APPROACHES TO OPTIMAL CONTROL UNDER UNCERTAINTY IN TRANSPORTATION AND BIOLOGY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Mallory Elizabeth Gaspard

August 2025

APPROACHES TO OPTIMAL CONTROL UNDER UNCERTAINTY IN TRANSPORTATION AND BIOLOGY

Mallory Elizabeth Gaspard, Ph.D.

Cornell University 2025

In this thesis, we study approaches to optimal control in situations where there is *transitional uncertainty* (i.e., uncertainty in the system's next state after initiating a transition) or *temporal uncertainty* (i.e., uncertainty in the length of the planning horizon). We begin by considering discrete optimal control problems on graphs where the transitional uncertainty is described by a probability distribution over possible successor nodes. We introduce a broad subclass of *shortest path problems* (SSPs) called *opportunistically stochastic shortest path problems* (OSSPs). In OSSPs, the availability of actions specifying probabilistic transitions to successor nodes implies the availability of actions resulting in deterministic transitions to each of those nodes. We prove a set of mathematical conditions on the OSSP's transition cost function which allows the problem to be treated with fast algorithms originally designed for deterministic cheapest path problems on graphs (e.g., Dijkstra's or Dial's method). We also illustrate the OSSP's connection to numerical discretizations of partial differential equations (PDEs) encoding continuous optimal control problems and usefulness in determining nuanced driving directions for autonomous vehicles on road networks amidst lane change uncertainty.

Next, we tackle two continuous optimal control problems in which the length of the planning horizon T is a random variable. In the first problem, we propose an approach to determine a driver's optimal braking / acceleration strategy through a signalized intersection where the light's turning yellow time T follows a

discrete probability distribution. The driver seeks to optimize multiple competing objectives (e.g., fuel use, time to destination, comfort) and we show a selection of numerical experiments which illustrate how the driver handles these tradeoffs, and how to adjust the driving strategy when new information is learned about the true value of T.

In the second problem, we develop a framework to determine when an energy-optimizing pursuer should use a movement concealment tactic called "motion camouflage" (MC) to appear less-threatening to an evader amidst uncertainty in the prey's escape attempt time, T. Here, T is a non-homogeneous exponentially distributed random variable governed by a rate function λ that is dependent on the pursuer's state and the evader's position. We motivate this approach through the biological model problem of hover fly pursuit-evasion interactions, and our simulations reveal that there is a specific parameter regime for λ where MC tactics are worthwhile.

BIOGRAPHICAL SKETCH

Mallory Gaspard was born on March 12, 1997 in Lafayette, Louisiana to parents Stephanie Gaspard (née Hebert) and Anthony Gaspard. Mallory's enthusiasm for learning and interacting with the world around her began at a very early age. She was reading books by age three, and her parents would carry her around Barnes and Noble to read the spines as entertainment. By age five, she wanted to be an astronaut, and by age seven, she was governing fictional countries with leaves as currency on the playground at Epiphany Day School in New Iberia, Louisiana. Her vibrant childhood was characterized by a love of school, art lessons, and piano lessons, all of which set the foundation for the development of her lifelong interests.

From sixth grade through twelfth grade, Mallory attended the Episcopal School of Acadiana (ESA) in Cade, Louisiana. Beneath the oaks, amid the cane, the late (and absolutely great) Coach Brian Rhoades was the first person to tell middle-school Mallory that she'd likely be a mathematician one day. In ninth grade, Mallory discovered her first true love - quantum mechanics - and was enthusiastically mentored in a nearly four year quest to internalize (as much as one can) the mysteries of quantum states by her physics teacher, Andy LeGoullon. Following the encouragement of Mr. LeGoullon, her math teacher Ashley Jankower Dugas, her advisor Kat Movassaghi, and so many other faculty at ESA, Mallory took off full speed ahead along a physics and math trajectory.

In August of 2015, Mallory followed this trajectory to Rensselaer Polytechnic Institute (RPI) in Troy, NY where she spent four of the most transformative years of her life. Through the hallowed halls of the Jonsson-Rowland Science Center at RPI, Mallory met lifelong friends, learned science she dreamt about since childhood, and ultimately found herself at a crossroads between pursuing physics or math professionally. One cold, dark evening in early February 2017, Mallory's

Quantum Physics II professor, Peter Persans, pulled her aside after class and said, "I think you should check out Applied Math." Shortly thereafter, Mallory met RPI mathematics professors Gregor Kovačič and Peter Kramer. The rest was history. Mallory received a B.S. in Mathematics and Applied Physics from RPI in December 2018 and spent the subsequent semester in the Mathematics graduate program there serving as a teaching assistant (TA) for Advanced Calculus.

Mallory joined Cornell's Center for Applied Mathematics (CAM) as a PhD Student in August 2019. On February 10, 2020, Mallory began working with Alex Vladimirsky in the Cornell Mathematics Department on problems involving optimal control under uncertainty - one month before COVID changed the world. Despite such trying times for humanity, Mallory found support, inspiration, and true friendship within the CAM community as a whole. Mallory was a National Science and Engineering Graduate (NDSEG) Fellow from 2020-2023, and in the summer of 2024, Mallory was an intern Member of the Technical Staff at the Aerospace Corporation in El Segundo, CA. There, Mallory expanded her knowledge of multi-objective optimization methods, worked on projects of direct technical impact to the US Space Program, made new friends, learned to surf, and fell in love with Los Angeles County.

Following her graduation from Cornell, Mallory is thrilled to continue her scientific career as a Postdoctoral Research Associate at Princeton University, under the mentorship of Professor Naomi Leonard in Mechanical and Aerospace Engineering.

To my parents for giving me the world,

to Peter Kramer and Gregor Kovačič for opening my eyes to the world,

to my fellow CAMsters (current, past, and honorary), for brightening my world,

and to Alex Vladimirsky for teaching me (through his own ways) that I can take on the world.

ACKNOWLEDGEMENTS

First and foremost, I'd like to thank my family. Mom and Dad, I would be nothing without your love and support. You have supported my dreams and education since day one (literally). My appreciation for both of you could never be fully captured by words alone. I am so blessed that the universe chose you be my parents, and I love you both dearly. To my grandparents Kay and Larry, my grandparents Joyce and Ray, my Aunt Marci, and my cousins Davis and Henry - your constant love, enthusiasm for and encouragement of all of my endeavors has given me wind beneath my wings for as long as I can remember. I am so grateful for all of you.

To my RPI family - Gregor Kovačič, Peter Kramer, Peter Persans, Lakshmi, David Schmidt, Bella Peck, George Dowhan, Charles Martin, Andrew Kiselik, Liv Williams, Joey Souto, and Vera Titze: All of you showed me that family is a choice, and ours is bound by a shared passion for science and a commitment to caring for those around them. So much of who I am today has been deeply shaped by the time we have spent together, and I am forever grateful for our friendships.

To my Aerospace Corporation internship friends - Leah Ruckle, Brian Hansen, Sebi Torres, Eliya Pelton, and everyone else I crossed paths with in El Segundo. Getting to expand my technical knowledge in optimization while meaningfully contributing to the nation's space program alongside all of you was a true privilege.

My time at Cornell would not have been nearly as enriching without my years in the Russian Language Program. I extend my gratitude to Slava Paperno, Viktoria Tsimberov, and Raissa Krivitsky for their inspiring courses and constant support.

Erika Fowler-Decatur, thank you for being a ray of sunshine on earth. John Guckenheimer, thank you for your support and for bringing me into CAM with open arms. Phil Rusher, thank you for always being supportive. Lionel Levine, thank you for being such a great professor and a delightful and caring committee

member. Steve Strogatz, serving as your TA (twice!) was a highlight of my Cornell experience. I learned so much about teaching from you, and I am grateful for your mentorship. David Bindel, you are the definition of a "good egg." Thank you for serving on my committee and always being a warm, supportive presence. Jamol Pender, thank you for your fiercely supportive mentorship and your friendship. Your care for the CAMsters is touching, and I'm grateful to have been in your orbit at Cornell. I look forward to more Asian Chili Spot lunches when I visit.

My dearest CAMels, past, present, and honorary - This community has been my rock while we have all been living through a constant stream of "unprecedented events" over the last six years. From the legendary MMMutiny meetings with Marissa Gee and MingYi Wang, to all of the other incredible moments spent in CAM with Gokul Nair, Max Ruth, Shawn Ong, Shriya Nagpal, Misha Padidar, Kath Landgren, Carlos Martinez, Will Clark, Jackson Kulik, Liam Packer, Tayler Fernandes-Nunez, Maria Oprea, Arnav Gupta, Kyle Fridberg, Katerina Tang, Emma Roszkowski, Jonah Botvinick-Greenhouse, Ashira Mawji, Dennis Corraliza, Mai Pham, Esther Gallmeier, Poompol Buathong, Jennifer Zvonek, Robert Stephany, Amit Harlev, Robin Armstrong, and so many more. I am blessed to have you all in my life. Thank you for the light you bring into this world.

And last but not least, to my advisor, Alex. The PhD has been a journey, and there is nobody else that I could have imagined guiding me instead. Even through the triumphs and challenges of research and graduate school, there are very few folks on this earth who have left such a profound impact on me as you have. The beautiful mathematics you've introduced me to and the life lessons you've imparted on me have touched my mind and my soul. I will carry these gifts and a piece of you with me for the rest of my days. Thank you.

