for the MDP congestion game with modified rewards can be computed by solving (4.10) with $\bar{F}(y)$ as the objective.

The Lagrangian for (4.10) with $\bar{F}(y)$ is given by $\bar{L}(y,\mu,V) = L(y,\mu,V,\tau^\star)$. Again by strict convexity,

$$\min_{y\geq 0}\max_{\mu\geq 0,V}\bar{L}(y,\mu,V) = \min_{y\geq 0}\max_{\mu\geq 0,V} L(y,\mu,V,\tau^\star)$$

has a unique solution which we denote as $(\bar{y}^\star,\bar{\mu}^\star,\bar{V}^\star)$. It follows that $\bar{y}^\star = y^\star$. Thus the game equilibrium with modified rewards, $\bar{y}^\star$ satisfies desired constraints. ■

For the social planner, Theorem 4.1 has the following interpretation: in order to impose constraints of form (4.8) on an MDP congestion game, the planner could solve the constrained game (4.10) for optimal dual variables $\tau^\star$ and offer incentives of form (4.11).

4.3 Incentivizing Seattle Ride Hail Drivers to Satisfy Rider Demands

We demonstrate how a ride hail company can take on the role of social planner and shift the equilibrium of the driver game for the following objectives: 1) ensuring minimum driver density in various neighborhoods (Section 4.3.2), 2) improving the social welfare (Section 4.3.3).

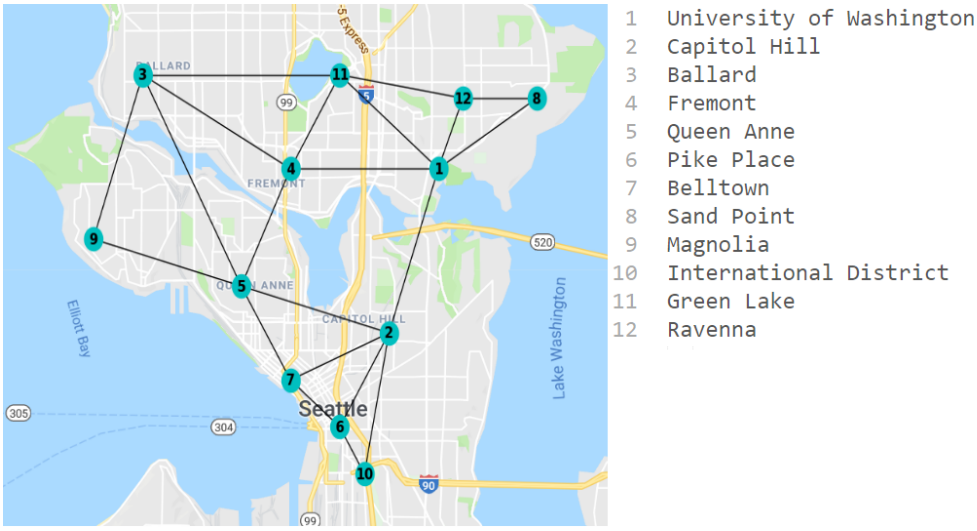


Figure 4.1: State representation of metro Seattle.

4.3.1 Ride Hail Driver Solving an MDP

Consider a fleet of ride hail drivers in metro Seattle. The rational drivers seek to earn maximum individual profit by repeatedly working Friday nights. Assume that the rider is constant for each Friday night. The *time interval* is set to $\Delta T = 15$ minutes over a period of 20 intervals, i.e. the average time for a ride, after which the driver needs to take a new action.

We model Seattle's individual neighborhoods as an abstract set of *states*, $s \in [S]$, as shown in Fig 4.1. Adjacent neighborhoods are connected by edges. The following states are characterized as residential: 'Ballard' (3), 'Fremont' (4), 'Sand Point' (8), 'Magnolia' (9), 'Green Lake' (11), 'Ravenna' (12). Assume drivers have equal probabilities of starting from any of the residential neighborhoods.

Because drivers cannot see a rider's destination until after accepting a ride, the game has stochastic transition dynamics. At each state s, drivers can choose from two actions: 1) a_r, wait for a rider in s, or 2) a_{s_j}, transition to an adjacent state s_j. When choosing a_r, we assume that the driver will eventually pick up a rider, although it may take longer if there are many drivers waiting for a rider in that neighborhood. Longer wait times increase the driver's cost for choosing a_r.

On the other hand, there are two possible scenarios when drivers choose a_{s_j}. The driver

either drives to s_j and pays the travel costs without receiving a fare, or picks up a rider in s_i. We allow the second scenario with a small probability to model the possibility of drivers deviating from their predetermined strategy during game play.

The *probability of transition* for each action at state s_i are given below, where N_i denotes the set of neighboring states, and $|N_i|$ the number of neighboring states for state s_i.

$$P(s,a,s_i) = \begin{cases} \frac{1}{|N_i|+1}, & \text{if } s \in N_i, a = a_r \\ \frac{1}{|N_i|+1}, & \text{if } s = s_i, a = a_r \\ \frac{0.1}{|N_i|}, & \text{if } s \in N_i, s \neq s_j, a = a_{s_j} \\ 0.9, & \text{if } s \in N_i, s = s_j, a = a_{s_j} \\ 0, & \text{otherwise} \end{cases}$$

The *cost function* for taking each action is given by

$$\ell_{tsa}(y_{tsa}) = \mathbb{E}_{s'}\left[c^{\text{trav}}_{ts's} - m_{ts's}\right] + c^{\text{wait}}_t \cdot y_{tsa} = \sum_{s'} P_{ts'sa}\left[c^{\text{trav}}_{ts's} - m_{ts's}\right] + c^{\text{wait}}_t \cdot y_{tsa}$$

where $m_{ts's}$ is the monetary cost for transitioning from state s to s', $c^{\text{trav}}_{ts's}$ is the travel cost from state s to s', c^{wait}_t is the coefficient of the cost of waiting for a rider. We compute these various parameters as

$$m_{ts's} = \big(\text{Rate}\big) \cdot \big(\text{Dist}\big) \tag{4.14a}$$

$$c^{\text{trav}}_{ts's} = \tau \underbrace{\big(\text{Dist}\big)}_{\text{mi}} \underbrace{\big(\text{Vel}\big)^{-1}}_{\text{hr/mi}} + \underbrace{\big(\tfrac{\text{Fuel}}{\text{Price}}\big)}_{\$/\text{gal}} \underbrace{\big(\tfrac{\text{Fuel}}{\text{Eff}}\big)^{-1}}_{\text{gal/mi}} \underbrace{\big(\text{Dist}\big)}_{\text{mi}} \tag{4.14b}$$

$$c^{\text{wait}}_{ta} = \begin{cases} \tau \cdot \Big(\underbrace{\tfrac{\text{Customer}}{\text{Demand Rate}}}_{\text{rides/hr}}\Big)^{-1}, & \text{if } a = a_r \\ \epsilon_{tsa_{s'}}, & \text{if } a = a'_s \end{cases} \tag{4.14c}$$

where $\epsilon_{tsa_{s'}}$ is the congestion effect from drivers who all decide to traverse from s to s', and τ is a time-money tradeoff parameter, computed as $\left(\frac{\text{Rate}\cdot D_{ave}}{\text{Time Step}}\right)$, where the average trip length, D_{ave}, is equivalent to the average distance between neighboring states. The values that are independent of specific transitions are listed in Table 4.1.

Rate	Velocity	Fuel Price	Fuel Eff	τ	D_{ave}
\$6 /mi	8 mph	\$2.5/gal	20 mi/gal	\$27 /hr	1.25 mi

Table 4.1: Parameters for the driver reward function.

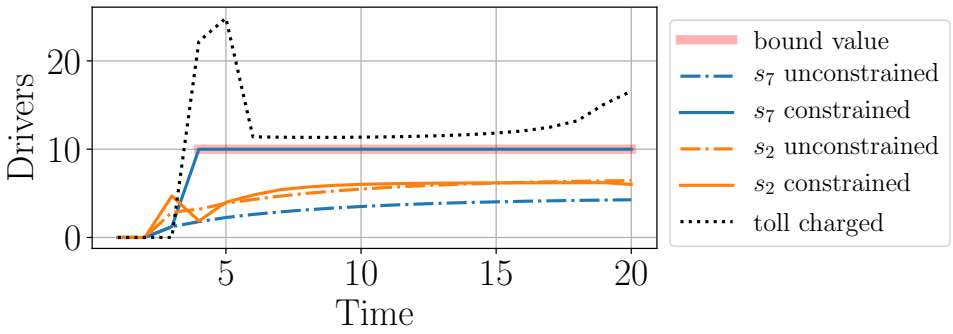


Figure 4.2: Optimal state density of (4.10).

4.3.2 Ensuring Minimum Driver Density

To ensure rider satisfaction, the ride hail company aims to achieve a minimum driver density of 10 drivers in 'Belltown', $s = 7$, a neighborhood with highly variable rider demand. To this end, they solve the optimization problem in (4.10) where (4.10a) for $t \in \{3, \dots, T\}$, $s = 7$, take on form $g_i(y) = \sum_a y_{tsa} - 10$. The modified rewards from Theorem 4.1 are given by $\hat{\ell}_{tsa}(y) = \ell_{tsa}(y_{tsa}) + \tau^\star_{ts}$, where each $\tau^\star_{ts}$ is the optimal dual variable corresponding to each new constraint.

The optimal population distribution in 'Belltown' (state 7) and an adjacent neighborhood, 'Capitol Hill' (state 2), are shown in Fig. 4.2. The imposed constraints also affect the optimal population distribution of adjacent states, as shown by the population distribution of Capitol Hill. Note that the incentive $\tau^\star_{ts}$ is applied to all actions of state s. Furthermore, if the solution to the unconstrained problem is feasible for the constrained problem, then $\tau^\star_{ts} = 0$—i.e. no incentive is offered. We simulate drivers' behavior with Algorithm 3.1, as a function of decreasing termination tolerance ϵ. In Fig. 4.3, the result shows that the optimal population distribution from the FW algorithm converges to Wardrop equilibrium as the approximation tolerance ϵ decreases.

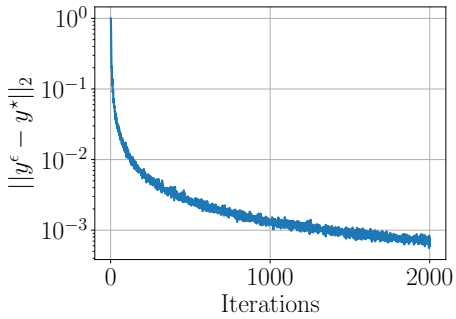


Figure 4.3: Convergence of y^ϵ to $y^\star$ normalized by $\|y^\star\|_2$.

4.3.3 Increasing Average Social Welfare of Ride Hail Drivers

In most networks with congestion effects, the population does not achieve the maximum *social welfare*, which can be achieved by optimizing (4.5) with objective $J(y) = \sum_{t\in[T]} \sum_{s\in[S]} \sum_{a\in[A]} y_{tsa}\ell_{tsa}(y)$. In general, a gap exists between $J(x^\star)$ and $J(y^\star)$, where $y^\star = \{y^\star_{tsa}\}$ is the optimal solution to (4.5), and $x^\star = \{x^\star_{tsa}\}$ is the optimal solution to (4.5) with swapped objective $J(y)$.

The typical approach to closing the social welfare gap is to impose mass-dependent incentives. An alternative method, perhaps under-explored, is to impose constraints. As opposed to congestion-dependent taxation methods for improving social welfare [106, 15], constraint-generated tolls are congestion independent.

We can compare the two distributions and generate upper/lower bound constraints with an ϵ threshold—see Algorithm 4.1 for the constraint selection method. The number of constraints increases with decreasing ϵ. Since the objective function in (4.5) is continuous in y_{tsa}, as ϵ approaches zero, the objective will also approach the socially optimal. In Fig. 4.4, we compare the optimal social welfare to the social welfare at Wardrop equilibrium of the unconstrained congestion game, modeled in Section 4.3.1, as a function of the population size. We use CVXPY [35] to solve the optimization problem.

We utilize Algorithm 4.1 to generate incentives for the congestion game. Then, we simulate (4.10) and compare the game output to the social objective in Fig. 4.4. For a population size of 3500, there is a discernible gap between the social and user-selected optimal values. Note that with only 200 (t, s, a) constraints, the gap between the social optimal and the user-selected equilibrium is already less than 5%.

An interesting question is how much of the total market worth is affected by the incentives. In Fig. 4.5, we demonstrate how payouts vary based on the number of constraints imposed.

Algorithm 4.1 Constraint Generation

Input: $x^\star$, $y^\star$.
Output: $\mathcal{U} = \{(u_i, t, s, a) \in \mathbb{R} \times [T] \times [S] \times [A]\}$
$\mathcal{L} = \{(l_i, t, s, a) \in \mathbb{R} \times [T] \times [S] \times [A]\}$
for each $s \in [S]$, $a \in [A]$, $t \in [T]$ **do**
if $y^\star_{tsa} - x^\star_{tsa} > \epsilon$ **then**
$(x^\star_{tsa}, t, s, a) \to \mathcal{U}$
else if $y^\star_{tsa} - x^\star_{tsa} < \epsilon$ **then**
$(x^\star_{tsa}, t, s, a) \to \mathcal{L}$
end if
end for

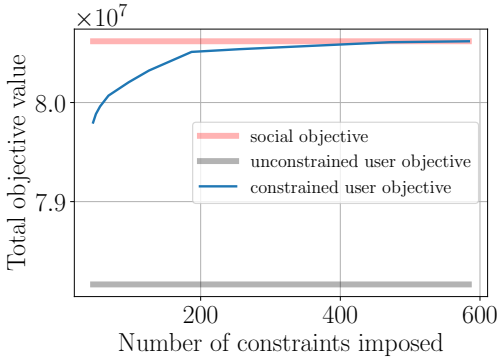


Figure 4.4: Constrained user optimal as a function of ϵ.

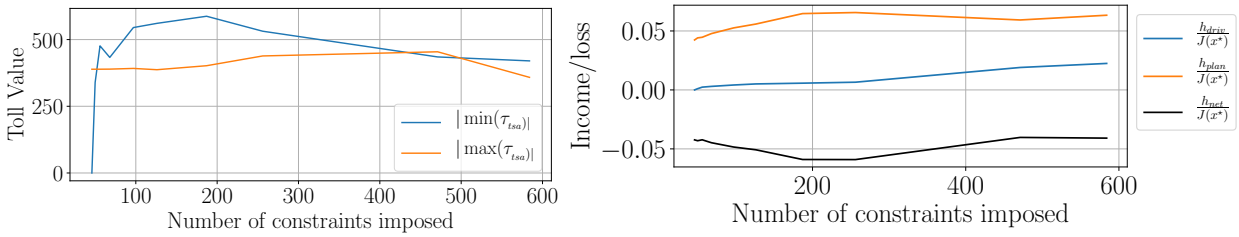


Figure 4.5: Maximum and minimum tolls as a function of the number of constraints (top). The income/loss required to increase social welfare as a function of constraints imposed (bottom).

Let $(\cdot)_- = \min\{0, \cdot\}$ and $(\cdot)_+ = \max\{0, \cdot\}$. The total payout from the drivers to the social planner and vice versa are given by $h_{\text{driv}} = \sum_{tsa} y_{tsa}|(\tau_{tsa})_-|$ and $h_{\text{plan}} = \sum_{tsa} y_{tsa}(\tau_{tsa})_+$. The net revenue the social planner receives from tolls is $h_{\text{net}} = \sum_{tsa} y_{tsa}(\tau_{tsa}) = h_{\text{plan}} - h_{\text{driv}}$. Fig. 4.5(b) shows how these quantities change as the total number of constraints is increased.

4.4 Constraining Irrational Players With Unknown Congestion Costs

In this section, we formulate system-level constraints using affine population distribution inequalities and relate the inexact oracle of the tolled MDP congestion game to an ϵ-MDP Wardrop equilibrium. Affine constraints cover many design requirements for large-scale networks. For example, certain roads may pass through residential neighborhoods in a city's traffic network, and a city planner may wish to limit traffic levels to ensure residents' well-being artificially.

Definition 4.3 (Affine Constraints) *The set of population distribution constraints is given by*

$$\mathcal{C} = \left\{ y \in \mathbb{R}_+^{(T+1)SA} \mid Ay - b \leq 0 \right\} \tag{4.15}$$

where $A \in \mathbb{R}^{C \times (T+1)SA}$, $b \in \mathbb{R}^C$, and $0 \leq C < \infty$ denotes the total number of constraints imposed.

Let $A_i \in \mathbb{R}^{(T+1)SA}$ be the i^{th} row of A. Instead of searching over all possible tolls, we only consider tolls of the form $\tau_i A_i \in \mathbb{R}^{(T+1)SA}$ for $\tau_i \in \mathbb{R}_+$. This formulation ensures that τ_i only affects the (t, s, a) component of ℓ when $A_{i,tsa}$ is non-zero, where the toll magnitude is controlled by τ_i. We denote the toll-augmented game cost vector as

$$\ell_\tau(y) := \ell(y) + A^\top \tau, \quad \tau \in \mathbb{R}_+^C. \tag{4.16}$$

When ℓ satisfies Assumption 4.1, we denote ℓ_τ's potential as $L(\cdot, \tau)$, such that $\nabla_y L(y, \tau) = \ell_\tau$ and L augments F (4.4) as

$$L(y, \tau) = F(y) + \tau^\top (Ay - b). \tag{4.17}$$

Given a toll value τ, the toll-augmented game $d(\tau)$ and the tolled MDP Wardrop equilibrium, $y_\tau \in \mathcal{W}(\ell_\tau)$, are given by

$$d(\tau) = \min_{y \in \mathcal{Y}(P,p)} L(y,\tau), \quad y_\tau \in \operatorname*{argmin}_{y \in \mathcal{Y}(P,p)} L(y,\tau). \tag{4.18}$$

Under cost vector (4.16), any feasible affine population constraint will hold for large values of τ [73]. We specifically want to compute the minimum toll value to enforce $\mathcal{C}$ (4.15) on the MDP Wardrop equilibrium of the toll-augmented game.

Definition 4.4 (Minimum toll value) *Given a constraint set $\mathcal{C}$ (4.15), the minimum toll value $\tau^\star \in \mathbb{R}^C_+$ is the smallest non-negative toll that ensures that the MDP congestion game has constraint-satisfying MDP Wardrop equilibria—i.e.,*

$$\tau^\star = \min \left\{ \tau \in \mathbb{R}^C_+ \mid \mathcal{W}(\ell_\tau) \subseteq \mathcal{C} \right\}. \tag{4.19}$$

The minimum toll value exists under the following sufficient condition [73].

Proposition 4.2 *[73]: When $\mathcal{C}$ is convex, $\mathcal{C} \cap \mathcal{Y}(P, p_0)$ is non-empty, and the cost vector ℓ satisfies Assumption 4.1, a unique minimum toll value $\tau^\star$ (4.19) maximizes $d(\tau)$.*

$$\tau^\star = \operatorname*{argmax}_{\tau \in \mathbb{R}^C_+} \left[\min_{y \in \mathcal{Y}(P,p)} L(y,\tau) \right] = \operatorname*{argmax}_{\tau \in \mathbb{R}^C_+} \; d(\tau). \tag{4.20}$$

When ℓ is known, (4.20) directly computes $\tau^\star$. When ℓ is unknown, we cannot explicitly solve for either $\tau^\star$ or $d(\tau)$.

Problem 4.1 *For MDP congestion games with unknown but strictly increasing congestion costs, find the minimum toll value $\tau^\star$ (4.19) that ensures the resulting MDP Wardrop equilibrium $y_{\tau^\star}$ (4.18) satisfies the desired affine constraints $\mathcal{C}$ (4.15).*

As summarized in Figure 4.6, we compute $\tau^\star$ by querying the ϵ-MDP Wardrop equilibria of $d(\tau)$, $\hat{y}_\tau(\epsilon)$, and performing gradient descent with $\hat{y}_\tau(\epsilon)$. To see how the ϵ-MDP Wardrop equilibrium of a tolled game induces an inexact oracle for $\nabla d(\tau)$, we first derive the analytical expression of $\nabla d(\tau)$.

Proposition 4.3 *If the cost vector ℓ satisfies Assumption 4.1 and $\mathcal{C}$ satisfies Definition 4.3, d (4.18) has the following properties.*

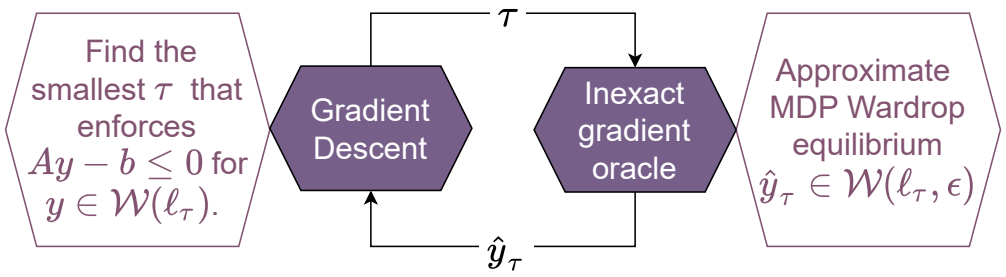


Figure 4.6: Using approximate MDP Wardrop equilibrium, we perform inexact gradient descent on τ to find the minimum toll value.

- *d is concave.*
- *d is $\bar{\alpha}$-smooth with $\bar{\alpha} = \frac{\|A\|_2^2}{\alpha}$. I.e., for any $\sigma, \tau \in \mathbb{R}^C$,*

$$d(\tau) + \nabla d(\tau)^\top (\sigma - \tau) - \tfrac{\bar{\alpha}}{2} \|\sigma - \tau\|_2^2 \leq d(\sigma). \tag{4.21}$$

- *Let y_τ be defined as (4.18), then $\nabla d(\tau)$ is given by*

$$\nabla d(\tau) = A y_\tau - b. \tag{4.22}$$

Proof: $d(\tau)$ is the dual function of the optimization problem $\min_{y \in \mathcal{Y}(P,p)} F(x)$ s.t. $Ay \leq b$. As the dual function of a convex optimization problem with linear constraints, it is concave [18, Prop 5.1.2]. The smoothness constant of $d(\tau)$ follows from [92, Thm 1], where α is the strong convexity factor of F_0. Finally, the computation of $\nabla d(\tau)$ follows directly from [18, Prop.B.25]. ■

When the costs ℓ are unknown, we can compute the ϵ-MDP Wardrop equilibrium via learning algorithms [145, 59, 142]. When applied to tolled games, the ϵ-MDP Wardrop equilibria form *ϵ-inexact oracles* of d.

Definition 4.5 (ϵ-inexact oracle) *The ϵ-inexact oracles of $\nabla d(\tau)$ and $d(\tau)$ are given by*

$$\hat{\nabla} d(\tau) = A \hat{y}_\tau(\epsilon) - b, \quad \hat{d}(\tau) = L(\hat{y}_\tau(\epsilon), \tau), \tag{4.23}$$

where $\hat{y}_\tau(\epsilon) \in \mathcal{W}(\ell_\tau, \epsilon)$ (4.6) *is an* ϵ*-MDP Wardrop equilibrium satisfying*

$$L(\hat{y}_\tau(\epsilon), \tau) \leq L(y_\tau, \tau) + \epsilon. \tag{4.24}$$

When $\epsilon = 0$, the oracle is exact. When $\epsilon > 0$, the oracle's accuracy directly affects the concavity and the smoothness of d.

Lemma 4.1 (Concavity) *Under Assumption 4.1, all* ϵ*-MDP Wardrop equilibria* $\hat{y}_\tau(\epsilon)$ *given by* (4.24) *will generate* ϵ*-inexact oracles* (4.23) *that satisfy* $d(\sigma) \leq \hat{d}(\tau) + \hat{\nabla} d(\tau)^\top (\sigma - \tau), \quad \forall\ \sigma,\ \tau \in \mathbb{R}^C_+$.

Proof: We denote $\hat{y}_\tau(\epsilon)$ by $\hat{y}_\tau$ for simplicity. Since $\hat{y}_\tau \in \mathcal{W}(\ell_\tau, \epsilon) \subset \mathcal{Y}(P, p_0)$, using (4.18) we can show that

$$d(\sigma) \leq L(\hat{y}_\tau, \sigma). \tag{4.25}$$

Combining (4.25) with the fact that $L(\hat{y}_\tau, \sigma) = L(\hat{y}_\tau, \tau) + \hat{\nabla} d(\tau)^\top (\sigma - \tau)$, we obtain Lemma 4.1. ■

Lemma 4.2 (ϵ-approximate smoothness) *Under Assumption 4.1, all* ϵ*-MDP Wardrop equilibria* $\hat{y}_\tau(\epsilon)$ (4.24) *will generate inexact oracles* (4.23) *that satisfy*

$$\hat{d}(\tau) + \hat{\nabla} d(\tau)^\top (\sigma - \tau) - \tfrac{\|A\|_2^2}{\alpha} \|\sigma - \tau\|_2^2 \leq d(\sigma) + 2\epsilon,\ \forall\ \tau, \sigma \in \mathbb{R}^C. \tag{4.26}$$

Proof: We denote $\hat{y}_\tau(\epsilon)$ by $\hat{y}_\tau$ for simplicity and recall y_τ from (4.18). From Proposition 4.3, we know that

$$\nabla d(\tau) = \hat{\nabla} d(\tau) + A(y_\tau - \hat{y}_\tau). \tag{4.27}$$

Substituting (4.27) into (4.21), we obtain the following

$$0 \leq d(\sigma) - d(\tau) - \hat{\nabla} d(\tau)^\top (\sigma - \tau) + \tfrac{\|A\|_2^2}{2\alpha} \|\sigma - \tau\|_2^2 - \big(A(y_\tau - \hat{y}_\tau)\big)^\top (\sigma - \tau). \tag{4.28}$$

Furthermore, we can show

$$\Big|\big(A(y_\tau - \hat{y}_\tau)\big)^\top (\sigma - \tau)\Big| \leq \|\hat{y}_\tau - y_\tau\|_2 \cdot \|A\|_2 \cdot \|\sigma - \tau\|_2 \leq \tfrac{\alpha}{2} \|\hat{y}_\tau - y_\tau\|_2^2 + \tfrac{\|A\|_2^2}{2\alpha} \|\sigma - \tau\|_2^2, \tag{4.29}$$

where the first inequality is due to the Cauchy–Schwarz inequality, and the second inequality is due to the inequality of arithmetic and geometric inequalities.

Next, we note that F (4.4) and subsequently $L(y,\tau)$ (4.17) are strongly convex under Assumption 4.1. We combine this with the fact that $L(y_\tau,\tau) = d(\tau)$ from (4.18) to obtain

$$\tfrac{\alpha}{2}\left\|\hat{y}_\tau - y_\tau\right\|_2^2 \leq L(\hat{y}_\tau,\tau) - d(\tau). \tag{4.30}$$

From (4.24), $\hat{y}_\tau$ satisfies

$$L(\hat{y}_\tau,\tau) - d(\tau) \leq \epsilon. \tag{4.31}$$

Summing up (4.24), (4.28), (4.29), (4.30), and 2×(4.31), we obtain (4.26), which completes the proof. ■

Lemma 4.3 *Under Assumption 4.1, if* $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, τ^s *from Algorithm 4.2 satisfies*

$$\left\|\tau^{s+1} - \tau\right\|_2^2 \leq \left\|\tau^s - \tau\right\|_2^2 + 2\gamma\big(d(\tau^{s+1}) - L(y^s,\tau^s) + 2\epsilon^s + \hat{\nabla}d(\tau^s)^\top(\tau^s - \tau)\big), \quad \forall\ \tau \in \mathbb{R}_+^C,\ k \geq 0. \tag{4.32}$$

Proof: Given $\tau \in \mathbb{R}_+^C$, let $r^s = \left\|\tau^s - \tau\right\|_2^2$. We compute $r^{s+1} - r^s$ using the law of cosine as

$$r^{s+1} - r^s = 2(\tau^{s+1} - \tau^s)^\top(\tau^{s+1} - \tau) - \left\|\tau^{s+1} - \tau^s\right\|_2^2. \tag{4.33}$$

From line 3 of Algorithm 4.2, $\tau^{s+1} = [\tau^s + \gamma(Ay^s - b)]_+$. Using [25, Lem 3.1], the projection onto $\mathbb{R}_+^C$ implies that

$$0 \leq (\tau^s + \gamma\hat{\nabla}d(\tau^s) - \tau^{s+1})^\top(\tau^{s+1} - \tau) \tag{4.34}$$

From (4.34), we can upper bound $(\tau^{s+1} - \tau^s)^\top(\tau^{s+1} - \tau)$ and combine with (4.33) to obtain

$$r^{s+1} - r^s \leq 2\gamma\hat{\nabla}d(\tau^s)^\top(\tau^{s+1} - \tau) - \left\|\tau^{s+1} - \tau^s\right\|_2^2 \tag{4.35}$$

From Lemma 4.2, we recall

$$L(y^s,\tau^s) - d(\tau^{s+1}) - 2\epsilon^s \leq d(\tau^s)^\top(\tau^s - \tau^{s+1}) + \tfrac{\|A\|_2^2}{\alpha}\left\|\tau^{s+1} - \tau^s\right\|_2^2. \tag{4.36}$$

We can then combine (4.35) and 2γ×(4.36) to derive

$$r^{s+1} - r^s + 2\gamma(L(y^s,\tau^s) - d(\tau^{s+1}) - 2\epsilon^s) \leq 2\gamma\hat{\nabla}d(\tau^s)^\top(\tau^s - \tau) + (\tfrac{2\|A\|_2^2}{\alpha}\gamma - 1)\left\|\tau^{s+1} - \tau^s\right\|_2^2, \tag{4.37}$$

and use the fact that $\gamma\frac{2\|A\|_2^2}{\alpha} \leq 1$ to eliminate the $\left\|\tau^{s+1} - \tau^s\right\|_2^2$ term and complete the proof. ■

4.5 *Tolling Algorithm for ϵ-sub-optimal Players*

The convergence of first-order gradient methods relies on the objective's convexity and smoothness. If an inexact gradient preserves concavity and smoothness, its gradient descent will also converge [34]. In this section, we apply the same concept to tolling in MDP congestion games and analyze how ϵ-MDP Wardrop equilibria affect constraint violations.

Algorithm 4.2 Iterative toll synthesis

Input: ℓ, P, p_s, τ_0.
Output: τ^N, y^N.
1: **for** $k = 0, 1, \ldots$ **do**
2: $\quad y^k \in \mathcal{W}(\ell + A^\top \tau^k, \epsilon^k)$
3: $\quad \tau^{k+1} = [\tau^k + \gamma^k (A y^k - b)]_+$
4: **end for**

In Algorithm 4.2, we denote the k^{th} toll charged, the k^{th} ϵ-MDP Wardrop equilibrium, and the ϵ in the k^{th} ϵ-inexact oracle as τ^k, y^k, and ϵ^k, respectively. When $\epsilon^k = 0\ \forall k \in \mathbb{N}$, Algorithm 4.2 is a projected gradient ascent on $d(\tau)$ with sublinear convergence rates [25]. We analyze Algorithm 4.2's convergence when $\epsilon^k > 0$ through the following quantities:

$$\bar{\tau}^k = \frac{1}{k}\sum_{s=1}^{k} \tau^s, \quad \bar{y}^k = \frac{1}{k}\sum_{s=0}^{k-1} y^s, \quad E^k = \sum_{s=0}^{k-1} \epsilon^s, \tag{4.38}$$

where $\bar{\tau}^k$/ $\bar{y}^k$/E^k is the average toll/average ϵ-MDP Wardrop equilibrium/accumulated ϵ up to iteration k, respectively.

Theorem 4.2 *If the cost vector ℓ satisfies Assumption 4.1, and $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$ for each $k \in \mathbb{N}$, then $\bar{\tau}^k$ from (4.38) satisfies*

$$d(\tau^\star) - d(\bar{\tau}^k) \leq \tfrac{1}{k}(\tfrac{1}{2\gamma}\|\tau^0 - \tau^\star\|_2^2 + 2E^k), \tag{4.39}$$

where $\tau^\star$ is the minimum toll value (4.19). and E^k (4.38) is the total approximation error.

Proof: Our proof is inspired by [34, 91]. Let $r^s = \|\tau^s - \tau^\star\|_2^2$. From Lemma 4.3, when $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, we have

$$r^{s+1} \leq r^s + 2\gamma\big(d(\tau^{s+1}) - L(y^s, \tau^s) + 2\epsilon^s + \hat{\nabla} d(\tau^s)^\top (\tau^s - \tau^\star)\big) \tag{4.40}$$

From Lemma 4.1, we have

$$\hat{\nabla} d(\tau^s)^\top (\tau^s - \tau^\star) \leq L(y^s, \tau^s) - d(\tau^\star) \tag{4.41}$$

Summing up (4.40) and $2\gamma\times$(4.41), we obtain

$$r^{s+1} - r^s \leq 2\gamma(d(\tau^{s+1}) - d(\tau^\star) + 2\epsilon^s) \tag{4.42}$$

Summing over (4.42) for $s = 0\ldots, k-1$, we obtain $0 \leq r^k \leq r^0 - 2\gamma\sum_{s=1}^{k}\big(d(\tau^\star) - d(\tau^s)\big) + 4\gamma\sum_{s=0}^{k-1}\epsilon^s$. Finally, the concavity of d from Proposition 4.3 implies that $-kd(\bar{\tau}^k) = -kd(\sum_{s=1}^{k}\tau^s) \leq -\sum_{s=1}^{k} d(\tau^s)$. This completes the proof. ■

Remark 4.2 *When $\epsilon^k = \epsilon$ is constant, (4.39) becomes $d(\tau^\star) - d(\bar{\tau}^k) \leq \frac{1}{k}(\frac{1}{2\gamma}\|\tau^0 - \tau^\star\|_2^2) + 2\epsilon$. Similar to exact gradient descent, $\frac{1}{2\gamma}\|\tau^0 - \tau^\star\|_2^2$ converges sublinearly in k. However, the term $2\epsilon > 0$ causes a constant convergence error.*

Constraint violation of $\bar{y}^k$ (4.38) is similarly bounded.

Corollary 4.1 *If the cost vector ℓ satisfies Assumption 4.1 and $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, then the constraint violation of the average population distribution $\bar{y}^k$ from (4.38) satisfies*

$$\big\|[A\bar{y}^k - b]_+\big\|_2 \leq \tfrac{1}{\gamma k}\Big(\|\tau^\star\|_2 + \|\tau^0 - \tau^\star\|_2 + 2\sqrt{\gamma E^k}\Big). \tag{4.43}$$

Proof: We first derive an upper bound for $\big\|\tau^k\big\|_2$ and then bound the left hand side of (4.43) by $\big\|\tau^k\big\|_2$. Recall (4.42), we use $d(\tau^\star) - d(\tau^k) \geq 0$ to derive $r^{s+1} \leq r^s + 4\gamma\epsilon^s$. Summing over $s = 0, \ldots, k-1$, we have

$$\big\|\tau^k - \tau^\star\big\|_2^2 \leq \|\tau^0 - \tau^\star\|_2^2 + 4\gamma E^k. \tag{4.44}$$

Taking the square root of both sides of (4.44) and noting the identity $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we obtain

$$\big\|\tau^k - \tau^\star\big\|_2 \leq \big\|\tau^0 - \tau^\star\big\|_2 + \sqrt{4\gamma E^k}. \tag{4.45}$$

We add $\|\tau^\star\|_2$ to both sides of (4.45) and use the triangle inequality $\big\|\tau^k\big\|_2 \leq \big\|\tau^k - \tau^\star\big\|_2 + \|\tau^\star\|_2$ to obtain

$$\big\|\tau^k\big\|_2 \leq \|\tau^\star\|_2 + \big\|\tau^0 - \tau^\star\big\|_2 + \sqrt{4\gamma E^k}. \tag{4.46}$$

Next, we bound $\left\|[A\bar{y}^k - b]_+\right\|_2$ using $\left\|\tau^k\right\|_2$. From line 3 of Algorithm 4.2, $\tau^{s+1} \geq \tau^s + \gamma(Ay^s - b)$. We sum over $s = 0, \ldots, k-1$ to obtain $\tau^k \geq \tau^0 + \gamma k(A\bar{y}^k - b)$. Noting $\tau^0 \in \mathbb{R}^C_+$ can be dropped, $\gamma k[A\bar{y}^k - b]_+ \leq \tau^k$ combined with (4.46) completes the proof. ■

Remark 4.3 *With a constant error oracle $\epsilon^k = \epsilon$, the average constraint violation will still asymptotically reduce to zero.*

Unlike $\bar{\tau}^k$, E^k's effect on the average constraint violation can be reduced with *larger* step sizes as $2\sqrt{E^k\gamma^{-1}}$. We note that Corollary 4.1 shows that Algorithm 4.2 is not appropriate for enforcing safety-critical system constraints.

Algorithm 4.2 also ensures that the average population distribution $\bar{y}^k$ (4.38) converges to the optimal equilibrium for $\tau^\star$.