The work presented in this thesis was supported by a Cornell Dean's Excellence Fellowship, a National Defense Science and Engineering Graduate (NDSEG) Fellowship, and the National Science Foundation (awards DMS-1645643, DMS-2111522).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

> "Plans should be ephemeral, so be prepared to move away from them."
>
> Anthony Bourdain

From the control module in an autonomous taxi needing to plan the fastest route to the customer's destination, to an animal predator attempting to capture its prey in the most energy efficient manner, determining the "best" course of action for the scenario at hand is crucial to both the functionality and survivability of countless real world systems. Optimal control provides a mathematical mechanism of determining *how* a system should behave in order for it to best satisfy its objectives according to some specified metric. However, an unexpected event that occurs nearby may render the original "best course of action" obsolete. For example, the autonomous taxi that was following the pre-planned fastest route may be unsuccessful in its lane change attempt to take a necessary exit from the highway. Or, the escaping evader may suddenly change to a drastically different path, and the predator's pre-planned capture trajectory will no longer be energy-optimal in response to the evader's new route.

When the the likelihood of potentially plan-disrupting events can be described by a suitable probability distribution, it is possible to formulate a *stochastic realization* of the optimal control problem which takes the possibility of these events into account *during* the planning process. As such, this thesis discusses approaches to optimal control under uncertainty which are motivated by a range of specific

applications. While there are many different types of uncertainty that may arise in real world settings, we narrow our focus to scenarios which involve one of two types:

1) *transitional* - uncertainty in *where* exactly the system will end up after choosing to transition to a new state.
2) *temporal* - uncertainty in *when* a (possibly) plan-altering event will take place, or *when* more information about the surroundings will be revealed.

For the first type of uncertainty, we focus on scenarios which can be modeled mathematically by *discrete* cheapest path problems on graphs. For the second type of uncertainty, we focus on where it arises in situations suitably modeled by optimal control problems in *continuous* domains. The following subsections contain overviews of the relevant types of optimal control problems in their *deterministic* form and an introduction to their stochastic versions which explicitly deal with uncertainty.

1.1 An Overview of Discrete Cheapest Path Problems

> Life can only be understood going backwards, but it must be lived going forwards.
>
> Kierkegaard

We begin[1] with an overview of deterministic, discrete "cheapest path prob-

[1]The eipgraph quote above is used by *many* folks - from Bertsekas in his books to my advisor,

lems." Discrete cheapest path problems aim to determine the *route* or *path* in a discrete state space from a starting state $\boldsymbol{x}_i$ to a specified target state $\boldsymbol{t}$ that has the *least total cost* in comparison to all other possible paths to $\boldsymbol{t}$ starting from $\boldsymbol{x}_i$. The total cost of a path may refer to the total distance traveled along that path, the total time taken to traverse that path, the amount of fuel needed to follow that path, or any other quantification of an expenditure related to capturing the controller's optimization objectives.

To formalize this, let $X := \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{N+1} = \boldsymbol{t}\}$ be the state space. At the initial stage $k = 0$, the system starts at state (node) $\boldsymbol{y}_0 = \boldsymbol{x}_i \in X$. To move to the next stage, the controller selects a successor node $\boldsymbol{y}_{k+1} = \boldsymbol{x}_j$ to transition to from a set of possible successors $\mathcal{N}(\boldsymbol{y}_k)$, and pays the nonnegative cost $C(\boldsymbol{x}_i, \boldsymbol{x}_j) = C_{ij}$ for this choice. This process continues until the system reaches $\boldsymbol{t}$. The full path to $\boldsymbol{t}$ is encoded in a sequence of nodes $(\boldsymbol{y}_0 = \boldsymbol{x}_i, \dots, \boldsymbol{y}_k, \dots, \boldsymbol{y}_M = \boldsymbol{t})$ through which the system must pass in order to reach $\boldsymbol{t}$ after M stages. Thus, the total cost of the specific path taken is the sum of the transition costs incurred at each stage.

At stage k, a control mapping $\mu_k(\cdot)$ provides a rule for *how* to select which of the possible successors to transition to from the current node. A control *policy* $\pi := (\mu_0, \dots, \mu_k, \dots)$ is a sequence of control mappings to be used at each stage, and the set of *all* polices is denoted Π. If the control mappings do not change from stage to stage, we say that the policy is *stationary* (i.e., $\pi := (\mu, \mu, \dots)$). The total cost incurred while traveling to $\boldsymbol{t}$ starting from $\boldsymbol{y}_0 = \boldsymbol{x}_i$ using policy π is

$$\mathcal{J}(\boldsymbol{x}_i, \pi) = \sum_{k=0}^{\infty} C(\boldsymbol{y}_k, \mu_k(\boldsymbol{y}_k)). \tag{1.1}$$

The value function $u(\boldsymbol{x}_i)$ (i.e., *minimum cost to reach* $\boldsymbol{t}$ starting from $\boldsymbol{x}_i$) is defined

Alex in his lectures - when introducing optimal control and the dynamic programming principle. It would be a shame to not continue such a tradition here.

as

$$u(\boldsymbol{x}_i) = \inf_{\pi \in \Pi} \{\mathcal{J}(\boldsymbol{x}_i, \pi)\}. \tag{1.2}$$

Naturally, some control policies are going to be more favorable candidates for cost minimization than others. Depending on the set of available successors at each node, it may be possible to *revisit* the same nodes multiple times. If this is the case, some of the policies available to the controller may accumulate an infinite cost as a result of cycling forever between a few states belonging to a subset of $X \setminus \{\boldsymbol{t}\}$. As such, the optimal policy π^* must allow the system to reach $\boldsymbol{t}$ in *finitely many* stages from each state in $X \setminus \{\boldsymbol{t}\}$.

The dynamic programming principle (DP) provides a recursive means of determining the value function defined in (1.2). Pioneered by Richard Bellman at the RAND Corporation amidst the postwar Operations Research boom in the 1950's, the principle relies on the notion that "the tail of an optimal policy is optimal" [11]. That is, for optimal policy π^*, the truncated policy $\pi_k^* = (\mu_k, \mu_{k+1}, \dots)$ must also be the optimal policy starting from node $\boldsymbol{y}_k$ at stage k. Since π^* specifies the optimal path from each node in X to $\boldsymbol{t}$, an optimal path starting from $\boldsymbol{x}_i$ that passes through $\boldsymbol{x}_j$ must also contain the full optimal path starting from $\boldsymbol{x}_j$. This notion of "tail optimality" results in a system of coupled DP equations, one for the value function at each $\boldsymbol{x}_i \in X$,

$$\begin{cases} u(\boldsymbol{x}_i) = \min_{\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i)} \{C(\boldsymbol{x}_i, \boldsymbol{x}_j) + u(\boldsymbol{x}_j)\}, & \boldsymbol{x}_i \in X \setminus \{\boldsymbol{t}\} \\ u(\boldsymbol{t}) = 0. \end{cases} \tag{1.3}$$

Several algorithmic approaches have been developed to solve the system of DP equations for the deterministic cheapest path problem quickly and efficiently. These include the "label-correcting" methods (e.g., Bellman-Ford [12, 46] and d'Esopo-Pape [78]), and the "label-setting" methods (e.g., Dijkstra's algorithm

[37], Dial's method [36] (i.e., parallelizable Dijkstra's), and the heuristic-driven A* algorithm [56]). In these methods, every node in X is assigned a label whose value is the current estimate of the value function at that node. At the start, the target's label is set to 0 (its true value), while the other nodes typically receive "tentative" labels $\tilde{u} = +\infty$. In some algorithms, such as Dijkstra's and Dial's method, the tentative labels at nodes $\boldsymbol{x}$ from which it is possible to transition to $\boldsymbol{t}$ are initialized to $\tilde{u}(\boldsymbol{x}) = C(\boldsymbol{x}, \boldsymbol{t})$. At the start of each subsequent round, a node $\boldsymbol{x}_i$ with a tentative label is selected [2] according to algorithm-specific criteria. For every node $\boldsymbol{x}_j$ for which $\boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_j)$, the method updates $\boldsymbol{x}_j$'s label according to $\tilde{u}(\boldsymbol{x}_j) = \min\{\tilde{u}(\boldsymbol{x}_j), C_{ij} + \tilde{u}(\boldsymbol{x}_i)\}$. As such, labels are progressively improved throughout the process, and the algorithm terminates once *all* of the labels can no longer be improved. Once this happens, $\tilde{u}(\boldsymbol{x}) = u(\boldsymbol{x})$ for all $\boldsymbol{x} \in X$, and the system in (1.3) has been solved.

The algorithmic structure outlined above lies at the heart of many approaches used to solve deterministic cheapest path problems arising in application areas such as robotic path planning [86] and producing turn-by-turn directions for vehicle navigation [54]. In such settings, the state spaces are typically large (e.g., nodes representing fine-grained positions within the robot's operational domain or along the highway that the vehicle is traveling on), and these algorithms can compute the optimal policy quickly and efficiently. However, neither the deterministic cheapest path problem framework nor its associated algorithms are equipped to directly handle any transitional *uncertainty* that may arise due to fluctuating traffic patterns or other changing conditions in the robot's surroundings. As a concrete example, consider an autonomous vehicle attempting to change lanes in

[2] When label-setting methods select a node, $\tilde{u} = u$ upon selection, and as a result, that node is never selected again. On the other hand, label-correcting methods simply select nodes from a specified position in a queue, and a node may be selected several times before $\tilde{u} = u$ there.

heavy traffic. At the start of the trip, suppose the vehicle's routing system solved a cheapest path problem on a road network to determine a set of turn-by-turn directions along the most fuel-efficient path to the target. In order to follow this route, the vehicle must take the next exit from the highway, but heavy traffic is making it difficult to transition to the node in the exit lane. If the vehicle is *unsuccessful* in changing lanes to take the exit, how should the route be adjusted? What happens if these lane change failures happen repeatedly?