Theorem 4.3 *If the cost vector ℓ satisfies Assumption 4.1 and $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, then the average player population distribution given by $\bar{y}^k$ (4.38) satisfies*

$$\left\|\bar{y}^k - y^\star\right\|_2^2 \leq \tfrac{\alpha}{2\gamma k} D(\tau^0, \tau^\star, E^k), \tag{4.47}$$

where $\tau^\star$ is the minimum toll value, $y^\star$ is the optimal population distribution for $d(\tau^\star)$, and $D(\tau^0, \tau^\star, E^k)$ is given by

$$D(\tau^0, \tau^\star, E^k) = \max\left\{\tfrac{1}{2}\left\|\tau^0\right\|_2^2 + 2E^k, \left\|\tau^\star\right\|_2^2 + \left\|\tau^\star\right\|_2\left\|\tau^0 - \tau^\star\right\|_2 + 2\sqrt{\gamma E^k}\right\}. \tag{4.48}$$

Proof: We bound the term $F(\bar{y}^k) - F(y^\star)$. First, consider the upper bound. From Lemma 4.3, let $\tau = 0$,

$$\left\|\tau^{s+1}\right\|_2^2 \leq \left\|\tau^s\right\|_2^2 + 2\gamma\big(d(\tau^{s+1}) + 2\epsilon^s - L(y^s, \tau^s) + \hat{\nabla} d(\tau^s)^\top \tau^s\big). \tag{4.49}$$

Recall from (4.22) and (4.17), $\hat{\nabla} d(\tau^s) = Ay^s - b$ and $L(y^s, \tau^s) = F(y^s) + (\tau^s)^\top(Ay^s - b)$. Therefore $L(y^s, \tau^s) - \hat{\nabla} d(\tau^s)^\top \tau^s = F(y^s)$. Then (4.49) becomes

$$\left\|\tau^{s+1}\right\|_2^2 + 2\gamma(F(y^s) - d(\tau^{s+1})) \leq \left\|\tau^s\right\|_2^2 + 4\gamma\epsilon^s \tag{4.50}$$

Summing over $s = 0, \ldots k-1$, $\sum_{s=0}^{k-1} F(y^s) - d(\tau^{s+1}) \leq \frac{1}{2\gamma}\left\|\tau^0\right\|_2^2 + 2E^k$. Taking the average $\bar{y}^k$ and noting that $d(\tau^k) \leq d(\tau^\star) = F(y^\star)$ for all $\tau^k \in \mathbb{R}^C_+$,

$$F(\bar{y}^k) - F(y^\star) \leq \tfrac{1}{2\gamma k}\left\|\tau^0\right\|_2^2 + \tfrac{2E^k}{k}. \tag{4.51}$$

Next, consider the lower bound of $F(\bar{y}^k) - F(y^\star)$. By definition, $y^\star$ solves $\min_{y \in \mathcal{Y}(P,p)} F(y) + (Ay - b)^\top \tau^\star$ where $(Ay^\star - b)^\top \tau^\star = 0$. This implies that $F(y^\star) \leq L(\bar{y}^k, \tau^\star)$. We expand $L(\bar{y}^k, \tau^\star)$ with (4.17) to obtain

$$F(y^\star) - F(\bar{y}^k) \leq (A\bar{y}^k - b)^\top \tau^\star \leq [A\bar{y}^k - b]_+^\top \tau^\star.$$

We can then bound the difference $F(y^\star) - F(\bar{y}^k)$ by $\|\tau^\star\|_2 \left\|[A\bar{y}^k - b]_+\right\|_2$. From Corollary 4.1,

$$F(y^\star) - F(\bar{y}^k) \leq \tfrac{\|\tau^\star\|_2}{\gamma k}(\|\tau^\star\|_2 + \|\tau^0 - \tau^\star\|_2 + 2\sqrt{\gamma E^k}). \tag{4.52}$$

Together, (4.51) and (4.52) imply $|F(y^\star) - F(\bar{y}^k)| \leq \frac{1}{\gamma k} D(\tau^\star, \tau^0, E^k)$. Strong convexity of F follows from Assumption 4.1, such that $\left\|\bar{y}^k - y^\star\right\|_2^2 \leq \frac{\alpha}{2}|F(\bar{y}^k) - F(y^\star)|$. This completes the proof. ■

Remark 4.4 *Similar to $\bar{\tau}^k$, the convergence of $\bar{y}^k$ to $y^\star$ is sublinear in k and induces error that scales linearly in E^k. However, taking* larger *step sizes minimize this error.*

Fast first-order gradient method. When $\epsilon^k = \epsilon$ for all $k \in \mathbb{N}$, the fast gradient method [34] augments Algorithm 4.2 with the following update after Step 3,

$$\tau^{k+1} = \tfrac{\|A\|_2^2}{\alpha(k+3)}\left[\textstyle\sum_{i=1}^{k+1} \sqrt{i(i+1)}(A\hat{y}^k - b)\right]_+ + \tfrac{k+1}{k+3}\tau^{k+1}.$$

In large networked systems with low-accuracy oracles, the fast gradient method theoretically and empirically diverges from $d(\tau^\star)$ [34]. Since its constraint satisfaction results are comparable to Algorithm 4.2 [91], we focus on the standard first-order gradient descent instead.

4.6 *Alleviating Congestion for Irrational Drivers in a Stochastic Ride Hail Network*

In this section, we model competition among NYC's ride hail drivers as an MDP congestion game and apply Algorithm 4.2 to demonstrate how ride hail companies can implicitly enforce constraints by utilizing tolls.[1] Since origin-destination-specific trip data for ride hail companies are not publicly available, we use the rider demand distribution provided by the NYC

[1]Code for Manhattan's ride hail MDP congestion game is available at github.com/lisarah/manhattan_MDP_queue_game.

TLC as a proxy for Uber's rider demand distribution. In [118], the overall rider demand for TLC is estimated to be about 40% of the rider demand for Uber.

4.6.1 Geographical Network of Stochastic Queues Modeled as an MDP Congestion Game

We consider a cohort of competitive ride hail drivers in Manhattan, NYC repeatedly operating between 9 am and noon. Using six hundred thousand trip data from the yellow taxi data during January, 2019 [97], we model individual drivers as a finite time horizon MDP in a queuing network.

Modeling assumptions. 1) All trips take discretized times of $\{15, 20, 30, \ldots\}$ minutes based on the trip distance. 2) The initial driver distribution is uniform across all Manhattan zones. We find that varying the initial distribution does not significantly impact the time-averaged MDP Wardrop equilibrium or the toll norm, as long as the constraints are satisfied. 3) From [82], the Uber driver population in NYC is approximately 50000. We assume that 20% of the driver population works in Manhattan between 9 am and noon.

States. Each state is given by $s = (z, q) \in [Z] \times [Q]$, where $z \in [Z]$ is one of the sixty-three ($Z = 63$) Manhattan zones are visualized in the right plot of Figure 4.7 (islands excluded), and $q \in [Q]$ is the queue level, with the maximum level being $Q = 7$. At $q = 0$, the driver is in zone z without a rider. At $q > 0$, the driver is q time steps away from completing a ride to zone z. Zone z's geographically adjacent (sharing one or more edges) zones are given by $\mathcal{N}(z)$.

Actions. The action set of state (z, q) is q-dependent and is given by $\mathcal{A}(z, q)$. When $q > 0$, the driver is completing a ride. Therefore, the only action, a_z, is to finish the ride and $\mathcal{A}(z, q)$ is the singleton set given by $\mathcal{A}(z, q) = \{a_z\}, \forall z \in [Z],\ q > 0$. When $q = 0$, the driver can either go to a neighboring zone ($a_{z'}$) or pick up a rider in the current zone (a_z). The action set of $(z, 0)$ is $\mathcal{A}(z, 0) = \left\{a_{z'} |\ z' \in \mathcal{N}(z) \cup \{z\}\right\},\ \forall z \in [Z]$. The q-dependent action model follows the MDP model in Section 2.2 if we define $A = \max_{z,q} |\mathcal{A}(z, q)|$, and for all (z, q) where $|\mathcal{A}(z, q)| < A$, insert $A - |\mathcal{A}(z, q)|$ actions with infinite costs.

Time. The average trip time from the TLC data is 12.02 minutes. We add buffer time for drivers to locate and drop off riders, such that the MDP time interval is 15 minutes between 9 am and noon for a total of $T = 12$ time steps.

Transition Dynamics. Transition dynamics is q-dependent. When $q > 0$, the driver is completing a ride (a_z). Then, for all $z \in [Z]$ and $t \in [T]$, the transition dynamics of (z, q, a_z)

is given by

$$P(t, s', a_z, z, q) = \begin{cases} 1 & s' = (z, q-1) \\ 0 & \text{otherwise} \end{cases}, \forall\ q \geq 1. \tag{4.53}$$

When $q = 0$, the driver may go to an adjacent zone ($\{a_{z'} | z' \in \mathcal{N}(z)\}$) or pick up a rider ($a_z$). For $a_{z'}$, the transition dynamics is given by

$$P(t, s', a_{z'}, z, q) = \begin{cases} 1-\delta, & \text{if } s = (z', 0), \\ \frac{\delta}{|\mathcal{N}(z)|-1}, & \text{if } s = (\bar{z}, 0), \bar{z} \in \mathcal{N}(z)/\{z\}, \\ 0, & \text{otherwise}, \end{cases} \tag{4.54}$$

where $\delta \in [0, 1)$ models the driver's probability of real-time deviation from a chosen strategy. We set $\delta = 0.01$.

For action a_z from state $(z, 0)$, drivers will find a ride and transition to the appropriate queue in the destination zone. The transition dynamics for (z, q, a_z, t) is derived using the TLC ride demand distribution at $(z, 0)$ [97]. Let $N(z, z', q, t)$ be the number of trips with origin-destination (z, z') at time step t that took between $15q$ and $15(q+1)$ minutes. For all $z' \in [Z]$ and $q \in [Q]$ at time $t \in [T]$, the transition probability to state (z', q) is given by

$$P((z', q), (z, 0), a_z, t) = \frac{N(z, z', q, t)}{\sum_{\bar{q} \in [Q]} \sum_{\bar{z} \in [Z]} N(z, \bar{z}, \bar{q}, t)}.$$

Note that z' need not be an adjacent zone to z.

Driver costs. Driver costs are q-dependent at each state $s = (z, q)$. When $q > 0$, the driver cost is given by

$$\ell_{tsa}(y_{tsa}) = \beta y_{tsa},\ \forall\ t \in [T],\ a \in \mathcal{A}(z, q),$$

where β models the minor congestion effect of drivers entering zone z at queue level q. We set $\beta = 0.001$.

When $q = 0$, we follow the cost model in [73], given by

$$\ell_{tsa}(y_{tsa}) = \mathbb{E}_{s'}\left[c^{\text{trav}}_{ts's} - m_{ts's}\right] + c^{\text{wait}}_t \cdot y_{tsa} = \sum_{s'} P_{ts'sa}\left[c^{\text{trav}}_{ts's} - m_{ts's}\right] + c^{\text{wait}}_t \cdot y_{tsa} \tag{4.55}$$

The parameters in (4.55) are action-dependent: $m_{ts's}$ is the monetary reward, c^{wait}_t is the congestion scaling coefficient, and $c^{\text{trav}}_{ts's}$ is the fuel cost.

1. For $a_{z'}$ and $s' = (z', 0)$, $m_{ts's} = 0$ and $c^{wait}_t = 0.01$. The term $c^{\text{trav}}_{ts's}$ is the fuel cost of

reaching z', given by

$$c^{\text{trav}}_{ts's} = \mu \underbrace{d_{zz'}}_{\text{mi}} \underbrace{\left(\text{Vel}\right)^{-1}}_{\text{hr/mi}} + \underbrace{\left(\text{Fuel Price}\right)}_{\$/\text{gal}} \underbrace{\left(\text{Fuel Eff}\right)^{-1}}_{\text{gal/mi}} \underbrace{d_{zz'}}_{\text{mi}}. \tag{4.56}$$

The parameter $d_{zz'}$ is the estimated trip distance between z and z'. When $z' = z$, d_{zz} is the average distance (mi) for all (z, z) trips from the TLC data. When $z' \neq z$, $d_{zz'}$ is the Haversine distance (mi) between z and z'. The parameter μ is a time-money trade-off parameter, given in Table 4.2 along with other parameters.

μ	Velocity	Fuel Price	Fuel Eff
\$15 /mi	8 mph	\$2.5/gal	20 mi/gal

Table 4.2: Parameters for the driver cost function.

2. For $a = a_z$ and $s' = (z', 0)$, $m_{ts's}$ is the monetary reward, defined using Uber's NYC pay rate [9] as

$$m_{ts's} = \max\Big(\$7, \$2.55 + \$0.35 \cdot \Delta t + \$1.75 \cdot \Delta d\Big), \tag{4.57}$$

where Δt is the trip time (min) and Δd is the trip distance (mi). We set $\Delta t = 12$ as the average trip time from the TLC data and Δd as the estimated Haversine distance between (z, z'). The parameter c^{wait}_t is the coefficient of congestion, scaled linearly by the portion of drivers who are waiting for a rider, and is given by

$$c^{\text{wait}}_{tsa} = \mathop{\mathbb{E}}_{s'}\left[m_{ts's}\right] \cdot \Bigg(\underbrace{\text{Customer Demand Rate}}_{\text{rides}/\Delta t}\Bigg)^{-1}, \tag{4.58}$$

where $m_{ts's}$ is given by (4.57) and the customer demand rate is derived from TLC data per time interval per day. We estimate the Uber ride demand to be 2.5 times greater than Yellow Taxi's ride demand in January 2019 [26] and scale the TLC data accordingly.

4.6.2 Online Learning via Conditional Gradient Descent

When drivers optimize their strategies for the ride hail game in Section 4.6.1, we assume that

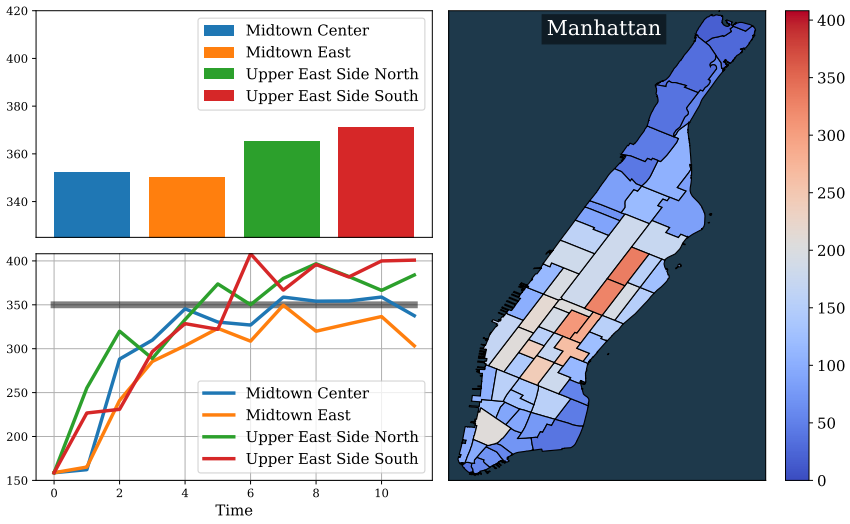


Figure 4.7: Predicted ride hail traffic in Manhattan.

they cannot directly access the congestion costs model and transition dynamics. Instead, they collectively receive costs for a chosen joint policy, and iterate to find the equilibrium policy.

We implement the learning method from [73, Alg. 3]. Inspired by conditional gradient descent (Frank-Wolfe), [73, Alg.3] implicitly enforces $y \in \mathcal{Y}(P, p_0)$ by solving linearized game potentials (4.7) via dynamic programming. Frank-Wolfe converges rapidly to low-accuracy solutions. Based on Frank-Wolfe's stopping criterion, the ϵ in the ϵ-MDP Wardrop equilibrium is given by

$$\epsilon^k = \left(\ell(y^k) + A^\top \tau^k\right)^\top (y^k - y^{k+1}). \tag{4.59}$$

We set $\epsilon^k = 1e3$, which is approximately equal to a normalized error of 0.5% for the unconstrained game potential. The corresponding ϵ-MDP Wardrop equilibrium and the driver densities of the most congested zones are shown in Figure 4.7.

4.6.3 Reducing Driver Presence in Congested Taxi Zones

Suppose the ride hail company wishes to reduce the driver density in congested zones to below 350 per zone per time step via Algorithm 4.2. This constraint can be formulated as

$$\textstyle\sum_a y_{tsa} \leq 350, \;\; s = (z, 0), \;\; \forall \; (t, z) \in [T+1] \times [Z]. \tag{4.60}$$

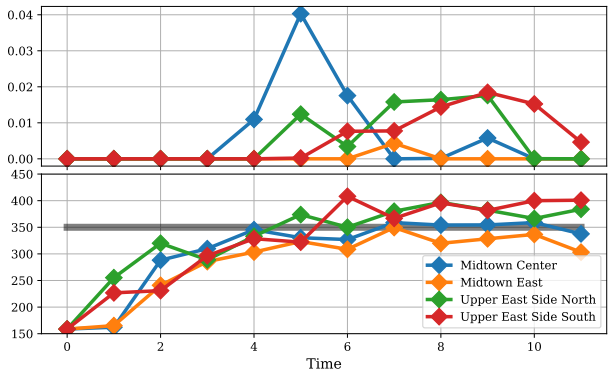


Figure 4.8: Manhattan game under constraint (4.60). The top line plot shows congestion tolls ($/15min). The bottom line plot shows the congested driver distributions (drivers/15min).

Each $(t, z) \in [T+1] \times [Z]$ corresponds to a constraint $A_i \in \mathbb{R}^{(T+1)SA}$, where for all $a \in \mathcal{N}(z, 0)$, the $(t, (z, 0), a)^{th}$ entry is 1 and all other entries are 0. Thus, we enforce a total of $63 \times 12 = 1008$ constraints of form (4.60). The constraint matrix is $A = [A_1, \ldots, A_{(T+1)S}]^\top \in \mathbb{R}^{(T+1)S \times (T+1)SA}$.

4.6.4 Discussion

We run Algorithm 4.2 for 2000 iterations at $\epsilon^k = 0.5\%$ of the unconstrained potential value. The results are shown in Figure 4.8. The resulting constraint violation has a 2 norm of 10.24 for the whole time horizon. The toll's 2 norm is 2.94.

In Figure 4.8 top, we see that for tolls around $1 per time step per state, the average constraint violation decreases from over 200 drivers to less than 10 drivers for the whole time horizon. This is comparable to the proposed toll value for lower Manhattan ($2.25 per entry) [41]. For each of the zones shown, the highest toll does not occur at the time of the largest constraint violation. In Figure 4.9 left, we evaluate Algorithm 4.2 by its toll value convergence and the constraint violation during tolling. Note that the average constraint violation $\left\|[A\bar{y}^k - b]_+\right\|_2$ differs from the last-iterate constraint violation $\left\|[Ay^k - b]_+\right\|_2$, and the last iterate constraint violation does not converge in part due to drivers' imprecision in finding the equilibrium strategy.

Social cost. A major concern is the effects of tolling on the average drivers' earnings, measured by the *social cost* [73]. When the social cost increases significantly, drivers may

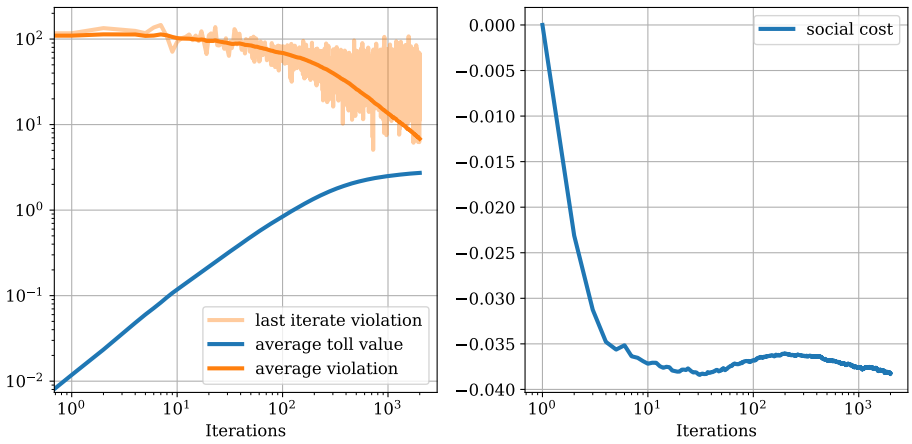


Figure 4.9: Left: average toll value and constraint violation during toll synthesis. Right: average driver earnings during toll synthesis.

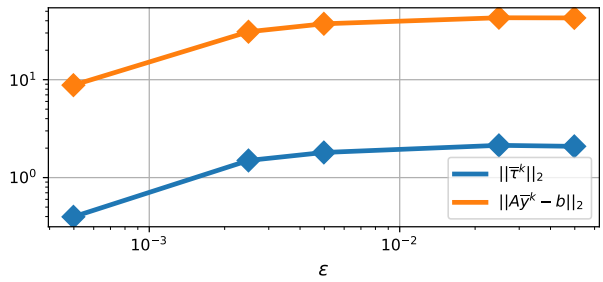


Figure 4.10: ϵ vs average toll and constraint violation at $k = 1000$.

quit, thus reducing the ride hail workforce. We show empirically in Figure 4.9 (right) that tolling does not significantly impact driver earnings: the average driver earnings during the tolling process are normalized against the untolled average earnings. During tolling, the social cost decreased, implying that the average driver's earnings increased. Therefore, congestion-based tolling is unlikely to cause quitting among the driver population.

Equilibrium accuracy. The effect of ϵ^k on the toll value $\|\bar{\tau}\|_2$ and the average constraint violation $\left\|[A\bar{y}^k - b]_+\right\|_2$ are shown in Figure 4.10 after 1000 iterations for $\epsilon = [100, 1000, 5000, 10000, 50000]$. The increased accuracy in ϵ decreases both the toll value and the constraint violation during the tolling process, thus providing more incentive to accurately compute the minimum toll value.

Chapter 5

MITIGATING DISRUPTION PROPAGATION IN NETWORKED LEARNING DYNAMICS

In the previous chapters, much of the emphasis is placed on how to model large-scale competitive players under shared resource uncertainty, and how to characterize and constrain the playing population at Nash equilibria. In this chapter, we focus on how the player can *learn* the Nash equilibrium using disturbance-prone observations of the shared environment.

As the application of learning in multi-agent settings gains traction, game theory has emerged as an informative abstraction for understanding the coupling between algorithms employed by individual players (see, e.g., [43, 84, 31]). For settings in which scalability to a large number of players is crucial, a commonly employed class of algorithms in both games and modern machine learning approaches to multi-agent learning is *gradient-based learning*, in which players update their actions using the gradient of their objective with respect to their action. In the gradient-based learning paradigm, continuous *quadratic games* stand out as a benchmark due to their simplicity and ability to exemplify state-of-the-art multi-agent learning methods such as policy gradient and alternating gradient-descent-ascent [83].

Despite the resurgence of interest in learning in games, a gap exists between algorithmic performance in simulation and physical application in part due to disturbances in measurements [121]. Robustness to environmental noise has been analyzed in a wide variety of learning paradigms [75, 21]. Most analysis focuses on independent and identically distributed stochastic noise drawn from a stationary distribution.

In contrast, we study *adversarial disturbance* without any assumptions on its dynamics or bounds on its magnitude. Though some work exists on the effects of bounded adversarial disturbance in multi-agent learning [57], there is limited understanding of how gradient disturbance propagates through the network structure as determined by the coupling of the players' objectives. Does gradient-based learning fundamentally contribute to or reduce the propagation of disturbance through player actions? Our analysis aims to answer this question for gradient-based multi-agent learning dynamics. The insights we gain provide desiderata to support algorithm synthesis and incentive design, and will lead to improved robustness of

multi-agent learning dynamics.

Contributions. We provide a novel graph-theoretical perspective for analyzing disturbance decoupling in multi-agent learning settings. For quadratic games, we obtain a necessary and sufficient condition, which can be verified in polynomial time, that ensures complete decoupling between the corrupted gradient of one player and the learned actions of another player, stated in terms of algebraic and graph-theoretic conditions. The latter perspective leads to greater insight on the types of cost coupling structures that enjoy disturbance decoupling, and hence, provides a framework for designing agent interactions, e.g., via incentive design or algorithm synthesis. Applied to LQ games, a benchmark for multi-agent policy gradient algorithms, we show that disturbance decoupling enforces necessary constraints on the controllable subspace to the unobservable subspace of individual players. Applied to bilinear games, we show that disturbance decoupling enforces necessary constraints on the players' payoff matrices.

5.1 Continuous Games and the Game Graph Model

Consider an N-player continuous game $(f_1, \ldots, f_N)$ where for each $i \in [N]$, $f_i \in C^r(\mathbb{R}^n, \mathbb{R})$ with $r \geq 2$ is player i's cost function and $\mathbb{R}^n = \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_N}$ is the joint action space, with $\mathbb{R}^{n_i}$ denoting player i's action space and $n = \sum_{i=1}^{N} n_i$. Each player's goal is to select an action $x_i \in \mathbb{R}^{n_i}$ to minimize its cost $f_i : \mathbb{R}^n \to \mathbb{R}$ given the actions of all other players. That is, player i seeks to solve the following optimization problem:

$$\min_{x_i \in \mathbb{R}^{n_i}} \; f_i(\underbrace{x_1, \ldots, x_i, \ldots, x_N}_{:= x}). \tag{5.1}$$

One of the most common characterizations of the outcome of a continuous game is a Nash equilibrium.

Definition 5.1 (Nash equilibrium) *For an N–player continuous game $(f_1, \ldots, f_N)$, a joint action $x^\star = (x_1^\star, \ldots, x_N^\star) \in \mathbb{R}^n$ is a Nash equilibrium if for each $i \in [N]$,*

$$f_i(x^\star) \leq f_i(x_1^\star, \ldots, x_{i-1}^\star, x_i, x_{i+1}^\star, \ldots, x_N^\star), \; \forall \; x_i \in \mathbb{R}^{n_i}.$$

Gradient-based learning. We consider a class of simultaneous play, gradient-based multi-agent learning techniques such that at iteration k, player i receives $h_i(x^k)$ from an oracle to

update its action as follows:

$$x_i^{k+1} = x_i^k - \gamma_i h_i(x_1^k, \ldots, x_N^k), \tag{5.2}$$

where $\gamma_i > 0$ is player i's step size,

$$h_i(x^k) = \frac{\partial f_i(x^k)}{\partial x_i} + d_i^k \tag{5.3}$$

is player i's gradient evaluated at the current joint action x^k and affected by a player-specific, arbitrary additive disturbance $d_i^k \in \mathbb{R}^{n_i}$. In the setting we analyze, d_i^k can modify x_i^k to any other action within $\mathbb{R}^{n_i}$.

Under reasonable assumptions on step sizes—e.g., relative to the spectral radius of the Jacobian of h_i in a neighborhood of a critical point—it is known that the undisturbed dynamics converge [84, 31]. While such a guarantee cannot be given for arbitrary disturbances as considered in this paper, we provide conditions under which a subset of players still equilibrates and follows the undisturbed dynamics.

Quadratic games. For an N–player continuous game $(f_1, \ldots, f_N)$, the behavior of gradient-based learning around a local Nash equilibrium can be approximated by linearizing the learning dynamics, where the *linearization* corresponds to a quadratic game.

Definition 5.2 (Quadratic game) *For each $i \in [N]$, $f_i : \mathbb{R}^n \to \mathbb{R}$ is defined by*

$$f_i(x) = \tfrac{1}{2} x_i^\top P_i x_i + x_i^\top \left(\textstyle\sum_{j \neq i} P_{ij} x_j + r_i\right). \tag{5.4}$$

Quadratic games encompass potential games [90] with $P_{ij} = P_{ji}^\top$, and zero sum games [45] with $P_{ij} = -P_{ji}^\top$. We give further examples of quadratic games below.

Game graph. To highlight how an individual player's action updates depend on others' actions, we associate a directed graph to the gradient-based learning dynamics defined in (5.2).

We consider a directed graph $([N], \mathcal{E})$, where $[N]$ is the index set for the nodes in the graph, and $\mathcal{E}$ is the set of edges. Each node $i \in [N]$ is associated with action x_i of the i^{th} player. A directed edge (j, i) points from j to i and has weight matrix $W_{ij} \in \mathbb{R}^{n_i \times n_j}$, such that $(j, i) \in \mathcal{E}$ if $W_{ij} \neq 0$ element-wise. For each node i, we assume the self loop edge (i, i) always exists and has weight $W_{ii} \in \mathbb{R}^{n_i \times n_i}$. The composite matrix $W \in \mathbb{R}^{n \times n}$ with entries W_{ij} is the adjacency matrix of the *game graph*.

On a game graph, we define a path $p = (i, v_1, \ldots, v_{k-1}, j)$ as a sequence of nodes connected by edges. The set of paths $\mathcal{P}_{ij}^k$ includes all paths starting at i and ending at j, traversing $k+1$ nodes in total. For a path $p = (i, v_1, \ldots, v_{k-1}, j)$, we define its *path weight* as the product of consecutive edges on the path, given by $W_{j,v_{k-1}} \ldots W_{v_1,i} = \prod_{l=0}^{k-1} W_{v_{l+1},v_l}$.

In the absence of disturbances d_i, the update in (5.2) for a quadratic game reduces to

$$x^{k+1} = Wx^k - \Gamma\bar{r}, \tag{5.5}$$

where $\bar{r} = \begin{bmatrix} r_1^\top & \ldots & r_N^\top \end{bmatrix}^\top$, $W_{ii} = I_{n_i} - \gamma_i P_i$, $W_{ij} = -\gamma_i P_{ij}$, and $\Gamma = \mathbf{diag}(\gamma_1 I_{n_1}, \ldots, \gamma_N I_{n_N})$.

To both illustrate the breadth of quadratic games and provide exemplars of the game graph concept, we describe two important subclasses of games and their game graphs.

Example 5.1 (Finite horizon LQ game) *Given initial state $z^0 \in \mathbb{R}^m$ and horizon T, each player i in an N-player, finite-horizon LQ game selects an action sequence $(u_i^0, \ldots, u_i^{T-1})$ with $u_i^t \in \mathbb{R}^{m_i}$ in order to minimize a cumulative state and control cost subjected to state dynamics:*

$$\begin{aligned} \min_{u_i^t \in \mathbb{R}^{m_i}} \ & \tfrac{1}{2}\big(\textstyle\sum_{t=0}^{T} (z^t)^\top Q_i z^t + \sum_{t=0}^{T-1} (u_i^t)^\top R_i u_i^t \big) \\ s.t. \ & z^{t+1} = Az^t + \textstyle\sum_{i=1}^{N} B_i u_i^t, \ t = 0, \ldots, T-1. \end{aligned} \tag{5.6}$$

The LQ game defined by the collection of optimization problems (5.6) for each $i \in [N]$ is equivalent to a one-shot quadratic game in which each player selects $U_i = [(u_i^0)^\top, \ldots, (u_i^{T-1})^\top]^\top \in \mathbb{R}^{n_i}$ with $n_i = Tm_i$, in order to minimize their cost $f_i(U)$ defined by $\frac{1}{2}(\sum_{j=1}^N G_j U_j + Hz^0)^\top \bar{Q}_i (\sum_{j=1}^N G_j U_j + Hz^0) + \frac{1}{2} U_i^\top \bar{R}_i U_i$, where $U = (U_1, \ldots, U_N)$ is the joint action profile, and the cost matrices are given by $\bar{Q}_i = \mathbf{diag}\{Q_i, \ldots, Q_i\}$,

$$G_i = \begin{bmatrix} 0 & \ldots & 0 \\ B_i & \ldots & 0 \\ \vdots & \ddots & \vdots \\ A^{T-1}B_i & \ldots & B_i \end{bmatrix}, H = \begin{bmatrix} I \\ \vdots \\ A^T \end{bmatrix}, \tag{5.7}$$

and $\bar{R}_i = \mathbf{diag}\{R_i, \ldots, R_i\}$. This follows precisely from observing that the dynamics are equivalent to $Z = \sum_{i=1}^N G_i U_i + Hz^0$ where $Z = [(z^0)^\top, \ldots, (z^T)^\top]^\top$. From here, it is straight forward to rewrite the optimization problem in (5.6) as $\min_{U_i} f_i(U)$. The LQ game is a potential game if and only if $Q_i = Q_j$ and $R_i = R_j$ for all $i, j \in [N]$.

LQ Game Graph. *Suppose each player uses step size γ_i. Since, $D_i f_i(U)$ is given by*

$$(G_i^\top \bar{Q}_i G_i + \bar{R}_i)U_i + G_i^\top \bar{Q}_i(\textstyle\sum_{j\neq i} G_j U_j + Hz^0), \tag{5.8}$$

the learning dynamics (5.5) are equivalent to

$$U^{k+1} = WU^k - \Gamma[\bar{Q}_1 G_1, \ldots, \bar{Q}_N G_N]^\top H z^0, \tag{5.9}$$

where $W = I_n - M$, with $M \in \mathbb{R}^{n\times n}$ a blockwise matrix having entries $M_{ij} = \gamma_i G_i^\top \bar{Q}_i G_j$ if $i \neq j$ and $M_{ij} = \gamma_i(G_i^\top \bar{Q}_i G_i + \bar{R}_i)$ otherwise.

Another important class of games is bilinear games. In adversarial learning, a number of game formulations have a hidden bilinear structure [136]. In evaluating and selecting hyper-parameter configurations in so-called *test suites*, pairwise comparisons between algorithms are formulated as bimatrix games [13, 12].

Example 5.2 (Bilinear game) *A* two player bilinear game[1]*, a subclass of continuous quadratic games, is defined by $f_1(x_1, x_2) = x_1^\top A x_2$ and $f_2(x_1, x_2) = x_1^\top B^\top x_2$ where $A \in \mathbb{R}^{n_1\times n_2}$ and $B \in \mathbb{R}^{n_2\times n_1}$ and $x_i \in \mathbb{R}^{n_i}$. Common approaches to learning in games [136, 11], simultaneous and alternating gradient descent both correspond to a linear system.*

Game graph for simultaneous gradient play. *Players update their strategies simultaneously by following the gradient of their own cost with respect to their choice variable:*

$$x_1^{k+1} = x_1^k - \gamma_1 A x_2^k,\ x_2^{k+1} = x_2^k - \gamma_2 B x_1^k \tag{5.10}$$

The simultaneous gradient play game graph is given by

$$W_s = \begin{bmatrix} I & -\gamma_1 A \\ -\gamma_2 B & I \end{bmatrix}. \tag{5.11}$$

Game graph for alternating gradient play. *In zero-sum bilinear games, it has been shown that alternating gradient play has better convergence properties [11]. Alternating gradient play is defined by*

$$x_1^{k+1} = x_1^k - \gamma_1 A x_2^k,\ x_2^{k+1} = x_2^k - \gamma_2 B x_1^{k+1} \tag{5.12}$$

[1]The bilinear game formulation and corresponding game graph for different gradient-based learning rules easily extend to an N-player setting, however the results in Sec. 5.2 are presented for two player games.

Examining the second player's update, we see that $x_2^{k+1} = (I + \gamma_1\gamma_2 BA)x_2^k - \gamma_2 Bx_1^k$. *The game graph in this case is defined by*

$$W_a = \begin{bmatrix} I & -\gamma_1 A \\ -\gamma_2 B & I + \gamma_1\gamma_2 BA \end{bmatrix}. \tag{5.13}$$

Remark 5.1 *Convergence of* (5.10) *and boundedness of* (5.12) *depend on choosing appropriate step sizes* γ_1 *and* γ_2 *[31, 11]. We consider disturbance decoupling for settings such as these where the undisturbed dynamics are convergent.*

5.2 Disturbance Decoupling on Game Graph

In this section, we derive the necessary and sufficient condition that ensures the decoupling of gradient disturbance from the learning trajectory of a subset of players. We emphasize that the condition holds for disturbances with arbitrary magnitudes and functions. This is a useful result because it provides guarantees on both the equilibrium behavior and the learning trajectory under adversarial disturbance.

Definition 5.3 (Complete disturbance decoupling) *Given initial joint action* $x^0 \in \mathbb{R}^n$, *game costs* $(f_1, \ldots, f_N)$, *step sizes* $\Gamma \in \mathbb{R}^{n\times n}$, *suppose that player* i*'s gradient update is corrupted as in* (5.3), *then for player* $j \neq i$, *action* x_j *is decoupled from the disturbance in player* i*'s gradient if the uncorrupted and corrupted dynamics, given respectively by*

$$x^{k+1} = Wx^k - \Gamma\bar{r},\ y^{k+1} = Wy^k - \Gamma\bar{r} - \Gamma d^k \tag{5.14}$$

result in identical trajectories for player j *when* $y^0 = x^0$. *That is,* $y_j^k = x_j^k$ *holds for all* $k \geq 0$, $d^k \in \mathcal{D}_i$, *where*

$$\mathcal{D}_i = \{d = [d_1, \ldots, d_N]^\top \in \mathbb{R}^n \mid d_j = 0, \forall\ j \neq i\}.$$

5.2.1 Algebraic condition

We first derive an algebraic condition on the joint action space for disturbance decoupling. Define $\mathcal{M}^\perp = \{x \in \mathbb{R}^n \mid x^\top \tilde{x} = 0,\ \forall\ \tilde{x} \in \mathcal{M}\}$ and let $\mathcal{R}(A) = \{Ax \mid x \in \mathbb{R}^n\}$ denote the image of $A \in \mathbb{R}^{m\times n}$.