One option to dealing with the uncertainty is to *replan* after route-altering events occur. This entails solving a new deterministic cheapest path problem starting from the location where the event occurred. Although this is a viable approach in some settings, the process of frequent replanning every time the system ends up at a state different from what the original plan required can become burdensome in highly dynamic and uncertain environments. Instead, an alternative approach is to directly account for the uncertainty in where the system will end up after initiating a transition from its current state *during* the planning process.

When the transitional uncertainty at each node $\boldsymbol{x}_i \in X$ can be described by *probability distributions* over its possible successor nodes, *stochastic shortest path problems* (SSPs), a special subset of *Markov decision processes* (MDPs), provide a suitable framework for determining the best course of action which directly takes this uncertainty into account. At each stage in an SSP, the controller selects an *action* which specifies a transition probability distribution over the possible successors and incurs a known cost. The goal then is to determine an optimal policy which leads the system to the target state with probability 1 from every state with minimum *expected* cost. Such problems are formally introduced in the early sections of Chapter 2.

The remainder of Chapter 2 is devoted to developing a broad subclass of SSPs called *opportunistically stochastic shortest path problems* (OSSPs). OSSPs have the restriction that when the controller has access to an action which results in a probabilistic transition to the possible successor nodes from their current node, they must also have access to actions which result in *deterministic* transitions to every one of those possible successor nodes as well. We present the OSSP's formal definition and show that the associated DP equations can be solved with Dijkstra's or Dial's method, provided that some simple conditions on the transition cost are satisfied at every state. After proving these results mathematically, we illustrate the OSSP's usefulness in analyzing fast numerical methods used to solve a special class of partial differential equations (PDEs) and in determining more nuanced driving instructions under lane change uncertainty.

1.2 An Overview of Continuous "Exit Time" Problems

Continuous exit time problems or *indefinite horizon problems* [3] are a natural analogue to the discrete cheapest path problems presented in the previous subsection. Here, the controller aims to minimize a cumulative cost while moving through a continuous state space to reach a closed target set, $\mathcal{T}$. To formalize this, consider a system starting from state $\boldsymbol{y}_0 = \boldsymbol{x} \in \Omega$ at time $t = 0$, where Ω is an open and bounded subset of $\mathbb{R}^n$. Depending on the specific problem at hand[4], the target set $\mathcal{T}$ might be taken to be the *boundary* of the domain $\partial\Omega$, while in others, it might be chosen to be a compact subset of Ω. The system's state at time $t > 0$

[3] These two names will be used interchangeably throughout this thesis.

[4] For simplicity, the definitions and motivating examples presented in this chapter will implicitly assume that $\mathcal{T} = \partial\Omega$. However, examples where $\mathcal{T}$ is a subset of $\partial\Omega$ or a compact subset of Ω itself are considered in Chapters 3 and 4 respectively.

is denoted by $\boldsymbol{y}(t)$, and the state evolves according to the ordinary differential equation (ODE),

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{f}(\boldsymbol{y}(t), \boldsymbol{a}(t)) \tag{1.4}$$

where $\boldsymbol{f}$ is bounded and Lipschitz continuous, $\boldsymbol{a}(t) : \mathbb{R}_{+,0} \to A$ is a measurable control function, and A is a compact set of admissible control values. The system is subject to a Lipschitz continuous running cost $K(\boldsymbol{y}(t), \boldsymbol{a}(t))$ and a (possibly nonzero) Lipschitz continuous terminal cost $\delta(\boldsymbol{y}(T))$ charged upon reaching the target. The goal is to select $\boldsymbol{a}(t)$ which minimizes the total cost incurred until the process ends at time $T := \min\{t \mid \boldsymbol{y}(t) \in \mathcal{T}\}$. The cumulative cost of following a path to $\mathcal{T}$ from $\boldsymbol{x}$ while using control function $\boldsymbol{a}(\cdot)$ is

$$\mathcal{J}(\boldsymbol{x}, \boldsymbol{a}(\cdot)) = \int_0^T K(\boldsymbol{y}(s), \boldsymbol{a}(s))ds + \delta(\boldsymbol{y}(T)). \tag{1.5}$$

The value function is defined as

$$u(\boldsymbol{x}) = \inf_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \{\mathcal{J}(\boldsymbol{x}, \boldsymbol{a}(\cdot))\} \tag{1.6}$$

where $\mathcal{A}$ is the set of admissible control functions.

An application of the dynamic programming principle to Equation (1.6) suggests that for all sufficiently small $\tau > 0$,

$$u(\boldsymbol{x}) = \inf_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \left\{ \int_0^\tau K(\boldsymbol{y}(s), \boldsymbol{a}(s))ds + u(\boldsymbol{y}(\tau)) \right\}. \tag{1.7}$$

Appropriately Taylor expanding the right hand side of Equation (1.7) and taking the limit as $\tau \to 0$ yields the following *stationary Hamilton-Jacobi-Bellman (HJB) equation* for the value function,

$$0 = \min_{\boldsymbol{a}\in A} \{K(\boldsymbol{x}, \boldsymbol{a}) + \nabla u(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a})\}, \quad \text{for } \boldsymbol{x} \in \Omega \tag{1.8}$$

with the boundary condition $u(\boldsymbol{x}) = \delta(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathcal{T}$.

When the controller's goal is to select the direction of motion at each time in order to reach $\mathcal{T}$ as quickly as possible while moving with *isotropic* speed $f(\boldsymbol{x})$, Equation (1.8) reduces to an *Eikonal equation*. The control value at each time in this setting is a unit vector $\boldsymbol{a} \in S_1$ that specifies the direction of motion, and the running cost is $K(\boldsymbol{x}, \boldsymbol{a}) = 1$. Equation (1.8) then becomes

$$0 = 1 + \min_{\boldsymbol{a} \in S_1} \{ f(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \cdot \boldsymbol{a} \} .$$

Here, the optimal direction of motion $\boldsymbol{a}_*$ *minimizes the directional derivative* of u. Thus, $\boldsymbol{a}_* = -\nabla u(\boldsymbol{x}) / |\nabla u(\boldsymbol{x})|$, and (1.8) fully simplifies to

$$|\nabla u(\boldsymbol{x})| f(\boldsymbol{x}) = 1,$$

where $|\cdot|$ is the standard Euclidean norm.

If the value function were always *smooth*, the Taylor expansion carried out on the right hand side of (1.7) to derive the stationary HJB equation would suffice to show that it is a classical solution of that PDE. However, even for the simplest problems, the value function may not be differentiable everywhere, which would make it ineligible to be a *classical* solution. As an illustrative example, consider the problem of a robot trying to determine the quickest escape path out of $\Omega = (-1, 1)$. The robot can only move left or right along the real line with constant speed $f = 1$, and the process ends immediately once it reaches one of the two endpoints. The corresponding Eikonal equation for this problem is

$$0 = 1 - |u'(x)| \tag{1.9}$$

with $u(-1) = u(1) = 0$. For any solution to Equation (1.9) to be classical, it must be continuous, satisfy both (1.9) and the boundary conditions, and it must have a continuous first derivative on the entire interval. To see that there are no classical solutions of (1.9), assume that a solution $u(x)$ is continuous and differentiable on

$(-1, 1)$. Then, the Mean Value Theorem guarantees the existence of $y \in (-1, 1)$ such that $u'(y) = 0$. This directly contradicts the requirement that $|u'(x)| = 1$ on $(-1, 1)$, so Equation (1.9) has no classical solutions.

The difficulties above can be surmounted by instead considering a suitable type of *weak* solution which satisfies the original PDE almost everywhere. Since it is straightforward to show that the value function is *Lipschitz continuous* under our assumptions about the Lipschitz continuity of K and δ [8], it may seem that merely requiring the PDE's solution to be Lipschitz continuous and satisfy the PDE at all points where the solution is differentiable would be sufficient. Under these requirements, such a solution to the example problem posed in Equation (1.9) should be 0 at $x = \pm 1$, have a measure-zero set of points where it is not differentiable (Rademacher's Theorem), and have a derivative equal to ± 1 at all other points. Clearly, the function $u(x) = 1 - |x|$ is a viable candidate. But so is *any* piecewise continuous affine function with $u(-1) = u(1) = 0$ and whose line segments also all have a slope of ± 1. As an illustrative example, $u(x) = 1 - |x|$ and another suitable piecewise affine function $q(x)$ are shown in Figure 1.1. With so many possibilities, how do we know which of these Lipschitz continuous functions actually corresponds to the value function of the optimal control problem we are trying to solve?

In the 1980's, Crandall and Lions pioneered a solution mechanism for PDEs of the Hamilton-Jacobi type that answers the "which one" question posed in the paragraph above. They proposed a new type of weak solution called a *viscosity solution* which admits both existence and uniqueness theorems, *and* they showed that the value function of a continuous optimal control problem is the viscosity solution of the corresponding HJB PDE [31].