Proposition 5.1 *Consider an* N*-player quadratic game* $(f_1, \ldots, f_N)$ *as in Definition 5.2 under learning dynamics as given by* (5.2), *where player* i *experiences gradient disturbance*

as given by (5.3). Let $\mathcal{S}(i) = \{x = [x_1, \ldots, x_N]^\top \in \mathbb{R}^n \mid x_j = 0,\ \forall\ j \neq i\}$ *be the joint action subset. For player* $j \neq i$*, the following statements are equivalent:*

(i) Player j *is disturbance decoupled from player* i*.*

(ii) $W^k v \in \mathcal{S}(j)^\perp,\ \forall\ v \in \mathcal{S}(i),\ \forall\ 0 \leq k < n$*.*

(iii) $\mathcal{R}(W^k E) \subseteq \mathcal{R}(Y),\ \forall\ 0 \leq k < n$*, where* $E \in \mathbb{R}^{n \times n_i}$ *and* $Y \in \mathbb{R}^{n \times (n - n_j)}$ *are matrices such that* $\mathcal{R}(E) = \mathcal{S}(i)$ *and* $\mathcal{R}(Y) = \mathcal{S}(j)^\perp$*.*

Proof: For a quadratic game $(f_1, \ldots, f_N)$, the learning dynamics without and with disturbances reduce to the equations in (5.14). Given initial joint action x^0,

$$x^k = W^k x^0 - \begin{bmatrix} W^{k-1} & \ldots & W^0 \end{bmatrix} \Gamma \begin{bmatrix} \bar{r}^\top & \ldots, & \bar{r}^\top \end{bmatrix}^\top,$$
$$y^k = x^k - \begin{bmatrix} W^{k-1} & \ldots & W^0 \end{bmatrix} \Gamma \begin{bmatrix} (d^0)^\top & \ldots, & (d^{k-1})^\top \end{bmatrix}^\top.$$

Then, Definition 5.3 is equivalent to $\sum_{l=0}^{M-1} W^{M-l-1} d^l \in \mathcal{S}(j)^\perp$ satisfied for $M \geq 1$ and $d^l \in \mathcal{S}(i)$. Since the condition holds for all $M \geq 1$, it is equivalent to $W^k d^l \in \mathcal{S}(j)^\perp$ for all $k \geq 0$ and $d^l \in \mathcal{S}(i)$. This is then equivalent to $W^k d^l \in \mathcal{S}(j)^\perp$ for all $0 \leq k < n$ and $d^l \in \mathcal{S}(i)$. To see this equivalence, consider the following result from Cayley-Hamilton theorem, $W^k = \sum_{l=0}^{n-1} \alpha_l W^l$ for some $\alpha_l \in \mathbb{R}$. Thus, for $k \geq n$ and any $d \in \mathcal{S}(i)$, $W^k d = \sum_{l=0}^{n-1} W^l \alpha_l d = \sum_{l=0}^{n-1} W^l \hat{d}_l$ where $\hat{d}_l = \alpha_l d \in \mathcal{S}(i)$ for $l = 0, \ldots, n-1$, which implies that $W^k d \in \mathcal{S}(j)^\perp$. This concludes the equivalence.

Finally, we note that *(iii)* is a restatement of *(ii)*. Furthermore, *(iii)* can be verified in polynomial time. ■

Remark 5.2 *In connection to geometric control theory, condition (iii) of Proposition 5.1 is equivalent the fact that* $\mathcal{R}([E, \ldots, W^{n-1}E])$*, the smallest* W*-invariant subspace containing* $\mathcal{R}(E)$*, must be a subset of* $\mathcal{S}(j)^\perp$ *[133, Thm 4.6].*

5.2.2 Graph-theoretic condition

Next, we derive the graph-theoretic condition on the joint action space for disturbance decoupling.

Theorem 5.1 *Consider an* N*-player quadratic game* $(f_1, \ldots, f_N)$ *as in Definition 5.2 under learning dynamics as given by (5.2), where player* i *experiences gradient disturbance as given*

by (5.3). Player $j \neq i$ is disturbance decoupled if and only if the path weights of paths with length k satisfy

$$\sum_{p \in \mathcal{P}_{ij}^k} \prod_{l=0}^{k-1} W_{v_{l+1}, v_l} = 0, \ \forall\ 0 < k < n, \tag{5.15}$$

where (v_l, v_{l+1}) denotes consecutive nodes on path $p = (i, v_1, \ldots, v_{k-1}, j)$.

Proof: The result follows from the equivalence between Proposition 5.1 condition (ii) and (5.15). Note that $x \in \mathcal{S}(i)$ is equivalent to $x_\ell = 0$ for all $\ell \neq i$, and $W^k x \in \mathcal{S}(j)^\perp$ is equivalent to $(W^k x)_j = 0$ for all $n > k \geq 0$. We prove the result by induction. For $k = 0$, $(W^0 x)_j = 0\ \forall\ x \in \mathcal{S}(i)$ holds if and only if $i \neq j$. For $k > 0$, $(W^k x)_j = 0\ \forall\ x \in \mathcal{S}(i)$ is equivalent to $i \neq j$ and $(W^k)_{ji} = 0$. Suppose that for $i, j \in [N]$, $(W^k)_{ji}$ is the sum of path weights over all paths of length k, originating at i and ending at j, then $(W^{k+1})_{ji}$ is the sum of path weights over all paths of length $k+1$, originating at i and ending at j. Let $W^k = M$, then $(W^{k+1})_{ji} = \sum_{q \in [N]} M_{jq} W_{qi}$, where $M_{jq} W_{qi}$ is the sum of path weights over all paths of length $k+1$ from i to j each of which contains $v_1 = q$. Since we sum over $q \in [N]$, we conclude that $(W^{k+1})_{ji}$ is the sum of all paths weights of length $k+1$ from i to j, i.e., $(i, v_1, \ldots, v_k, j) \in \mathcal{P}_{ij}^{k+1}$. ■

The concept of disturbance decoupling is quite counter-intuitive: any change in player i's action does not affect player j's action, despite f_j being implicitly dependent on x_i through the network of player cost functions. As we see from the proof of Theorem 5.1, this situation arises when the dependencies 'cancel' each other out, i.e. the sum of path weights from i to j is always zero for equally lengthed paths.

Example 5.3 (Disturbance decoupled players) *Consider a 4 player quadratic game where $x_i \in \mathbb{R}$ and the game graph is given in Figure 5.1. Edge weights α, β, γ, and $\delta \in \mathbb{R}$, while each self loop has weight $w_i > 0$. Paths of length $k \leq 4$ from player 1 to player 4 are enumerated as $\mathcal{P}_{14}^1 = \{\emptyset\}$, $\mathcal{P}_{14}^2 = \{(1,2,4), (1,3,4)\}$, and $\mathcal{P}_{14}^3 = \{(1,1,2,4), (1,1,3,4), (1,2,2,4), (1,3,3,4)$, $(1,2,4,4), (1,3,4,4)\}$. To satisfy Theorem 5.1, the sum of path weights for each $\mathcal{P}_{14}^k$ must be 0 for $0 < k < 4$. There are no paths of length one, summation for $k = 2$ implies the criteria $\alpha\gamma + \beta\delta = 0$, and summation for $k = 3$ implies the criteria $(w_1 + w_2 + w_4)\alpha\gamma + (w_1 + w_3 + w_4)\beta\delta = 0$. If $w_2 = w_3$, $\alpha\gamma + \beta\delta = 0$ is necessary and sufficient for disturbance decoupling between player 1 and player 4.*

Remark 5.3 *Disturbance decoupling is a structural property of the game in terms of disturbance propagation and attenuation. An open research problem is linking this structural*

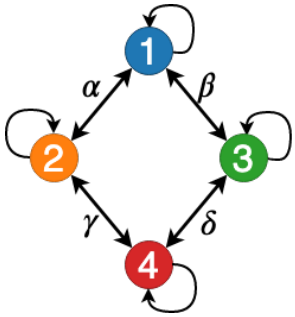


Figure 5.1: A simple game graph between four players

property to robust decision-making under uncertainties in cost parameters P_i, P_{ij}, and step sizes γ_i.

The following corollary specializes to the class of potential games [90], which arise in many applications [100, 80, 6].

Corollary 5.1 *Consider an N-player quadratic potential game under learning dynamics as given by (5.2), where player i experiences gradient disturbance as given by (5.3). Player i is disturbance decoupled from player $j \neq i$ if and only if player j is also disturbance decoupled from player i.*

Proof: In a potential game graph, $W_{ij} = \frac{\gamma_i}{\gamma_j} W_{ji}^\top$. Therefore, a path p with path weight $W_{j,v_{k-1}} \dots W_{v_1,i}$ is equivalent to

$$\frac{\gamma_j}{\gamma_{v_{k-1}}} W_{v_{k-1},j}^\top \frac{\gamma_{v_{k-1}}}{\gamma_{v_{k-2}}} W_{v_{k-2},v_{k-1}}^\top \dots \frac{\gamma_{v_1}}{\gamma_i} W_{i,v_1}^\top = \frac{\gamma_j}{\gamma_i} W_{i,v_1} \dots W_{v_{k-1},j}, \tag{5.16}$$

where $\frac{\gamma_j}{\gamma_i}$ scales all path weights from i to j. Since $\gamma_j, \gamma_i > 0$, $\frac{\gamma_j}{\gamma_i} > 0$. Therefore, (5.15) holds from player i to player j if and only if it holds from player j to player i. ■

Corollary 5.2 *Consider an N-player finite horizon LQ game as in (5.6) under learning dynamics as given by (5.9), where player i experiences gradient disturbance as given by (5.3), if disturbance decoupling holds between player j and gradient disturbance from player i, then*

$$\begin{bmatrix} B_j^\top \\ \vdots \\ B_j^\top (A^\top)^{T-1} \end{bmatrix} Q_j \begin{bmatrix} B_i & \cdots & A^{T-1} B_i \end{bmatrix} = 0. \tag{5.17}$$

If Q_j is positive definite and $T \geq m$, the controllable subspace of $(\tilde{A}, \tilde{B}_i)$ must lie in the unobservable subspace of $(\tilde{B}_j^\top, \tilde{A}^\top)$ where $\tilde{A} = Q_j^{1/2} A Q_j^{-1/2}$, $\tilde{B}_i = Q_j^{1/2} B_i$, and $\tilde{B}_j = Q_j^{1/2} B_j$.

Proof: For player j to be disturbance decoupled from player i, edge (i, j) cannot exist, i.e. $-\gamma_j G_j^\top \bar{Q}_j G_i = 0$ from (5.7). Expanding $G_j^\top \bar{Q}_j G_i = M \in \mathbb{R}^{n_j \times n_i}$, $M_{pq} \in \mathbb{R}^{m_j \times m_i}$ is given by $\sum_{t=\min\{p,q\}}^{T-1} B_j^\top (A^\top)^{t-p} Q_j A^{t-q} B_i$. We unwrap these conditions starting from $p = T-1$, $q = T-1$; in this case $M_{pq} = B_j^\top Q_j B_i = 0$ is necessary. Then we consider $M_{T-2,T-2} = B_j^\top A^\top Q_j A B_i + B_j^\top Q_j B_i = 0$, which implies that $B_j^\top A^\top Q_j A B_i$ is necessary. Subsequently, this implies that all $B_j^\top (A^\top)^t Q_j A^t B_i = 0$ is necessary for $t \in [0, T)$. Similarly, we note that $M_{T-1,q} = B_j^\top Q_j A^q B_i = 0$ and $M_{p,T-1} = B_j^\top (A^\top)^p Q_j B_i = 0$. From these we can use the rest of M to conclude that $B_j^\top (A^\top)^p Q_j A^q B_i = 0$ for any $p, q \in [0, T)$. This condition is equivalent to (5.17). ■

We apply Theorem 5.1 to two player bilinear games and prove a necessary condition for disturbance decoupling between different coordinates of each player's action space that is *independent* of players' step sizes.

Corollary 5.3 *Consider a two player bilinear game under learning dynamics (5.10) and (5.12), where coordinates $x_{1,i}$ and $x_{2,i}$ experience gradient disturbance as given by (5.3). If $j \neq i$ and coordinate $x_{1,j}$ is disturbance decoupled from coordinate $x_{1,i}$, (A, B) must satisfy $\sum_{\ell=1}^{n_2} b_{\ell i} a_{j\ell} = 0$, where a_{pq} and b_{pq} denote the $(p,q)^{th}$ elements of A and B, respectively. Similarly, if $j \neq i$ and coordinate $x_{2,j}$ is disturbance decoupled from coordinate $x_{2,i}$, (A, B) must satisfy $\sum_{\ell=1}^{n_1} b_{j\ell} a_{\ell i} = 0$.*

Proof: We construct games played by $n_1 + n_2$ players with actions $\{x_{1,1}, \ldots, x_{1,n_1}, x_{2,1}, \ldots, x_{2,n_2}\}$ and whose game graphs are identical to W_s (5.11) and W_a (5.13). First consider disturbance decoupling of $x_{1,j}$ from $x_{1,i}$. In both learning dynamics, $\{x_{1,1}, \ldots, x_{1,n_1}\}$ do not have any edges between players. Therefore, paths between $x_{1,i}$ and $x_{1,j}$ with length 2 is given by $\mathcal{P} = \{(x_{1,i}, x_{2,\ell}, x_{1,j}) \mid \ell \in [n_2]\}$. We sum path weights over $\mathcal{P}$ to obtain $\sum_{\ell=1}^{n_2} b_{\ell i} a_{j\ell} = 0$ for disturbance decoupling of $x_{1,j}$ from $x_{1,i}$ in (5.10) and (5.12). A similar argument follows for disturbance decoupling of $x_{2,j}$ from $x_{2,i}$ in (5.10). For disturbance decoupling of $x_{2,j}$ from $x_{2,i}$ in (5.12), we note that an edge from $x_{2,i}$ to $x_{2,j}$ exists with weight $\gamma_1 \gamma_2 (BA)_{ji}$ when $j \neq i$. Disturbance decoupling requires $\gamma_1 \gamma_2 (BA)_{ji} = 0$, therefore $\sum_{\ell=1}^{n_1} b_{j\ell} a_{\ell i} = 0$. ■

Corollary 5.4 *Consider a two player bilinear game under learning dynamics (5.10) and (5.12), where coordinates $x_{1,i}$ and $x_{2,i}$ experience gradient disturbance as given by (5.3). If coordinate $x_{2,j}$ is disturbance decoupled from coordinate $x_{1,i}$, (A, B) must satisfy $b_{ji} = 0$ and*

$\sum_{q=1}^{n_2} b_{qi} \sum_{\ell=1}^{n_1} a_{\ell q} b_{j\ell} = 0$, *where* a_{pq} *and* b_{pq} *denote the* $(p,q)^{th}$ *elements of* A *and* B, *respectively. If coordinate* $x_{1,j}$ *is disturbance decoupled from coordinate* $x_{2,i}$, (A, B) *must satisfy* $a_{ji} = 0$ *and* $\sum_{q=1}^{n_1} a_{qi} \sum_{\ell=1}^{n_2} b_{\ell q} a_{j\ell} = 0$.

Proof: We construct games played by n_1+n_2 players with actions $\{x_{1,1}, \ldots, x_{1,n_1}, x_{2,1}, \ldots, x_{2,n_2}\}$ and whose game graphs are identical to W_s (5.11) and W_a (5.13). In both learning dynamics, disturbance decoupling requires no direct path between the decoupled players. Therefore $a_{ji} = 0$ or $b_{ji} = 0$.

Consider disturbance decoupling of $x_{1,j}$ from $x_{2,i}$ in (5.10), paths of length 3 from $x_{2,i}$ to $x_{1,j}$ without self loops is given by $\mathcal{P} = \{(x_{2,i}, x_{1,q}, x_{2,\ell}, x_{1,j}) \mid q \in [n_1],\ \ell \in [n_2]\}$. A path of length 3 with self loops must also include $(x_{2,i}, x_{1,j})$, whose weight is 0. We sum path weights over $p \in \mathcal{P}$ to obtain $\sum_{q=1}^{n_1} a_{qi} \sum_{\ell=1}^{n_2} b_{\ell q} a_{j\ell} = 0$. A similar argument is made for disturbance decoupling of $x_{2,j}$ from $x_{1,i}$ in (5.10).

Consider disturbance decoupling of $x_{2,j}$ from $x_{1,i}$ in (5.12), paths of length 2 from $x_{1,i}$ to $x_{2,j}$ without self loops is given by $\mathcal{Q} = \{(x_{1,i}, x_{2,q}, x_{2,j}) \mid q \in [n_2]\}$. A path of length 2 with self loops must also include $(x_{1,i}, x_{2,j})$, whose weight is 0. Weight of $(x_{2,q}, x_{2,j})$ is given by $\gamma_1\gamma_2(BA)_{jq} = \gamma_1\gamma_2 \sum_{\ell=1}^{n_1} b_{j\ell} a_{\ell q}$. We sum path weights over $p \in \mathcal{Q}$ to obtain $\sum_{q=1}^{n_2} b_{qi} \sum_{\ell=1}^{n_1} a_{\ell q} b_{j\ell} = 0$. A similar argument is made for disturbance decoupling of $x_{1,j}$ from $x_{2,i}$ in (5.12). ■

5.3 Disturbance Decoupling for a Tug-of-war Game

We provide an example of disturbance decoupling in a LQ game. Consider a tug-of-war game in which a single target $z \in \mathbb{R}^2$ is controlled by four players. We assume that player i can move z along vector $B_i \in \mathbb{R}^2$ by $u_i \in \mathbb{R}$, and that z is stationary without any player input, i.e., $A = I$. Starting with a randomized initial condition z^0, at each step t, the target moves according to the dynamics $z^{t+1} = z^t + \sum_{i=1}^{4} B_i u_i^t$ where $B_1 = [1, 0]^\top$, $B_2 = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^\top$, $B_3 = [\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^\top$, $B_4 = [0, 1]^\top$. Each player i's cost function is given by

$$\tfrac{1}{2} \left\| z^9 - c_i \right\|_2^2 + \sum_{t=0}^{8} \tfrac{1}{2} \left\| z^t - c_i \right\|_2^2 + 10 \left\| u_i^t \right\|_2^2$$

which describes player i's objective to move target z towards $c_i \in \mathbb{R}^2$ in a finite time $T = 10$ by using minimal amount of control. By designing the game dynamics to satisfy Theorem 5.1, we ensure that player 4's action is disturbance decoupled from player 1's.

Using the equivalent formulation as described in Section 5.1,

$$\frac{\partial f_i(U)}{\partial u_i} = (G_i^\top \bar{Q}_i G_i + \bar{R}_i)U_i + \sum_{j \neq i} G_i^\top \bar{Q}_i (G_j U_j + H z^0 - C_i),$$

where $C_i = [c_i^\top, \ldots, c_i^\top]^\top$. Hence, the learning dynamics are $U^{k+1} = WU^k + \Gamma \bar{Q}_i [G_1, \ldots, G_N]^\top [(Hz^0 - C_1)^\top, \ldots, (Hz^0 - C_N)^\top]^\top$, where $W_{ij} = G_i^\top \bar{Q}_i G_j = E \otimes B_i^\top B_j$ with $B_1^\top B_2 = B_1^\top B_3 = B_2^\top B_4 = \frac{1}{\sqrt{2}}$, $B_2^\top B_3 = B_1^\top B_4 = 0$, $B_3^\top B_4 = -\frac{1}{\sqrt{2}}$, $B_1^\top B_1 = B_2^\top B_2 = B_3^\top B_3 = B_4^\top B_4 = 1$, and

$$E = \begin{bmatrix} 9 & 8 & 7 & \ldots & 1 \\ 8 & 8 & 7 & \ldots & 1 \\ 7 & 7 & 7 & \ddots & 1 \\ \vdots & & \ddots & & 1 \\ 1 & & \ldots & & 1 \end{bmatrix} \in \mathbb{R}^{9 \times 9}.$$

To ensure convergence of the undisturbed learning dynamics [31], we use uniform step sizes such that $\Gamma = \text{blkdiag}(\gamma_1 I, \ldots, \gamma_4 I)$ with $\gamma_i = \frac{\sqrt{\alpha}}{\beta}$, where $\alpha = \lambda_{\min}\big(\frac{1}{4}(W + W^\top)^\top (W + W^\top)\big)$ and $\beta = \lambda_{\max}\big(W^\top W\big)$ with $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denoting the maximum and minimum eigenvalues of their arguments, respectively. The associated game graph is given in Figure 5.1, where $\alpha = \beta = \gamma = \frac{1}{\sqrt{2}} E$ and $\delta = -\frac{1}{\sqrt{2}} E$. A path $p = (1, v_1, \ldots, v_{k-1}, 4)$ of length k must have path weight $(\frac{-1}{\sqrt{2}})^{m_\delta} (\frac{1}{\sqrt{2}})^{m_\gamma} E^k$, where m_δ (m_γ) denotes the number of times the edge with weight δ (γ) is traversed in p.

Disturbance decoupling between players 1 and 4 is guaranteed if all paths of length $k \in (0, 36)$ satisfy (5.15). We can numerically verify that Proposition 5.1 is satisfied or make the following graph-theoretic observations based on Theorem 5.1. First, due to the symmetry within the game graph, the existence of path $p = (1, v_1, \ldots, v_{k-1}, 4)$ with path weight $L = (\frac{-1}{\sqrt{2}})^{m_\delta} (\frac{1}{\sqrt{2}})^{m_\gamma} E^k$ implies the existence of path $\hat{p} = (1, \hat{v}_1, \ldots, \hat{v}_{k-1}, 4)$ with path weight $\hat{L} = (\frac{-1}{\sqrt{2}})^{\hat{m}_\delta} (\frac{1}{\sqrt{2}})^{\hat{m}_\gamma} E^k$, where $m_\gamma = \hat{m}_\delta$ and $m_\delta = \hat{m}_\gamma$. Second, since edges $(3, 4)$ and $(2, 4)$ form a cut between player 1 and player 4 in the game graph, any path between them has the property that $m_\gamma + m_\delta$ is odd. From these observations, we can conclude that $L = -\hat{L}$. Since each path p of length k and weight L can be paired with path $\hat{p}$ of equivalent length k and weight $\hat{L} = -L$, we conclude that all path sets $\mathcal{P}_{14}^k$ where $k > 0$ must satisfy Theorem 5.1.

To numerically verify disturbance decoupling, we simulate the uncorrupted learning tra-

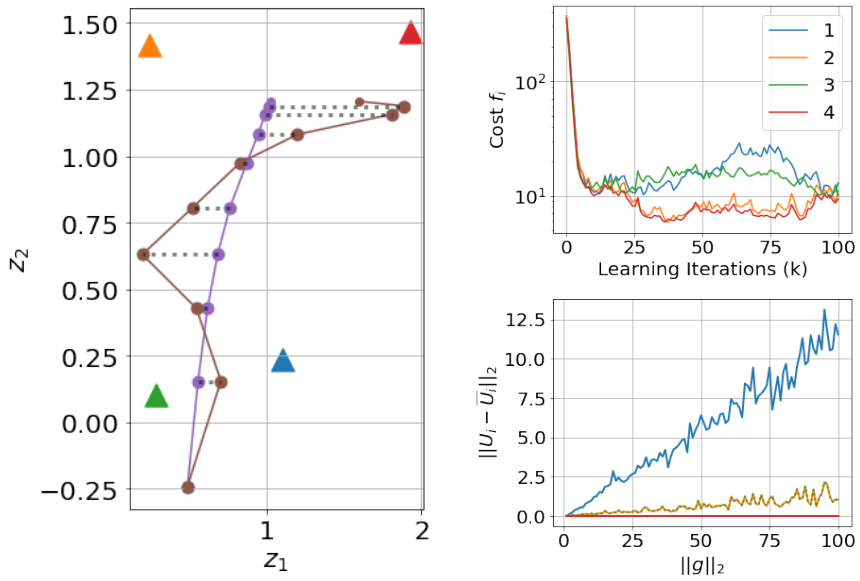


Figure 5.2: Left: Trajectory of z with and without disturbances. Players' preferred destinations are given by triangles. Top right: Players' game costs during learning. Bottom right: Players' control error as a function of disturbance magnitude.

jectory of z, shown in the left plot of Figure 5.2 in purple. We then inject a random disturbance into player 1's gradient updates as given by (5.3) with increasing magnitude, and observe its effects on each player's action. A sample corrupted trajectory is shown in the left plot of Figure 5.2 in brown. In the bottom right plot of Figure 5.2, we show the total error in each player's action from to the uncorrupted optimal action. We observe that player 4 does not deviate from the optimal action, while player 1's action error increases as the disturbance magnitude increases. We note that these results hold despite the fact that gradient-based learning no longer converges. In the top right plot of Figure 5.2, individual player costs are compared in one round of gradient-based learning where $\|d_i\| \leq 50$ is injected. Interestingly, despite action remaining uncorrupted, player 4's cost *is* disturbance affected. Note that the disturbance decoupling in actions does not necessarily imply disturbance decoupling in costs.

Chapter 6

COMPUTING GAME EQUILIBRIUM SENSITIVITY TO RESOURCE DISRUPTIONS

In Chapters 3 and 4, we used Markov games to model distributions of selfish decision makers when competing for finite and uncertain resources. In particular, MDP congestion games allow for stochastic dynamics in congestion games by mapping user inputs through stochastic dynamics to probabilistic outcomes. An equilibrium concept similar to Wardrop equilibrium of routing games [29], an *MDP Wardrop equilibrium*, describes steady-state population behavior at which no players can optimize their expected state-action costs through further changes in their decision strategies.

In modeling a physical process as an MDP congestion game, the game equilibrium is an *approximation* to the true steady state of the physical process; this is because models inherently cannot predict the physical process with full accuracy. The underlying assumption is that the modeling errors cause negligible deviations in prediction from physical equilibrium. However, this is false if the steady state distribution is *sensitive* to changes in the modeling parameters. This motivates our study of the sensitivity of the MDP congestion game to state-action costs.

In this chapter, we quantify sensitivity for the occurrence of *stochastic Braess paradox*, and relate the paradox to its deterministic counterpart. We also define and derive conditions for MDP dynamics and state-action costs under which our sensitivity analysis is valid. Finally, we found the sensitivity of a stochastic MDP congestion game in terms of the sensitivity of its deterministic counterpart.

Here we'd also like to emphasize why we consider the sensitivity of Wardrop equilibrium to the state-action cost parameters. In utilizing MDP congestion game models to forecast the steady state behavior of a physical system, state-action costs are often parameterized by experimental data, which typically has uncertainty. When the cost parameter uncertainty is bounded, it is natural to consider bounding the deviation of true equilibrium from the predicted equilibrium. Secondly, the sensitivity of game equilibrium is highly relevant to Stackelberg games for the leader, who may utilize the sensitivity information to derive an

optimal action sequence for its own objective [123]. Finally, we can consider a game designer who has a certain 'budget' for changing the cost function and wishes to alter the existing game equilibrium to maximize an external objective. In such settings, it's valuable to know the optimal direction of change that will achieve the most impact with respect to the designer's alternative objective.

We review existing literature on sensitivity and hypergraphs in section 6.1. In section 6.2, we relate the MDP congestion game, game equilibria definition, and KKT characterization to the hypergraph interpretation. Sensitivity results and stochastic Braess paradox characterizations are given in section 6.3. We analyze stochasticity's effect on paradox sensitivity in section 6.4. Finally, we demonstrate the stochastic Braess paradox via sensitivity analysis in section 6.5.

6.1 Related Literature

Our analysis resembles sensitivity work on Wardrop equilibria in traffic assignment literature [132, 108, 103], where the analysis is closely related to Braess paradox [23]. The occurrence of Braess paradox is known to be linked to the underlying graph of routing games [89]. The sensitivity of other games to modeling parameters has also been studied [102]. We note that while similar techniques are used, our work is fundamentally different due to our MDP network structure and focus on stochastic effects on the game equilibrium.

6.2 Directed Hypergraph for Infinite Horizon MDP Congestion Games

In this section, we define a variational inequality-style game equilibrium for the infinite horizon MDP congestion game introduced in Section 2.4. We then show how hypergraphs can be used to describe an MDP.

First, recall the infinite horizon MDP congestion game from Section 2.4. This model was first introduced in [29]. A continuous population of players experiences identical transition dynamics in a shared state-action space. We assume that this transition dynamic is time-invariant and can be represented by the probability kernel, $P \in \mathbb{R}^{S \times SA}$. Instead of the

simplex representation, we formulate P as a matrix in this section, element-wise defined as

$$P := \begin{pmatrix} P_{s_1 s_1 a_1} & P_{s_1 s_1 a_2} & \dots & P_{s_1 s_2 a_1} & \dots & P_{s_1 s_S a_A} \\ P_{s_2 s_1 a_1} & P_{s_2 s_1 a_2} & \dots & P_{s_2 s_2 a_1} & \dots & P_{s_2 s_S a_A} \\ \vdots & & & & & \\ P_{s_n s_1 a_1} & P_{s_n s_1 a_2} & \dots & P_{s_n s_2 a_1} & \dots & P_{s_n s_S a_A} \end{pmatrix},$$

where $P_{ss'a}$ denotes the transition probability from state s' to s when taking action a.

We exclusively consider time-invariant player population distributions with respect to the transition kernel P. Recall the set of feasible population distributions are given by the set $\mathcal{Y}(P) = \left\{ y \in \mathbb{R}_+^{SA} \;\middle|\; \sum_a x_{sa} = M, \sum_a x_{sa} = \sum_{a \in [A]} \sum_{s' \in [S]} P_{ss'a} x_{s'a},\ \forall s \in [S] \right\}$.

We denote $\ell : \mathbb{R}_+^{SA} \to \mathbb{R}^{SA}$ as the vector of state-action costs. Each element ℓ_{sa} is the cost function that relates the population distribution y to the expected incurred cost at state-action (s, a). The population dependency of ℓ reflects *congestion effects*: the greater the population in a given state-action pair, the greater the cost of taking that state-action for all decision-makers. This assumption is consistent with practical networked interactions in traffic and telecommunications [98] where, e.g., the cost of traversing a road increases for each driver when the number of cars on the road increases.

Assumption 6.1 *The state-action costs $\ell : \mathbb{R}_+^{SA} \to \mathbb{R}^{SA}$ are continuously differentiable and $\nabla_y \ell$ is positive definite.*

6.2.1 Directed Hypergraphs

We introduce the hypergraph structure of MDP in order to highlight the role that network topology plays in an MDP congestion game. Hypergraphs have been previously used to define a stochastic shortest path problem in finite horizon MDPs [40, 94].

We consider a weighted directed hypergraph [44], $\mathcal{G} = ([S], \mathcal{E})$, where $[S]$ is the set of states considered in MDP congestion game. $\mathcal{E}$ denotes the set of *hyperarcs*. While hyperarcs are defined with multiple heads and tails in general, we consider hyperarcs that have a single tail and multiple heads. A *hyperarc* $(s, a) \in \mathcal{E}$ is defined for each state-action pair, such that the tail of the hyperarc is always at s, and the head of the hyperarc is the set of states $\mathcal{H}(s, a)$ that can be reached from state s taking action a—i.e., $\mathcal{H}(s, a) = \{s' \in [S] \mid P_{s'sa} > 0\}$. Each hyperarc is equivalent to a state-action pair. To simplify notation, we represent each hyperarc by $(s, a) \in \mathcal{E}$.

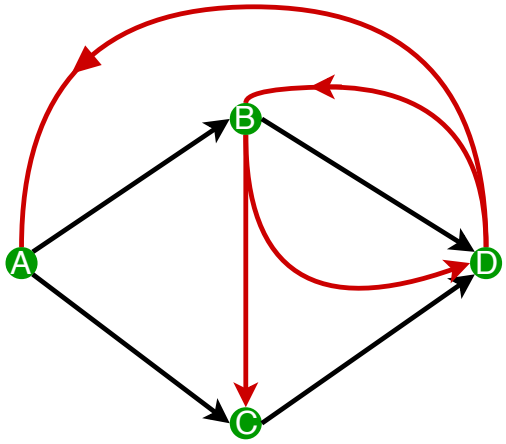


Figure 6.1: A directed hypergraph with 4 states. Black edges can be considered as both directed edges on a graph or hyperarcs with one tail. Red hyperarcs have one tail but multiple heads, denoting the possible outcomes that the hyperarc can result in next.

We define a *hypergraph incidence* matrix $E \in \mathbb{R}^{S \times |\mathcal{E}|}$ as

$$(E)_{s',(s,a)} = \begin{cases} 1 & s' = s, \\ -P_{s'sa} & s' \neq s. \end{cases} \tag{6.1}$$

Another formulation of the incidence matrix is $E = (I_S \otimes \mathbb{1}_A^T - P)$. In this form, we can see that the difference between the probability in each state (i.e., $\sum_a x_{sa}$) before and after a probabilistic transition (i.e., Px) can be written as $\sum_a x_{sa} - Px = Ex$. Therefore a stationary distribution $\hat{x}$ of an infinite horizon MDP satisfies $E\hat{x} = 0$.

The incidence matrix of a directed graph can be considered as a limiting case of the incidence matrix defined here: when each action leads to a deterministic state, $P_{s'sa} = 1$ for a hyperarc (s, a) at a unique s', and zero elsewhere. Alternatively, the hypergraph incidence matrix can be considered as a convex relaxation of directed graph incidence matrices where the tail of each hyperarc is fixed. Furthermore, like incidence matrices for directed graphs, the $\mathbb{1}_S$ vector is always in the null space of the incidence matrix for any hypergraph, $E^\top \mathbb{1}_S = 0$.

A directed hypergraph is *strongly connected* if every non-empty subset $\mathcal{R} \subset [S]$ has at least one incoming hyperarc from the set $[S]/\mathcal{R}$. We consider hypergraphs whose incidence matrix has rank $S - 1$.

Assumption 6.2 (Incidence Rank) *The hypergraph that corresponds to probability tran-*

sition kernel P is strongly connected, and its incidence matrix E has row rank $S-1$.

When an infinite time horizon MDP congestion game has an associate universal potential function F (3.13), it is given by

$$\min_{y} \quad \sum_{s\in[S]}\sum_{a\in[A]}\int_0^{y_{sa}} \ell_{sa}(u)du \tag{6.2a}$$

$$\text{s.t.} \quad Ey = 0, \tag{6.2b}$$

$$\mathbb{1}^\top y = M, \tag{6.2c}$$

$$y \geq 0, \tag{6.2d}$$

where constraints on y in (6.2) can be derived from the feasibility conditions of individual decision-makers.

Let $\nu \in \mathbb{R}^S$, $\lambda \in \mathbb{R}$, $\mu \in \mathbb{R}_+^{SA}$ be dual variables corresponding to (6.2b), (6.2c), (6.2d), respectively. Then the optimal solution of (6.2) satisfies

$$\begin{aligned} H(y^\star, \nu^\star, \lambda^\star, \mu^\star) &= 0, \\ \mu^\star &\geq 0, \\ y^\star &\geq 0, \end{aligned} \tag{6.3}$$

where

$$H(y, \nu, \lambda, \mu) = \begin{bmatrix} \ell(y) - E^\top\nu - \lambda\mathbb{1} - \mu \\ Ey \\ \mathbb{1}^\top y - M \\ \mu^\top y \end{bmatrix}.$$

The optimal dual variable $\lambda^\star \in \mathbb{R}$ quantifies the *average expected cost* for each decision maker. In addition, $\nu^\star \in \mathbb{R}^S$ can be interpreted as *state potentials* on the hypergraph such that $E^\top\nu^\star \in \mathbb{R}^{|\mathcal{E}|}$ gives the expected potential for hyperarc (s,a) relative to the average expected cost, λ. A detailed Karush-Khun-Tucker (KKT) analysis of an MDP congestion game is given in [30].

Given the non-negativity of μ, the first component of the KKT conditions can be written as

$$\ell(y^\star) \geq E^\top\nu^\star + \lambda^\star\mathbb{1}.$$

where equality holds at $y^\star_{sa} > 0$. Therefore, the optimal congestion costs $\ell(y^\star)$ can be

characterized by the dual variables $\nu^\star$ and $\lambda^\star$.

When the cost functions satisfy Assumption 6.1, uniqueness of the tuple $(y^\star, \lambda^\star, \mu^\star)$ is guaranteed [29]. We note that due to the rank deficiency of $E^\top$, $\nu^\star$ must be non-unique. However, when the hypergraph satisfies assumption 6.2, we can show that the feasibility constraint $Ey = 0$ is equivalent to $\tilde{E}y = 0$ where $\tilde{E}$ is the incidence matrix with one row removed and has full rank— i.e. the null space of $\tilde{E}^\top = \emptyset$. We'll see in the next section that this is required to derive a unique sensitivity for MDP Wardrop equilibrium.

Lemma 6.1 (Full Row Rank Incidence Matrix) *An MDP congestion game (4.5) which satisfies Assumption 6.2 and (4.4) can be equivalently formulated as*

$$\begin{aligned} \min_{y} \quad & F(y) \\ s.t. \quad & \tilde{E}y = 0, \\ & \mathbb{1}^\top y = M, \\ & y \geq 0, \end{aligned} \tag{6.4}$$

where $E = \begin{bmatrix} \tilde{E} \\ e^\top \end{bmatrix}$ *and* $\tilde{E}$ *has full row rank.*

Proof: Consider removing row vector $e^\top$ from the incidence matrix E. By Assumption 6.2, $e^\top$ is not identically 0. Clearly, $Ey = 0$ implies $\tilde{E}y = 0$. To see that the opposite implication is also true, we observe that $E^\top \mathbb{1} = 0$ is always true due to the conservation of mass. Therefore, $\mathbb{1}^T \tilde{E} = -e^\top$, so that $e^\top y = 0$ since $\tilde{E}y = 0$. ■

6.3 Sensitivity Analysis

In this section, we derive a sensitivity characterization of the stochastic Braess paradox. We *implicitly* characterize the optimal distribution of an MDP congestion game by perturbing its latency functions and performing a sensitivity analysis with respect to these perturbations.