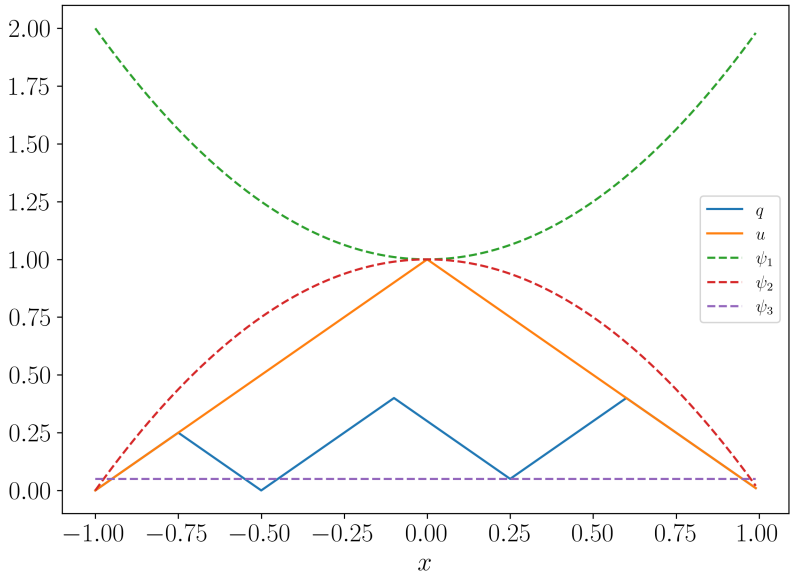


Figure 1.1: Plots of $u(x) = 1 - |x|$, a piecewise affine function $q(x)$ that satisfies Equation (1.9) in a weak sense based on Lipschitz continuity, and three infinitely differentiable test functions ψ_1, ψ_2, and ψ_3.

Following notation used in standard references (e.g., [7, 40]), we define the *Hamiltonian* as

$$H(\boldsymbol{x}, p) := \min_{\boldsymbol{a} \in A} \{K(\boldsymbol{x}, \boldsymbol{a}) + \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}) \cdot p\} \tag{1.10}$$

where H is assumed to be concave in p[5], continuous, and real-valued on $\Omega \times \mathbb{R}^N$. In the special case of the Eikonal equation encoding minimum time-to-target optimal control problems, $H = 1 - f(\boldsymbol{x})|\nabla u(\boldsymbol{x})|$.

Definition 1 (Continuous Viscosity Solution - Indefinite Horizon)**.** *Suppose that the value function $u(\boldsymbol{x})$ is continuous but not necessarily differentiable throughout Ω.*

1. u is a viscosity subsolution of the stationary HJB equation if for every test

[5]Many standard references actually define $H(\boldsymbol{x}, p) := \max_{\boldsymbol{a} \in A} \{-K(\boldsymbol{x}, \boldsymbol{a}) - \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}) \cdot p\}$ which is convex in p. However, this version is equivalent to negating (1.10), and the HJB equation remains unchanged.

function $\psi(\boldsymbol{x}) \in \mathcal{C}^\infty(\Omega)$ *such that* $u - \psi$ *has a local maximum at* $\boldsymbol{x}_0 \in \Omega$*, the following holds:*

$$H(\boldsymbol{x}_0, \nabla\psi) \geq 0.$$

2. u *is a viscosity supersolution of the stationary HJB equation if for every test function* $\psi(\boldsymbol{x}) \in \mathcal{C}^\infty(\Omega)$ *such that* $u - \psi$ *has a local minimum at* $\boldsymbol{x}_0 \in \Omega$*, the following holds:*

$$H(\boldsymbol{x}_0, \nabla\psi) \leq 0.$$

If both 1 and 2 simultaneously hold, then u *is a continuous viscosity solution of the stationary HJB PDE given in* (1.8).

We remark here that the definition above is presented *without* boundary conditions. When applicable, boundary conditions are enforced in the *viscosity sense*. For technical simplicity, we omit a detailed discussion of this here; Chapter 5, Section 4 of [7] for more information.

It is straightforward to check using Definition 1 that $u(x) = 1 - |x|$ is a viscosity solution of the Eikonal equation, while $q(x)$ shown in Figure 1.1 is not. For $u(x)$, we notice that it is not possible to find a test function $\psi(x)$ such that $u(0) = \psi(0) = 1$ while $u - \psi$ has a local minimum at $x_0 = 0$. To see this, take $\psi(x) = 1$. Even though this simple horizontal line touches the graph of u at $x = 0$, $\psi(x)$ will still be *larger* than $u(x)$ at all $x \neq 0$. Since there is no $\psi \in \mathcal{C}^\infty(\Omega)$ such that $u - \psi$ has a local minimum at 0, $u(x) = 1 - |x|$ is trivially a viscosity supersolution of (1.9). To check that u is a viscosity subsolution, pick any $\psi \in \mathcal{C}^\infty(\Omega)$ such that $u(0) = \psi(0) = 1$. Regardless if ψ is locally convex or concave, $u - \psi$ will have a local maximum at $x_0 = 0$, since $\psi(x) \geq u(x)$ around that point. Plugging ψ' into

the subsolution inequality requires

$$1 + |\psi'(x)| \geq 0,$$

which holds at any value of $\psi'(x)$. The boundary requirement is trivially satisfied since $u(x) - 0 = 0$.

On the other hand, $q(x)$ *cannot* be a viscosity solution of Equation (1.9), since the supersolution inequality condition on H *fails* for horizontal line test functions at any of the local minima of q. An example of this is depicted by the dashed horizontal line ψ_3 in Figure 1.1 for the local minimum at $x = 0.25$. To this end, it is straightforward to see that *any* of the other piecewise affine candidate functions with local minima in $(-1, 1)$ will fail the supersolution test, and $u(x) = 1 - |x|$ is the unique viscosity solution. Hence, $u(x) = 1 - |x|$ is the value function of the original optimal control problem. Formal theorems establishing uniqueness of the viscosity solution for the general stationary HJB equation are proved in [31].

In many real-world applications, the value function may actually be discontinuous. In Chapter 5 of [7], Bardi and Capuzzo-Dolcetta present an extension upon the continuous framework by showing that the upper and lower semicontinuous envelopes of u are indeed viscosity super and subsolutions (respectively) of the HJB equation. Let $\boldsymbol{y} \in \mathbb{R}^N$. The *upper semicontinuous envelope* of u is $\overline{u}(\boldsymbol{x}) := \limsup_{\boldsymbol{y} \to \boldsymbol{x}} u(\boldsymbol{y})$, and the *lower semicontinuous envelope* of u is $\underline{u}(\boldsymbol{x}) := \liminf_{\boldsymbol{y} \to \boldsymbol{x}} u(\boldsymbol{y})$.

Definition 2 (Discontinuous Viscosity Solution - Indefinite Horizon)**.** *Suppose that the value function $u(\boldsymbol{x})$ is locally bounded. If $\overline{u}$ is a viscosity supersolution of* (1.8) *by Definition 1, and $\underline{u}$ is a viscosity subsolution of* (1.8) *by Definition 1, then u is a discontinuous viscosity solution of* (1.8)*.*

Until now, we have formulated continuous exit time problems in a fully *deterministic* sense. But what happens if a potentially plan-altering event occurs nearby *before* reaching the target set $\mathcal{T}$? For example, pedestrians arriving at a crosswalk at some rate may force the driver of an oncoming vehicle to come to an unplanned stop, thereby adding extra time to the trip duration. Or, a predator stalking prey along a well-known trajectory to their feeding location may have to abruptly change their course if the prey gets spooked and decides to run away from the predator. In situations where the time at which a (potentially) disruptive event occurs can be modeled as the result of a Poisson point process with rate λ, it is possible to formulate a stochastic version of the continuous indefinite horizon problem which directly takes into account the probability of disruption in the planning process itself.

In [5], Andrews and Vladimirsky introduce a mathematical framework for minimizing the *expected cumulative cost* incurred until the process terminates at time $\tilde{T}$ (i.e., the time of disruption) or upon reaching $\mathcal{T}$, whichever comes first. They specifically consider cases where the termination time follows an *exponential* distribution (i.e., $P(\tilde{T} \geq t) = e^{-\lambda t}$), and show that for an undiscounted indefinite horizon problem, the goal is to find a control $\boldsymbol{a} \in \mathcal{A}$ which satisfies

$$u(\boldsymbol{x}) = \inf_{\boldsymbol{a}\in\mathcal{A}} \mathbb{E}_{\tilde{T}\sim \mathrm{Exp}(\lambda)} \left\{ \int_0^{\tilde{T}} K(\boldsymbol{y}(s), \boldsymbol{a}(s)) ds + \delta(\boldsymbol{y}(\tilde{T})) \right\}. \tag{1.11}$$

For small $\tau > 0$, the value function satisfies the dynamic programming principle, and standard Taylor expansion arguments can then be used to derive the following stationary PDE,

$$0 = \min_{\boldsymbol{a}\in A} \{K(\boldsymbol{x}, \boldsymbol{a}) + \nabla u(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a})\} + \lambda(\delta(\boldsymbol{x}) - u(\boldsymbol{x})). \tag{1.12}$$

Although λ is assumed to be constant in their original analysis, this approach is suitable for time and state dependent rates as well. The stationary HJB encoding

the control problem in such cases is of the same form as Equation (1.12), with the state and / or time dependence included in λ as appropriate (see e.g., [51, 50] for additional examples).

In Chapter 4, we apply this framework to determine energy-optimal flight trajectories for a pursuer attempting to capture an evader whose escape attempt time $\tilde{T}$ occurs as a result of a non-homogeneous Poisson point process. The rate function $\lambda(\boldsymbol{z}, t)$, is *both* state and time dependent as $\boldsymbol{z}$ is the pursuer's position at time t. The relevant HJB equations, the details of the numerical methods implemented to solve the PDEs, and simulation results are discussed in that portion of the thesis.

1.3 An Overview of Continuous Finite Horizon Problems

We conclude our survey of deterministic optimal control with a discussion of continuous finite horizon problems. The setup of such problems is very similar to the continuous exit time setting, but now, the planner aims to minimize the cumulative cost up to a specified *time*, T. The cumulative cost incurred during the time period $[t, T]$ while using control $\boldsymbol{a}(t)$ is

$$\mathcal{J}(\boldsymbol{x}, t, \boldsymbol{a}(\cdot)) = \int_t^T K(\boldsymbol{y}(s), \boldsymbol{a}(s))ds + \delta(\boldsymbol{y}(T)), \tag{1.13}$$

where just as before, $\delta(\boldsymbol{y}(T))$ is a (possibly) nonzero penalty paid at the terminal time. Thus, the value function is defined as

$$u(\boldsymbol{x}, t) = \inf_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \left\{\mathcal{J}(\boldsymbol{x}, t, \boldsymbol{a}(\cdot))\right\}, \tag{1.14}$$

where $\boldsymbol{x}$ is the starting state at time t and $\mathcal{A}$ is the set of admissible control functions. For small $\tau > 0$, the value function adheres to the dynamic programming

principle

$$u(\boldsymbol{x},t) = \inf_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \left\{ \int_t^{t+\tau} K(\boldsymbol{y}(s),\boldsymbol{a}(s))ds + u(\boldsymbol{y}(t+\tau),t+\tau) \right\}. \tag{1.15}$$

Taylor expanding the right hand side of (1.15) yields the governing *time-dependent HJB equation*,

$$0 = u_t + \min_{\boldsymbol{a}\in A} \left\{ K(\boldsymbol{x},\boldsymbol{a}) + \nabla u(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{a}) \right\}, \tag{1.16}$$

with the terminal condition $u(\boldsymbol{x},T) = \delta(\boldsymbol{x})$.

As is the case for the continuous exit time problems, the value function may not be differentiable everywhere in $\Omega \times [0,T]$. The value function has been shown to be a time-dependent viscosity solution of (1.16) [7]. It is straightforward to extend the definitions of continuous and discontinuous viscosity solutions from the stationary case to the time-dependent scenarios. Using the Hamiltonian[6] defined in (1.10), we define the time-dependent viscosity solution in the following sense.

Definition 3 (Continuous Viscosity Solution - Finite Horizon)**.** *Suppose that the value function $u(\boldsymbol{x},t)$ is continuous but not necessarily differentiable throughout $\Omega \times \mathbb{R}_{0,+}$.*

1. *u is a viscosity subsolution of the time-dependent HJB equation if for any test function $\psi(\boldsymbol{x},t) \in \mathcal{C}^\infty(\Omega \times \mathbb{R}_{0,+})$ such that $u - \psi$ has a local maximum at $(\boldsymbol{x}_0,t_0) \in \Omega \times [0,T]$, the following holds:*

$$\psi_t + H(\boldsymbol{x}_0, \nabla\psi) \geq 0.$$

2. *u is a viscosity supersolution of the time-dependent HJB equation if for any test function $\psi(\boldsymbol{x},t) \in \mathcal{C}^\infty(\Omega \times \mathbb{R}_{0,+})$ such that $u - \psi$ has a local minimum*

[6] In the time-dependent setting, the Hamiltonian is usually defined as a function of $\boldsymbol{x}$, t, and p. This is done when K and f are time-dependent. However, none of the finite horizon problems considered in this thesis have time-dependent cost or dynamics, so we omit the t-dependence in our Hamiltonian here.

at $(\boldsymbol{x}_0, t_0) \in \Omega \times [0, T]$*, the following holds:*

$$\psi_t + H(\boldsymbol{x}_0, \nabla\psi) \leq 0.$$

If both 1 and 2 simultaneously hold, then u *is a continuous viscosity solution of the time-dependent HJB PDE given in* (1.16).

In the time-dependent setting, the *upper semicontinuous envelope* of u is $\overline{u}(\boldsymbol{x}, t) := \limsup_{(\boldsymbol{y},t)\to(\boldsymbol{x},t)} u(\boldsymbol{y}, t)$, and the *lower semicontinuous envelope* of u is $\underline{u}(\boldsymbol{x}, t) := \liminf_{(\boldsymbol{y},t)\to(\boldsymbol{x},t)} u(\boldsymbol{y}, t)$. The discontinuous viscosity solution of the time-dependent HJB equation is then defined as follows.

Definition 4 (Discontinuous Viscosity Solution - Finite Horizon)**.** *Suppose that the value function* $u(\boldsymbol{x}, t)$ *is a locally bounded function from* $\Omega \times \mathbb{R}_{0,+}$ *to* $\mathbb{R}$*. If* $\overline{u}(\boldsymbol{x}, t)$ *is a viscosity supersolution of* (1.8) *by Definition 3, and* $\underline{u}(\boldsymbol{x}, t)$ *is a viscosity subsolution of* (1.8) *by Definition 3, then* u *is a discontinuous viscosity solution to the time dependent HJB equation in* (1.8).

In the real world, the termination time T may not be known with certainty at the beginning of the process, but instead, it will be revealed later on. In this case, T can be treated as a random variable which follows a probability distribution that reflects the likelihood of a specific plan-disrupting event occurring, or it could reflect the amount of information that the controller has about the possible values for T at the present time. In the former setting, if the termination event occurs as the result of a non-homogeneous Poisson point process with a time and control-dependent rate, the value function will be time-dependent. It can be shown that the value function satisfies a time-dependent version of the HJB equation given in (1.12). We present an example of this case in Chapter 4 as it arises in determining

energy-optimal flight trajectories for a pursuer trying to capture an evader who may be spooked and attempt to escape at an unknown time T.

In the latter setting, T may not be known with certainty at the beginning but the controller knows a probability distribution over the possible termination times. Each instance that termination does *not* happen at one of the possible times, the controller gains new insight into which of the possibilities is the true value of T, and the probability distribution is updated accordingly. This approach was first studied by Qi et al in [83] for discretely distributed T and exponentially distributed T. In the discretely distributed case, they showed that the value function can be determined by solving a *sequence* of time-dependent HJB PDEs defined in the time period between successive pairs of possible T values.

In Chapter 3, we illustrate a novel application of this case by considering the problem of a driver trying to optimize their route to a destination through a signalized intersection whose green light duration is unknown. We assume that the driver faces a steady green light at the start of the planning process and only knows a *discrete* distribution over possible times at which it will turn yellow, $\tilde{T}_Y$. The driver balances multiple objectives (e.g., fuel use, time to destination, and comfort) while aiming to select an optimal braking / acceleration policy that yields the minimum expected cost to their destination with respect to the discrete distribution on $\tilde{T}_Y$. We show that the optimal driving policy can be recovered by solving the appropriate *sequence* of time-dependent HJB equations that hold in between each of the possible values of $\tilde{T}_Y$.

Thesis Chapters and their Associated Publications

Chapter 2 is based on a paper [49] co-authored with Alex Vladimirsky which was accepted for publications in INFORMS Mathematics of Operations Research (MOR) in March 2025. Chapter 3 is based on a paper [48] co-authored with Alex Vladimirsky which was published in IFAC-PapersOnline in September 2022. Chapter 4 is based on the paper [47] published in IEEE Control Systems Letters in November 2024, although we note that the material discussed in the thesis chapter presents significant modifications to the original journal version.

CHAPTER 2

MONOTONE CAUSALITY IN OPPORTUNISTICALLY STOCHASTIC SHORTEST PATH PROBLEMS

Sequential decision making under uncertainty is ubiquitous in science and engineering. If the duration of the process is not known in advance and the termination results from entering a specific part of state space, this is often modeled as a Stochastic Shortest Path (SSP) problem [18]. SSPs arise naturally in diverse applications including aircraft routing [58] and optimal scheduling in wireless sensor networks [27]. Unfortunately, solving most SSPs is computationally expensive and is usually accomplished iteratively. In this work, we introduce a subclass of SSPs called *Opportunistically Stochastic Shortest Path* (OSSP) problems, in which one can always use deterministic transitions at each stage, but opting for stochastic transitions can often yield a lower cumulative cost. We derive a number of sufficient conditions on the OSSP's transition cost function to guarantee the applicability of non-iterative label-setting methods. Our conditions cover problems with finite and infinite action spaces, make no assumptions about the smoothness of transition cost functions, and are naturally satisfied by many OSSPs arising in very different applications (e.g., autonomous vehicle routing models and discretizations of partial differential equations in optimal control theory).

The state space of an SSP is usually identified with nodes of a finite graph, with a separate set of actions available at each of them. Each chosen action incurs an instantaneous cost and yields a probability distribution over possible successor nodes. The goal is to choose actions that minimize the expected total cost accumulated by the time we reach the pre-specified target. The *value function* is defined by minimizing the expected cumulative cost starting from each of the n nodes, and it

can be recovered by solving a system of n coupled non-linear optimality equations. But unlike in the case of fixed-horizon Markov Decision Processes, for general SSPs the direction of information flow is not a priori obvious (i.e., the dependence of the nodes/equations usually does not have an obvious tree-like structure). As a result, iterative methods are generally unavoidable in computing the SSP value function. This is expensive even in "causal" SSPs, for which the convergence is obtained in finitely many iterations: even if we assume that the optimal action at any single node can be found in $O(1)$ operations once the value function is already known at all possible successor nodes, the iterative process yields the overall computational cost of $O(n^2)$. In deterministic shortest path problems with non-negative transition costs and bounded node outdegrees, such iterations are avoided by using more efficient *label-setting* algorithms: most notably, Dijkstra's [37] and Dial's [36], achieving the computational cost of $O(n \log n)$ and $O(n)$ respectively. It is thus natural to consider whether these non-iterative methods are applicable to any broad class of SSPs as well. The previous attempts to answer this question yielded either implicit conditions that could be only verified a posteriori [14] or a priori verifiable sufficient conditions that were fairly restrictive (e.g., relying on a very rich set of available actions and smoothness of the cost function) [105]. The OSSPs studied in this paper significantly broaden this class, with simple explicit criteria for "monotone causality" (the property that ensures the applicability of label-setting) and far fewer assumptions on the problem data.