To facilitate the analysis, we introduce *perturbation dependent* cost functions $\ell : \mathbb{R}^{SA} \times \mathbb{R}^{SA} \to \mathbb{R}^{SA}$, where the additional second parameter represents cost *perturbation*. An MDP congestion game is played with respect to a given perturbation ϵ and a corresponding cost $\ell(\cdot, \epsilon)$. Parameterized by ϵ, KKT conditions given by (6.3) implicitly define the optimal distribution $y^\star$ and the corresponding dual variables, which we define as a *point-to-set* mapping

given by

$$Q : \epsilon \mapsto \{(y, \nu, \lambda, \mu) |\ H(\epsilon, \lambda, \nu, y, \mu) = 0,\ \mu \geq 0,\ y \geq 0\}\,.$$

The point-to-set mapping, $Q(\epsilon)$, generalizes local differentiability of $y^\star$ as a function of ϵ [36]. When $\ell(\cdot, \epsilon)$ satisfies Assumption 6.1 in the first argument, the associated optimization formulation (6.4) has a unique solution and $Q(\epsilon)$ is a *single valued set mapping*; in this case we denote the unique optimal distribution by $y^\star(\epsilon)$. Unless otherwise stated, assumption 6.1 is assumed to hold from this point on.

Consider an MDP congestion game played with costs $\ell(y, 0)$ and its optimal solution $y^\star(0)$. When $Q(\epsilon)$ is a single-valued set mapping for an open set of ϵ containing zero, the Jacobian $\nabla_\epsilon y^\star(0)$ exists and defines the *sensitivity* of MDP Wardrop equilibria—i.e., it describes how much $y^\star(0)$ changes when cost ℓ is perturbed by ϵ.

We restrict our attention to MDP congestion games whose unique equilibrium $y^\star(0)$ is positive everywhere; such an assumption is equivalent to the fact that every state-action is optimal due to having a sufficiently large total mass.

Assumption 6.3 (Positivity Condition) *The optimal primal solution to the unperturbed MDP congestion game, $y^\star$, is strictly positive.*

We note that such an assumption is not restrictive in the following sense: when state-action costs satisfy Assumption 6.1, Assumption 6.3 will always be satisfied for some total mass M. Consider cost functions ℓ_{sa} which evaluates to $b_{sa} \in \mathbb{R}$ with no mass density. If a hyperarc from a state with positive mass is not optimal, i.e. has no mass, then b_{sa} must be at least $\max_{a' \in [A]} \ell_{sa'}(y^\star_{sa'}, 0)$. However, all other actions' costs must increase as total mass M increases, therefore a total mass threshold exists for which $\ell_{sa'}(y^\star_{sa'}, 0) \geq b_{sa}$, past which (s, a) will become optimal. Furthermore, if the transition kernel results in a strongly connected hypergraph, every state will have a positive mass.

Proposition 6.1 (Perturbation Map) *If an MDP congestion game (6.2) satisfies Assumptions 6.1 and 6.2 with costs $\ell(y, \epsilon)$ given ϵ, and optimal distribution $y^\star(\epsilon)$ satisfies assumption 6.3, then the following mapping is a single-valued mapping at ϵ,*

$$Q : \epsilon \mapsto \{y \in \mathbb{R}^{SA}, \nu \in \mathbb{R}^S, \lambda \in \mathbb{R} \mid f(y, \nu, \lambda, \epsilon) = 0\}, \tag{6.5}$$

where $f : \mathbb{R}^{SA\times(S(A+1))} \mapsto \mathbb{R}^{S(A+1)}$ *is defined as*

$$f\left(\begin{pmatrix} y \\ \nu \\ \lambda \end{pmatrix}, \epsilon\right) = \begin{bmatrix} \ell(\epsilon, y) + \tilde{E}^T\nu + \lambda\mathbb{1} \\ \tilde{E}y \\ \mathbb{1}^T y - M \end{bmatrix}.$$

Proof: From strict convexity assumptions, there exists a unique $y^\star(\epsilon) > 0$ solving the KKT conditions (6.3) for costs $\ell(\cdot, \epsilon)$. The dual variable $\mu^\star = 0$ from complementary slackness, and other dual variables can be uniquely determined. From $(y^\star)^\top(\ell(y^\star) - E^\top\nu^\star - \lambda^\star\mathbb{1}) = 0$, $\lambda^\star = (y^\star)^\top\ell(y^\star)/M$. Furthermore, unique $y^\star$ and unique $\lambda^\star$ implies $E^\top\nu^\star = \ell(y^\star) - \lambda^\star\mathbb{1}$. Since E^T has full rank $\nu^\star$ is unique. Finally, the KKT conditions can be simplified to f given above. ■

Proposition 6.1 implies that when ℓ is continuously differentiable at $y^\star$ and $\epsilon = 0$, there exists a continuously differentiable and invertible function of the optimal distribution y in terms of ϵ. We note that similar sensitivity results which do not consider stochastic congestion effects exist for routing games [132]. However, our results for MDP congestion games are less restrictive due to the lack of dual route/link space.

Theorem 6.1 (MDP Congestion Game Flow Sensitivity) *Consider an MDP congestion game with costs* $\ell(y, \epsilon)$*, such that* ℓ *is a continuously differentiable function of* (y, ϵ) *and satisfies Assumption 6.1, and the associated hypergraph satisfies Assumption 6.2. If the equilibrium distribution* $y^\star(\epsilon^\star) > 0$*, the sensitivity of the MDP Wardrop equilibrium is given by*

$$\nabla_\epsilon y^\star = G^{-1}N(N^\top G^{-1}N)^{-1}N^\top G^{-1}J - G^{-1}J.$$

Moreover, the sensitivity of optimal state-action costs is

$$\nabla_\epsilon \ell(y^\star, \epsilon^\star) = N(N^\top G^{-1}N)^{-1}N^\top G^{-1}J,$$

where $N = \begin{bmatrix} \tilde{E}^\top & \mathbb{1} \end{bmatrix}$*,* $\tilde{E}$ *as given by Lemma 6.1,* $G = \nabla_y \ell(y^\star(\epsilon^\star), \epsilon^\star)$*, and* $J = \nabla_\epsilon \ell(y^\star(\epsilon^\star), \epsilon^\star)$*.*

Proof: From Proposition 6.1, the game with costs $\ell(\cdot, \epsilon)$ has associated single valued mapping $Q(\epsilon)$ in a neighborhood of $\epsilon^\star$; let $v = Q(\epsilon)$. Then $f(v, \epsilon) = 0$ implies the total derivative $df(Q(\epsilon), \epsilon)/d\epsilon = 0$ for $\|\epsilon - \epsilon^\star\| \leq \delta$. Furthermore, f is continuously differentiable in all of its inputs. From the implicit function theorem [36, Sec.1B], when $\nabla_v f(v^\star, \epsilon^\star)$ is

invertible,

$$\nabla_\epsilon v^\star = \left(\nabla_v f(v^\star, \epsilon^\star)\right)^{-1} \nabla_\epsilon f(v^\star, \epsilon^\star).$$

We wish to show the non-singularity of

$$\nabla_v f(v^\star, \epsilon^\star) = \begin{pmatrix} G & -N \\ N^\top & 0 \end{pmatrix}.$$

The Schur complement of $\nabla_v f(v^\star, \epsilon^\star)$ with respect to the lower block diagonal component 0 is $N^\top G^{-1} N$. From assumptions 6.2 and 6.1, $N^\top$ has full row rank and $G \succ 0$. Therefore $N^\top G^{-1} N$ is positive definite and non-singular and equivalently, $\nabla_v f(v^\star, \epsilon^\star) \succ 0$ and non-singular.

The partial gradient of $f(v^\star, \epsilon^\star)$ with respect to ϵ is

$$\nabla_\epsilon f(v^\star, \epsilon^\star) = \begin{pmatrix} J & 0 \\ 0 & 0 \end{pmatrix}.$$

We use Gaussian elimination to invert $\nabla_v f(v^\star, \epsilon^\star)$ and get

$$(\nabla_v f(v^\star, \epsilon^\star))^{-1} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

where A, B, C, D are defined as follows:

$$A = G^{-1} - G^{-1} N (N^\top G^{-1} N)^{-1} N^\top G^{-1},$$

$$B = -G^{-1} N (N^\top G^{-1} N)^{-1},$$

$C = B^\top$, and $D = (N^\top G^{-1} N)^{-1}$. We decompose $v^\top = \begin{pmatrix} y^\top & \nu^\top & \lambda^\top \end{pmatrix}$ and solve for $\nabla_\epsilon y^\star$,

$$\nabla_\epsilon \begin{bmatrix} y^\star \\ \nu^\star \\ \lambda^\star \end{bmatrix} = - \begin{pmatrix} G^{-1}(J - N(N^\top G^{-1} N)^{-1} N^\top G^{-1} J) & 0 \\ -(N^\top G^{-1} N)^{-1} N^\top G^{-1} J & 0 \end{pmatrix},$$

where the first row corresponds to $\nabla_\epsilon y^\star(\epsilon^\star)$ and the second row corresponds to $\nabla_\epsilon \begin{bmatrix} \nu^\star & \lambda^\star \end{bmatrix}^\top$. The first block corresponds to $\nabla_\epsilon y^\star$. Note that because $y^\star(\epsilon^\star) > 0$, we can express the optimal

cost as

$$\ell^\star = \begin{bmatrix} \tilde{E}^\top & \mathbb{1} \end{bmatrix} \begin{bmatrix} \nu^\star \\ \lambda^\star \end{bmatrix} = N \begin{bmatrix} \nu^\star \\ \lambda^\star \end{bmatrix}.$$

The sensitivity of the costs $\ell^\star$ with respect to perturbation is

$$\nabla_\epsilon \ell^\star = N \nabla_\epsilon \begin{bmatrix} \nu^\star \\ \lambda^\star \end{bmatrix} = N(N^\top G^{-1} N)^{-1} N^\top G^{-1} J.$$

■

6.3.1 Stochastic Braess Paradox

The sensitivity of the optimal edge costs and distribution is important from a game design perspective. In the routing game literature, a well-known phenomenon that is related to the sensitivity of optimal distribution is *Braess paradox* [23]. The phenomenon refers to the paradoxical effect that occurs when costs of traversing edges are *decreased*, resulting in an increase in the player's average cost. The occurrence of the Braess paradox has been shown to be related to a routing game's underlying network structure [89].

Here, we show that not only does a similar behavior occur in MDP congestion games, but that a stochastic Braess paradox can be linked to the underlying hypergraph structure through sensitivity analysis. Consider the *social cost* of an MDP congestion game,

$$J(y, \ell) = y^\top \ell(y).$$

The stochastic Braess paradox can be defined by the sensitivity of the social cost of MDP congestion games.

Definition 6.1 (Stochastic Braess Paradox) *Consider two MDP congestion games* (6.2) *defined on the same hypergraph with costs ℓ and $\bar{\ell}$, respectively, such that both satisfies assumption 6.1 and for any feasible stationary mass distribution,*

$$\ell(y) - \bar{\ell}(y) \geq 0, \quad \forall \{y \mid Ey = 0,\ \mathbb{1}^\top y = M,\ y \geq 0\}.$$

Let the optimal distribution be $y^\star$ and $\bar{y}^\star$, a stochastic Braess paradox *occurs when the social cost satisfies $J(y^\star, \ell) < J(\bar{y}^\star, \bar{\ell})$.*

When ℓ is continuously differentiable, the existence of the Braess paradox suggests that there is a perturbation that increases the state-action costs from $\bar{\ell}$ to ℓ such that $J(y^\star, \ell) < J(\bar{y}^\star, \bar{\ell})$. We derive sufficient conditions for the stochastic Braess paradox using the sensitivity of $y^\star$ and $\ell^\star$.

Corollary 6.1 (Sufficient Conditions for stochastic BP) *Consider a feasible MDP congestion game* (6.2) *which satisfies Assumptions 6.1 and 6.2 with an optimal distribution* $y^\star > 0$. *Its* social cost sensitivity *can be defined as*

$$\begin{aligned}\nabla_\epsilon J =& (G^{-1}N(N^\top G^{-1}N)^{-1})^{-1}N^\top G^{-1} - G^{-1})\ell(y^\star) \\ &+ N(N^\top G^{-1}N)^{-1}N^\top G^{-1}y^\star.\end{aligned} \tag{6.6}$$

Then, $\nabla_\epsilon J \notin \mathbb{R}_+^{|\mathcal{S}||\mathcal{A}|}$ *is a sufficient condition for the occurrence of* stochastic Braess paradox.

Proof: J is bilinear and therefore continuously differentiable in $\ell^\star$ and $y^\star$. From Theorem 6.1, there exists a neighbourhood $\|\epsilon\| \leq \delta$ within which J is continuously differentiable in ϵ, and the Jacobian is given as

$$\nabla_\epsilon J(\ell^\star, y^\star) = \nabla_{y^\star} J \nabla_\epsilon y^\star + \nabla_{\ell^\star} J \nabla_\epsilon \ell^\star.$$

For any $\nabla_\epsilon J \notin \mathbb{R}_+^{|\mathcal{S}||\mathcal{A}|}$, there exists $\epsilon \in \mathbb{R}_+^{|\mathcal{S}||\mathcal{A}|}$ such that $\|\epsilon\| \leq \delta$ and $\epsilon^\top \nabla_\epsilon J < 0$. We then consider the MDP congestion game with costs $\bar{\ell}$ and equilibrium $\bar{y}^\star$, where $\bar{\ell}$ is defined by

$$\bar{\ell} = \ell + \epsilon.$$

By the mean value theorem, there exists $k \in (0, 1]$ where

$$J(\bar{y}^\star, \bar{\ell}^\star) = J(y^\star, \ell^\star) + (k\epsilon)^\top \nabla_\epsilon J.$$

Since $k\epsilon^\top \nabla_\epsilon J(\delta) < 0$, $J(\bar{y}^\star, \bar{\ell}) < J(y^\star, \ell^\star)$ holds. ■

6.4 Role of Stochasticity

In this section, we consider the deterministic counterpart of MDP congestion games to evaluate how the introduction of *stochasticity* influences social cost sensitivity.

6.4.1 Cycle Game

A directed *primal graph* [4] $\mathcal{G}_d = ([S], \mathcal{E}_d)$ can be derived from a hypergraph $\mathcal{G} = ([S], \mathcal{E})$, by considering the same set of states and define *edge set* $\mathcal{E}_d$ defined by

$$e = (s_1, s_2) \in \mathcal{E}_d \text{ if } \exists\ (s_1, a) \text{ s.t. } P_{s_2 s_1 a} > 0.$$

Its incidence matrix $D \in \mathbb{R}^{S \times \mathcal{E}_d}$ is given by

$$[D]_{ie} = \begin{cases} 1, & \text{if edge e starts at state i,} \\ -1, & \text{if edge e ends at state i,} \\ 0, & \text{otherwise.} \end{cases}$$

An MDP congestion game (6.2) can be played on $\mathcal{G}_d$ for a given cost ℓ. The constraint $Dy = 0$ implies that any feasible distribution must be a combination of cycles of $\mathcal{G}_d$ [49]. Therefore, we call a deterministic MDP congestion game where all state-action pairs lead to deterministic outcomes, a *cycle game* [29].

The edge set of a primal graph dictates *allowable* transitions over state space $[S]$. A hypergraph's hyperarc set $\mathcal{E}$ corresponds to a discrete set of particular probability distributions assignments to *state-actions* as given by $\mathcal{E}_d$. We consider a transformation $T \in \mathbb{R}_+^{|\mathcal{E}_d| \times |\mathcal{E}|}$ between the incidence matrix of a hypergraph E, and that of its host graph, D, such that $E = DT$. Columns of T denote how an action a distributes mass over edges adjacent to s of the primal graph,

$$T_{(s_1,s_2),(s,a)} = \begin{cases} P_{s_2 a s}, & s_1 = s, \\ 0, & \text{otherwise.} \end{cases} \tag{6.7}$$

In addition to being element-wise non-negative, T is also column stochastic—i.e.,

$$\sum_{e \in \mathcal{E}_d} T_{e,(s,a)} = \sum_{s' \in \mathcal{S}} P_{s' a s} = 1.$$

We provide an example in Fig. 6.2 in which four labeled edges are defined between three

states, $\{A, B, C\}$. The incidence matrix corresponding to Fig. 6.2 is given by

$$D = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 1 & -1 & 0 \\ -1 & 0 & 1 & -1 \end{bmatrix}$$

and the transformation is given by

$$T = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0.6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The eigenvalues of T characterize the amount of stochasticity introduced by the MDP dy-

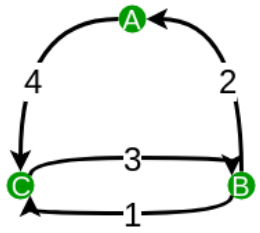


Figure 6.2: Example graph structure of a cycle game.

namics. When $T = I$, the MDP congestion game is itself a cycle game with no stochasticity. When each state-action pair uniformly distributes the probability over available edges, T has a block diagonal structure with eigenvalues less than 1 if a state has two or more actions available. Fig. 6.2 also provides an example of a feasible transformation that is invertible.

6.4.2 Effects of Stochasticity

When the incidence matrix of a hypergraph is related to the incidence matrix of the corresponding primal graph by an invertible transformation T, there is a direct relationship between the equilibria of the MDP congestion game and cycle game played on these graphs.

Assumption 6.4 (Invertible Transformation T) *A directed hypergraph $\mathcal{G} = ([S], \mathcal{E})$ can be induced from its directed primal graph $\mathcal{G}_d = ([S], \mathcal{E}_d)$, such that $|\mathcal{E}| = |\mathcal{E}_d|$, and the incidence*

matrices, E and D, of the two graphs, respectively, are related by an invertible transformation T.

$$E = DT, \qquad T \in \mathbb{R}_+^{|\mathcal{E}_d| \times |\mathcal{E}|},\ \mathbb{1}^\top T = \mathbb{1}^\top.$$

Proposition 6.2 (Equilibria Relationship) *If the graph $\mathcal{G}$ of an MDP congestion game satisfies Assumption 6.4, $y^\star > 0$ is an MDP Wardrop equilibrium if and only if $Ty^\star$ is an equilibrium of the cycle game defined on $\mathcal{G}_d$ with costs ℓ_e on its edges where*

$$\ell_e(\cdot) = T^{-\top} \ell_{sa} \circ T^{-1}(\cdot).$$

Proof: Consider an MDP Wardrop equilibrium $y^\star$ that satisfies Assumption 6.3, then there exists dual variables $\nu^\star$, $\lambda^\star$ that satisfy the KKT conditions (6.3) with $\mu^\star = 0$. We can re-write $H(y, \nu, \lambda, \mu) = 0$ from (6.3) with transformations $DT = E$ and $z = Ty^\star$, and $\mu^\star = 0$,

$$\begin{aligned} T^{-\top} \ell(T^{-1} z) - D^\top \nu^\star - \lambda^\star T^{-\top} \mathbb{1} &= 0, \\ Dz &= 0, \\ \mathbb{1}^\top T^{-1} z - M &= 0. \end{aligned} \tag{6.8}$$

Since T is element-wise non-negative and invertible, and $y > 0$, $Ty = z > 0$. By construction, T^{-1} is column stochastic, therefore $T^{-\top}\mathbb{1} = \mathbb{1}$. Therefore (6.8) is equivalent to

$$\begin{aligned} T^{-\top} \ell(T^{-1} z) - D^\top \nu^\star - \lambda^\star \mathbb{1} &= 0, \\ Dz &= 0, \\ \mathbb{1}^\top z - M &= 0, \\ z &> 0. \end{aligned} \tag{6.9}$$

We note that $T^{-\top}(\nabla \ell) T^{-1}$ is positive definite, and while an individual state-action cost $(T^{-\top} \circ \ell \circ T^{-1})_{sa}$ requires *multiple* hyperarcs' population distribution to define the congestion cost at (s, a), it defines a potential game [27] consistent with Assumption 6.1. This implies that (6.9) coincides with the KKT conditions of a cycle game formulation with costs $T^{-\top} \circ \ell \circ T^{-1}$, incidence matrix D, and mass M. Since $z > 0$ satisfies the KKT conditions of this cycle game, z is the cycle game's unique optimal distribution. ■

The relationship between the equilibria of the deterministic game and the equilibria of the game allows for a direct comparison between the sensitivity of the social cost in the two games. We show next that the social cost sensitivity of an MDP congestion game can be

directly bounded by the eigenvalues of T, ie the amount of stochasticity introduced.

Theorem 6.2 (Effects of Stochasticity) *We consider an MDP congestion game (6.2) and a cycle game (Section 6.4.1) whose graphs satisfy Assumption 6.2. Let the social cost of the cycle game be J_c, and the social cost of the MDP congestion game be J, the sensitivity of the cycle game can be bounded by*

$$\|\nabla_\epsilon J_c\|_2 \leq \|T\|_2 \|\nabla_\epsilon J\|_2 .$$

Proof: Let $N_c = \begin{bmatrix} \bar{D}^\top & \mathbb{1} \end{bmatrix}$, where $\bar{D}$ is D with any row of D removed. We note that from Assumption 6.2, the removed row cannot be identically zero as that would ensure $\mathbf{rank}(D) \leq S-2$, then N_c is related to $N = \begin{bmatrix} \bar{E}^\top & \mathbb{1} \end{bmatrix}$ by $T^\top N_c = N$ where $\bar{E}$ has the same row removed.

Since $z^\star = Ty^\star$, the sensitivity of the cycle game social cost $J_c = (z^\star)^\top T^{-\top} \ell(T^{-1} z^\star)$ can be evaluated at $(y^\star, \ell^\star)$,

$$\nabla_\epsilon J_c \begin{pmatrix} y^\star \\ \ell(y^\star) \end{pmatrix} = \begin{pmatrix} T^{-\top} A T^\top T & 0 \\ 0 & TB \end{pmatrix} \begin{pmatrix} y^\star \\ \ell(y^\star) \end{pmatrix}.$$

where $A = N(N^\top G^{-1} N)^{-1} N^\top G^{-1}$ and $B = G^{-1} - G^{-1} N (N^\top G^{-1} N)^{-1})^{-1} N^\top G^{-1}$. In comparison, the sensitivity of the MDP congestion game's social cost is

$$\nabla_\epsilon J \begin{pmatrix} y^\star \\ \ell(y^\star) \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} y^\star \\ \ell(y^\star) \end{pmatrix}.$$

We can compare the social cost sensitivity Jacobian for the cycle game and the MDP congestion game, denoted by M_c and M respectively.

$$\begin{aligned} \|M_c\|_2 &= \sigma_{max}\{T^{-\top} A T^\top T, TB\} \\ &= \|A\|_2 \|T\|_2 + \|T\|_2 \|B\|_2 \\ &\leq \|T\|_2 \|M\|_2 . \end{aligned} \tag{6.10}$$

■

Intuitively, Theorem 6.2 states that given equivalent MDP Wardrop equilibria, the sensitivity of the social cost in the deterministic cycle game is always bounded by the sensitivity of the MDP congestion game and the amount of stochasticity introduced. Since $\|T\|_2 \leq 1$,

Theorem 6.2 states that introducing stochasticity *increases* effects of Braess paradox.

6.5 Simulations

In this section, we use the results of sensitivity analysis on a hypergraph derived from a directed Wheatstone graph. Wheatstone structure is known to induce the Braess paradox for non-atomic routing games [89], we analyze its behavior under stochastic transitions and show that not only does stochastic Braess paradox also occur, but we can avoid the paradox by our sensitivity analysis. We demonstrate Theorem 6.1 by cost perturbations in both the negative and positive directions of the social cost sensitivity and validate the predictions with simulated results.

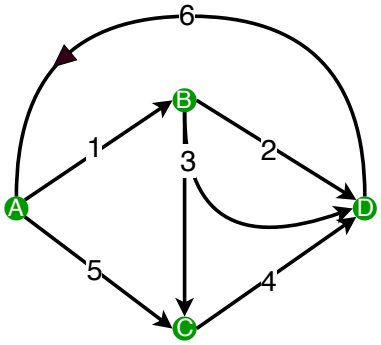


Figure 6.3: Hypergraph structure of MDP congestion game

Consider an MDP congestion game defined on the hypergraph shown in Figure 6.3. We play the MDP congestion game defined by (6.2), with a scaled mass distribution of $M = 1$. The cost functions are defined as $\ell_{sa}(y_{sa}) = A_{sa}y_{sa} + b_{sa}$.

All state-action pairs correspond to hyperarcs, but all state-action pairs except for hyperarc 3 define deterministic actions. The stochastic incidence matrix is defined by

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & -1 \\ -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.9 & 1 & -1 & 0 \\ 0 & -1 & -0.1 & -1 & 0 & 1 \end{pmatrix}.$$

Note that when a hyperarc has one head state, its corresponding column of incidence matrix E is identical to that of the cycle game incidence matrix D (Section 6.4.1). Stochastic

	A_{sa}	b_{sa}
ℓ_1	9	1
ℓ_2	0.1	1
ℓ_3	0.1	0
ℓ_4	9	1
ℓ_5	0.1	0.1
ℓ_6	0.1	0

Table 6.1: Distribution dependent hyperarc costs

hyperarcs are convex combinations of the deterministic edges that correspond to allowable state transitions originating from the same tail state.

We simulate each MDP congestion game by solving the convex optimization formulation given by (6.2) with cvxpy. First, we verify in Figure 6.4 that at given costs ℓ, the optimal distribution $y^\star$ is strictly positive.

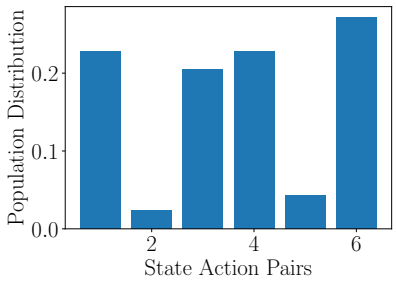


Figure 6.4: Optimal Distribution at with link costs from table 6.1

We consider perturbing the hyperarc costs modeled by

$$\bar{\ell}(\cdot, \epsilon) = \ell(\cdot) + \epsilon.$$

The sensitivity of social cost can be analytically derived from theorem 6.1 based on the hypergraph structure as

$$\nabla_\epsilon J = \begin{pmatrix} 0.023 & 0.501 & -0.478 & 0.023 & 0.454 & 0.477 \end{pmatrix}^\top.$$

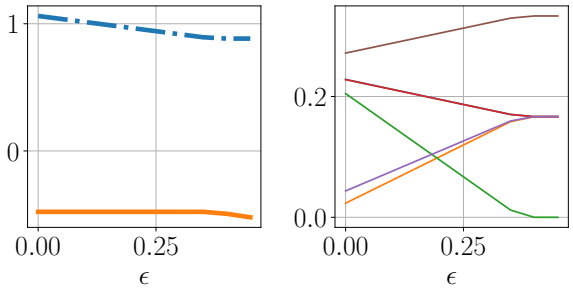


Figure 6.5: Braess Paradox: Perturbing game costs with $\epsilon[0, 0, 1, 0, 0, 0]$, where $\epsilon \in \mathbb{R}_+$ increases along x-axis. The right shows the game's optimal distribution on each hyperarc. Left shows the social cost at optimal distribution (blue) and the sensitivity for hyperarc 3 varying with ϵ (orange).

The sensitivity vector $\nabla_\epsilon J$ implies that increasing the third hyperarc cost would result in the most decrease in social cost while increasing the second hyperarc cost would result in the most increase in social cost. We verify both scenarios by successively increasing ϵ and re-evaluating the social cost at the optimal distribution $y^\star(\epsilon)$, as solved by cvxpy. The results are shown in Figures 6.5 and 6.6.

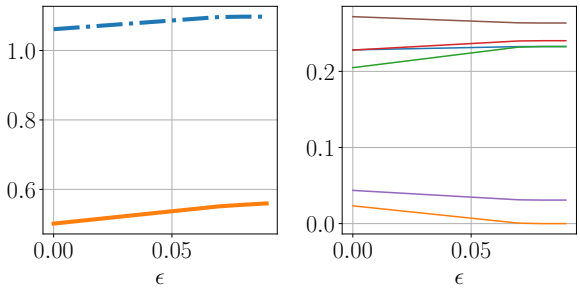


Figure 6.6: No Braess Paradox: Perturbing the game costs with $\epsilon[0, 1, 0, 0, 0, 0]$, where $\epsilon \in \mathbb{R}_+$ increases along x-axis. The right shows the game's optimal distribution on each hyperarc. Left shows the social cost at optimal distribution (blue) and the sensitivity value for hyperarc 2 at given ϵ (orange).

A couple of conclusions can be drawn from Figures 6.5 and 6.6. First, we see that there

exists a continuous region around ϵ where $y^\star(\epsilon) > 0$, and therefore renders this sensitivity analysis valid. Figure 6.5 shows a negative sensitivity value for the third hyperarc as we increase ϵ, which implies the stochastic Braess paradox. Then as predicted, the social cost *decreases* as ϵ is increased. In contrast, Figure 6.6 shows a positive sensitivity value for the second hyperarc as we increase ϵ, therefore the social cost should not decrease as ϵ increases. This is also confirmed as the social cost obtained from the output of cvxpy increases with ϵ. Both the Braess paradox and the absence of the Braess paradox are correctly predicted for the regions where positive mass exists on every hyperarc.

Conclusion We derived sensitivity analysis for MDP congestion games when the optimal mass distribution is strictly positive. From the sensitivity of optimal cost and distribution to changes in state-action cost, we derived sufficient conditions for the occurrence of stochastic Braess paradox defined in terms of network and cost structure. Finally, we considered the effects of stochasticity on the magnitude of the Braess paradox. Our simulations explicitly show the occurrence of stochastic Braess paradox on MDP congestion games. Future work include generalizing the analysis to MDP congestion games whose optimal mass distribution is not strictly positive.

Chapter 7

VALUE FUNCTION SET INVARIANCE FOR SET-BASED VALUE OPERATORS

MDP is a versatile model for decision-making in stochastic environments and is widely used in trajectory planning [5], robotics [134], and operations research [37]. Given state-action costs and transition probabilities, finding an optimal policy of the MDP is equivalent to solving for the fixed point of the corresponding Bellman operator. This method is known as dynamic programming and extends to the model-free setting via value-based reinforcement learning [127].

Many applications of MDPs, including traffic light control, motion planning, and dexterous manipulation, experience *environmental non-stationarity*—dynamically changing MDP cost and transition probabilities due to external factors or the presence of interfering decision makers. This environmental non-stationarity differs from stochasticity which is already modeled by the MDP. For example, we can use a stationary Gaussian distribution to model the arrival time uncertainty of a given flight. However, if a known windstorm disrupts the flight's trajectory, then the mean and variance of the arrival time distribution will be different.

To protect performance output against environmental non-stationarity, a common approach is to optimize the worst-case performance via robust control. Robust control-type approaches for MDPs include robust dynamic programming [54, 95], risk-sensitive reinforcement learning [88], and zero-sum stochastic games [51]. By assuming that the MDP parameters are chosen adversarially, the worst-case approach can derive asymptotic bounds of the value function trajectory affected by the parameter non-stationarity.

Within robust dynamic programming, much of the focus is on the worst-case analysis of the *MDP policy improvement* and *MDP policy evaluation* problem under a rectangularity assumption on a set of MDP parameters. In the model-based setting, policy improvement and policy evaluation correspond to the Bellman and policy evaluation operators on the space of value functions. However, recent progress in learning-based methods utilizes other operators such as Q-learning [138] and temporal difference [129]. To broaden the application of our results, we ground our analysis in a more general class of value operators, and ask

the following question from a less adversarial perspective: is it possible to characterize the transient behavior of a value function-based contraction operator with dynamically changing MDP parameters?

Contributions. For compact sets of MDP parameter uncertainty, we propose a set-extension of *value operators*: a general class of contraction operators that are order-preserving on the space of value functions and Lipschitz in the space of MDP parameters. We prove the existence of compact *fixed point sets* of the set-based value operators and show that the set-based value iteration converges. In a non-stationary Markovian environment, standard value iteration may not converge. However, we can show that the point-to-set distance of the resulting value function trajectory to the fixed point set always goes to zero in the limit. We derive a *containment condition* that is sufficient for the fixed point sets to contain their extremal elements. Within robust MDPs, we show that the containment condition generalizes the rectangularity condition, such that the optimal worst-case policy, or the robust policy, exists when the containment condition is satisfied. We then derive the relationship between the fixed point sets of 1) the set-based optimistic policy evaluation operator, 2) the set-based robust policy evaluation operator, and 3) the set-based Bellman operator. Given a value operator and a compact MDP parameter uncertainty set, we present an algorithm that computes the bounds of the corresponding fixed point set and derives its convergence guarantees.

Related research. MDP with parameter uncertainty is well-studied in robust control and reinforcement learning. In control theory, the worst-case cost-to-go under state-decoupled parameter uncertainties is derived via the min-max variation of the Bellman operator in [46, 54, 95, 139]. The cost-to-go under parameter uncertainty with coupling between states and time steps is similarly bounded in [81, 50]. The effect of statistical uncertainty on the optimal cost-to-go is studied in [95, 81, 139, 141]. Recently, MDP with parameter uncertainty has gained traction in the reinforcement learning community due to the presence of uncertainty in real-world problems such as traffic signal control and multi-agent coordination [60, 65, 101]. Most RL research extends the min-max worst-case analysis to methods such as Q-learning and SARSA. Recently, methods for value-based RL using non-contracting operators have been investigated in [17].

As opposed to the worst-case approach to analyzing MDPs under parameter uncertainty, we do not assume adversarial MDP parameter selection. Instead, we derive a set of cost-to-gos that is invariant with respect to the compact parameter uncertainty sets for order-

preserving, α-contracting operators, a class that the Bellman operator belongs to.

Chapter-specific notation. Matrices, vectors, and some integers are denoted by capital letters, X, while sets are denoted by cursive typeset $\mathcal{X}$. The set of all *compact subsets* of $\mathbb{R}^d$ is denoted by $\mathcal{K}(\mathbb{R}^d)$. A vector $h \in \mathbb{R}^S$ has equivalent notation $(h_1, \ldots, h_s)$, where h_s is the value of h in the s^{th} coordinate, $s \in [S]$. Throughout the paper, $\|\cdot\|$ denotes the infinity norm in $\mathbb{R}^S$.

7.1 Discounted Infinite-horizon MDP

We re-introduce the discounted infinite horizon MDP definition from Chapter 2.2 using the parameters $([S], [A], P, C, \gamma)$, and augment the existing MDP parameter definition with matrix and vector-based notation to facilitate this chapter's discussion.

MDP costs. $C \in \mathbb{R}^{S\times A}$ is the matrix encoding the expected MDP cost. Each $C_{sa} \in \mathbb{R}$ is the cost of taking action $a \in [A]$ from state $s \in [S]$. We also denote the cost of all actions at state s by the vector $c_s = [C_{s1}, \ldots, C_{sA}] \in \mathbb{R}^A$, such that $C = [c_1, \ldots, c_S]^\top$.

MDP transition dynamics. The transition probabilities when action a is taken from state s are given by the vector $p_{sa} \in \Delta_S$. Collectively, all possible transition probabilities from state $s \in [S]$ are given by the matrix $P_s = [p_{s1}, \ldots, p_{sA}] \in \Delta_S^A \subset \mathbb{R}^{S\times A}$, and all possible transition probabilities in the MDP are given by the matrix $P = [P_1, \ldots, P_S] \in \Delta_S^{SA} \subset \mathbb{R}^{S\times SA}$.

MDP objective. We want to minimize the expected cost-to-go, or the **value vector** $V \in \mathbb{R}^S$, defined per state as

$$V_s := \mathbb{E}_s\Big\{ \textstyle\sum_{t=0}^{\infty} \gamma^t C_{s^t a^t} \mid s^0 = s \Big\}, \ \forall\ s \in [S], \tag{7.1}$$

where $\mathbb{E}_s\{\cdot\}$ is the expected value of the input with respect to initial state s, and (s^t, a^t) are the state and action at time t. The value vector can be minimized by the choice of actions a^t at every time step t:

$$V_s^\star := \min_{a^t \in [A]}\ \mathbb{E}_s\Big\{ \textstyle\sum_{t=0}^{\infty} \gamma^t C_{s^t a^t} \mid s^0 = s \Big\}, \ \forall\ s \in [S], \tag{7.2}$$

Remark 7.1 *Although* value function *is the standard term for the expected cost-to-go state values, we use the term value vector instead to emphasize that the cost-to-go state values of finite MDPs belong in a finite-dimensional space.*

MDP Policy. We optimize the objective (7.2) via a *policy*, denoted as $\pi = [\pi_1, \ldots, \pi_S] \in \Delta_A^S$, where the a^{th} element of $\pi_s \in \Delta_A$ is the conditional probability of action a being chosen from state s. Under policy π_s, the expected immediate cost at s is given by $c_s^\top \pi_s \in \mathbb{R}$ and the expected transition probabilities from s is given by $P_s \pi_s \in \Delta_S$.

7.2 Value Operators

Solving an MDP is equivalent to finding the value vector and the associated policy that minimizes the objective (7.2). Typical solution methods utilize *order preserving* [119, Def.3.1], *α-contractive operators* whose fixed points are the optimal value vectors (e.g. Bellman operator [107, Thm.6.2.3], Q-value operator [86]).

Definition 7.1 (α-Contraction) *Let $(\mathcal{X}, d)$ be a metric space with metric d. The operator $H : \mathcal{X} \mapsto \mathcal{X}$ is an α-contraction if and only if there exists $\alpha \in [0, 1)$ such that*

$$d(H(V), H(V')) \leq \alpha d(V, V'), \quad \forall\ V,\ V' \in \mathcal{X}. \tag{7.3}$$

Definition 7.2 (Order Preservation) *Let $(\mathcal{X}, \leq)$ be an ordered space with partial order $\leq$. The operator $H : \mathcal{X} \mapsto \mathcal{X}$ is order preserving if for all $V, V' \in \mathcal{X}$ such that $V \leq V'$, $H(V) \leq H(V')$.*

These operators are typically locally Lipschitz in MDP parameter space.