We begin §2.1 with a quick introduction to general SSPs (and the "pruning process", which allows reducing their action sets without altering the value function) followed by a formal definition of OSSPs. For the latter, the sufficient conditions for applicability of Dijkstra's and Dial's methods are then developed in §2.2. We then proceed to demonstrate the usefulness of OSSPs in analyzing the

computational (causal) properties of semi-Lagrangian discretizations of stationary Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDEs) with different types of anisotropy. The fact that many semi-Lagrangian PDE discretizations can be re-interpreted as SSPs is well-known [67]. But our analysis in §2.3 yields (a) a simplification and geometric interpretation of prior disparate criteria for applicability of Dijkstra's method and (b) new sufficient conditions for the applicability of Dial's method to HJB discretizations.

OSSPs also provide a particularly attractive modeling framework for discrete problems in which the key choice is between maintaining the status quo or pushing (with some degree of urgency) to enact a specific change. This corresponds to OSSPs in which every action yields at most two possible successor nodes. One very natural example of this is a decision on whether (and when) to change lanes while driving. In §2.4, we show how OSSPs can be used in autonomous vehicle routing problems. The standard driving directions provide turn-by-turn instructions for reaching the target. But our OSSP model also answers finer-scale tactical questions based on heterogeneous traffic conditions: where, when, and with what urgency should a car try to change lanes along the way? And what should be done if an attempted lane change fails? The success of lane change maneuvers is inherently uncertain, and failure to take this into account during the planning process may result in sub-optimal performance or even an unsafe driving experience. Our work in that section provides an alternative to a model recently introduced by Jones, Haas-Heger, and van den Berg [62]. That earlier paper has showed the applicability of Dijkstra's method to a specific SSP-based routing model that allowed "tentative" and "forced" lane change maneuvers. The modeling framework in §2.4 goes much farther, allowing for a continuous spectrum of lane change urgency and a variety of cost functions treatable by label-setting methods.

We conclude by discussing the current limitations of the OSSP framework and possible directions for future work in §4.5.

2.1 SSPs and OSSPs

2.1.1 General Stochastic Shortest Path Problems

Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}_{n+1} = \boldsymbol{t}\}$ be the set of all states (nodes), where the last one encodes the target (destination). A sequence $(\boldsymbol{y}_k)_{k=0,1,\ldots}$ reflects a possible stochastic path, where each $\boldsymbol{y}_k \in X$ denotes the state of the process at the k-th stage (i.e., after k transitions). We use $A(\boldsymbol{x}_i)$ to denote the set of actions available at a state $\boldsymbol{x}_i \in X$ and $A = \bigcup_{\boldsymbol{x} \in X} A(\boldsymbol{x})$ to denote the set of all actions. Choosing any action $\boldsymbol{a} \in A(\boldsymbol{x}_i)$ incurs a known cost $C(\boldsymbol{x}_i, \boldsymbol{a})$ and yields a known probability distribution $p(\boldsymbol{x}_i, \boldsymbol{a}, \boldsymbol{x}_j) = p_{ij}(\boldsymbol{a})$ over the possible successor nodes for the next transition; i.e., if an action $\boldsymbol{a} \in A(\boldsymbol{y}_k)$ is chosen at the k-th stage of the process, then $\mathbb{P}(\boldsymbol{y}_{k+1} = \boldsymbol{x}_j) = p(\boldsymbol{y}_k, \boldsymbol{a}, \boldsymbol{x}_j)$ for all $\boldsymbol{x}_j \in X$.

The target node is assumed to be *absorbing* and no additional cost is incurred after we reach it; i.e., $p(\boldsymbol{t}, \boldsymbol{a}, \boldsymbol{t}) = 1$ and $C(\boldsymbol{t}, \boldsymbol{a}) = 0$ for all $\boldsymbol{a} \in A(\boldsymbol{t})$. The decision maker's goal is to choose actions to minimize the expected cumulative cost up to the target. This is done by optimizing over the set of control policies. A function $\mu : X \to A$ is a *control mapping* if $\mu(\boldsymbol{x}) \in A(\boldsymbol{x})$ for all $\boldsymbol{x} \in X$. A *control policy* is an infinite sequence of control mappings $\pi = (\mu_k)_{k=0,1,\ldots}$ to be used at respective stages of the process. The expected cost of using a policy π starting from any node

$\boldsymbol{y}_0 = \boldsymbol{x}$ is defined as

$$\mathcal{J}(\boldsymbol{x}, \pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} C\left(\boldsymbol{y}_k, \mu_k(\boldsymbol{y}_k)\right) \mid \boldsymbol{y}_0 = \boldsymbol{x}\right], \tag{2.1}$$

and the value function $U(\boldsymbol{x})$ denotes the result of minimizing the expected cost starting from $\boldsymbol{x}$:

$$U(\boldsymbol{x}) = \inf_{\pi} \mathcal{J}(\boldsymbol{x}, \pi). \tag{2.2}$$

With a slight abuse of notation, we also use the symbol μ to refer to a *stationary* policy $(\mu, \mu, \dots)$ that uses the same control mapping μ at each stage of the process.

The theory of SSPs was developed to describe a broad class of models, including those with infinite and even continuous action spaces. As shown in [18], the following five assumptions[1] guarantee that the infimum in (2.2) is actually attained and that there exists an optimal stationary policy μ^* which attains this minimum for every starting state $\boldsymbol{x} \in X$:

(A1) $A(\boldsymbol{x}_i)$ is compact for all $\boldsymbol{x}_i \in X$.

(A2) $C(\boldsymbol{x}_i, \boldsymbol{a})$ is lower semicontinuous over $A(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i \in X$.

(A3) $p_{ij}(\boldsymbol{a})$ is continuous over $A(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in X.

(A4) There exists at least one policy which results in arriving at the target with probability 1 from every starting state. We refer to policies of this type as *proper* and all others as *improper*.

(A5) Every improper policy yields an infinite expected cost starting from at least one node.

Throughout the paper, we will also impose a stronger cost-positivity assumption

[1] If the set of available actions is finite, the existence of an optimal stationary policy can be demonstrated without these additional assumptions [44].

(**A5'**) $C(\boldsymbol{x}, \boldsymbol{a}) > 0, \qquad \forall \boldsymbol{x} \in X \setminus \{\boldsymbol{t}\}, \boldsymbol{a} \in A(\boldsymbol{x})$.

This is a necessary condition for the applicability of label-setting methods, which will be discussed in detail in §2.2. We note that (**A1**) and (**A2**) make (**A5**) a trivial consequence of (**A5'**). If C is also bounded from above, the existence of a proper stationary policy implies that $U(\boldsymbol{x})$ is finite $\forall \boldsymbol{x} \in X$.

For simplicity of exposition, we will further assume the lack of self-transitions on $X \setminus \{\boldsymbol{t}\}$; i.e.,

(A6) $p(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{x}) = 0, \qquad \forall \boldsymbol{x} \in X \setminus \{\boldsymbol{t}\}, \boldsymbol{a} \in A(\boldsymbol{x})$.

This is not really restrictive since self-transitions can be removed without affecting the value function. Indeed, whenever $p_{ii}(\boldsymbol{a}) > 0$ for $\boldsymbol{x}_i \neq \boldsymbol{t}$, one can obtain an SSP with the same value function by switching to

$$\tilde{C}(\boldsymbol{x}_i, \boldsymbol{a}) = \frac{C(\boldsymbol{x}_i, \boldsymbol{a})}{1 - p_{ii}(\boldsymbol{a})}; \qquad \text{and} \qquad \tilde{p}_{ij}(\boldsymbol{a}) = \begin{cases} 0 & \text{if } j = i, \\ \frac{p_{ij}(\boldsymbol{a})}{1 - p_{ii}(\boldsymbol{a})} & \text{if } j \neq i; \end{cases} \qquad \text{for } j = 1, \cdots, n{+}1.$$

Replacing $p(\boldsymbol{a})$ by $\tilde{p}(\boldsymbol{a})$ is an example of "oblique projections," in which one of the possible successor nodes is excluded and its probability is redistributed among other successor nodes proportionally. We will later employ a similar approach in proving causal properties of SSP problems.