Definition 7.3 ($K(V)$-Lipschitz) *Let $(\mathcal{X}, d_\mathcal{X})$ be a metric space with metric $d_\mathcal{X}$ and $(\mathcal{Y}, d_Y)$ be a metric space with metric $d_\mathcal{Y}$. The operator $H : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{X}$ is $K(V)$-*Lipschitz *for $\mathcal{M} \subset \mathcal{Y}$ if for all $V \in \mathcal{X}$, there exists $K(V) \in \mathbb{R}_+$ such that*

$$d_\mathcal{X}(H(V, m), H(V, m')) \leq K(V) d_\mathcal{Y}(m, m'),\ \forall m, m' \in \mathcal{M}. \tag{7.4}$$

Remark 7.2 *The α-contraction property is a special instance of Lipschitz continuity in which the input and output spaces are identical and the Lipschitz constant is less than* 1.

To capture operators with these properties, we define a **value operator** that takes three inputs: a value vector, an MDP cost matrix, and an MDP transition probability matrix. The MDP cost and transition probability are selected from an MDP parameter set $\mathcal{M}$.

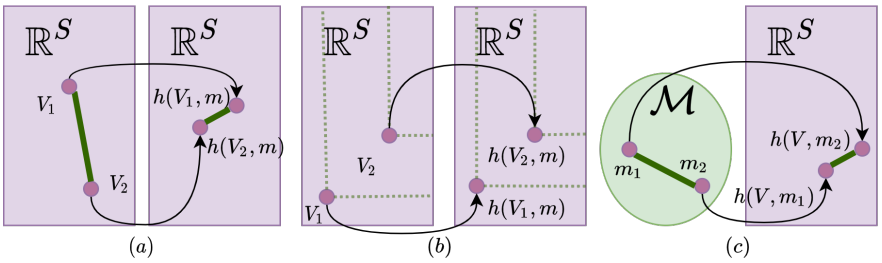


Figure 7.1: Illustration of the three value operator properties. (a) α-contraction on $\mathbb{R}^S$, (b) Order preservation on $\mathbb{R}^S$, and (c) $K(V)$-Lipschitz in input space $\mathcal{M}$.

Definition 7.4 (Value operator) *Consider the operator h,*

$$h : \mathbb{R}^S \times \mathcal{M} \mapsto \mathbb{R}^S, \ \mathcal{M} \subseteq \mathbb{R}^{S \times A} \times \Delta_S^{SA}. \tag{7.5}$$

*We say h (7.5) is a **value operator** on $\mathbb{R}^S \times \mathcal{M}$ if*

1. *For all $m \in \mathcal{M}$, $h(\cdot, m)$ is an α-contraction in $\mathbb{R}^S$.*
2. *For all $m \in \mathcal{M}$, $h(\cdot, m)$ is order preserving in $\mathbb{R}^S$.*
3. *For all $V \in \mathbb{R}^S$, $h(V, \cdot)$ is $K(V)$-Lipschitz on $\mathcal{M}$.*

Remark 7.3 *While we only consider value operators whose input's first component is $\mathbb{R}^S$, Definition 7.4 and the subsequent results can be extended to the space of Q-value functions by replacing $\mathbb{R}^S$ with $\mathbb{R}^{SA}$ in Definition 7.4 [86].*

An immediate consequence of the value operator h being an α-contractive and order-preserving operator on $\mathbb{R}^S$ is that h is continuous on $\mathbb{R}^S \times \mathcal{M}$.

Lemma 7.1 (Continuity) *If h (7.5) is a value operator on $\mathbb{R}^S \times \mathcal{M}$, then h is continuous on $\mathbb{R}^S \times \mathcal{M}$.*

Proof: Let $(V, m) \in \mathbb{R}^S \times \mathcal{M}$ and consider a sequence $\{(V_k, m_k)\}_{k \in \mathbb{N}} \subset \mathbb{R}^S \times \mathcal{M}$ that converges to (V, m). It holds that $\|h(V_k, m_k) - h(V, m)\| \leq \|h(V_k, m_k) - h(V, m_k)\| + \|h(V, m_k) - h(V, m)\|$, where from the α-contractive property of $h(\cdot, m^k)$, $\|h(V_k, m_k) - h(V, m_k)\| \leq \alpha \|V_k - V\|$. From the $K(V)$-Lipschitz property of $h(V, \cdot)$,

$$\|h(V, m_k) - h(V, m)\| \leq K(V) \|m_k - m\|.$$

As both $\lim_{k\to\infty} \|V_k - V\| \to 0$ and $\lim_{k\to\infty} \|m_k - m\| \to 0$, $\|h(V_k, m_k) - h(V, m)\| \to 0$ and h is continuous. ■

We make the following assumption on the MDP parameter set $\mathcal{M}$ with respect to h.

Assumption 7.1 (Containment condition) *The MDP parameter set $\mathcal{M}$ satisfies the containment condition with respect to h if $\mathcal{M}$ is compact and for all $V \in \mathbb{R}^S$,*

$$\bigcap_{s\in[S]} \operatorname*{argmin}_{m\in\mathcal{M}} h_s(V, m) \neq \emptyset, \quad \bigcap_{s\in[S]} \operatorname*{argmax}_{m\in\mathcal{M}} h_s(V, m) \neq \emptyset. \tag{7.6}$$

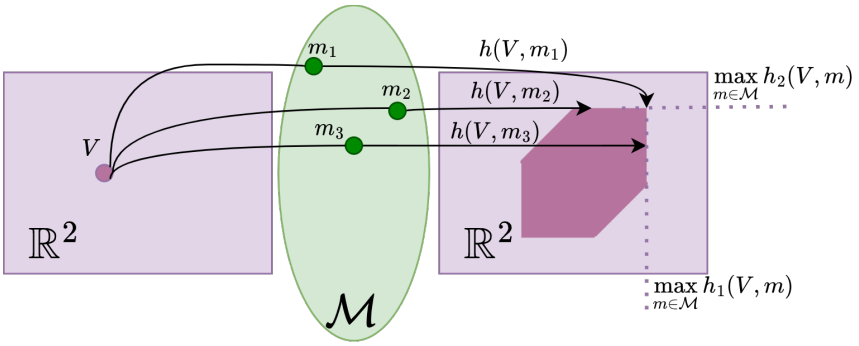


Figure 7.2: We illustrate $\operatorname{argmax}_{m\in\mathcal{M}} h_s(V, m)$ for a value operator h when $S = 2$. Here, $\operatorname{argmax}_{m\in\mathcal{M}} h_1(V, m) = \{m_2, m_3\}$, $\operatorname{argmax}_{m\in\mathcal{M}} h_2(V, m) = \{m_1, m_2\}$. Therefore, m_2 is the common parameter that achieves $\max_{m\in\mathcal{M}} h_s(V, m)$ for all $s \in [S]$.

Remark 7.4 *Assumption 7.1 defines the containment condition as a property of the set $\mathcal{M}$ and requires that $\mathcal{M}$ satisfies specific constraints with respect to the operator h. The containment condition is independent of $\mathcal{M}$'s convexity and connectivity.*

7.3 Bellman and Policy Evaluation Operators

Examples of value operators include the Bellman operator and the policy evaluation operators when the MDP cost and transition probability are input parameters rather than fixed parameters.

Definition 7.5 (Policy evaluation operator) *Given a policy $\pi \in \Pi$, the vector-valued operator $g^\pi = (g_1^\pi, \ldots, g_S^\pi) : \mathbb{R}^S \times \mathbb{R}^{S\times A} \times \Delta_S^{SA} \mapsto \mathbb{R}^S$ is defined per state as*

$$g_s^\pi(V, C, P) := c_s^\top \pi_s + \gamma \Big(P_s \pi_s\Big)^\top V, \ \forall s \in [S]. \tag{7.7}$$

Given (C, P), $g^\pi(\cdot, C, P) : \mathbb{R}^S \mapsto \mathbb{R}^S$ *is a vector-valued operator whose fixed point is the expected cost-to-go of the MDP* $([S], [A], C, P, \gamma)$ *under* π, *denoted as* $V^\pi(C, P)$ *[107, Thm.6.2.5].*

$$V^\pi(C, P) = g^\pi(V^\pi, C, P),\ V^\pi(C, P) \in \mathbb{R}^S. \tag{7.8}$$

When the context is clear, we denote $V^\pi(C, P)$ *as* V^π.

Definition 7.6 (Bellman operator) *The vector-valued operator* $f = (f_1, \ldots, f_S) : \mathbb{R}^S \times \mathbb{R}^{S\times A} \times \Delta_S^{SA} \mapsto \mathbb{R}^S$ *is defined per each state as*

$$f_s(V, C, P) := \inf_{\pi_s \in \Delta_A} g_s^\pi(V, C, P),\ \forall\, s \in [S]. \tag{7.9}$$

The corresponding optimal policy $\pi^\star = (\pi_1^\star, \ldots, \pi_s^\star)$ *is defined per state as* $\pi_s^\star \in \operatorname{argmin}_{\pi_s} g_s^\pi(V, C, P)$ (7.9) *and satisfies* $\mathbb{1}_S^\top \pi_s^\star = 1\ \forall s \in [S]$. *One such policy is defined for all* $(s, a) \in [S] \times [A]$ *by*

$$\pi_{s,a}^\star := \begin{cases} > 0 & a \in \underset{a\in[A]}{\operatorname{argmin}}\ C_{sa} + \gamma p_{sa}^\top V, \\ 0 & \textit{otherwise.} \end{cases} \tag{7.10}$$

where $\operatorname{argmin}_{a\in[A]}(h)$ *is the set of minimizing actions for the function* h. *An optimal policy in the form* (7.10) *always exists for a discounted infinite horizon MDP [107, Thm 6.2.10]. Given parameters* (C, P), $f(\cdot, C, P) : \mathbb{R}^S \mapsto \mathbb{R}^S$ *is a vector operator whose fixed point is the optimal cost-to-go for the MDP* $([S], [A], P, C, \gamma)$, *denoted as* $V^B(C, P)$.

$$V^B(C, P) = f(V^B, C, P),\ V^B(C, P) \in \mathbb{R}^S. \tag{7.11}$$

When the context is clear, we denote $V^B(C, P)$ *as* V^B.

We show that both (7.7) and (7.9) are value operators.

Lemma 7.2 *The Bellman operator* (7.9) *and the policy evaluation operators* (7.7) *for all* $\pi \in \Pi$ *are value operators on* $\mathbb{R}^S \times \mathcal{M}$ *where* $\mathcal{M} \subseteq \mathbb{R}^{S\times A} \times \Delta_S^{SA}$ (7.5).

Proof: We show that both the Bellman operator f and the policy evaluation operator g^π satisfy the contractive, order preserving and Lipschitz properties given in Definition 7.4. Contraction: given $(C, P) \in \mathcal{M}$, $g^\pi(\cdot, C, P)$ and $f(\cdot, C, P)$ are both γ-contractions [107, Prop.6.2.4] on the complete metric space $(\mathbb{R}^S, \|\cdot\|_\infty)$, where $\gamma < 1$ is the discount factor.

Order preservation: given $(C, P) \in \mathcal{M}$, the operator $g^\pi(\cdot, C, P)$ is order preserving [107, Lem.6.1.2]. Consider $U, V \in \mathbb{R}^S$ where $U \leq V$. If $g^\pi(\cdot, C, P)$ is order-preserving, $g^\pi(U, C, P) \leq g^\pi(V, C, P)$ for all $\pi \in \Pi$. Taking the infimum over Π, we have $f(U, C, P) = \inf_{\pi\in\Pi} g^\pi(U, C, P) \leq \inf_{\pi\in\Pi} g^\pi(V, C, P) = f(V, C, P)$.

$K(V)$-Lipschitz: given $(C, P), (C', P') \in \mathcal{M}$ and $V \in \mathbb{R}^S$, let $\hat{\pi}$ (7.10) be the optimal policy for $f(V, C', P')$ and $\pi^\star$ be the optimal policy for $f(V, C, P)$. For $s \in [S]$, suppose $f_s(V, C', P') \geq f_s(V, C, P)$, then $0 \leq f_s(V, C', P') - f_s(V, C, P) \leq (c'_s)^\top \hat{\pi}_s - c_s^\top \pi_s^\star + \gamma(P'_s\hat{\pi}_s)^\top V - \gamma(P_s\pi_s^\star)^\top V$. Since $\pi^\star$ is sub-optimal for $f(V, C', P')$, we can upper bound $|f_s(V, C', P') - f_s(V, C, P)| \leq (c'_s - c_s)^\top \pi_s^\star + \gamma[(P'_s - P_s)\pi_s{}^\star]^\top V$. We conclude that

$$|f_s(V, C', P') - f_s(V, C, P)| \leq \|c'_s - c_s\|_\infty + \gamma \|P'_s - P_s\|_\infty \max\left\{ \|\pi_s^\star\|_\infty, \|\hat{\pi}_s\|_\infty \right\} \|V\|_\infty . \tag{7.12}$$

Since $\pi_s^\star, \hat{\pi}_s \in \Delta_A$, $\|\pi_s^\star\|_\infty \leq 1$ and $\|\hat{\pi}_s\|_\infty \leq 1$. By similar arguments, (7.12) is true if $f_s(V, C', P') \leq f_s(V, C, P)$. We can upper bound $f(V, m) - f(V, m') = f - f'$ as

$$\|f - f'\|_\infty \leq \max_{s\in[S]} \{\|c'_s - c_s\|_\infty + \gamma \left\|(P_s - P'_s)^\top V\right\|_\infty\} \tag{7.13}$$

$$\leq \max(1, \gamma \|V\|_\infty) \|m - m'\|_\infty . \tag{7.14}$$

The policy evaluation operator g^π satisfies (7.12) if $\max\{\|\pi_s^\star\|_\infty, \|\hat{\pi}_s\|_\infty\}$ is replaced by $\|\pi_s\|_\infty$. Since $\|\pi_s\|_\infty \leq 1$, g^π is $K(V)$-Lipschitz. ■

Remark 7.5 *Beyond the policy evaluation operator and the Bellman operator, many algorithms in reinforcement learning can be reformulated using value operators. For example, it's not difficult to show that the Q-learning operator [86] is a value operator on the vector space* $\mathbb{R}^{SA}$.

7.4 Containment-satisfying MDP Parameter Sets

Assumption 7.1 restricts the structure of $\mathcal{M}$ with respect to the value operator h. Thus whether or not $\mathcal{M}$ satisfies Assumption 7.1 must always be determined with respect to the value operator h. When we take the value operator to be the Bellman operator f (7.9) and the policy evaluation operators g^π (7.7), the following conditions in robust MDP are sufficient for $\mathcal{M}$ to satisfy the containment condition laid out in Assumption 7.1.

Definition 7.7 ((s,a)-rectangular sets [54, 95]) *The MDP parameter set $\mathcal{M} \subset \mathbb{R}^{S\times A} \times \Delta_S^{SA}$ is (s,a)-rectangular if*

$$\mathcal{M} = \bigtimes_{(s,a)\in[S]\times[A]} \mathcal{M}_{sa},\ \mathcal{M}_{sa} \subset \mathbb{R}\times\Delta_S,\ \forall (s,a)\in[S]\times[A]. \tag{7.15}$$

Intuitively, (s,a)-rectangularity implies that the MDP parameter uncertainty is *decoupled* between each state-action. A more general condition is if the parameter uncertainty is decoupled between different states but not between different actions within the same state.

Definition 7.8 (s-rectangular sets) *The uncertainty set $\mathcal{M} \subset \mathbb{R}^{S\times A}\times\Delta_S^{SA}$ is s-rectangular if*

$$\mathcal{M} = \bigtimes_{s\in[S]} \mathcal{M}_s,\ \mathcal{M}_s \subset \mathbb{R}^A \times \Delta_S^A,\ \forall s\in[S]. \tag{7.16}$$

s-rectangularity generalizes (s,a)-rectangularity—i.e. (s,a)-rectangularity implies s-rectangularity.

Example 7.1 (Wind uncertainty) *Consider an MDP in which the states correspond to geographical coordinates, the actions correspond to navigation choices (up, down, left, right), and the transition probabilities correspond to the local wind patterns that vary between N major wind trends over time per state. At state $s\in[S]$, the transition probabilities of trend $i\in[N]$ are given by P_s^i. If the wind pattern strictly switches between the discrete wind trends, then the transition uncertainty at state $s\in[S]$ is $\mathcal{P}_s = \{P_s^1,\dots,P_s^N\}$. If the wind pattern is a mixture of the discrete wind trends, the transition uncertainty at state $s\in[S]$ is $\mathcal{P}_s = \{\sum_i \alpha_i P_s^i \mid \alpha\in\Delta_N\}$. Both wind patterns lead to s-rectangular uncertainty, given by $\mathcal{P} = \bigtimes_{s\in[S]}\mathcal{P}_s$.*

We show that the rectangularity conditions indeed are sufficient for satisfying Assumption 7.1 with respect to f (7.9) and g^π (7.7).

Proposition 7.1 *If $\mathcal{M}$ is compact and s-rectangular (Definition 7.8), $\mathcal{M}$ satisfies Assumption 7.1 with respect to the Bellman operator f (7.9) and the policy evaluation operator g^π (7.7) for all $\pi\in\Pi$.*

Proof: We first show that $\mathcal{M}$ satisfies Assumption 7.1 with respect to the Bellman operator. Given $s\in[S]$, $f_s(V,C,P)$ only depends on the s component of C and P. From Lemma 7.1, f_s is continuous in (c_s,P_s). Let $(c_s^\star,P_s^\star)$ be the solution to $\operatorname{argmin}_{(c_s,P_s)\in\mathcal{M}_s} f_s(V,C,P)$ for all $\forall\ s\in[S]$. If $\mathcal{M}_s$ is compact, $(c_s^\star,P_s^\star)\in\mathcal{M}_s$. We can construct $C^\star = [c_1^\star,\dots,c_S^\star]$ and

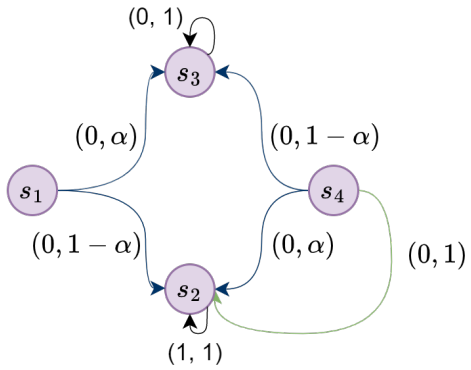


Figure 7.3: MDP with parameter coupling in transition probability across different states.

$P^\star = [P_1^\star, \ldots, P_S^\star]$. If $\mathcal{M}$ is s-rectangular, then $(C^\star, P^\star) \in \mathcal{M}$ and $(C^\star, P^\star) \in \operatorname{argmin}_{m \in \mathcal{M}} f_s(V, C, P)$ for all $s \in [S]$. We conclude that $\mathcal{M}$ satisfies Assumption 7.1.

Given $\pi \in \Pi$ and $s \in [S]$, g_s^π only depends on c_s and P_s as well. We can similarly show that there exists an optimal parameter $(C^\star, P^\star) \in \operatorname{argmin}_{(C,P) \in \mathcal{M}} g_s^\pi(V, C, P)$ for all $s \in [S]$ such that $(C^\star, P^\star) \in \mathcal{M}$. ■

There are sets which satisfy Assumption 7.1 with respect to specific value operators, but not s-rectangularity.

Example 7.2 (Beyond rectangularity) *In Figure 7.3, we visualize a four-state MDP with transition uncertainty $\mathcal{M}$ parameterized by α. MDP states are the nodes and MDP actions are the arrows. Actions that transition to multiple states are visualized by multi-headed arrows. Each head has an associated tuple $(c_{sa}, p_{sa,s'})$ denoting its state-action cost and transition probability. All states have a single action except for state s_4, where two actions exist and are distinguished by different colors. Both s_2 and s_3 are absorbing states with a unique action, such that $V_2 = \frac{1}{1-\gamma}$ and $V_3 = 0$ for both f and g^π for all $\pi \in \Pi$, where γ is the discount factor.*

The states s_1 and s_4 have transition uncertainty parametrized by $\alpha \in [0, 1]$. Therefore, $\mathcal{M}$ violates s-rectangularity (Definition 7.8). The optimal cost-to-go values V_1 and V_4 occur at different α's. Therefore, $\mathcal{M}$ violates Assumption 7.1 with respect to f. However, suppose that at s_4, we only consider policies that exclusively choose the action colored green in Fig. 7.3. Then the expected cost-to-go at s_4, V_4, is independent of α. The minimum and maximum values of V_1 under π occur at $\alpha = 1$ and $\alpha = 0$, respectively. Therefore, $\mathcal{M}$ satisfies Assumption 7.1 with respect to operator g^π for all $\pi = [\pi_{s_1}, \ldots, \pi_{s_4}]$ where $\pi_{s_4} = [1, 0]$.

7.5 Set-based Value Operators

Motivated by the uncertain MDP parameters encountered in robust MDP, stochastic games, and reinforcement learning in uncertain environments, we now consider value operators with respect to a compact set of uncertain MDP parameters. To understand the effect of both stationary and dynamic parameter uncertainty on the value vector, we extend value operators to set-based value operators, and prove the existence of fixed point sets on the space of compact subsets of $\mathbb{R}^S$.

To facilitate our set-based analysis, we first introduce Hausdorff-type set distances.

Definition 7.9 (Point-to-set Distance) *The distance between a value vector and a set $\mathcal{V} \subseteq \mathbb{R}^S$ is given by*

$$W \mapsto d(W, \mathcal{V}) := \inf_{V \in \mathcal{V}} \|W - V\| . \tag{7.17}$$

On the space of compact subsets of $\mathbb{R}^S$, given by $\mathcal{K}(\mathbb{R}^S)$, the distance between value vector sets extends (7.17) and is given by the Hausdorff distance [52].

Definition 7.10 (Set-to-set Distance) *The Hausdorff distance between two value vector sets $\mathcal{V}, \mathcal{W} \subseteq \mathbb{R}^S$ is given by*

$$d_{\mathcal{K}}(\mathcal{V}, \mathcal{W}) := \max \left\{ \sup_{V \in \mathcal{V}} d(V, \mathcal{W}), \sup_{W \in \mathcal{W}} d(W, \mathcal{V}) \right\} . \tag{7.18}$$

We use $(\mathcal{K}(\mathbb{R}^S), d_{\mathcal{K}})$ to denote the metric space formed by the set of all compact subsets of $\mathbb{R}^S$ under the Hausdorff distance $d_{\mathcal{K}}$. The induced Hausdorff space is complete if and only if the original metric space is complete [52, Thm 3.3]. Therefore, $(\mathcal{K}(\mathbb{R}^S), d_{\mathcal{K}})$ is a complete metric space.

For a value operator h (7.5), we ask the following question: what is the *set* of possible value vectors when the MDP has parameter uncertainty given by $\mathcal{M}$? To resolve this, we define the set-based value operator H.

Definition 7.11 (Set-based Value Operator) *The set-valued operator H is induced by h on $\mathbb{R}^S \times \mathcal{M}$ (7.5) and is defined as*

$$H(\mathcal{V}) := \{h(V, m) \mid (V, m) \in \mathcal{V} \times \mathcal{M}\} \subseteq \mathbb{R}^S, \tag{7.19}$$

where $\mathcal{V} \subseteq \mathbb{R}^S$ is a subset of the value vector space.

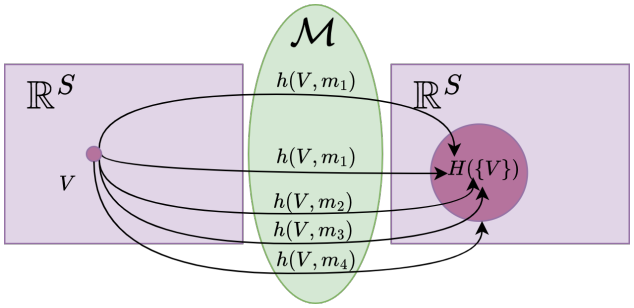


Figure 7.4: Illustration of the set-based operator $H(\mathcal{V})$ applied to the singleton set $\mathcal{V} = \{V\} \subset \mathbb{R}^S$, we compute $h(V, m)$ for every parameter $m \in \mathcal{M}$ and collect the output $h(V, m)$, such that $H(\mathcal{V}) = \cup_{m \in \mathcal{M}} h(V, m)$.

We denote the set-based value operator induced by the Bellman operator (7.9) and policy evaluation operators (7.7) as F and G^π, respectively, such that for any value vector set $\mathcal{V} \subseteq \mathbb{R}^S$,

$$F(\mathcal{V}) := \{f(V, C, P) \mid (V, C, P) \in \mathcal{V} \times \mathcal{M}\}, \tag{7.20}$$

$$G^\pi(\mathcal{V}) := \{g^\pi(V, C, P) \mid (V, C, P) \in \mathcal{V} \times \mathcal{M}\}, \ \forall\ \pi \in \Pi. \tag{7.21}$$

The set-based Bellman operator F is the union over all the *optimal* value vectors, where the optimal policy that corresponds to each $f(V, C, P) \in F(\mathcal{V})$ varies based on (C, P). On the other hand, G_π is the union over all value vectors that results from a constant π and all possible $(C, P) \in \mathcal{M}$ parameters.

We can ask the following question: is there a set of value vectors that is invariant with respect to H? Similar to the value operators h from Definition 7.4, we can affirmatively answer this question by showing that H is α-contractive on $\mathcal{K}(\mathbb{R}^S)$.

Theorem 7.1 *If h is a value operator on $\mathbb{R}^S \times \mathcal{M}$ (7.5) and $\mathcal{M}$ is compact, then the induced set value operator H (7.19) satisfies*

- *For all $\mathcal{V} \in \mathcal{K}(\mathbb{R}^S)$, $H(\mathcal{V}) \in \mathcal{K}(\mathbb{R}^S)$;*
- *H is an α-contractive on $(\mathcal{K}(\mathbb{R}^S), d_\mathcal{K})$ (7.18) with a unique fixed point set $\mathcal{V}^\star$ given by*

$$H(\mathcal{V}^\star) = \mathcal{V}^\star, \quad \mathcal{V}^\star \in \mathcal{K}(\mathbb{R}^S); \tag{7.22}$$

- *The sequence $\{\mathcal{V}^k\}_{k \in \mathbb{N}}$ where $\mathcal{V}^{k+1} = H(\mathcal{V}^k)$ converges to $\mathcal{V}^\star$ for any $\mathcal{V}^0 \in \mathcal{K}(\mathbb{R}^S)$.*

In particular, these hold for F (7.20) *and* G^π (7.21)*, whose fixed point sets are denoted as* $\mathcal{V}^B$ *and* $\mathcal{V}^\pi$*, respectively.*

$$F(\mathcal{V}^B) = \mathcal{V}^B \in \mathcal{K}(\mathbb{R}^S),\ G^\pi(\mathcal{V}^\pi) = \mathcal{V}^\pi \in \mathcal{K}(\mathbb{R}^S),\ \forall \pi \in \Pi. \tag{7.23}$$

Proof: The first statement follows from Lemma 7.1, since the image of a compact set by a continuous function is compact [115]. Let us prove the second statement: for some $\beta \in (0,1)$, for all, $\mathcal{V}, \mathcal{V}' \in \mathcal{K}(\mathbb{R}^S)$:

$$\begin{aligned} d_{\mathcal{K}}(H(\mathcal{V}), H(\mathcal{V}')) =& \max \left\{ \sup_{\substack{V\in\mathcal{V}\\ m\in\mathcal{M}}} d\left(h(V,m), H(\mathcal{V}')\right), \sup_{\substack{V'\in\mathcal{V}'\\ m'\in\mathcal{M}}} d(h(V',m'), H(\mathcal{V})) \right\} \\ \leq& \beta d_{\mathcal{K}}(\mathcal{V}, \mathcal{V}') \end{aligned}$$

Take $(V,m) \in \mathcal{V}\times\mathcal{M}$, then $d(h(V,m), H(\mathcal{V}')) \leq \inf_{V'\in\mathcal{V}'} \|h(V,m) - h(V',m)\| \leq \alpha \inf_{V'\in\mathcal{V}'} \|V - V'\|$ holds from the α-contractive property of h. Finally,

$$\sup_{\substack{V\in\mathcal{V}\\ m\in\mathcal{M}}} d(h(V,m), H(\mathcal{V}')) \leq \alpha \sup_{V\in\mathcal{V}} \inf_{V'\in\mathcal{V}'} \|V - V'\|_\infty \leq \quad \alpha d_{\mathcal{K}}(\mathcal{V}, \mathcal{V}')$$

We use the same technique to prove that

$$\sup_{\substack{V'\in\mathcal{V}'\\ m'\in\mathcal{M}}} d(h(V',m'), H(\mathcal{V})) \leq \alpha d_{\mathcal{K}}(\mathcal{V}, \mathcal{V}').$$

Finally, $d_{\mathcal{K}}(H(\mathcal{V}), H(\mathcal{V}')) \leq \alpha d_{\mathcal{K}}(\mathcal{V}, \mathcal{V}')$. From the Banach fixed point theorem and the completeness of $(\mathcal{K}(\mathbb{R}^S), d_{\mathcal{K}})$ [52, Thm 3.3], H has a unique fixed point $H^\star$ in $\mathcal{K}(\mathbb{R}^S)$.

The third point is a consequence of the Banach fixed point theorem. Finally, f and g^π are value operators (7.5) on $\mathbb{R}^S \times \mathcal{M}$, therefore this theorem's statements apply. ■

Remark 7.6 (Set-based value iteration) *An important consequence of Theorem 7.1 is the existence of the* set-based value iteration*, given by*

$$\mathcal{V}^{k+1} = H(\mathcal{V}^k),\ \mathcal{V}^0 \in \mathcal{K}(\mathbb{R}^S). \tag{7.24}$$

Analogous to standard value iteration, (7.24) *is a sequence of value vector sets in* $\mathcal{K}(\mathbb{R}^S)$ *that converges to the fixed point set* $\mathcal{V}^\star \in \mathcal{K}(\mathbb{R}^S)$*.*

7.6 Properties of the Fixed Point Set

For the MDP parameters (C, P), the fixed point of $h(\cdot, C, P)$ is typically only meaningful for the corresponding MDP. For example, the fixed point of a policy evaluation operator $g^\pi(\cdot, C, P)$ (7.7) is the expected cost-to-go under policy π, and the fixed point of the Bellman operator $f(\cdot, C, P)$ (7.9) is the minimum cost-to-go when π can be freely chosen. In this section, we derive properties of the fixed point set $\mathcal{V}$ of H (7.19) in the context of non-stationary value iteration.

7.6.1 Elements of the Fixed Point Set for the Bellman Operator

In the case of the Bellman operator f on metric space $\mathbb{R}^S$, the fixed point V^B corresponds to the optimal value vector of the MDP associated with stationary MDP parameters. However, there is no direct association of any individual MDP to the set of MDP parameters $\mathcal{M}$, we cannot claim the same for the set-based Bellman operator and $\mathcal{V}^\star$.

We consider the Bellman operator-based value iteration under a dynamic parameter uncertainty model discussed in [95], where at every iteration, a new set of MDP parameters m^k is chosen from $\mathcal{M}$ as

$$V^{k+1} = f(V^k, m^k),\ V^0 \in \mathbb{R}^S,\ m^k \in \mathcal{M}, \forall k \in \mathbb{N}. \tag{7.25}$$

In general, $\lim_{k\to\infty} f(V^k, m^k)$ may not exist. We build on the results of Theorem 7.1 that the sequence $\{V^k\}$ converges to the value function set $\mathcal{V}^B$ (7.23) in the Hausdorff distance.

Proposition 7.2 *The value vector sequence $\{V^k\}_{k\in\mathbb{N}}$ in (7.25) satisfies*

$$\lim_{k\to\infty} \inf_{V\in\mathcal{V}^B} \left\| f(V^k, m^k) - V \right\|_\infty = 0,$$

where $\mathcal{V}^B$ (7.23) is the unique fixed point set of the set-based Bellman operator F with respect to MDP parameter set $\mathcal{M}$.

Proof: Define $\mathcal{V}^0 = \{V^0\}$, then from the set-based value operator Definition 7.11 for the Bellman operator (7.20), $V^{k+1} = f(V^k, m^k) \in F(\mathcal{V}^k)$ for all $k \geq 0$. From Theorem 7.1, $\mathcal{V}^k$ converges to $\mathcal{V}^\star$ in Hausdorff distance, $\lim_{k\to\infty} d_H(\mathcal{V}^k, \mathcal{V}^B) = 0$. Therefore for every $\delta > 0$, there exists K such that for all $k \geq K$, $d_H(\mathcal{V}^k, \mathcal{V}^B) < \delta$. Since $f(V^k, m^k) \in \mathcal{V}^{k+1}$,

$\inf_{V\in\mathcal{V}^B} \left\|f(V^k,m^k)-V\right\|_\infty \leq d_H(\mathcal{V}^{k+1},\mathcal{V}^B) < \delta$ must also be true for all $k \geq K$. Therefore $\lim_{k\to\infty} \inf_{V\in\mathcal{V}^B} \left\|f(V^k,m^k)-V\right\|_\infty = 0$. ■

Proposition 7.2 implies that regardless of whether or not the sequence $\{f(V^k,m^k)\}_{k\in\mathbb{N}}$ converges, the sequence $\{V^k\}$ must become arbitrarily close in Hausdorff distance to the set $\mathcal{V}^B$. This has important interpretations in the game setting that is further explored in Section 7.9. On the other hand, Proposition 7.2 also implies that if $\lim_{k\to\infty} V^k$ does converge, its limit point must be an element of $\mathcal{V}^\star$.

Corollary 7.1 *We define the set of fixed points of $f(\cdot,m)$ for each $m\in\mathcal{M}$ as*

$$\mathcal{U} = \bigcup_{m\in\mathcal{M}} \{V\in\mathbb{R}^S \mid f(V,m)=V\},$$

i.e., $\mathcal{U}$ is the set of optimal value functions for the set of MDPs $([S],[A],P,C,\gamma)$ where $(C,P)\in\mathcal{M}$. Furthermore, we consider all sequences $\{m^k\}_{k\in\mathbb{N}}\subseteq\mathcal{M}$ such that for $V^0\in\mathbb{R}^S$, the iteration (7.25) converges to a limit point $V=\lim_{k\to\infty}V^k$, and define the set of all such limit points as

$$\mathcal{W} = \bigcup_{\{m^k\}_{k\in\mathbb{N}}\subseteq\mathcal{M}} \{V\in\mathbb{R}^S \mid \lim_{k\to\infty} f(V^k,m^k)=V, \text{ where } V^0\in\mathbb{R}^S, V^{k+1}=f(V^k,m^k),\ k=0,1,\ldots\}, \tag{7.26}$$

then $\mathcal{U}\subseteq\mathcal{W}\subseteq\mathcal{V}^B$ (7.20).

Proof: For any $V\in\mathcal{W}$ and $V^B\in\mathcal{V}^B$,

$$\left\|V^B-V\right\|_\infty \leq \left\|V^B-f(V^k,m^k)\right\|_\infty + \left\|f(V^k,m^k)-V\right\|_\infty$$

is satisfied for all $k\in\mathbb{N}$. Furthermore, by assumption, each $V\in\mathcal{W}$ has an associated iteration $V^{k+1}=f(V^k,m^k)$ whose limit point is equal to V, i.e. $\lim_{k\to\infty}\left\|f(V^k,m^k)-V\right\|_\infty = 0$. Additionally,

$$\lim_{k\to\infty} \inf_{V^B\in\mathcal{V}^B} \left\|f(V^k,m^k)-V^B\right\|_\infty = 0,$$

follows from Proposition 7.2. Therefore,

$$\inf_{V^B\in\mathcal{V}^B} \left\|V^B-V\right\|_\infty \leq 0, \quad \forall\, V\in\mathcal{W}.$$

From the fact that the infimum over a compact set is always achieved for an element of the

set [115], $V = V^B \in \mathcal{V}^B$. Therefore $\mathcal{W} \subseteq \mathcal{V}^B$. To see that $\mathcal{U} \subseteq \mathcal{W}$, take $m^k = m$ for all $k = 0, 1, \ldots$, then $\mathcal{U} \subseteq \mathcal{W}$. ■

Remark 7.7 *We make the distinction between $\mathcal{V}^B$, $\mathcal{W}$, and $\mathcal{U}$ to emphasize that $\mathcal{V}^B$ is not simply the set of fixed points corresponding to $f(\cdot, m)$ for all possible $M \in \mathcal{M}$, given by $\mathcal{U}$, or the limit points of the iteration (7.25) for all possible sequences of parameters m^k, given by $\mathcal{W}$. The fixed point set $\mathcal{V}^B$ contains all possible limiting trajectories of the non-stationary value iteration (7.25) without assuming a limit point exists.*

In Corollary 7.1, $\mathcal{U}$ can be easily understood as the set of optimal value functions for the set of standard MDPs $([S], [A], P, C, \gamma)$ generated by $m \in \mathcal{M}$. An interpretation for $\mathcal{W}$ is perhaps less obvious. We use the following example to illustrate the differences between these three sets.

Example 7.3 *Consider a single state, two action MDP with a discount factor $\gamma = 0.9$, where there is uncertainty in the cost parameter, given by $\mathcal{C} = \left\{\begin{bmatrix}0 & 1\end{bmatrix}, \begin{bmatrix}0 & 2\end{bmatrix}, \begin{bmatrix}1 & 1\end{bmatrix}\}$. Here, $\mathcal{U} = \{0, 10\}$ corresponds to the three optimal value functions when the cost is fixed —i.e., where $C^k = C \in \mathcal{C}$. We note that if $\{C^k\} \subseteq \{\begin{bmatrix}0 & 1\end{bmatrix}, \begin{bmatrix}0 & 2\end{bmatrix}\}$, then $V^B = 0$ regardless of how C^k is chosen. Therefore $\mathcal{W} = \{0\} \cup \mathcal{U} = \mathcal{U}$. Finally, if C^k is randomly chosen from $\mathcal{C}$ and $V^0 = 0$, V^k will randomly fluctuate but satisfy $V^k \in \mathcal{V}^B \subseteq [0, 10]$.*

In the context of robust MDPs, $\mathcal{U}$ contains all the fixed point value functions of regular MDPs. The value function set $\mathcal{W}$ contains the fixed point value functions that are *invariant* to fluctuating MDP parameters within any subset of $\mathcal{M}$. On the other hand, if the value functions do not converge, the value function trajectory will still converge to $\mathcal{V}^B$, even if $V^0 \notin \mathcal{V}^B$. Therefore if the goal is to bound the asymptotic behavior of V^k, it is more useful to determine $\mathcal{V}^B$.