For notational ease, denote $U(\boldsymbol{x}_i)$ as U_i and $A(\boldsymbol{x}_i)$ as A_i. The value function can be computed by solving a coupled system of dynamic programming equations

$$U_i = \min_{\boldsymbol{a} \in A_i} \left\{ C(\boldsymbol{x}_i, \boldsymbol{a}) + \sum_{j=1}^{n+1} p_{ij}(\boldsymbol{a}) U_j \right\}, \qquad \text{for } i = 1, \ldots n;$$

$$U_{n+1} = 0, \tag{2.3}$$

and the optimal control mapping μ^* can be obtained by using any action from the argmin in (2.3). In vector notation, $\boldsymbol{U} = \begin{bmatrix} U_1, & \ldots, & U_n \end{bmatrix}^T$ is a fixed point of the operator $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ defined componentwise as

$$(\mathcal{T}\boldsymbol{W})_i = \min_{\boldsymbol{a} \in A_i} \mathcal{F}_i(\boldsymbol{a}, \boldsymbol{W}), \quad \text{where } \mathcal{F}_i(\boldsymbol{a}, \boldsymbol{W}) = C(\boldsymbol{x}_i, \boldsymbol{a}) + \sum_{j=1}^{n} p_{ij}(\boldsymbol{a}) W_j. \quad (2.4)$$

This $\mathcal{F}_i(\boldsymbol{a}, \boldsymbol{W})$ represents the expected cost-to-go from $\boldsymbol{x}_i$ when using $\boldsymbol{a} \in A_i$ for the first transition and under the assumption that the remaining expected cost-to-go is encoded in $\boldsymbol{W} \in \mathbb{R}^n$.

It is often useful to consider "stochastic connectivity" of various nodes in an SSP. Focusing on any specific $\boldsymbol{x}_i$, we define the set of potential successor nodes $\mathcal{I}(\boldsymbol{a}) = \{\boldsymbol{x}_j \in X \mid p_{ij}(\boldsymbol{a}) > 0\}$ for each action $\boldsymbol{a} \in A_i$. The set of *all* possible successor nodes from $\boldsymbol{x}_i$ is then defined as $\mathcal{N}(\boldsymbol{x}_i) = \cup_{\boldsymbol{a} \in A_i} \mathcal{I}(\boldsymbol{a}) = \{\boldsymbol{x}_j \in X \mid \exists \boldsymbol{a} \in A_i \text{ s.t. } p_{ij}(\boldsymbol{a}) > 0\}$. The SSP's *transition digraph* G can then be defined as a network representation of all possible transitions to successor nodes from each $\boldsymbol{x}_i \in X$ under all available actions in A_i. We provide two simple examples in Figure 2.1, with each possible $\boldsymbol{x}_i \to \boldsymbol{x}_j$ transition indicated by a dashed arrow.

Since this perspective aggregates transitions possible under *all* available actions, a transition digraph captures only the global (topological) structure of the SSP, which is typically insufficient to infer the optimal policy. But the practicality of a computational method based on fixed-point iterations to determine $\boldsymbol{U}$ depends heavily on whether G is acyclic (as in Figure 2.1(a)). Even for large acyclic SSPs, it is easy to compute $\boldsymbol{U}$ efficiently via an iterative process. But if G contains cycles (as in Figure 2.1(b)), the number of iterations can be prohibitive, particularly when n is large. In such cases, faster non-iterative algorithms described in §2.2 provide an attractive alternative, whenever they are applicable. Unfortunately, their applicability to each specific SSP problem can be hard to verify a priori. In

this paper, we show how this can be done by checking simple, explicit conditions on $C(\boldsymbol{x}, \boldsymbol{a})$ for a broad subclass of SSPs defined in §2.1.3.

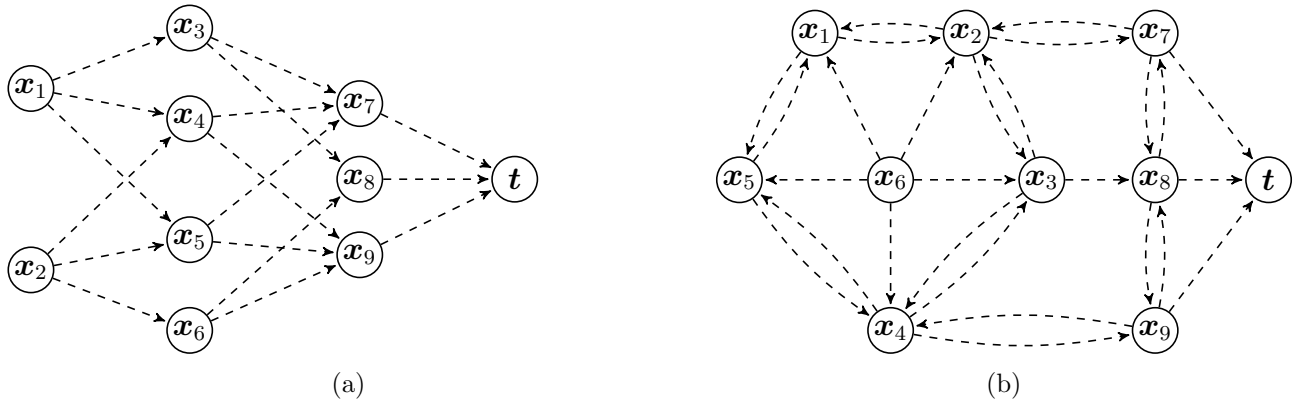


Figure 2.1: Transition digraphs for two simple SSP examples. The target node is marked by $\boldsymbol{t}$, and the possibility of a transition from node $\boldsymbol{x}_i$ to a successor node $\boldsymbol{x}_j$ is indicated by a dashed arrow.

2.1.2 Randomized Policies, Dominated Actions, and Convexified Cost

The above definition of control policies selects state-dependent actions deterministically. It is well known that extending SSPs to *randomized control policies* (which select actions probabilistically) does not lower the value function. Nevertheless, this generalization is very useful in proving the properties of OSSPs in §2.2. In addition, it can be also used to justify a reduction of the action space in general SSP problems.

In this section, we will refer to all actions available in the original SSP as "pure" to distinguish them from the "relaxed"/mixed actions allowed here. Suppose the set of pure actions A_i is finite and $\lambda = (\lambda_1, \ldots, \lambda_{|A_i|})$ is some probability distribution over it. Preparing for a transition from $\boldsymbol{y}_k = \boldsymbol{x}_i$, one can implement a mixed

(or *relaxed*) action $\boldsymbol{a}_\lambda$ by selecting each $\boldsymbol{a}_r \in A_i$ with probability λ_r. This requires extending the definition in (2.4) to use

$$\mathcal{F}_i(\boldsymbol{a}_\lambda, \boldsymbol{W}) = \sum_{r=1}^{|A_i|} \lambda_r \mathcal{F}_i(\boldsymbol{a}_r, \boldsymbol{W})$$

and minimizing over all such λ's when computing $(\mathcal{T}\boldsymbol{W})_i$. Of course, this also includes all pure actions since λ might be prescribing a single $\boldsymbol{a}_r \in A_i$ with probability one. We will refer to the problem based on such relaxed actions as rSSP. But it's easy to see that the value function of this rSSP is actually the same as in the original SSP since

$$\boldsymbol{a}_* \in \operatorname*{argmin}_{\boldsymbol{a} \in A_i} \mathcal{F}_i(\boldsymbol{a}, \boldsymbol{W}) \qquad \Longrightarrow \qquad \mathcal{F}_i(\boldsymbol{a}_*, \boldsymbol{W}) \leq \mathcal{F}_i(\boldsymbol{a}_\lambda, \boldsymbol{W}) \tag{2.5}$$

for all probability distributions λ. Thus, there exists an optimal stationary policy for rSSP that uses only pure actions. (The same conclusions also hold for a general compact A_i, except that λ would need to be a probability measure.)

We will say that two pure actions $\boldsymbol{a}, \tilde{\boldsymbol{a}} \in A_i$ are *transition-equivalent* if $p_{ij}(\boldsymbol{a}) = p_{ij}(\tilde{\boldsymbol{a}})$. Clearly, if they are transition-equivalent and $C(\boldsymbol{x}_i, \boldsymbol{a}) \geq C(\boldsymbol{x}_i, \tilde{\boldsymbol{a}})$, then $\boldsymbol{a}$ can be removed from A_i without affecting the value function. This pruning idea can also be extended to use relaxed actions.

Definition 5 (Transition-equivalent Relaxed Actions)**.** *Given a set of pure actions* $\{\boldsymbol{a}_1, ..., \boldsymbol{a}_l\} \subset A_i$ *and a probability distribution over them* $\lambda = (\lambda_1, \ldots, \lambda_l)$*, we say that the corresponding relaxed action* $\boldsymbol{a}_\lambda$ *is* transition-equivalent *to a pure action* $\boldsymbol{a} \in A_i$ *if*

$$p_{ij}(\boldsymbol{a}) \;=\; p_{ij}(\boldsymbol{a}_\lambda) \;=\; \sum_{r=1}^{l} \lambda_r p_{ij}(\boldsymbol{a}_r), \qquad \forall \boldsymbol{x}_j \in X. \tag{2.6}$$

We will also use $\Lambda(\boldsymbol{a})$ *to denote the set of all relaxed actions that are transition-equivalent to* $\boldsymbol{a}$*.*

Definition 6 (Dominated, Replaceable, and Useful Actions)**.** *An action* $\boldsymbol{a} \in A_i$ *is* probabilistically dominated *if there exists a transition-equivalent relaxed action* $\boldsymbol{a}_\lambda \in \Lambda(\boldsymbol{a})$ *such that the resulting expected cost of the first transition is lower with* $\boldsymbol{a}_\lambda$*; i.e.,*

$$C(\boldsymbol{x}_i, \boldsymbol{a}) \; > \; C(\boldsymbol{x}_i, \boldsymbol{a}_\lambda) \; = \; \sum_{r=1}^{l} \lambda_r C(\boldsymbol{x}_i, \boldsymbol{a}_r). \tag{2.7}$$

We will say that an action $\boldsymbol{a} \in A_i$ *is* replaceable *if there exists a set of actions* $\{\boldsymbol{a}_1, ..., \boldsymbol{a}_l\} \subset A_i$ *and a probability distribution over them* $\lambda = (\lambda_1, \ldots, \lambda_l)$ *such that*

- *none of the* $\boldsymbol{a}_r$*'s are individually transition-equivalent to* $\boldsymbol{a}$ *but* $\boldsymbol{a}_\lambda \in \Lambda(\boldsymbol{a})$;
- *the expected transition cost is the same; i.e.,* $C(\boldsymbol{x}_i, \boldsymbol{a}) \; = \; C(\boldsymbol{x}_i, \boldsymbol{a}_\lambda) \; = \; \sum_{r=1}^{l} \lambda_r C(\boldsymbol{x}_i, \boldsymbol{a}_r)$.