We summarize our results on set-based Bellman operator as the following: given a compact set of MDP parameters $\mathcal{M}$, the set-based Bellman operator F converges to a unique compact set $\mathcal{V}^B$. The set $\mathcal{V}^B$ contains all the fixed points of $f(\cdot, m)$ for $m \in \mathcal{M}$. Furthermore, $\mathcal{V}^B$ also contains the limit points of $f(V^k, m^k)$ for any $\{m^k\}_{k \in \mathbb{N}} \subseteq \mathcal{M}$, $V^0 \in \mathbb{R}^S$, given that $\lim_{k \to \infty} V^k$ converges. Even if the limit does not exist, V^k must asymptotically converge to $\mathcal{V}^B$ in the Hausdorff distance.

7.6.2 Relationship to Non-stationary Value Iteration

Next, we extend the analysis on the non-stationary value iteration introduced in (7.25) to the general value operators h on $\mathbb{R}^S \times \mathcal{M}$.

$$V^{k+1} = h(V^k, m^k),\ V^0 \in \mathbb{R}^S,\ m^k \in \mathcal{M}, \forall k \in \mathbb{N}. \tag{7.27}$$

In robust MDP literature [54, 95], m^k is modified by an adversarial opponent of the MDP decision maker such that (7.27) converges to a worst-case value vector. We consider a more general scenario in which m^k is chosen from the closed and bounded set $\mathcal{M}$ without any probabilistic prior. In this scenario, convergence of V^k in $\mathbb{R}^S$ will not occur for all possible sequences of $\{m^k\}_{k\in\mathbb{N}}$. However, we can show convergence results on the set domain by leveraging our fixed point analysis of the set-based operator H (7.19).

Proposition 7.3 *Let $\mathcal{V}^\star$ be the fixed point set of the set-based value operator H (7.19) induced by h on $\mathbb{R}^S \times \mathcal{M}$ (7.5). If the non-stationary value iteration (7.27) satisfies $\{m^k\}_{k\in\mathbb{N}} \subset \mathcal{M}$, then the sequence $\{V^k\}_{k\in\mathbb{N}}$ defined by (7.27) satisfies*

1. $\lim_{k\to+\infty} d(V^k, \mathcal{V}^\star) = 0$,
2. *there exists a sub-sequence $\{V^{\varphi(k)}\}_{k\in\mathbb{N}}$ that converges to a point in $\mathcal{V}^\star$ as $\lim_{k\to\infty} V^{\varphi(k)} \in \mathcal{V}^\star$.*

Proof: Let $\{V^k\}_{k\in\mathbb{N}}$ be a sequence defined by $\mathcal{V}^0 = \{V^0\}$ and $\mathcal{V}^{k+1} = H(\mathcal{V}^k)$, where H (7.19) is the set operator induced by h on $\mathbb{R}^S \times \mathcal{M}$. We first show statement 1). From Theorem 7.1, $\lim_{k\to\infty} \mathcal{V}^k$ converges to $\mathcal{V}^\star$ in $d_\mathcal{K}$. Therefore, $0 \leq d(V^k, \mathcal{V}^\star) = \inf_{y\in\mathcal{V}^\star} \left\|V^k - y\right\|_\infty \leq \sup_{x\in\mathcal{V}^k} \inf_{y\in\mathcal{V}^\star} \|x - y\|_\infty \leq d_H(\mathcal{V}^k, \mathcal{V}^\star) \to 0$ as k tends to $+\infty$.

Next, for all $k \in \mathbb{N}$, there exists $N \in \mathbb{N}$ such that for all $n \geq N$, $d(V^n, \mathcal{V}^\star) \leq (k+1)^{-1}$. We define the strictly increasing function $\psi_1 : \mathbb{N} \to \mathbb{N}$, such that $\psi_1(0) = 0$ and for all $k \neq 0$, $\psi_1(k) := \min\{N > \psi_1(k-1) : \forall n \geq N,\ d(V^n, \mathcal{V}^\star) < (k+1)^{-1}\}$. Then, for all $k \in \mathbb{N}^\star$, there exists $y^{\psi_1(k)} \in \mathcal{V}^\star$ such that $\left\|V^{\psi_1(k)} - y^{\psi_1(k)}\right\| < (k+1)^{-1}$. As $\mathcal{V}^\star$ is compact, there exists $\psi_2 : \mathbb{N} \to \mathbb{N}$ strictly increasing such that $(y^{\psi_1(\psi_2(k))})_k$ converges to some $y^\star \in \mathcal{V}^\star$ [115, Thm 3.6]. Finally, let $\varepsilon > 0$, there exist $K_1, K_2 \in \mathbb{N}$ such that for all $l \geq K_1$, $(\psi_2(l))^{-1} < \varepsilon/2$ and for all $l' \geq K_2$, $\left\|y^{\psi_1(\psi_2(l'))} - y^\star\right\| < \varepsilon/2$. So, taking $k \geq \max\{K_1, K_2\}$, we have $\left\|V^{\psi_1(\psi_2(k))} - y^\star\right\| \leq \left\|V^{\psi_1(\psi_2(k))} - y^{\psi_1(\psi_2(k))}\right\| + \left\|y^{\psi_1(\psi_2(k))} - y^\star\right\| \leq \varepsilon$ and $(V^{\psi_1(\psi_2(k))})_k$ converges to $y^\star \in \mathcal{V}^\star$. ■

In addition to containing all asymptotic behavior of value vector trajectories under time-varying value iteration, the fixed point set $\mathcal{V}$ also contains all fixed points of the value operator $h(\cdot, C, P)$ when $(C, P) \in \mathcal{M}$ (7.5) are fixed.

Corollary 7.2 *Let h (7.5) be a value operator on $\mathbb{R}^S \times \mathcal{M}$ where $\mathcal{M}$ is compact. For all $m \in \mathcal{M}$, if $V = h(V, m) \in \mathbb{R}^S$ and $\mathcal{V}^\star$ is the fixed point set of the induced set-based value operator H (7.19), $V \in \mathcal{V}^\star$.*

Proof: We construct sequence $\{V^k\}$ where $V^{k+1} = h(V^k, m)$ and $V^0 = V$. Then $V^k = V$ for all $k \in \mathbb{N}$. From the second point of Proposition 7.3, $V \in \mathcal{V}^\star$ follows. ■

Going further, we can bound the transient behavior of (7.27) when V^0 is an element of the fixed point set $\mathcal{V}^\star$.

Corollary 7.3 (Transient behavior) *Let $\mathcal{V}^\star$ be the fixed point of the set-based value operator H (7.19) induced by h on $\mathbb{R}^S \times \mathcal{M}$. If $\mathcal{M}$ is compact and $V^0 \in \mathcal{V}^\star$, then the sequence generated by (7.27) satisfies $\{V^k\}_{k\in\mathbb{N}} \subseteq \mathcal{V}^\star$.*

Proof: As a fixed point set of H (7.19), $\mathcal{V}^\star$ (7.22) satisfies $\mathcal{V}^\star = H(\mathcal{V}^\star)$, then the following is true by definition of H: if $V^k \in \mathcal{V}^\star$, then $V^{k+1} = h(V^k, m^k) \in \mathcal{V}^\star$. If $V^0 \in \mathcal{V}^\star$, then $\{V^k\}_{k\in\mathbb{N}} \subseteq \mathcal{V}^\star$ follows by induction. ■

Remark 7.8 *Proposition 7.3 and Corollary 7.3 bound the asymptotic and transient behavior of the sequence $\{h(V^k, m^k)\}_{k\in\mathbb{N}}$ generated from (7.27), regardless of the convergence of the value vector sequence. This is a more general result than the classic convergence results for MDPs and robust MDPs.*

Remark 7.9 *Corollary 7.3 also implies that $\mathcal{V}^\star$ is invariant in the non-stationary value iteration (7.27), and may prove useful in the analysis and design of MDPs with known parameter uncertainties.*

7.6.3 Bounds of the Fixed Point Set

Given compact sets of MDP costs and parameters, a natural question is how to bound the resulting fixed point set. In Theorem 7.1, the compactness of $\mathcal{M}$ implied the compactness of $\mathcal{V}^\star$. This relationship carries over to the supremum and infimum elements of $\mathcal{M}$ and $\mathcal{V}^\star$—i.e., if $\mathcal{M}$ satisfies Assumption 7.1 with respect to h, then $\mathcal{V}^\star$ contains its own supremum and infimum elements.

Greatest and least elements. We define the supremum and infimum elements of a value vector set $\mathcal{V} \in \mathcal{K}(\mathbb{R}^S)$ element-wise as follows,

$$\overline{V}_s := \sup_{V \in \mathcal{V}} V_s, \ \underline{V}_s := \inf_{V \in \mathcal{V}} V_s, \forall\ s \in [S]. \tag{7.28}$$

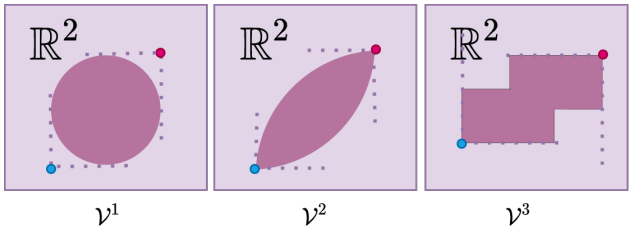


Figure 7.5: The greatest least bounds of three different value function sets $\mathcal{V}^i \in \mathbb{R}^2$, where $(0, 0)$ the origin is located on the lower left. Note that $\mathcal{V}^2$ and $\mathcal{V}^3$ contain their own greatest and least elements, but $\mathcal{V}^1$ does not. In $\mathcal{V}^1$, the coordinate-wise greatest and least elements are achieved by some elements in $\mathcal{V}^1$ but not at the same time.

If a set $\mathcal{V} \subseteq \mathbb{R}^S$ is compact, the projection of $\mathcal{V}$ on each state s is compact. Then, the coordinate-wise supremum and infimum values for each state s are achieved by $\mathcal{V}$. However, in general, no single element of the set $\mathcal{V}$ may simultaneously achieve the minimum over all the states—i.e., $\overline{V}(\underline{V})$ may not be an element of $\mathcal{V}$. This is illustrated in Figure 7.5.

Given h and parameter uncertainty set $\mathcal{M}$, we wish to 1) bound the supremum and infimum elements of the fixed point set $\mathcal{V}^\star$ (7.22) and 2) derive sufficient conditions for when they are elements of $\mathcal{V}^\star$. To facilitate bounding $\mathcal{V}^\star$, we introduce the following bound operators.

Definition 7.12 (Bound Operators) *The bound operators induced by the value operator h on $\mathbb{R}^S \times \mathcal{M}$ are coordinate-wise defined at each $s \in [S]$ as*

$$\underline{h}_s(V) = \inf_{m \in \mathcal{M}} h_s(V, m), \ \overline{h}_s(V) = \sup_{m \in \mathcal{M}} h_s(V, m). \tag{7.29}$$

We want to bound the fixed point set $\mathcal{V}$ of the set-based value operator H (7.19) by the bound operators $\underline{h}/\overline{h}$ (7.29). First, we show that $\underline{h}/\overline{h}$ are themselves α-contractive and order preserving on $\mathbb{R}^S$.

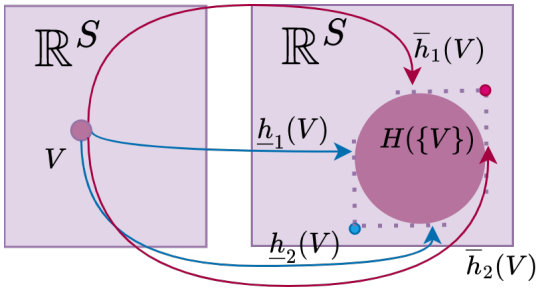


Figure 7.6: We visualize the bound operator for $H(\mathcal{V})$ for a given value operator h on $\mathbb{R}^S \times \mathcal{M}$. The input set $\mathcal{V}$ is a singleton $\{V\}$ in $\mathbb{R}^2$. Here, because $\underline{h}_1$ and $\underline{h}_2$are reached for two different parameters $m \in \mathcal{M}$, the resulting $\underline{h}(V)$ lies outside of the fixed point set.

Lemma 7.3 (α-Contraction) *If h (7.5) is a value operator on $\mathbb{R}^S \times \mathcal{M}$ and $\mathcal{M}$ is compact, then $\underline{h}$ and $\overline{h}$ (7.29) are α-contractions with fixed points $\underline{X}, \overline{X}$, respectively.*

$$\overline{h}(\overline{X}) = \overline{X}, \quad \underline{h}(\underline{X}) = \underline{X}, \ \underline{X}, \overline{X} \in \mathbb{R}^S. \tag{7.30}$$

Proof: From Lemma 7.1, h is continuous and $\mathcal{M}$ is compact, then for all $X, Y \in \mathbb{R}^S$, there exists $\hat{m}(s) \in \mathcal{M}$ such that $\underline{h}_s(Y) = h_s(Y, \hat{m}(s))$ and $\underline{h}_s(X) \leq h_s(X, \hat{m}(s))$. We upper-bound $\underline{h}_s(X) - \underline{h}_s(Y)$ by $h_s(X, \hat{m}(s)) - h_s(Y, \hat{m}(s))$, and use the α-contraction property of h to derive

$$\begin{aligned} \underline{h}_s(X) - \underline{h}_s(Y) &\leq |h_s(X, \hat{m}(s)) - h_s(Y, \hat{m}(s))| \\ &\leq \alpha \left\| X - Y \right\|_\infty . \end{aligned}$$

Since X and Y are arbitrarily ordered, we conclude that $\left\| \underline{h}(X) - \underline{h}(Y) \right\|_\infty \leq \alpha \left\| X - Y \right\|_\infty$. The proof for $\overline{h}$ follows a similar reasoning and takes $\hat{m}(s) = \sup_{m \in \mathcal{M}} h_s(X, m)$. The existence of $\underline{X}(\overline{X})$ follows from applying Banach's fixed point theorem. ■

Lemma 7.4 (Order Preservation) *The bound operators $\underline{h}$ and $\overline{h}$ (7.29) are order-preserving on $\mathbb{R}^S$ (Definition 7.2).*

$$\forall \ U, V \in \mathbb{R}^S, \ U \leq V \Rightarrow \underline{h}(U) \leq \underline{h}(V), \quad \overline{h}(U) \leq \overline{h}(V).$$

Proof: The lemma statement follows directly from the fact that order preservation is conserved through composition with inf and sup. If $h(U, m) \leq h(V, m)$, then

$\inf_{m\in\mathcal{M}} h(U,m) \leq \inf_{m\in\mathcal{M}} h(V,m)$. A similar argument follows for $\overline{h}(\cdot) = \sup_{m\in\mathcal{M}} h(\cdot,m)$. ■

To prove the next result, we use the following lemma.

Lemma 7.5 *Let $\{\mathcal{V}_n\} \subseteq \mathcal{K}(\mathbb{R}^S)$ be a converging sequence for $d_{\mathcal{K}}$ with $\mathcal{V}_n \to \mathcal{V}$ as $n \to \infty$. For all $V \in \mathcal{V}$, there exists a converging subsequence $\{V^{\varphi(n)}\}_{n\in\mathbb{N}}$ whose limit is V for $\|\cdot\|$.*

Proof: Let $V \in \mathcal{V}$. We can define the strictly increasing function φ on $\mathbb{N}$ as follows: $\varphi(0) := 0$ and for all $n \in \mathbb{N}$, $\varphi(n+1) := \min\{j > \varphi(n) \mid \exists\, V^j \in \mathcal{V}^j, \|V - V^j\| = d(V,\mathcal{V}^j) \leq (n+1)^{-1}\}$. Finally, as for all $n \in \mathbb{N}^*$, $\left\|V - V^{\varphi(n)}\right\| \leq (\varphi(n)+1)^{-1}$, the result holds. ■

We show that the fixed points $\underline{X}$ and $\overline{X}$ bounds the fixed point set $\mathcal{V}^\star$ of the set-based value operator H (7.19).

Theorem 7.2 (Bounding fixed point sets) *If h (7.5) is a value operator on $\mathbb{R}^S \times \mathcal{M}$ and $\mathcal{M}$ is compact,*

$$\underline{X} \leq V \leq \overline{X},\ \forall\, V \in \mathcal{V}^\star, \tag{7.31}$$

where $\underline{X}$ and $\overline{X}$ (7.30) are the fixed points of the bound operators $\underline{h}$ and $\overline{h}$ (7.29), and $\mathcal{V}^\star$ is the fixed point set of the set-based value operator H (7.19) induced by h (7.5) on $\mathbb{R}^S \times \mathcal{M}$.

Proof: For $\mathcal{V}^0 = \{\underline{X}, \overline{X}\}$ and $\mathcal{V}^{k+1} = H(\mathcal{V}^k)$ (7.24), we first show

$$\underline{X} \leq V \leq \overline{X},\ \forall\, V \in \mathcal{V}^k, \tag{7.32}$$

via induction. Suppose that (7.32) is satisfied for $\mathcal{V}^k$. The order preserving property of $h(\cdot,m)$ implies that $h(\underline{X},m) \leq h(V,m) \leq h(\overline{X},m)$ holds for all $(V,m) \in \mathcal{V}^k \times \mathcal{M}$. We take the infimum and supremum over $h(\underline{X},m)$ and $h(\overline{X},m)$, respectively, to show that for all $(V,m) \in \mathcal{V}^k \times \mathcal{M}$ and $s \in [S]$,

$$\inf_{m'\in\mathcal{M}} h_s(\underline{X},m') \leq h_s(V,m) \leq \sup_{m'\in\mathcal{M}} h_s(\overline{X},m').$$

Since $\underline{X}$ and $\overline{X}$ are the fixed points of $\inf_{m'\in\mathcal{M}} h_s(\cdot,m')$ and $\sup_{m'\in\mathcal{M}} h_s(\cdot,m')$ for all $s \in [S]$, respectively, we conclude that (7.32) holds for $\mathcal{V}^{k+1}$.

Next, we show that $\underline{X}$ and $\overline{X}$ bounds the fixed point set $\mathcal{V}^\star$ for the h-induced operator H (7.19). From Lemma 7.5, we know that for all $V \in \mathcal{V}^\star$, there exists a strictly increasing sequence $\phi : \mathbb{N} \mapsto \mathbb{N}$ and corresponding value vectors $\{W^{\phi(n)}\}$ such that $\lim_{n\to\infty} W^{\phi(n)} = V$

and $W^{\phi(n)} \in \mathcal{V}^{\phi(n)}$ for the sequence of value vector sets generated from $\mathcal{V}^0 = \{\underline{X}, \overline{X}\}$. Since $\underline{X} \leq W^{\phi(n)} \leq \overline{X}$ holds for all n, we conclude (7.31) holds. ■

When Assumption 7.1 is satisfied, the fixed point of H (7.19) contains its supremum and infimum.

Theorem 7.3 *If h (7.5) on $\mathbb{R}^S \times \mathcal{M}$ satisfies Assumption 7.1, then there exists $\underline{m}, \overline{m} \in \mathcal{M}$ such that $\underline{h}$ and $\underline{h}$ (7.29) and their fixed points $\underline{X}$ and $\overline{X}$ (7.30) satisfies*

$$\underline{h}(\underline{X}) = h(\underline{X}, \underline{m}) = \underline{X}, \ \overline{h}(\overline{X}) = h(\overline{X}, \overline{m}) = \overline{X}. \tag{7.33}$$

Additionally, $\underline{X}$ and $\overline{X}$ are the least and the greatest elements of H's fixed point set $\mathcal{V}^\star$, $\underline{V}^\star, \overline{V}^\star$ (7.28) respectively, and both belong to $\mathcal{V}^\star$ (7.22).

$$\underline{X} = \underline{V}^\star, \ \overline{X} = \overline{V}^\star, \ \underline{X}, \overline{X} \in \mathcal{V}^\star.$$

Proof: From Theorem 7.2, $\underline{X}$ and $\overline{X}$ are the lower and upper bounds on the fixed point set $\mathcal{V}^\star$. We show that these are the infimum and supremum elements of $\mathcal{V}^\star$ by showing that they are also elements of $\mathcal{V}^\star$. From Assumption 7.1, there exists $\underline{m}, \overline{m} \in \mathcal{M}$ such that $h_s(\underline{X}, \underline{m}) = \min_{m \in \mathcal{M}} h_s(\underline{X}, \underline{m})$ and $h_s(\overline{X}, \overline{m}) = \min_{m \in \mathcal{M}} h_s(\overline{X}, \overline{m})$ for all $s \in [S]$. Since $\underline{X}$ and $\overline{X}$ are fixed points of $h(\cdot, \underline{m})$ and $h(\cdot, \overline{m})$, we apply Corollary 7.2 to conclude that $\underline{X}, \overline{X} \in \mathcal{V}^\star$. ■

7.7 Revisiting Robust MDP

We re-examine robust MDP with the set-theoretical analysis in this section, and show that Assumption 7.1 generalizes the rectangularity assumption made in robust MDPs, thus enabling robust dynamic programming techniques to be available to a wider class of MDP problems and contraction operators.

Recall the optimistic value vector $W^o \in \mathbb{R}^S$ and robust value vectors $W^r \in \mathbb{R}^S$ of a discounted MDP $([S], [A], C, P, \gamma)$ from [54, 95] as the fixed points of the following operators.

$$W^o_s = \min_{\pi_s \in \Delta_A} \min_{(C,P) \in \mathcal{M}} g^\pi_s(W^o, C, P), \forall s \in [S] \tag{7.34}$$

$$W^r_s = \min_{\pi_s \in \Delta_A} \max_{(C,P) \in \mathcal{M}} g^\pi(W^r, C, P), \forall s \in [S] \tag{7.35}$$

The optimistic policy π^o and robust policy π^r are the optimal policies corresponding to (7.34)

and (7.35), respectively.

$$\pi_s^o \in \underset{\pi_s \in \Delta_A}{\operatorname{argmin}} \min_{(C,P)\in\mathcal{M}} g_s^\pi(W^o, C, P), \forall s \in [S] \tag{7.36}$$

$$\pi_s^r \in \underset{\pi_s \in \Delta_A}{\operatorname{argmin}} \max_{(C,P)\in\mathcal{M}} g_s^\pi(W^r, C, P), \forall s \in [S] \tag{7.37}$$

For readability, we denote the policy evaluation operator under π^o as g^o and the policy evaluation operator under π^r as g^r.

When $\mathcal{M}$ is (s,a)-rectangular (7.15), the set of policies satisfying (7.36) and (7.37) is non-empty and includes deterministic policies [54, Thm 3.1]. When $\mathcal{M}$ is s-rectangular and convex, the set of policies satisfying (7.37) is non-empty but may be mixed [139, Thm 4]. When $\mathcal{M}$ is convex, we show that policies (7.36) and (7.37) exist.

Proposition 7.4 *If the MDP parameter set $\mathcal{M}$ is compact and convex, then*

1. *W^o (7.34) and W^r (7.35) exist and satisfy $\overline{f}(W^r) = W^r$, $\underline{f}(W^o) = W^o$, where $\overline{f}$ and $\underline{f}$ (7.29) are the bound operators of the Bellman operator (7.9).*
2. *π^o (7.36) and π^r (7.37) exist.*

Proof: Recall the Bellman operator f (7.5). When $\mathcal{M} \times \Delta_A$ is compact, the formulation of the fixed point of $\underline{f}$ (7.29) is equivalently given by

$$\underline{f}(\underline{X}) = \min_{(C,P)\in\mathcal{M}} \min_{\pi_s \in \Delta_A} g_s^\pi(\underline{X}, C, P), \ \forall s \in [S]. \tag{7.38}$$

We note that (7.38) is identical to the formulation of W^o (7.34). Therefore, $W^o = \underline{X}$ is the fixed point of $\underline{f}$. When $\mathcal{M}$ is compact, W^o exists due to Lemma 7.3. From (7.36), π_s^o is the optimal argument of $g_s^\pi(W^o, C, P)$, a continuous function in π_s, C, P minimized over compact sets $\Delta_A \times \mathcal{M}$ for all $s \in [S]$. Therefore π_s^o exists. Since $\pi^o = (\pi_1^o, \ldots, \pi_S^o)$, the optimal $\pi^o \in \Pi$ exists.

For the robust scenario: when $\mathcal{M}$ is compact, the fixed point of $\overline{f}$ (7.29), $\overline{X}$, exists from Lemma 7.3 and is given by

$$\overline{X}_s = \max_{(C,P)\in\mathcal{M}} \min_{\pi_s \in \Delta_A} g_s^\pi(\overline{X}, C, P), \ \forall s \in [S]. \tag{7.39}$$

The function $g_s^\pi(\overline{X}, C, P)$ is concave in (C, P) and convex in π. If $\mathcal{M}$ is convex, then we

apply the minimax theorem [93] to switch the order of min and max in (7.39) to derive

$$\overline{X}_s = \min_{\pi_s \in \Delta_A} \max_{(C,P)\in\mathcal{M}} g_s^{\pi}(\overline{X}, C, P), \ \forall s \in [S]. \tag{7.40}$$

Equation (7.40) is identical to (7.35), therefore $W^r = \overline{X}$ and exists by Lemma 7.3. In the definition of $\overline{X}$ (7.40), $\max_{(C,P)\in\mathcal{M}} g_s^{\pi}(\overline{X}, C, P)$ is piece-wise linear in π_s and Δ_A is compact for all $s \in [S]$, thus $\operatorname{argmin}_{\pi_s\in\Delta_A} \max_{(C,P)\in\mathcal{M}} g_s^{\pi}(\overline{X}, C, P)$ is non-empty. Finally since $\pi^r = (\pi_1^r, \ldots, \pi_S^r)$, π^r exists. ■

Remark 7.10 *Since* $\max_{(C,P)\in\mathcal{M}} g_s^{\pi}(\overline{X}, C, P)$ *is piecewise linear in* π_s*, the optimal* π_s^r *is a mixed policy in general. This is consistent with the results in [139].*

Proposition 7.4 generalizes the results from [139] to show that (7.35) exists when $\mathcal{M}$ is compact and convex instead of s-rectangular and convex. From Theorem 7.2, W^o and W^r bound of the fixed point sets of the π^o and π^r. They become infimum and supremum elements when $\mathcal{M}$ satisfies Assumption 7.1 with respect to g^o and g^r. We explicitly derive this result next. First, we introduce some notations: let $G^o = G^{\pi_o}$, the fixed point of G^o be $\mathcal{V}^o$, $G^r = G^{\pi^r}$, and the fixed point of G^r be $\mathcal{V}^r$.

$$\mathcal{V}^o = \{g^o(V, C, P) \mid (C, P) \in \mathcal{M}, V \in \mathcal{V}^o\}, \tag{7.41}$$

$$\mathcal{V}^r = \{g^r(V, C, P) \mid (C, P) \in \mathcal{M}, V \in \mathcal{V}^r\}. \tag{7.42}$$

Additionally, the supremum elements of $\mathcal{V}^o$ and $\mathcal{V}^r$ are $\overline{V}^o$ and $\overline{V}^r$ respectively and the infimum elements are $\underline{V}^o$ and $\underline{V}^r$, respectively.

$$\underline{V}_s^r = \min_{V\in\mathcal{V}^r} V_s, \ \overline{V}_s^r = \max_{V\in\mathcal{V}_r} V_s, \ \forall s \in [S]. \tag{7.43}$$

$$\underline{V}_s^o = \min_{V\in\mathcal{V}^o} V_s, \ \overline{V}_s^o = \max_{V\in\mathcal{V}^o} V_s, \ \forall s \in [S]. \tag{7.44}$$

We compare these with the fixed point set of the Bellman operator, $\mathcal{V}^B = \{\min_\pi g^\pi(V, C, P) \mid (C, P) \in \mathcal{M}, V \in \mathcal{V}^B\}$ (7.22), denoted by $\overline{V}^B$ and $\underline{V}^B$ as

$$\underline{V}_s^B = \min_{V\in\mathcal{V}^B} [V]_s, \ \overline{V}_s^B = \max_{V\in\mathcal{V}^B} V_s, \ \forall s \in [S]. \tag{7.45}$$

Our next result proves the relationship between $\underline{V}^B, \underline{V}^o, \underline{V}^r, \overline{V}^B, \overline{V}^o, \overline{V}^r$ when f, g^o, and g^r

on $\mathbb{R}^S \times \mathcal{M}$ satisfy Assumption 7.1.

Theorem 7.4 *If f, g^o, g^r satisfy Assumption 7.1 on $\mathbb{R}^S \times \mathcal{M}$, then the bounding elements (7.44) (7.43) (7.45) of the corresponding fixed point sets $\mathcal{V}^B$,$\mathcal{V}^o$ (7.41) and $\mathcal{V}^r$ (7.42) are ordered as*

$$\underline{V}^B = \underline{V}^o \leq \underline{V}^r, \ \overline{V}^B = \overline{V}^r \leq \overline{V}^o. \tag{7.46}$$

Proof: Since $\underline{V}^o$ is the infimum element for the fixed point set $\mathcal{V}^o$ (7.44), we can apply Theorem 7.3 to derive

$$\underline{V}^o = \min_{(C,P)\in\mathcal{M}} g^o(\underline{V}^o, C, P). \tag{7.47}$$

By definition of π_o (7.36), $\min_{(C,P)\in\mathcal{M}} g^o(\underline{V}^o, C, P) = \min_{(C,P)\in\mathcal{M}} \min_{\pi\in\Pi} g^\pi(\underline{V}^o, C, P)$. As the two minima commute,

$$\min_{(C,P)\in\mathcal{M}} g^o(\underline{V}^o, C, P) = \min_{(C,P)\in\mathcal{M}} \min_{\pi\in\Pi} g^\pi(\underline{V}^o, C, P). \tag{7.48}$$

Combining (7.47) and (7.48), $\underline{V}^o$ is exactly the unique fixed point of $\min_{(C,P)\in\mathcal{M}} \min_{\pi\in\Pi} g^\pi(\cdot, C, P)$. However, by applying Theorem 7.3 to f on $\mathbb{R}^S \times \mathcal{M}$, $\underline{V}^B$ is also the unique fixed point of $\min_{(C,P)\in\mathcal{M}} \min_{\pi\in\Pi} g^\pi(\cdot, C, P)$. Therefore $\underline{V}^o = \underline{V}^B$.

From (7.43), $\underline{V}^r = \min_{(C,P)\in\mathcal{M}} g^r(\underline{V}^r, C, P)$, we can minimize over the policy space to lower bound $\underline{V}^r$ as

$$\underline{V}^r \geq \min_{\pi\in\Pi} \min_{(C,P)\in\mathcal{M}} g^r(\underline{V}^r, C, P). \tag{7.49}$$

Since the right hand side of (7.49) is equivalent to $\underline{f}(\underline{V}^r)$, (7.49) is equivalent to $\underline{V}^r \geq \underline{f}(\underline{V}^r)$. From Lemma 7.4, $\underline{f}$ is order-preserving in V, we conclude that $\underline{V}^o = \underline{V}^\star \leq \underline{V}^r$.

From Theorem 7.3, $\overline{V}^r$ is the fixed point of $\overline{g^r}$, such that

$$\overline{V}^r = \max_{(C,P)\in\mathcal{M}} g^r(\overline{V}^r, C, P). \tag{7.50}$$

We apply $\min_\pi$ to both sides of (7.50) and use the definition of π_r to derive that $\overline{V}^r$ is the fixed point of $\min_{\pi\in\Pi} \max_{(C,P)\in\mathcal{M}} g^\pi(V^r, C, P)$. From Assumption 7.1, there exists $(\overline{C}, \overline{P}) \in \mathcal{M}$ that maximizes $g^\pi(\overline{V}, C, P)$, so $\overline{V}^r$ equivalently satisfies

$$\overline{V}^r = \min_{\pi\in\Pi} g^\pi(\overline{V}^r, \overline{C}, \overline{P}).$$

From Corollary 7.2, this implies that $\overline{V}^r \in \mathcal{V}^B$ and therefore $\overline{V}^r \leq \overline{V}^B$. Next we show $\overline{V}^B \leq$

$\overline{V}^r$. From Theorem 7.3, $\overline{V}^B$ is the fixed point of $\overline{f}$, such that $\overline{V}^B = \max_{(C,P)\in\mathcal{M}} \min_\pi g^\pi(\overline{V}^B, C, P)$, From the min-max inequality,

$$\overline{V}^B \leq \min_{\pi\in\Pi} \max_{(C,P)\in\mathcal{M}} g^\pi(\overline{V}^B, C, P).$$

Since $\pi_r \in \Pi$,

$$\overline{V}^B \leq \max_{(C,P)\in\mathcal{M}} g^r(\overline{V}^B, C, P). \tag{7.51}$$

The right-hand side of (7.51) is $\overline{g}^r(\overline{V}^B)$ (7.29), such that (7.51) is equivalent to $\overline{V}^B \leq \overline{g}^r(\overline{V}^B)$. We consider the sequence $V^{k+1} = \overline{g}^r(V^k)$ where $V^1 = \overline{V}^B$. Since $\overline{g}^r$ is a contraction, $\lim_{k\to\infty} V^k = V^r$, the fixed point of $\overline{g}^r$. From Lemma 7.4, $\overline{g}^r$ is order preserving. Therefore $\overline{V}^B = V^1 \leq V^r$.

Finally, Theorem 7.3 implies that $\overline{V}^o$ is the fixed point of $\overline{g}^o$: $\overline{V}^o = \max_{(C,P)\in\mathcal{M}} g^o(\overline{V}^o, C, P)$. By construction, $\overline{V}^o \geq \min_{\pi\in\Pi} \max_{(C,P)\in\mathcal{M}} g^\pi(\overline{V}^o, C, P)$. From the min-max inequality,

$$\min_{\pi\in\Pi} \max_{(C,P)\in\mathcal{M}} g^\pi(\overline{V}^o, C, P) \geq \max_{(C,P)\in\mathcal{M}} \min_{\pi\in\Pi} g^\pi(\overline{V}^o, C, P),$$

such that the right-hand side of the inequality is equivalent to $\overline{f}(\overline{V}^o)$. Following the monotonicity properties of the Bellman operator f [107, Thm.6.2.2], we conclude that $\overline{V}^o \geq \overline{V}^B$. ■

Remark 7.11 *Through our fixed-point analysis, we see that in addition to having the best worst-case performance among $\{\mathcal{V}^o, \mathcal{V}^B, \mathcal{V}^r\}$, $\mathcal{V}^r$ also has the smallest variation in performance for the same uncertainty set $\mathcal{M}$.*

Finally, we generalize the s-rectangularity condition by showing that optimistic and robust policies exist when the MDP parameter set $\mathcal{M}$ satisfies Assumption 7.1.

Corollary 7.4 (Robust MDP under Assumption 7.1) *If $\mathcal{M}$ is compact and convex, and f, g^o, g^r satisfy Assumption 7.1 on $\mathbb{R}^S \times \mathcal{M}$, then W^o (7.34) and W^r (7.35) are the infimum and supremum value vectors for the policy evaluation operator under π^o (7.36) and π^r (7.37), respectively.*

$$W_s^o = \inf_{V\in\mathcal{V}^o} [V]_s, W_s^r = \sup_{V\in\mathcal{V}^r} [V]_s, \ \forall s \in [S], \tag{7.52}$$

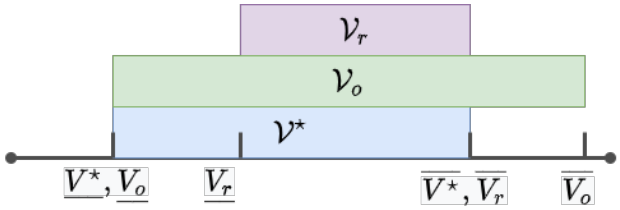


Figure 7.7: Illustration of Theorem 7.4. The purple, green, and blue regions indicate the ranges of $\mathcal{V}^r$, $\mathcal{V}^o$, and $\mathcal{V}^B$, respectively.

where $\mathcal{V}^o$ (7.41) *and* $\mathcal{V}^r$ (7.42) *are the fixed point sets of policies* π^o *and* π^r *under parameter uncertainty* $\mathcal{M}$*, respectively.*

Proof: When f satisfies Assumption 7.1 on $\mathbb{R}^S \times \mathcal{M}$, Theorem 7.3 shows that $\underline{V}^B = W^o$, $\overline{V}^B = W^r$. If f, g^o, and g^r also satisfy Assumption 7.1 on $\mathbb{R}^S \times \mathcal{M}$, then we apply Theorem 7.4 to derive $W^o = \underline{V}^o$ and $W^r = \overline{V}^r$. This proves the corollary statement. ■

Remark 7.12 *When Assumption 7.1 is not satisfied,* W^o *and* W^r *still provide a bound for* $\underline{V}^o$ *and* $\overline{V}^r$*. This result is also stated in [139].*

7.8 Value Iteration for Fixed Point Set Computation

In the previous sections, we proved the existence of a fixed point set for value operators with compact parameter uncertainty sets and re-interpreted robust control through our techniques. Next, we derive an iterative algorithm for computing the bounds of the fixed point set $\mathcal{V}$ given a value operator h and parameter uncertainty set $\mathcal{M}$.

Algorithm Sketch. Based on the set-based value iteration (7.24), we iteratively find the one-step bounds of $H(\mathcal{V}^k)$ to converge the bounds of the fixed point set.

For any compact set $\mathcal{V} \in \mathcal{K}(\mathbb{R}^S)$, the one step bounds of $H(\mathcal{V})$ are equivalent to the one-step output of the bound operators $\underline{h}$ and $\overline{h}$ (7.29) applied to the extremal points of $\mathcal{V}$.

Theorem 7.5 (One step H bounds) *Consider a set operator* H (7.19) *and its bound operators* $\underline{h}$ *and* $\overline{h}$ (7.29) *induced by* h *on* $\mathbb{R}^S \times \mathcal{M}$ (7.5)*. For a compact set* $\mathcal{V} \subset \mathbb{R}^S$*,* $H(\mathcal{V})$ *is bounded by* $\underline{h}(\underline{V})$ *and* $\overline{h}(\overline{V})$ (7.29) *as*

$$\underline{h}(\underline{V}) \leq V \leq \overline{h}(\overline{V}), \quad \forall\ V\ \in H(\mathcal{V}). \tag{7.53}$$

where $\underline{V}$ *and* $\overline{V}$ (7.28) *are the extremal elements of* $\mathcal{V}$. *If* h *satisfies Assumption 7.1 on* $\mathbb{R}^S \times \mathcal{M}$ *and* $\underline{V}, \overline{V} \in \mathcal{V}$, *then* $\underline{h}(\underline{V})$ *and* $\overline{h}(\overline{V})$ *are the supremum and infimum elements of* $H(\mathcal{V})$, *respectively— for all* $s \in [S]$, $\underline{h}_s(\underline{V})$ *and* $\overline{h}_s(\overline{V})$ *satisfy*

$$\underline{h}_s(\underline{V}) = \inf_{(V,m)\in\mathcal{V}\times\mathcal{M}} h_s(V,m),\ \overline{h}_s(\overline{V}) = \sup_{(V,m)\in\mathcal{V}\times\mathcal{M}} h_s(V,m). \tag{7.54}$$

Proof: For all $s \in [S]$, $h_s(V,m) \leq \overline{h}_s(V)$ for all $m \in \mathcal{M}$. If h is $K(V)$-Lipschitz and α-contractions in $\mathcal{M}$, then $\overline{h}$ is order-preserving (Lemma 7.4) such that $\overline{h}_s(V) \leq \overline{h}_s(\overline{V})$ for all $V \in \mathcal{V}$. We conclude that

$$h(V,m) \leq \overline{h}(\overline{V}),\ \forall (V,m) \in \mathcal{V} \times \mathcal{M}. \tag{7.55}$$

Since $\overline{h}$ is an upper bound, and sup is the least upper bound, it holds that $\sup_{V,m}[h(V,m)]_s \leq \overline{h}(\overline{V})$. We use the definition of $H(\mathcal{V})$ (7.19) to conclude that $V \leq \overline{h}(\overline{V})$ for all $V \in H(\mathcal{V})$. The inequality $\underline{h}(\underline{V}) \leq V\ \forall V \in H(\mathcal{V})$ can be similarly proved.

If h satisfies Assumption 7.1 on $\mathbb{R}^S \times \mathcal{M}$ and $\underline{V}, \overline{V} \in \mathcal{V}$, Assumption 7.1 states that there exists $\underline{m} \in \mathcal{M}$ such that $h(\underline{V}, \underline{m}) = \underline{h}(\underline{V})$. Therefore, $\underline{h}(\underline{V}) \in H(\mathcal{V})$. Since $\underline{h}(\underline{V})$ also lower bounds all the elements of $H(\mathcal{V})$, it is the infimum element of $H(\mathcal{V})$. The fact that the greatest element of $H(\mathcal{V})$ is $\overline{h}(\overline{V})$ can be similarly proved. ■

Based on Theorem 7.5, we propose the following bound approximation algorithm of the fixed point set $\mathcal{V}^\star$ (7.22) for a set-valued operator H (7.5).

Algorithm 7.1 Bound approximation of the fixed point set $\mathcal{V}$

Input: $\mathcal{C}$, $\mathcal{P}$, V^0, ϵ.
Output: $\underline{V}$, $\overline{V}$

1: $\underline{V}^0 := \overline{V}^0 := V^0$
2: $e^0 = \frac{1-\gamma}{\gamma}\epsilon$
3: **while** $\frac{\gamma}{1-\gamma}e^k \geq \epsilon$ **do**
4: $\quad \underline{V}_s^{k+1} = \min_{m\in\mathcal{M}} h_s(\underline{V}^k, m), \quad \forall s \in [S]$
5: $\quad \overline{V}_s^{k+1} = \max_{m\in\mathcal{M}} h_s(\overline{V}^k, m), \quad \forall s \in [S]$
6: $\quad e^{k+1} = \max\left\{ \left\|\underline{V}^{k+1} - \underline{V}^k\right\|, \left\|\overline{V}^{k+1} - \overline{V}^k\right\| \right\}$
7: $\quad k = k+1$
8: **end while**

7.8.1 Computing One-step Optimal Parameters

Algorithm 7.1 is stated for a general MDP parameter set $\mathcal{M}$ and does not specify how to compute lines 4 and 5. Here we discuss solution methods for different shapes of $\mathcal{M}$.

1. **Finite $\mathcal{M}$.** If $\mathcal{M} = \{m_1, \ldots, m_N\}$ is a set with a finite number of elements, we can directly compute line 4 as

$$\underline{V}^{k+1} = \min \Big\{ h_s(\underline{V}^k, m_i) \mid i = \{1, \ldots, N\} \Big\}. \tag{7.56}$$

   For line 5, we replace min with max in (7.56).

2. **Convex $\mathcal{M}$.** When $\mathcal{M}$ is a convex set, the computation depends on h. If $h = g^\pi$ is the policy operator, lines 4 and 5 can be solved as convex optimization problems. If h is the Bellman operator f, lines 4 and 5 take on min-max formulation and is NP-hard to solve in the general form [139]. When $\mathcal{M}$ can be characterized by an ellipsoidal set of parameters, the solutions to lines 4 and 5 is given in [139].

We recall the stochastic path planning problem from Example 7.1 with the two different parameter uncertainty scenarios. When the wind field uncertainty is discrete, $\mathcal{M}$ is finite, when the wind field is a combination of the major wind trends, $\mathcal{M}$ is convex.

7.8.2 Algorithm Convergence Rate

When lines 4 and 5 are solvable, Algorithm 7.1 asymptotically converges to *approximations* of the bounding elements of $\mathcal{V}^\star$. If $\mathcal{M}$ satisfies Assumption 7.1 with respect to h, Algorithm 7.1 derives the exact bounds of $\mathcal{V}$. Algorithm 7.1 has similar rates of convergence in Hausdorff distance as standard value iteration using h on $\mathbb{R}^S$.

Theorem 7.6 *Consider the value operator h, compact uncertainty set $\mathcal{M}$, and the fixed point set $\mathcal{V}^\star$ of the set-based operator H (7.19) induced by h on $\mathbb{R}^S \times \mathcal{M}$. If $\mathcal{M}$ satisfies Assumption 7.1 with respect to h, then at each iteration k,*

$$\left\| \underline{V}^{k+1} - \underline{V}^\star \right\| \leq \alpha \left\| \underline{V}^k - \underline{V}^\star \right\|, \left\| \overline{V}^{k+1} - \overline{V} \right\| \leq \alpha \left\| \overline{V}^k - \overline{V}^\star \right\|, \tag{7.57}$$

where all norms are infinity norms, and $\underline{V}^\star, \overline{V}^\star$ are the infimum and supremum bounds of $\mathcal{V}$,

respectively. At Algorithm 7.1's termination, $\underline{V}^k, \overline{V}^k$ *satisfies*

$$\max\{\left\|\underline{V}^k - \underline{V}^\star\right\|, \left\|\overline{V}^k - \overline{V}^\star\right\|\} < \epsilon. \tag{7.58}$$

Proof: From Algorithm 7.1, $\overline{V}^{k+1} = \overline{h}(\overline{V}^k)$. From Lemma 7.3, $\overline{h}$ is an α-contraction. We obtain $\left\|\overline{V}^{k+1} - \overline{V}^\star\right\| \leq \alpha \left\|\overline{V}^k - \overline{V}^\star\right\|$ and note that (7.57) holds by induction. Next, we apply triangle inequality to $\left\|\overline{V}^k - \overline{V}^\star\right\|$ to derive

$$\left\|\overline{V}^k - \overline{V}^\star\right\| \leq \left\|\overline{V}^k - \overline{V}^{k+1}\right\| + \left\|\overline{V}^{k+1} - \overline{V}^\star\right\|. \tag{7.59}$$

We can then use $\left\|\overline{V}^{k+1} - \overline{V}^\star\right\| \leq \alpha \left\|\overline{V}^k - \overline{V}^\star\right\|$ to bound (7.59) as $\left\|\overline{V}^k - \overline{V}^\star\right\| \leq \frac{1}{1-\alpha} \left\|\overline{V}^k - \overline{V}^{k+1}\right\|$. A similar argument can show that $\left\|\underline{V}^k - \underline{V}^\star\right\| \leq \frac{1}{1-\alpha} \left\|\underline{V}^k - \underline{V}^{k+1}\right\|$. When Algorithm 7.1's while condition is satisfied, $\max\left\{\left\|\overline{V}^k - \overline{V}^\star\right\|, \left\|\underline{V}^k - \underline{V}^\star\right\|\right\} \leq \epsilon$. This concludes our proof. ■

In particular, the Bellman operator f and policy operator g^π are γ-contractive on $\mathbb{R}^S$, where γ is the discount factor, therefore Theorem 7.5 applies with $\alpha = \gamma$.

Remark 7.13 *Theorem 7.6 implies that at the termination of Algorithm 7.1, the fixed point set* $\mathcal{V}^\star$ *can be* over-approximated *by*

$$\mathcal{V}^\star \subseteq \mathcal{V}_{approx} := \prod_{s\in[S]} [\underline{V}_s^{k+1} - \epsilon, \overline{V}_s^{k+1} + \epsilon],$$

where k *is the last iterate before Algorithm 7.1 terminates.*

7.9 Relationship to Nash equilibria Sets in Single Controller Stochastic Games

In this section, we consider the setting in which the MDP dynamics is stationary, but the MDP cost has uncertainty characterized by the set $\mathcal{C} \subseteq \mathbb{R}^{SA}$, relate this parameter uncertainty to single controller stochastic games, and elaborate on the properties of the resulting fixed point set $\mathcal{V}^B$ in the context of single controller stochastic games. We show that with an appropriate over-approximation of the Nash equilibria cost parameters, $\mathcal{V}^B$ contains the optimal value functions for player one at Nash equilibria.

We note that the stochastic games we discuss here implicitly assume *imperfect information* [42, Def. 6.3.6] — at every state, both players have multiple actions to choose from.

Therefore, each player's choice of action induces uncertainty in their opponent's costs.

7.9.1 Stochastic Game

In a two-player stochastic game, both players solve their own MDP while sharing the same states and dynamics. As opposed to standard MDPs, each player's cost and transition kernel depends on the *joint policy*, $\pi = (\pi_1, \pi_2)$, where π_1 and π_2 are respectively player one and player two's policies as defined for MDPs in Section 2.2. The set of joint policies is given by Π, while player one's and player two's sets of policies are given by Π_1 and Π_2, respectively. We denote the actions of player one by a and the actions of player two by b. Players share a common state, given by $s \in [S]$. The transition kernel of the shared dynamics is determined by the tensor $Q \in \mathbb{R}^{S \times S \times A_1 \times A_2}$, where Q satisfies

$$\sum_{s' \in [S]} Q_{s'sab} = 1, \ \forall \ (s, a, b) \in [S] \times [A_1] \times [A_2],$$

$$Q_{s'sab} \geq 0, \ \forall \ (s', s, a, b) \in [S] \times [S] \times [A_1] \times [A_2].$$

Each player's cost is given by $D^i \in \mathbb{R}^{S \times A_1 \times A_2}$, where D^1_{sab} and D^2_{sab} denote player one and player two's cost when the joint action (a, b) is taken from state s, respectively.

When player two applies policy π_2, player one's transition kernel is given by

$$P^1(\pi_2) \in \mathbb{R}^{S \times SA_1}, \ P^1_{s',sa}(\pi_2) = \sum_{b \in [A_2]} (\pi_2)_{sb} Q_{s'sab}. \tag{7.60}$$

Similarly, player one's cost is given by

$$C^1(\pi_2) \in \mathbb{R}^{S \times A_1}, \ C^1_{sa}(\pi_2) = \sum_{b \in [A_2]} (\pi_2)_{sb} D^1_{sab}. \tag{7.61}$$

For a specific π_1 adopted by player one, player two's cost $C^2(\pi_1)$ and transition kernel $P^2(\pi_1)$ can be similarly defined. Each player then solves a discounted MDP given by $([S], [A_i], P^i(\pi_j), C^i(\pi_j), \gamma_i)$. Since each player only controls a part of the joint action space, the generalization to joint action space introduces *non-stationarity* in the transition and cost, when viewed from the perspective of an individual player solving an MDP.

Given a joint policy (π_1, π_2), each player attempts to minimize its value function. Player

i's optimal discounted infinite horizon expected cost is given by

$$V_s^i = \min_{\pi_i \in \Pi_i} \mathbb{E}_s^{\pi_i} \Big\{ \sum_{t=0}^{\infty} \gamma_i^t C^i_{s^t a^t}(\pi_j) \Big\}, \quad \forall\ s \in [S]. \tag{7.62}$$

As formulated by (7.62), we denote the value function of player one, V^1, by $V \in \mathbb{R}^S$ and the value function of player two, V^2, by $W \in \mathbb{R}^S$. Given a joint policy π, both players have unique stationary value functions $\big(V(\pi_1, \pi_2), W(\pi_1, \pi_2)\big)$ given by

$$V(\pi_1, \pi_2) = \nu^1(\pi_1, \pi_2) + \gamma_1 M_{\pi_1} P^1(\pi_2)^\top V(\pi_1, \pi_2), \tag{7.63a}$$
$$W(\pi_1, \pi_2) = \nu^2(\pi_1, \pi_2) + \gamma_2 M_{\pi_2} P^2(\pi_1)^\top W(\pi_1, \pi_2), \tag{7.63b}$$

where $\nu^1(\pi_1, \pi_2) = \sum_{i \in [S]} e_i e_i^\top M_{\pi_1} (\mathbb{1}_s \otimes I_{A_1}) C^1(\pi_2)^\top e_i$ and $\nu^2(\pi_1, \pi_2) = \sum_{i \in [S]} e_i e_i^\top M_{\pi_2} (\mathbb{1}_s \otimes I_{A_2}) C^2(\pi_1)^\top e_i$. Since a stochastic game can be viewed as coupled MDPs, the MDP notion of optimality must be expanded to reflect the dependency of a player's optimal policy on the joint policy space. We define a *Nash equilibrium* in terms of each player's value function [42, Sec.3.1].

Definition 7.13 *[Two Player Nash Equilibrium] A joint policy $\pi^\star = (\pi_1^\star, \pi_2^\star)$ is a Nash equilibrium if the corresponding value functions as given by (7.63) satisfy*

$$V(\pi_1^\star, \pi_2^\star) \leq V(\pi_1, \pi_2^\star), \quad \forall\ \pi_1 \in \Pi_1,$$

$$W(\pi_1^\star, \pi_2^\star) \leq W(\pi_1^\star, \pi_2), \quad \forall\ \pi_2 \in \Pi_2.$$

We denote the Nash equilibrium value functions as $V^\star = V(\pi_1^\star, \pi_2^\star)$, $W^\star = W(\pi_1^\star, \pi_2^\star)$ and the set of Nash equilibria for a stochastic game as $\Pi_{NE} \subset \Pi$.

Definition 7.13 implies that a Nash equilibrium is achieved when the joint policy simultaneously generates both value functions $V^\star$ and $W^\star$, which are the fixed points of the Bellman operator with respect to parameters $\Big(C^1(\pi_2), P^1(\pi_2)\Big)$ and $\Big(C^2(\pi_1), P^2(\pi_1)\Big)$, respectively — i.e. $V^\star = \min_{\pi_1 \in \Pi_1} \Big\{ \nu^1(\pi_1, \pi_2^\star) + \gamma_1 M_{\pi_1} P^1(\pi_2^\star)^\top V^\star \Big\}$, and $W^\star = \min_{\pi_2 \in \Pi_2} \Big\{ \nu^2(\pi_1^\star, \pi_2) + \gamma_2 M_{\pi_2} P^2(\pi_1^\star)^\top W^\star \Big\}$.

A Nash equilibrium is not unique in general sum stochastic games. Furthermore, Nash equilibrium policies are not necessarily composed of deterministic individual policies. Therefore while each player's Nash equilibrium value function is always the fixed point of the

associated Bellman operator, the Nash equilibrium policy for each player is *not* the optimal deterministic policy associated with the Nash equilibrium value function in general. The existence of at least one Nash equilibrium for any general sum stochastic game is given in [42]. When the stochastic game is also zero sum, all Nash equilibria correspond to a unique value function.

Since the technical content of this paper does not address non-stationarity in the transition kernel, we focus on analyzing non-stationarity in the cost term. Specifically, we constrain our analysis to a *single controller game* [42], i.e. when the transition kernel is controlled by player one only. Single controller stochastic games form an important class of games that models dynamic control in queueing networks [7] and attacker-defender games with stochastic transitions [10, 39]. Similar to our discussion of a two player Nash equilibrium, we exclusively consider a two player single controller game. However, we note that the following definition can be extended to an N player single controller stochastic game in which the transition kernel is independent of all but one player's actions.

Definition 7.14 (Single controller game) *A single controller game is a two player stochastic game where the probability transition kernel is independent of player two's actions, i.e., for each* $(s', s, a) \in [S] \times [S] \times [A]$

$$Q_{s'sab} = Q_{s'sab'}, \quad \forall\ b, b' \in [A],$$

i.e. $P^1(\pi_2) = P$, $\forall\ \pi_2 \in \Pi_2$ *and* $P^2(\pi_1)_{s',sb} = P^2(\pi_1)_{s',sb'}$, $\forall\ b, b' \in [A]$, $\pi_1 \in \Pi_1$.

Although both players are still optimizing their value functions in a single controller game, player two's policy only affects its immediate cost at each state, while its transition dynamic becomes a time-varying Markov chain. However, player two's policy still affects player one's MDP through cost matrix $C^1(\pi_2)$.

We analyze a single controller game from the set-based MDP perspective by utilizing Proposition 7.2. Suppose we are given a compact set $\mathcal{C} \subset \mathbb{R}^{S \times A}$ that over-approximates the set of $\mathcal{C}^{NE}$ — i.e. cost parameters that player one observes at Nash equilibria,

$$\mathcal{C}^{NE} = \{C^1(\pi_2^\star) \in \mathbb{R}^{S \times A} \mid (\pi_1^\star, \pi_2^\star) \in \Pi_{NE}\} \subseteq \mathcal{C}. \tag{7.64}$$

Then we show that the Nash equilibria value functions belong to the fixed point set of $F_{\mathcal{C}}$.

Valid over-approximations to $\mathcal{C}^{NE}$ can be easily found — the simplest being the interval set of all feasible costs.

Example 7.4 (Interval set approximation) *An approximation to $\mathcal{C}^{NE}$ can always be given by interval sets. At each state-action pair (s,a), the MDP cost parameter for player one is given by (7.61). Then we can take the maximum and minimum elements of the set $\{D^1_{sab}\}_{b\in[A_2]}$ for all state actions pairs (s,a) to form an interval set $\mathcal{C} = \mathcal{C}_{11} \times \ldots \times \mathcal{C}_{SA_1} \in H(\mathbb{R})^{S\times A_1}$, such that*

$$\mathcal{C}_{sa} = \{D^1_{sab}\}_{b\in[A_2]} = [\underline{C}_{sa}, \overline{C}_{sa}],$$

where $\underline{C}_{sa} = \min_{b\in[A_2]} D^1_{sab}$ and $\overline{C}_{sa} = \max_{b\in[A_2]} D^1_{sab}$ can be directly observed.

Interval sets will always give an admissible approximation. However, more general sets such as polytopes allow for more precise representations of the limiting value function trajectories for the game player.

Example 7.5 (Polytope set approximation) *Consider the set of costs at a particular state s in a two player single controller stochastic game, for which $A_1 = 2$ and $A_2 = 3$. Player one's costs corresponding to player two's deterministic policies are given by points $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$ in Figure 7.8. Any mixed policy from player two will result in an expected cost for player one that corresponds to a point within the blue region in Figure 7.8. On the other hand, the interval set approximation from Example 7.4 is given by the yellow polytope. In this example, we can observe that the interval set is a generous over-approximation of player one's feasible costs.*

An over-approximation of the set of feasible costs also over-approximates possible limiting trajectories for a player's learning algorithm. We consider the point $\times_1 = (C'_2, C'_1)$ in Figure 7.8. Fixed at this cost, value iteration would choose a_2 corresponding to C'_2, and return the corresponding discounted value function and transition kernel from state s. However, the feasible cost *when action a_2 has an equivalent cost C'_2 is at $\times_2 = (\bar{C}_1, C'_2)$ on the boundary of the blue polytope. Since $\times_2$ lies below the line $C_1 = C_2$, a_1 corresponding to $\bar{C}_1$ is the optimal action. Therefore, the resulting cost and transition kernel would have been different. This corresponds to a different value function trajectory that would have been infeasible.*

The set of feasible costs itself is an over-approximation of the set of Nash equilibria costs $\mathcal{C}^{NE}$. As Example 7.5 shows, the extension from interval sets to compact sets enables

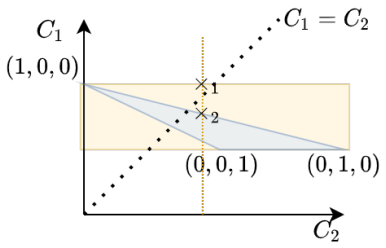


Figure 7.8: Feasible player costs vs interval set over-approximation.

additional information (feasible costs and knowledge of opponents' action constraints) to be used to approximate $\mathcal{C}^{NE}$ to greater accuracy.

Given a compact set $\mathcal{C}$ that over-approximates the set of player one's cost parameters at Nash equilibria, $\mathcal{C}^{NE}$, we now show that the Nash equilibria value functions for player one must lie within $\mathcal{V}^\star$, the fixed point set of $F_\mathcal{C}$.

Theorem 7.7 *In a single controller game, let $\mathcal{C} \subset \mathbb{R}^{S\times A}$ be an over-approximation of Nash equilibria costs for player one as in (7.64). If $\mathcal{C}$ is compact, then the set of stationary value functions for player one at Nash equilibria policies (7.63a) is a subset of $\mathcal{V}^\star$, the fixed point set of $F_\mathcal{C}$.*

Proof: We define the set of Nash equilibria value functions for player one as

$$\mathcal{V}^{NE} = \Big\{ V \in \mathbb{R}^S \mid V = f(V, C^1(\pi_2^\star), P) \Big\}. \tag{7.65}$$

For any $V^\star \in \mathcal{V}^{NE}$, there exists $C^\star = C^1(\pi_2^\star) \in \mathcal{C}$ such that $V^\star$ is the fixed point of $f(\cdot, C^\star, P)$. Then from Corollary 7.1, $V^\star \in \mathcal{V}^B$ (7.20).

■

Remark 7.14 *Although the Nash equilibrium value function $V^\star$ is always the unique fixed point of the Bellman operator $f(\cdot, C^\star, P)$, where $C^\star$ is player one's cost at Nash equilibrium, we note that in general, player one's policy at Nash equilibrium is not the optimal deterministic policy of $f(\cdot, C^\star, P)$ given by (7.10); this is because the joint policy at Nash equilibrium may not be composed of deterministic individual policies, while the solution to (7.10) is always deterministic.*

However, if we consider the set of all deterministic policies that solve (7.10), then player one's policy at Nash equilibrium must be a convex combination within this set [42].

We summarize the application of set-based MDP framework to single controller stochastic games as the following: when $\mathcal{C}$ over-approximates the set of costs at Nash equilibria, the fixed point set $\mathcal{V}^\star$ of the operator $F_\mathcal{C}$ contains all of the Nash equilibria value functions for player one in a single controller stochastic game.

7.10 Example: Single Controller Stochastic Games

We demonstrate this by applying interval set-based value iteration in a two-player single-controller stochastic game, and showing that both transient and asymptotic behaviors of player one's value function can be bounded, regardless of the opponent's learning algorithm.

We consider a two-player single controller stochastic game as defined in Definition 7.14, where each player solves a discounted MDP given by $([S], [A_{1,2}], P, C^{1,2}, \gamma_{1,2})$, where $A_1 = A_2 = A$. Both players share an identical state-action space $([S], [A])$ as well as the same transition probabilities P controlled by player one's actions. Player one's cost is given by

$$C^1_{sa}(\pi_2) = C_{sa} + J_{sb}\pi_2(s, b), \quad \forall\ (s, a) \in [S] \times [A],$$

while player two's cost is given by

$$C^2_{sb}(\pi_1) = C_{sb} - J_{sa}\pi_1(s, a), \quad \forall\ (s, b) \in [S] \times [A],$$

where the matrix $J \in \mathbb{R}_+^{S\times A}$ is the same for the two costs.

While algorithms that converge to Nash equilibrium exist [78, 53] for such single controller games, convergence is not guaranteed if players do not coordinate on which algorithm to use between themselves. In this section, we utilize the set-based Bellman operator to show that we can determine the value function set that player one's Nash equilibrium value function belongs to, and equivalently, the value function set that player one's value function trajectory converges to, regardless of what the opponent does.

We define the state space of a stochastic game on a 3×3 grid, shown in Figure 7.9 left, where the total number of states is $S = 9$ and the total number of actions per state is $A = 4$. State s' is a neighboring state of s if it is immediately connected to s by a green arrow in Figure 7.9 left. At state s, let $\mathcal{N}_s$ denote the set containing all neighboring states of s and let N_s denote the number of elements in $\mathcal{N}_s$.

As shown in Figure 7.9 right, the actions available in each state are labeled 'left', 'right', 'up', or 'down'. From each state s, an action is feasible if it coincides with a green arrow in

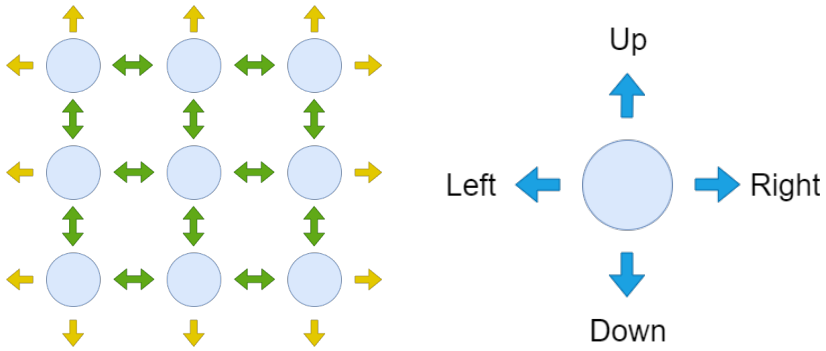


Figure 7.9: (a): Each player's state space $[S]$, $S = 9$. Green actions lead to a neighboring state and yellow actions are infeasible. (b): Actions space $[A]$, $A = 4$.

Figure 7.9 right, and infeasible if it coincides with a yellow arrow. For feasible actions, its transition probabilities are given as

$$P_{s'sa} = \begin{cases} 0.7 & s' = \text{ target state} \\ \frac{0.3}{N_s - 1} & s' \neq \text{ target state, } s' \in \mathcal{N}_s \\ 0 & \text{otherwise} \end{cases}. \tag{7.66}$$

In (7.66), we define the target state s' of state-action pair (s, a) to be the neighboring state of s in the direction of action a. If action a is infeasible, its transition probabilities are defined as

$$P_{s'sa} = \begin{cases} \frac{1}{N_s} & s' \in \mathcal{N}_s \\ 0 & \text{otherwise} \end{cases}. \tag{7.67}$$

We select matrices $C, J \in \mathbb{R}^{9\times 4}$ by randomly sampling each element C_{sa}, J_{sa} uniformly from the interval $[0, 1]$. As in Example 7.4, we derive an over-approximation of player one's feasible costs as interval set $\mathcal{C}$, given by

$$\left\{ C^1 \in \mathbb{R}^{9\times 4} \mid C^1_{sa} \in [C_{sa}, C_{sa} + J_{sa}],\ \forall\ (s, a) \in [9] \times [4] \right\}, \tag{7.68}$$

where the upper bound $C_{sa} + J_{sa}$ is achieved when player two's probability of taking action $b = a$ from state s is 1.

We consider a two-player value iteration algorithm presented in Algorithm 7.2 which

forms the basis of many dynamic programming-based learning algorithms for stochastic games [42, 78]. At each time step, player one takes the optimal policy π^{k+1} given by (7.10) that solves the Bellman operator $f(V^k, C^k, P)$, where C^k is player one's cost parameter at step k and V^k is player one's value function at step k — i.e. player one performs value iteration at every time step. Player two obtains its optimal policy using the function $g : \Pi_1 \to \Pi_2$, we do not make any assumptions of g, it may produce any policy π_2 in response to the policy π_1.

Algorithm 7.2 Two player VI

Input: $([S], [A], P, C^{1,2}, \gamma_{1,2})$, V_0.
Output: $V^\star, \pi_1^\star$
$\pi_1^0(s) = \pi_2^0(s) = 0, \forall\ s \in [S]$
for $k = 0, \ldots,$ **do**
$(V^{k+1}, \pi_1^{k+1}) = f(V^k, C^1(\pi_1^k, \pi_2^k), P)$
$\pi_2^{k+1} = g(\pi_1^{k+1})$
end for

Our analysis provides bounds on player one's value function when we do not know how player two is updating its policy — i.e. when g is unknown. In the simulation, we take g to be different strategies and show that player one's value functions are bounded by the interval set analysis and converge towards the fixed point set of the corresponding Bellman operator.

Suppose both players are updating their policies via value iteration (7.10). Player one performs value iteration with a discount factor of $\gamma_1 = 0.7$, while player two performs value iteration with an unknown discount factor $\gamma \in (0, 1)$. Assuming both players' value functions are initialized to be 0 in every state, we simulate player one's value function trajectories for different values of γ in Figure 7.10.

Figure 7.10 shows that when player two utilizes different discount factors, player one experiences *different* trajectories even though both players are utilizing value iteration to minimize their losses. However, the value function trajectory that player one follows is always bounded between the thresholds we derived from Proposition 7.2. As Figure 7.10 shows, there does not seem to be any direct correlation between player two's discount factor and player one's value function. However, the interval bounds we derived do tightly approximate resulting value function trajectories.

Alternatively, suppose we know that player two has the same discount factor as player one, but we do not know player two's initial value function or if it is minimizing or maximizing

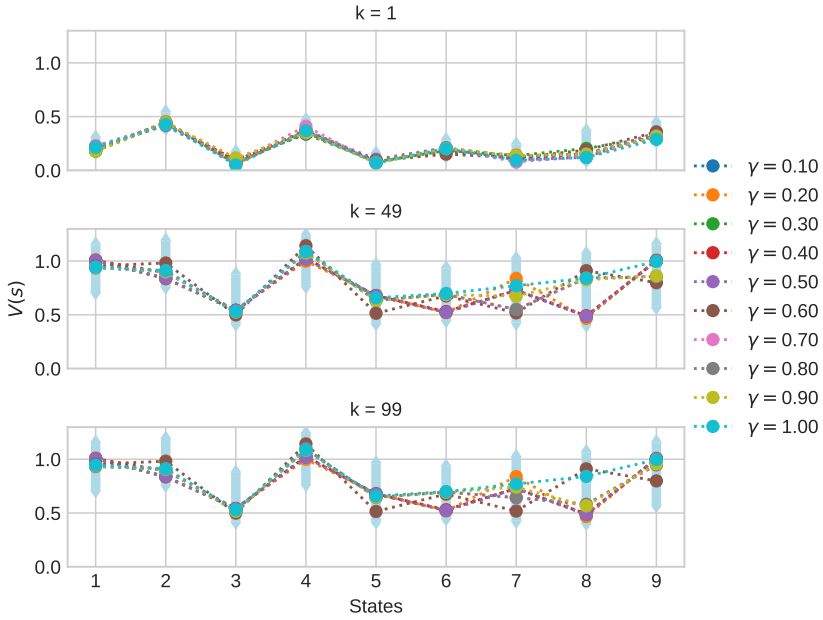


Figure 7.10: Player one's value function as a function of state at different iterations $k = \{1, 49, 99\}$. Range shown in blue is the bounded interval $\mathcal{V} = [\underline{V}^k, \overline{V}^k]$ at the corresponding iteration k.

its discounted objective. We analyze both scenarios: when player two is also minimizing its cost and when player two is maximizing its cost. In Figure 7.11, the infinity norm of player one's value function at each iteration k is shown with respect to these two scenarios. Both player one and player two's initial value function is randomly initialized as $V_s^0 \in [0, 1]$, $\forall\ s \in [9]$. Figure 7.11 plots player one's value function trajectory when player two utilizes value iteration towards different objectives: towards minimizing C^2 (player one's value functions shown in dotted lines) and towards maximizing C^2 (player one's value functions shown in solid lines). The grey region shows the predicted bounds as derived from Proposition 7.2.

As Figure 7.11 shows, player two's policy change causes a significant shift in player one's value function trajectory. When player two attempts to maximize its expected cost, player one's function achieves the absolute lower bound as predicted by $\mathcal{V}$. This is because at least four actions with different costs are available in each state. Since both players are only selecting from deterministic policies, they are bound to select different actions unless all actions have the same cost. On the other hand, if player two is minimizing its value

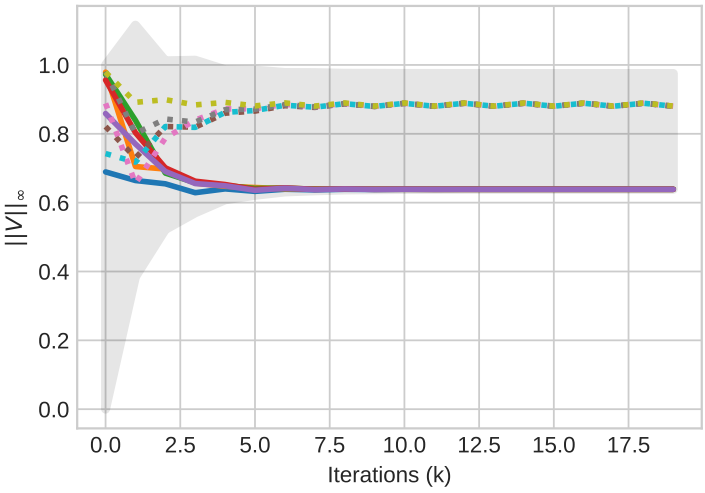


Figure 7.11: The infinity norm of player one's value functions as a function of iteration k.

function, then both players would precisely select the same state-actions at every time step. Then depending on the coupling matrix A, they may or may not choose a less costly action at the next step. This causes the limit cycle behavior that the dotted trajectories exhibits. In terms of the tightness of the bounds we derived in Proposition 7.2, we note that Figure 7.11 also shows the existence of trajectories that approach both the upper and lower bounds, therefore in practice the set-based bounds are shown to be tight.

7.11 Path Planning in Time-varying Wind Fields

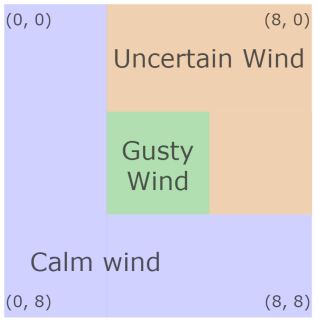
We apply set-based value iteration to wind-assisted probabilistic path planning of a balloon in strong, uncertain wind fields [140]. MDP as a model for wind-assisted path planning of balloons in the stratosphere and exoplanets has recently gained traction [140, 16]. Discrete state-action MDPs are a viable high-level path planning model [140] for such applications.

Mission Objective. In the two-dimensional wind field, we assume that the wind-assisted balloon is tasked with reaching the target state $(8, 8)$ in Figure 7.12 using minimum fuel.

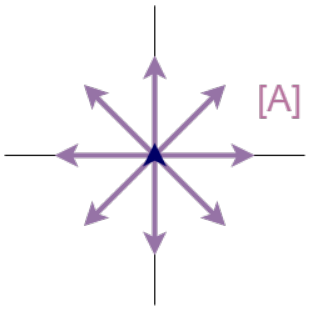
Uncertain Wind Fields. By collecting a set of wind data on the environment's wind field, an MDP can be created and a policy that handles stochastic planning can be deployed. However, wind can be a time-varying factor that causes the *expected* optimal policy to have *worse-than-expected* worst-case performance. We built an ideal uncertain wind field to

demonstrate how the set Bellman operator can be used to predict the best and worst-case behavior of a robust policy.

MDP Modeling Assumptions. Following the framework described in [140], we model the path planning problem in an uncertain wind field as an infinite horizon, discounted MDP with discrete state-actions in a two-dimensional space. While balloons typically traverse in three dimensions, we assume that the wind is consistent in the vertical direction and that the final target is any vertical position along the given two-dimensional coordinates. As a result, we can disregard the vertical position during planning.



(a) Wind field traversed by the balloon, discretized into 81 states.



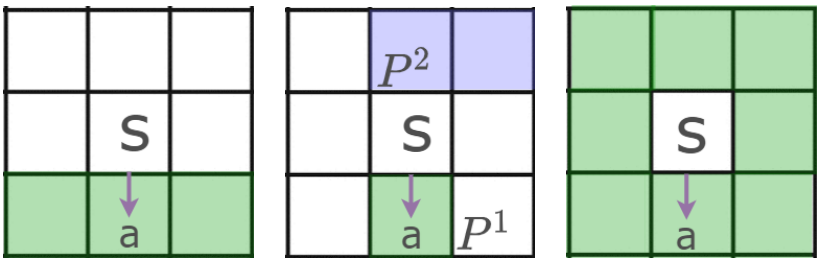
(b) At each state, 9 actions corresponding to different thrust vectors are available.

Figure 7.12

States. A total of 81 states represent the two-dimensional space, composed of three different regions characterized by their wind variability as shown in Figure 7.12.

1. **Calm wind**. In calm states S_{calm}, the wind magnitude varies uniformly between $[0, 0.5]$, and the wind direction is uniformly sampled between $[0, 2\pi]$. $S_{calm} = \{(i,j) \mid (0,0) \leq (i,j) \leq (2,8),\ (6,0) \leq (i,j) \leq (8,8)\}$.
2. **Gusty wind**. In states with gusts S_{gusty}, wind magnitude is consistently equal to 1, while the wind direction is uniformly sampled between $[0, 2\pi]$. $S_{gusty} = \{(i,j) \mid (3,3) \leq (i,j) < (6,6)\}$.
3. **Unreliable wind**. In unreliable states $S_{unreliable}$, a predictable wind front occasionally moves across an otherwise windless region. In other words, the wind magnitude is either 0 or 1 and the wind direction varies uniformly between $[\pi/4, \pi/2]$.

Actions. The balloon is equipped with an actuator that provides a constant thrust of 1 in 8 discretized directions shown in Figure 7.12b. The only stationary action vector with a zero magnitude is highlighted in blue in the center of Figure 7.12b. We assume that the actuation force is enough to move the balloon across one state in wind with a magnitude less than or equal to 0.5, and is otherwise not strong enough to overcome wind effects.



(a) State transition in calm wind. (b) State transition in unreliable wind. (c) State transition in gusty wind.

Figure 7.13: Transition probabilities for the three different wind regions.

Transition Probabilities. The transition probabilities in S_{calm} and S_{gusty} are certain. At each state s, we consider the following neighboring states.

1. $\mathcal{N}(s)$: all 8 neighboring states of state s.
2. $\mathcal{N}(s, a, 0)$: the neighboring state of s in the direction of a.
3. $\mathcal{N}(s, a, 1)$: the neighboring state of s in the direction of a plus the two adjacent states as shown in Figure 7.13a.
4. $\mathcal{N}(s, a, 2)$: the up and upper-right neighbors of s, as shown in Figure 7.13b.

In the calm wind region, the transition probabilities are given by

$$P_{sa,s'} = \begin{cases} \frac{1}{\mathcal{N}(s,a,1)}, & s' \in \mathcal{N}(s, a, 1) \\ 0 & \text{otherwise} \end{cases}, \ \forall\ s \in [S_{calm}]. \tag{7.69}$$

In the gusty wind region, the transition probabilities are given by

$$P_{sa,s'} = \begin{cases} \frac{1}{\mathcal{N}(s)}, & s' \in \mathcal{N}(s) \\ 0 & \text{otherwise} \end{cases}, \ \forall\ s \in [S_{gusty}],\ \forall\ a \in [A]. \tag{7.70}$$

In the unreliable wind region, the transition probabilities vary between transition dynamics P_s^1 and P_s^2.

$$P^1_{sa,s'} = \begin{cases} 1, & s' \in \mathcal{N}(s,a,0) \\ 0 & \text{otherwise} \end{cases}, \ \forall\ s \in [S_{gusty}],\ \forall\ a \in [A]. \tag{7.71}$$

$$P^2_{sa,s'} = \begin{cases} 0.5, & s' \in \mathcal{N}(s,a,2) \\ 0 & \text{otherwise} \end{cases}, \ \forall\ s \in [S_{gusty}],\ \forall\ a \in [A]. \tag{7.72}$$

Collectively, P_s^1 and P_s^2 collectively form the uncertainty set $\mathcal{P}_s \subset \Delta_S^A$ defined at each state.

$$\mathcal{P}_s = \{P^i_{sa} \mid i \in \{1,2\}, a \in [A]\},\ \forall s \in [S_{unreliable}]. \tag{7.73}$$

Cost. We define the following state-action cost to achieve the mission objective: at each state-action, the cost is the sum of the current distance from the target position $s_{targ} = (8,8)$, as well as the fuel expended by the given action.

$$C((i,j),a) = \sqrt{(i - s_{targ}[0])^2 + (j - s_{targ}[1])^2} + \frac{1}{2}\left\|a\right\|_2.$$

We take $a = 1$ for all actions except for the staying still action, where $a = 0$.

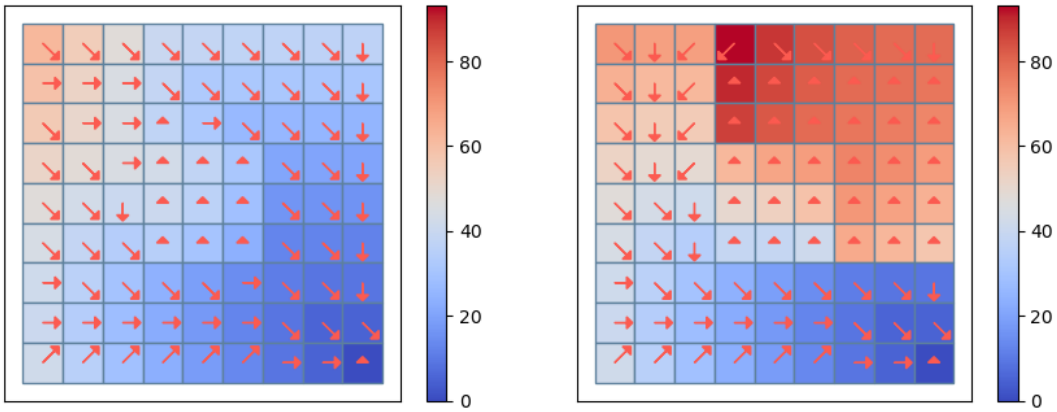
7.11.1 Bellman, Optimistic Policy, and Robust Policy

We first compute the optimistic and robust bounds of the MDP with parameter uncertainty in $\mathcal{P}$ when $s \in [S_{unreliable}]$ by running Algorithm 7.1. The results are shown in Figure 7.14.

We denote the optimistic policy as π^o and the robust policy as π^r, and derive the bounds of their respective value vector sets $\mathcal{V}^o$ (7.41) and $\mathcal{V}^r$ (7.42) using Algorithm 7.1. The output is compared against the bounds of the set-based Bellman operator's fixed point set $\mathcal{V}^\star$ in Table 7.1.

Set	Maximum value	Minimum value
$\mathcal{V}^\star$	70.61	62.25
$\mathcal{V}^o$	101.58	62.25
$\mathcal{V}^r$	70.63	70.52

Table 7.1: Bellman, optimistic policy, robust policy value bounds of the uncertain wind field.



(a) Optimistic case with an expected objective of 54.2

(b) Robust case with an expected objective of 96.7.

Figure 7.14

Time-varying wind field Next, we consider a time-varying wind field: at each time step k, the transition probability P^k is chosen at random from $\mathcal{P}$ (7.73). In this time-varying wind field, we compare three different policy deployments: 1) stationary optimistic policy π^o as policy operator g^o (7.41), 2) stationary robust policy π^r as policy operator g^r (7.42), and 3) dynamically changing policy that is optimal for the MDP $([S], [A], P^k, C, \gamma)$ as f (7.9). These three different policy deployments are given by

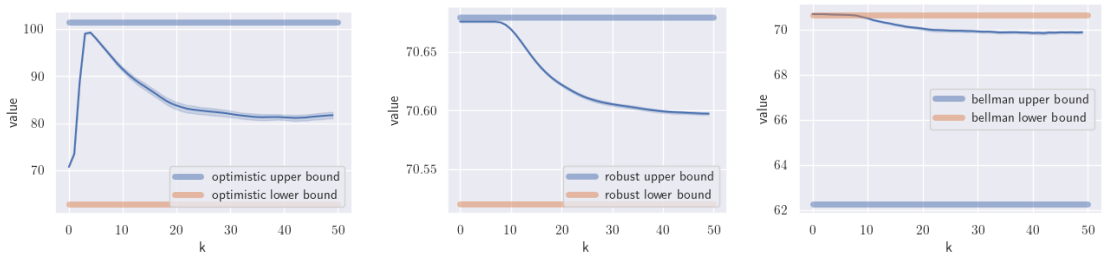
$$V^{k+1} = g^o(V^k, C, P^k), \tag{7.74}$$

$$V^{k+1} = g^r(V^k, C, P^k), \tag{7.75}$$

$$V^{k+1} = f(V^k, C, P^k). \tag{7.76}$$

The resulting cost-to-go at state $s_{orig} = [0, 0]$ is plotted in Figure 7.15. Here, we see that the optimistic policy deployment (7.74) has the greatest variation in value over the course of 50 MDP time steps. Both the robust policy deployment (7.75) and the dynamically changing policy deployment (7.76) achieve better upper-bound at each MDP iteration. The dynamically changing policy deployment (7.76) achieves less than 70 in cost-to-go on average, which is the best among all three deployments. As we discussed in Remark 7.11, the robust policy deployment has the smallest variance in value in the presence of wind uncertainty,

achieving a value difference of less than 0.1.



(a) Optimistic Policy with $\mathcal{V}^o$'s bounding values.

(b) Robust Policy with $\mathcal{V}^r$'s bounding values.

(c) Dynamically changing policy with $\mathcal{V}^B$'s bounding values.

Figure 7.15: Comparison of robust policy, optimistic policy, and Bellman policy's value trajectories in time-varying wind fields. The Center blue line is the average of over 50 trials. The shaded blue region denotes the standard variation. The top and bottom lines are the supremum and infimum values of the fixed points.

Sampled solutions. We can compute a sampled MDP model based on 50 samples of wind vectors for each state. Based on these samples, we add the action vector and compute the statistical distribution of state transitions. We then compute the value of this emph stationary sampled MDPs, and compare 9 randomly selected states' values. The resulting scatter plot is shown in Figure 7.16.

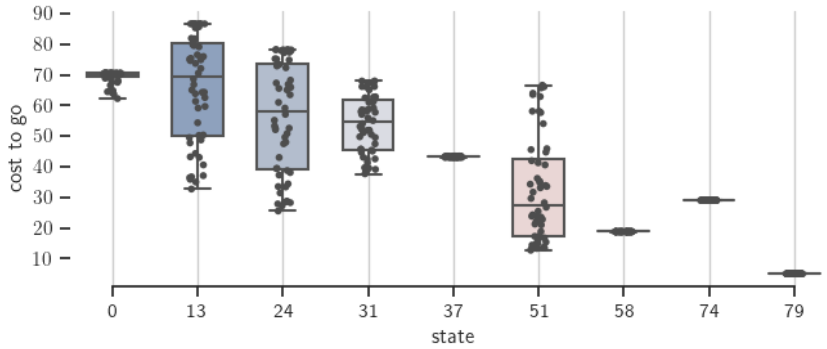


Figure 7.16: Comparison of different optimal value vectors under the Bellman operator for 50 randomly sampled MDPs. On the x-axis, the state number is computed as $i \times 9 + j$.

7.12 Conclusion

In this chapter, we categorized a class of operators utilized to solve Markov decision processes as value operators and lifted their input space from vectors to *compact sets* of vectors. We showed using fixed point analysis that the set extensions of value operators have fixed point sets that remain invariant given a compact set of MDP parameter uncertainties. These sets were applied to robust dynamic programming to further enrich existing results and generalize the k-rectangularity assumption for robust MDPs. Finally, we applied our results to a path-planning problem for time-varying wind fields. For future work, we plan on applying set-based value operators to stochastic games in the presence of uncoordinated players such as humans, as well as applying value operators to reinforcement learning to synthesize robust learning algorithms.

Chapter 8

CONCLUDING REMARKS

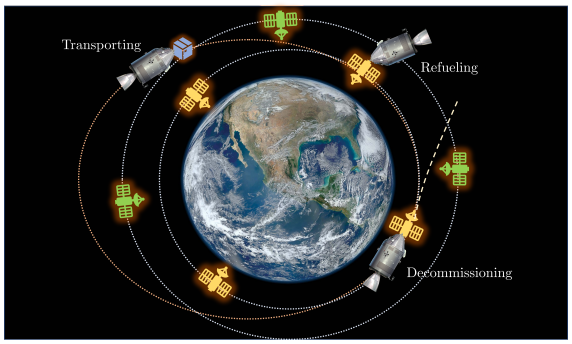
This dissertation applies MDPs, Markov games, and optimization techniques to enhance the safety, resilience, and robustness of large-scale autonomous systems in air traffic management and transportation. By formulating air traffic re-routing and competitive ride-hail as atomic MDP congestion games with specific potential functions, this dissertation opens up new possibilities for using reinforcement learning techniques to perform scalable trajectory planning with provable convergence guarantees. By analyzing the sensitivity of game-optimal solutions to changes in cost functions, we can develop urban air mobility infrastructure to be resilient to unexpected changes in operating conditions. Additionally, the analytical solutions to the gradient of the MDP Wardrop equilibrium and player cost coupling conditions offer a deeper understanding of how individual players' disturbances can propagate in gradient-based learning, which could inform the development of more efficient and stable autonomous systems. The exploration of the effects of parameter uncertainty on MDP learning operators provides insights into the challenges of implementing autonomous systems in real-world environments, paving the way for future research on developing robust and adaptable autonomous systems. Overall, this dissertation provides results that have the potential to advance the field of autonomous systems and contribute to the development of more efficient, safe, and sustainable transportation systems.

8.1 Future Directions

Today's autonomous capabilities are not ready for mass deployment in shared and congested spaces. While this dissertation analyzed a multi-agent model that addresses the fundamental resource limitations, competition, and task uncertainty features of multi-vehicle dynamics in large-scale transportation systems, much remains to be done. Based on the results from this dissertation, we discuss several interesting extensions to realize fully autonomous vehicle fleets in aerospace-based transportation systems

8.1.1 Sustainable Aerial and Orbital Autonomy at Scale

As UAV and satellite technologies mature, the adoption of UAV *fleets* and satellite *constellations* introduces distinct and domain-specific challenges to traditional aerial and orbital traffic management. Based on the tactical air traffic management example in Chapter 3.5, I propose an **autonomous air traffic control scheme** that explicitly accounts for vehicle heterogeneity, data-driven weather forecasts, and space usage efficiency. Moving away from human-based control also opens doors for noise compliance in real-time: by *adaptively* incentivizing urban air spaces for different urban topography and real-time aerial traffic patterns, the proposed air traffic scheme will autonomously uphold urban air traffic's sustainability metrics. Even the best satellites will eventually break down. With thousands of satellites in



orbit, a market is emerging for **satellite ride-share services**: spacecraft known as *space tugs* that specialize in refueling, de-orbiting, and transporting satellites among orbits. I propose an autonomous planning framework that combines orbital dynamics with ride-share forecasts to *pre-emptively* place space tugs in **optimal orbital positions**, from which space tugs can timely perform rendezvous with in-orbit satellites.

8.1.2 Conservative Autonomy in Human-shared Spaces

In a space congested with UAVs or robo-taxis, an immediate concern is how to share operational spaces without colliding with each other. I plan to extend existing game-theoretical equilibrium concepts to find the **trade-off between individual safety and fleet performance**, as well as to compute the level of conservatism that individual vehicles can adopt without compromising on efficiency and stability at the fleet level. Within this framework, I also plan to find inter-vehicle interactions that **prioritize the safety of human-piloted**

vehicles. Then, integrating optimization techniques, I plan to solve the individual policies that lead to stable and satisfactory fleet-level performance under customized safety constraints for autonomous and human-piloted vehicles.

BIBLIOGRAPHY

[1] Drones take to the sky, potentially disrupting last-mile delivery. https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/future-air-mobility-blog/drones-take-to-the-sky-potentially-disrupting-last-mile-delivery. Accessed: 2023-06-01.

[2] Spacex's 200th falcon 9 rocket launch looks absolutely gorgeous in these photos. https://www.space.com/spacex-falcon-9-200th-launch-photos. Accessed: 2023-03-01.

[3] This new drone is powerful enough to carry you and a friend. https://futurism.com/this-new-drone-is-powerful-enough-to-carry-you-and-a-friend. Accessed: 2023-05-03.

[4] Isolde Adler, Georg Gottlob, and Martin Grohe. Hypertree width and related hypergraph invariants. *European Journal of Combinatorics*, 28(8):2167–2181, 2007.

[5] Wesam H Al-Sabban, Luis F Gonzalez, and Ryan N Smith. Wind-energy based path planning for unmanned aerial vehicles using markov decision processes. In *2013 IEEE International Conference on Robotics and Automation*, pages 784–789. IEEE, 2013.

[6] Tansu Alpcan and Tamer Başar. *Network security: A decision and game-theoretic approach*. Cambridge University Press, 2010.

[7] Eitan Altman. Flow control using the theory of zero sum markov games. *IEEE Trans. Autom. Control*, 39(4):814–818, 1994.

[8] Eitan Altman. Constrained markov decision processes with total cost criteria: Occupation measures and primal lp. *Mathematical methods of operations research*, 43(1):45–72, 1996.

[9] Alvia. Uber new york. http://www.alvia.com/uber-city/uber-new-york/, 2021. Accessed: 2021-02-14.

[10] Samuel Ang, Hau Chan, Albert Xin Jiang, and William Yeoh. Game-theoretic goal recognition models with applications to security domains. In *Int. Conf. Decision Game Theory Secur.*, pages 256–272. Springer, 2017.

[11] James P Bailey, Gauthier Gidel, and Georgios Piliouras. Finite regret and cycles with fixed step-size via alternating gradient descent-ascent. *arXiv preprint arXiv:1907.04392*, 2019.

[12] David Balduzzi, Wojiech M Czarnecki, Thomas W Anthony, Ian M Gemp, Edward Hughes, Joel Z Leibo, Georgios Piliouras, and Thore Graepel. Smooth markets: A basic mechanism for organizing gradient-based learners. In *Int. Conf. Representation Learning (ICRL)*, 2020.

[13] David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating evaluation. In *Adv. Neural Inf. Process. Syst.*, pages 3268–3279, 2018.

[14] Amir Beck. *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*. SIAM, 2014.

[15] Martin Beckmann. A continuous model of transportation. *Econometrica*, pages 643–660, 1952.

[16] Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

[17] Marc G Bellemare, Georg Ostrovski, Arthur Guez, Philip Thomas, and Rémi Munos. Increasing the action gap: New operators for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[18] Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, 1999.

[19] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[20] Antonio Bicchi and Lucia Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221–231, 2000.

[21] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proc. Computational Statistics*, pages 177–186. Springer, 2010.

[22] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[23] Dietrich Braess. U about a paradox of traffic planning. *Oper. Res.*, 12(1):258–268, 1968.

[24] Peter Brooker. Longitudinal collision risk for atc track systems: a hazardous event model. *The Journal of Navigation*, 59(1):55–70, 2006.

[25] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015.

[26] Nicu Calcea. Nycdot's experience with big data and use in transportation projects. https://citymonitor.ai/transport/uber-lyft-rides-during-coronavirus-pandemic-taxi-data-5232, 2017. Accessed: 2021-02-14.

[27] Dan Calderone and S Shankar Sastry. Markov decision process routing games. In *2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*, pages 273–280. IEEE, 2017.

[28] Dan Calderone and S Shankar Sastry. Markov decision process routing games. In *Int. Conf. Cyber-Physical Syst.*, pages 273–280. IEEE, 2017.

[29] Dan Calderone and S Shankar. Infinite-horizon average-cost markov decision process routing games. In *Proc. Intell. Transp. Syst.*, pages 1–6. IEEE, 2017.

[30] Daniel Calderone. *Models of Competition for Intelligent Transportation Infrastructure: Parking, Ridesharing, and External Factors in Routing Decisions*. PhD thesis, U.C. Berkeley, 2017.

[31] Benjamin Chasnov, Lillian J. Ratliff, Eric Mazumdar, and Samuel Burden. Convergence analysis of gradient-based learning in continuous games. In *Proc. 35th Conf. Uncertainty Artif. Intell. (UAI)*, 2019.

[32] Man-Wah Cheung and Ratul Lahkar. Nonatomic potential games: the continuous strategy case. *Games and Economic Behavior*, 108:341–362, 2018.

[33] Nazlı Demir, Utku Eren, and Behçet Açıkmeşe. Decentralized probabilistic density control of autonomous swarms with safety constraints. *Auton. Robots*, 39(4):537–554, 2015.

[34] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Math. Program.*, 146(1):37–75, 2014.

[35] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *J. Mach. Learn. Res.*, 17(1):2909–2913, 2016.

[36] Asen L Dontchev and R Tyrrell Rockafellar. Implicit functions and solution mappings. *Springer Monographs in Mathematics. Springer*, 208, 2009.

[37] Prashant Doshi, Richard Goodwin, Rama Akkiraju, and Kunal Verma. Dynamic workflow composition: Using markov decision processes. *International Journal of Web Services Research (IJWSR)*, 2(1):1–17, 2005.

[38] Mahmoud El Chamie, Yue Yu, Behçet Açıkmeşe, and Masahiro Ono. Controlled markov processes with safety state constraints. *IEEE Trans. Autom. Control*, 64(3):1003–1018, 2018.

[39] Abdel Rahman Eldosouky, Walid Saad, and Dusit Niyato. Single controller stochastic games for optimized moving target defense. In *2016 IEEE Int. Conf Commun.*, pages 1–6. IEEE, 2016.

[40] Amir Epstein, Michal Feldman, and Yishay Mansour. Efficient graph topologies in network routing games. *Games and Economic Behavior*, 66(1):115–125, 2009.

[41] Lauren Aratani Erin Durkin. New york becomes first city in us to approve congestion pricing. https://www.theguardian.com/us-news/2019/apr/01/new-york-congestion-pricing-manhattan. Accessed: 2021-02-14.

[42] Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

[43] Drew Fudenberg, Fudenberg Drew, David K Levine, and David K Levine. *The theory of learning in games*, volume 2. MIT press, 1998.

[44] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2-3):177–201, 1993.

[45] Donald B Gillies. Solutions to general non-zero-sum games. *Contributions to the Theory of Games*, 4:47–85, 1959.

[46] Robert Givan, Sonia Leach, and Thomas Dean. Bounded-parameter markov decision processes. *Artificial Intelligence*, 122(1-2):71–109, 2000.

[47] GlobeNewsWire. Global on-demand transportation market report 2022: Major players include cabify, careem, curb mobility, daimler and europcar mobility.

[48] GlobeNewsWire. M5g core global market report 2023: High speed, large coverage, higher bandwidth, and higher reductions in network energy usage drive demand.

[49] Chris Godsil and Gordon Royle. Cuts and flows. In *Algebraic Graph Theory*, pages 307–339. Springer, 2001.

[50] Vineet Goyal and Julien Grand-Clement. Robust markov decision processes: Beyond rectangularity. *Mathematics of Operations Research*, 2022.

[51] Saıd Hamadene and Jean-Pierre Lepeltier. Zero-sum stochastic differential games and backward equations. *Systems & Control Letters*, 24(4):259–263, 1995.

[52] Jeff Henrikson. Completeness and total boundedness of the hausdorff metric. In *MIT Undergraduate J. Math.* Citeseer, 1999.

[53] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

[54] Garud N Iyengar. Robust dynamic programming. *Math. Op. Res.*, 30(2):257–280, 2005.

[55] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Int. Conf. Mach. Learning*, pages 427–435. PMLR, 2013.

[56] Jie Ji, Amir Khajepour, Wael William Melek, and Yanjun Huang. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. *IEEE Transactions on Vehicular Technology*, 66(2):952–964, 2016.

[57] Qiang Jiao, Hamidreza Modares, Shengyuan Xu, Frank L Lewis, and Kyriakos G Vamvoudakis. Multi-agent zero-sum differential graphical games for disturbance rejection in distributed control. *Automatica*, 69:24–34, 2016.

[58] Rudolf Emil Kalman et al. Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2):102–119, 1960.

[59] W Krichene, B Drighès, and A Bayen. Online Learning of Nash Equilibria in Congestion Games. *SIAM J. Control Optim.*, 53(2):1056–1081, 2015.

[60] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[61] N Vimal Kumar and C Selva Kumar. Development of collision free path planning algorithm for warehouse mobile robot. *Procedia comput. sci.*, 133:456–463, 2018.

[62] Simon Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

[63] Serge Lang. *Undergraduate analysis*. Springer Science & Business Media, 2013.

[64] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japan J. Math.*, 2(1):229–260, 2007.

[65] Erwan Lecarpentier and Emmanuel Rachelson. Non-stationary markov decision processes, a worst-case approach using model-based reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

[66] Sarah H. Q. Li, Yue Yu, Daniel Calderone, Lillian Ratliff, and Behcet Acikmese. Tolling for constraint satisfaction in markov decision process congestion games. In *Amer. Control Conf.*, pages 1238–1243. IEEE, 2019.

[67] Sarah HQ Li, Assalé Adjé, Pierre-Loïc Garoche, and Behçet Açıkmeşe. Bounding fixed points of set-based bellman operator and nash equilibria of stochastic games. *Automatica*, 130:109685, 2021.

[68] Sarah HQ Li, Assalé Adjé, Pierre-Loïc Garoche, and Behçet Açıkmeşe. Set-based value operators for non-stationary markovian environments. *arXiv preprint arXiv:2207.07271*, 2022.

[69] Sarah HQ Li, Daniel Calderone, and Behçet Açıkmeşe. Congestion-aware path coordination game with markov decision process dynamics. *IEEE Control Systems Letters*, 2022.

[70] Sarah HQ Li, Daniel Calderone, Lillian Ratliff, and Behçet Açıkmeşe. Sensitivity analysis for markov decision process congestion games. In *Conf. Decision Control (CDC)*, pages 1301–1306. IEEE, 2019.

[71] Sarah HQ Li, Avi Mittal, and Behcet Acikmese. Reducing collision risk in multi-agent path planning: Application to air traffic management. *arXiv preprint arXiv:2212.04122*, 2022.

[72] Sarah HQ Li, Lillian Ratliff, and Behçet Açıkmeşe. Disturbance decoupling for gradient-based multi-agent learning with quadratic costs. *IEEE Control Systems Letters*, 5(1):223–228, 2020.

[73] Sarah HQ Li, Yue Yu, Daniel Calderone, Lillian Ratliff, and Behçet Açıkmeşe. Tolling for constraint satisfaction in markov decision process congestion games. In *2019 American Control Conference (ACC)*, pages 1238–1243. IEEE, 2019.

[74] Sarah HQ Li, Yue Yu, Nicolas I Miguel, Dan Calderone, Lillian J Ratliff, and Behçet Açıkmeşe. Adaptive constraint satisfaction for markov decision process congestion games: Application to transportation networks. *Automatica*, 151:110879, 2023.

[75] Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *AAAI Conf. Artif. Intell.*, 2019.

[76] Zhi Li, Ali Vatankhah Barenji, Jiazhi Jiang, Ray Y Zhong, and Gangyan Xu. A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand. *J. Intell. Manuf.*, 31(2):469–480, 2020.

[77] X Lin. Environmental constraints in urban traffic management: Traffic impacts and an optimal control framework. 2018.

[78] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Mach. Learn. Proc. 1994*, pages 157–163. Elsevier, 1994.

[79] Yisha Liu, Qunxiang Wang, Huosheng Hu, and Yuqing He. A novel real-time moving target tracking and path planning system for a quadrotor uav in unknown unstructured outdoor scenes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2362–2372, 2018.

[80] Benny Lutati, Vadim Levit, Tal Grinshpoun, and Amnon Meisels. Congestion games for v2g-enabled ev charging. In *AAAI Conf. Artif. Intell.*, 2014.

[81] Shie Mannor, Ofir Mebel, and Huan Xu. Robust mdps with k-rectangular uncertainty. *Mathematics of Operations Research*, 41(4):1484–1509, 2016.

[82] Aarian Marshall. New york city flexes again, extending cap on uber and lyft. https://www.wired.com/story/new-york-city-flexes-extending-cap-uber-lyft/. Accessed: 2021-02-14.

[83] Eric Mazumdar, Lillian J Ratliff, Michael I Jordan, and S Shankar Sastry. Policy-gradient algorithms have no guarantees of convergence in continuous action and state multi-agent settings. *Int. Conf. Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2020.

[84] Eric Mazumdar, Lillian J Ratliff, and S. Shankar Sastry. On the convergence of gradient-based learning in continuous games. *SIAM J. Mathematics of Data Science*, 2019.

[85] Jonathan C McDowell. The low earth orbit satellite population and impacts of the spacex starlink constellation. *The Astrophysical Journal Letters*, 892(2):L36, 2020.

[86] Francisco S Melo. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pages 1–4, 2001.

[87] Enrique Lobato Miguélez, Luis Rouco Rodríguez, TGS Roman, FM Echavarren Cerezo, Ma Isabel Navarrete Fernández, Rosa Casanova Lafarga, and Gerardo López Camino. A practical approach to solve power system constraints with application to the spanish electricity market. *IEEE Trans. Power Syst.*, 19(4):2029–2037, 2004.

[88] Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49:267–290, 2002.

[89] Igal Milchtaich. Network topology and the efficiency of equilibrium. *Games and Economic Behavior*, 57(2):321–346, 2006.

[90] Dov Monderer and Lloyd S Shapley. Potential games. *Games Econ. Behav.*, 14(1):124–143, 1996.

[91] Ion Necoara and Valentin Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, 59(5):1232–1243, 2013.

[92] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[93] J v Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

[94] Lars Relund Nielsen and Anders Ringgaard Kristensen. Finding the k best policies in a finite-horizon markov decision process. *Eur. J. Oper. Res.*, 175(2):1164–1179, 2006.

[95] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[96] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[97] City of New York. Tlc trip record data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Accessed: 2021-02-14.

[98] Bureau of Public Roads. Traffic assignment manual. *US Department of Commerce*, 1964.

[99] Jun Ota. Multi-agent robot systems as distributed autonomous systems. *Advanced engineering informatics*, 20(1):59–70, 2006.

[100] D. Paccagnan, B. Gentile, F. Parise, M. Kamgarpour, and J. Lygeros. Distributed computation of generalized Nash equilibria in quadratic aggregative games with affine coupling constraints. In *Proc. IEEE 55th Conf. Decision and Control*, pages 6123–6128, 2016.

[101] Sindhu Padakandla, Prabuchandran KJ, and Shalabh Bhatnagar. Reinforcement learning algorithm for non-stationary environments. *Applied Intelligence*, 50(11):3590–3606, 2020.

[102] Francesca Parise and Asuman Ozdaglar. A variational inequality framework for network games: Existence, uniqueness, convergence and sensitivity analysis. *Games and Economic Behavior*, 2019.

[103] Michael Patriksson. Sensitivity analysis of traffic equilibria. *Transportation Science*, 38(3):258–281, 2004.

[104] Michael Patriksson. *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015.

[105] David W Peterson. A review of constraint qualifications in finite-dimensional spaces. *Siam Review*, 15(3):639–654, 1973.

[106] Arthur Pigou. *The economics of welfare*. Routledge, 2017.

[107] Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[108] Yuping Qiu and Thomas L Magnanti. Sensitivity analysis for variational inequalities. *Math. Op. Res.*, 17(1):61–76, 1992.

[109] Shankarachary Ragi and Edwin KP Chong. Uav path planning in a dynamic environment via partially observable markov decision process. *IEEE Transactions on Aerospace and Electronic Systems*, 49(4):2397–2412, 2013.

[110] Lillan J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. On the characterization of local nash equilibria in continuous games. *IEEE Trans. Autom. Control*, 61(8):2301–2307, Aug. 2016.

[111] Lillian J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. On the characterization of local nash equilibria in continuous games. *IEEE Transactions on Automatic Control*, 61(8):2301–2307, 2016.

[112] James B Rawlings. Tutorial overview of model predictive control. *IEEE control systems magazine*, 20(3):38–52, 2000.

[113] J Ben Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pages 520–534, 1965.

[114] Tim Roughgarden. *Selfish routing and the price of anarchy*, volume 174. MIT press Cambridge, 2005.

[115] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

[116] Oren Salzman and Roni Stern. Research challenges and opportunities in multi-agent path finding and multi-agent pickup and delivery problems. In *International Conference on Autonomous Agents and MultiAgent Systems*, pages 1711–1715, 2020.

[117] William H Sandholm. Potential Games with Continuous Player Sets. *J. Econ. Theory*, 97(1):81–108, 2001.

[118] Todd Schneider. Taxi and ridehailing usage in new york city. https://toddwschneider.com/dashboards/nyc-taxi-ridehailing-uber-lyft-data/, 2021. Accessed: 2021-02-14.

[119] Bernd SW Schröder. Ordered sets. *Springer*, 29:30, 2003.

[120] Gesualdo Scutari, Francisco Facchinei, Jong-Shi Pang, and Daniel P Palomar. Real and complex monotone communication games. *IEEE Transactions on Information Theory*, 60(7):4197–4231, 2014.

[121] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *Int. Conf. Machine Learning*, pages 3067–3075. JMLR, 2017.

[122] Lloyd S Shapley. Stochastic games. *Proc. Nat. Acad. Sci.*, 39(10):1095–1100, 1953.

[123] Ching-Shin Norman Shiau and Jeremy J Michalek. Optimal product design under price competition. *Journal of Mechanical Design*, 131(7):071003, 2009.

[124] Rob Shone, Kevin Glazebrook, and Konstantinos G Zografos. Applications of stochastic modeling in air traffic management: Methods, challenges and opportunities for solving air traffic problems under uncertainty. *European Journal of Operational Research*, 292(1):1–26, 2021.

[125] Matthew J Sobel. Continuous stochastic games. *Journal of Applied Probability*, 10(3):597–604, 1973.

[126] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. 2018.

[127] Csaba Szepesvári and Michael L Littman. A unified analysis of value-function-based reinforcement-learning algorithms. *Neural computation*, 11(8):2017–2060, 1999.

[128] Tatiana Tatarenko and Maryam Kamgarpour. Learning nash equilibria in monotone games. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3104–3109. IEEE, 2019.

[129] Gerald Tesauro et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[130] David P Thipphavong, Rafael Apaza, Bryan Barmore, Vernol Battiste, Barbara Burian, Quang Dao, Michael Feary, Susie Go, Kenneth H Goodrich, Jeffrey Homola, et al. Urban air mobility airspace integration concepts and considerations. In *2018 Aviation Technology, Integration, and Operations Conference*, page 3676, 2018.

[131] The Times. Is elon musk's starlink winning the war for ukraine? https://www.thetimes.co.uk/article/is-elon-musks-starlink-winning-the-war-for-ukraine-9s9rwgxb8, 2023. [Online; accessed 01-June-2023].

[132] RL Tobin and TL Friesz. Sensitivity analysis for equilibrium network flow. *Transportation Science*, 22(4):242–250, 1988.

[133] Harry L Trentelman, Anton A Stoorvogel, and Malo Hautus. *Control theory for linear systems*. Springer Science & Business Media, 2012.

[134] Herke Van Hoof, Tucker Hermans, Gerhard Neumann, and Jan Peters. Learning robot in-hand manipulation with tactile features. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 121–127. IEEE, 2015.

[135] Martijn Van Otterlo and Marco Wiering. Reinforcement learning and markov decision processes. *Reinforcement learning: State-of-the-art*, pages 3–42, 2012.

[136] Emmanouil-Vasileios Vlatakis-Gkaragkounis, Lampros Flokas, and Georgios Piliouras. Poincaré recurrence, cycles and spurious equilibria in gradient-descent-ascent for non-convex non-concave zero-sum games. In *Adv. Neural Inf. Process. Syst.*, pages 10450–10461, 2019.

[137] Reza Vosooghi, Joseph Kamel, Jakob Puchinger, Vincent Leblond, and Marija Jankovic. Robo-taxi service fleet sizing: assessing the impact of user trust and willingness-to-use. *Transport.*, 46(6):1997–2015, 2019.

[138] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[139] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Math. Op. Res.*, 38(1):153–183, 2013.

[140] Michael T Wolf, Lars Blackmore, Yoshiaki Kuwata, Nanaz Fathpour, Alberto Elfes, and Claire Newman. Probabilistic motion planning of balloons in strong, uncertain wind fields. In *2010 IEEE International Conference on Robotics and Automation*, pages 1123–1129. IEEE, 2010.

[141] Insoon Yang. A convex optimization approach to distributionally robust markov decision processes with wasserstein distance. *IEEE control systems letters*, 1(1):164–169, 2017.

[142] Yue Yu, Dan Calderone, Sarah HQ Li, Lillian J Ratliff, and Behçet Açıkmeşe. Variable demand and multi-commodity flow in markovian network equilibrium. *Automatica*, 140:110224, 2022.

[143] Dong Yue, Qing-Long Han, and Chen Peng. State feedback controller design of networked control systems. In *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.*, volume 1, pages 242–247. IEEE, 2004.

[144] Kyongsik Yun, Changrak Choi, Ryan Alimo, Anthony Davis, Linda Forster, Amir Rahmani, Muhammad Adil, and Ramtin Madani. Multi-agent motion planning using deep learning for space applications. In *ASCEND 2020*, page 4233. AIAA, 2020.

[145] Bo Zhou, Qiankun Song, Zhenjiang Zhao, and Tangzhi Liu. A reinforcement learning scheme for the equilibrium of the in-vehicle route choice problem based on congestion game. *Applied Mathematics and Computation*, 371:124895, 2020.