We will say that a pure action $\boldsymbol{a} \in A_i$ *is (potentially)* useful *in an SSP if it is not probabilistically dominated or replaceable, with* A_i^u *denoting the set of all such actions. In view of* (2.5)*, whenever* $\boldsymbol{a} \in A_i$ *is probabilistically dominated or replaceable, there will always exist another* $\tilde{\boldsymbol{a}} \in A_i$ *which is at least as good of a choice at* $\boldsymbol{x}_i$*.*

Observation 1. *Suppose* $\boldsymbol{a} \in A_i$ *is not useful. Then, for every* $\boldsymbol{W} \in \mathbb{R}^n$*, there exists another action* $\tilde{\boldsymbol{a}} \in A_i \backslash \{\boldsymbol{a}\}$ *such that* $\mathcal{F}_i(\boldsymbol{a}, \boldsymbol{W}) \geq \mathcal{F}_i(\tilde{\boldsymbol{a}}, \boldsymbol{W})$*. So, removing this* $\boldsymbol{a}$ *from* A_i *will not affect* U_i *regardless of the* U_j *values at any potential successor nodes. The SSP based on* A_i^u*'s instead of* A_i*'s will have exactly the same value function.*

Proof. Suppose $\boldsymbol{a}$ is probabilistically dominated (or replaceable) by some $\boldsymbol{a}_\lambda$ and $\tilde{\boldsymbol{a}} \in \operatorname{argmin}_{\boldsymbol{a}_r \in \mathcal{A}(\boldsymbol{a}_\lambda)} \mathcal{F}_i(\boldsymbol{a}_r, \boldsymbol{W})$, where $\mathcal{A}(\boldsymbol{a}_\lambda) \subset A_i$ is the set of pure actions selected

by $\boldsymbol{a}_\lambda$ with positive probability. Then

$$\mathcal{F}_i(\boldsymbol{a}, \boldsymbol{W}) \;=\; C(\boldsymbol{x}_i, \boldsymbol{a}) + \sum_{j=1}^{n} p_{ij}(\boldsymbol{a}) W_j \;\geq\; \sum_{r=1}^{l} \lambda_r C(\boldsymbol{x}_i, \boldsymbol{a}_r) + \sum_{j=1}^{n} p_{ij}(\boldsymbol{a}) W_j \;=$$

$$\sum_{r=1}^{l} \lambda_r \mathcal{F}_i(\boldsymbol{a}_r, \boldsymbol{W}) \geq \mathcal{F}_i(\tilde{\boldsymbol{a}}, \boldsymbol{W}).$$

□

Since the above definitions are fully based on the cost $C(\boldsymbol{x}_i, \cdot)$, the set of "potentially useful" pure actions is defined locally for each $\boldsymbol{x}_i$ and independently of the global properties of the SSP. The corresponding pruning of the action space can be performed as a pre-processing step to speed up the computation of the value function, which might be particularly worthwhile if one needs to solve a family of related problems; e.g., differing only by the identity of the target $\boldsymbol{t} \in X$.

Geometric Interpretation: Once A_i no longer includes any transition-equivalent elements, it is often convenient to identify the remaining pure actions with their corresponding probability distributions over successor nodes. Suppose $\mathcal{N}(\boldsymbol{x}_i) = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m\} \subset X$ is a set of all possible immediate successors of $\boldsymbol{x}_i$ and let

$$\Xi_m = \{\boldsymbol{\xi} = (\xi_1, ..., \xi_m) \mid \; \xi_1 + \ldots + \xi_m = 1 \text{ and } \xi_j \geq 0 \text{ for } j = 1, \ldots, m\}$$

denote the $(m-1)$-dimensional probability simplex. We can then take A_i to be a subset of Ξ_m and $\boldsymbol{\xi} \in A_i$ implies $p(\boldsymbol{x}_i, \boldsymbol{\xi}, \boldsymbol{z}_j) = \xi_j$ for $j = 1, \ldots, m$. The cost of the next transition is then $C(\boldsymbol{x}_i, \boldsymbol{\xi})$ though we will also use the notation $C(\boldsymbol{\xi})$ or $C(\xi_1, ..., \xi_m)$ whenever $\boldsymbol{x}_i \in X$ is clear from the context. Allowing the use of relaxed actions in rSSP is equivalent to switching from A_i to its convex hull $A_i^{co} \subset \Xi_m$. Using the best relaxed action in each transition equivalence class

essentially replaces C with its lower convex envelope

$$\check{C}(\boldsymbol{\xi}) = \min_{\boldsymbol{a}_\lambda \in \Lambda(\boldsymbol{\xi})} C(\boldsymbol{a}_\lambda) \qquad \forall \boldsymbol{\xi} \in A_i^{co}, \tag{2.8}$$

where $\boldsymbol{a}_\lambda$ is specified by some choice of $\lambda \in \Xi_m$ and $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m\} \subset A_i$ that satisfy $\xi_j = \sum_{r=1}^m \lambda_r \xi_{r,j}$ for $j = 1, \ldots, m$ while $C(\boldsymbol{a}_\lambda) = \sum_{r=1}^m \lambda_r C(\boldsymbol{\xi}_r)$. The resulting $\check{C}$ is convex[2] and thus continuous on the interior of A_i^{co} (and lower semi-continuous on ∂A_i^{co}). The pruning outlined above is possible because the minimum in (2.3) can only be attained at $\boldsymbol{\xi} \in A_i$ if $C(\boldsymbol{\xi}) = \check{C}(\boldsymbol{\xi})$. If an optimal $\boldsymbol{\xi}$ is replaceable, that same minimum will be also achieved by another (non-replaceable) action. An action $\boldsymbol{\xi} \in A_i$ is "useful" if and only if $(\boldsymbol{\xi}, C(\boldsymbol{\xi}))$ is an extreme point of the epigraph of $\check{C}$.

For a concrete example, consider an SSP whose transition digraph is represented in Figure 2.1(b). Focusing on $\boldsymbol{x}_1$, there are two possible successor nodes, $\boldsymbol{x}_2$ and $\boldsymbol{x}_5$. In this case, we can identify $\boldsymbol{\xi} = (1-p, p) \in \Xi_2$ with $p \in [0, 1]$, which makes it convenient to visualize $K(p) = C\,(1-p, p)$ and its convexified version $\check{K}(p) = \check{C}\,(1-p, p)$. In many SSPs, it is natural to consider a continuous spectrum of actions; in other applications, the set of actions is discrete and finite. To illustrate the most general case, we will suppose here that $\boldsymbol{x}_1$ has a "hybrid" action set; e.g., $A_1 = [\underline{\boldsymbol{a}}, \bar{\boldsymbol{a}}] \cup \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_4\}$. We will further suppose that some of these actions are transition-equivalent (yielding the same probabilities of transition to $\boldsymbol{x}_2$ and $\boldsymbol{x}_5$), which makes the graph of $K(p)$ multivalued and possibly discontinuous. Figure 2.2 illustrates the deterministic and probabilistic pruning process at $\boldsymbol{x}_1$. Once the transition-equivalent pure actions are removed (crossed in red in Figure 2.2(a)), we can perform additional pruning based on relaxed actions (Figure 2.2(b)). Although the value function remains unchanged during the convexification procedure, it is clear now that the optimal action at $\boldsymbol{x}_1$ will come from the set A_1^u.

[2]The same "cost convexification" approach has also been used in [41, Lemma 6].

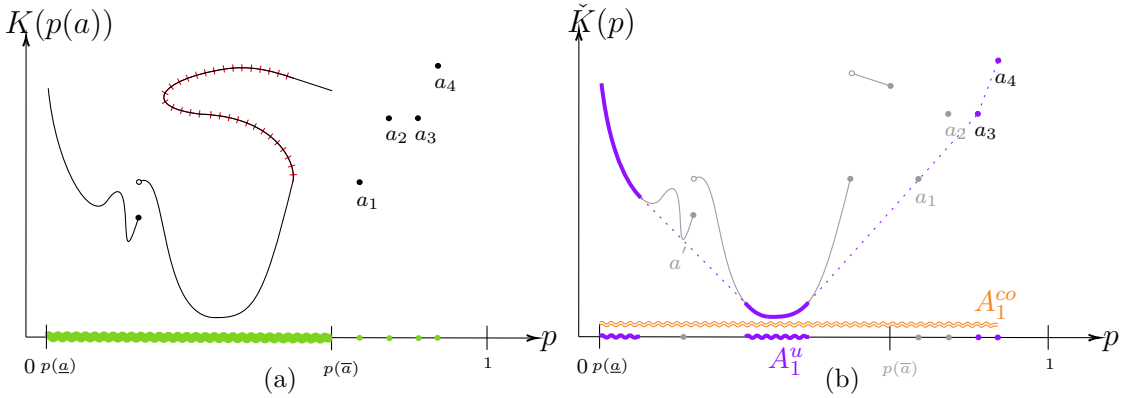


Figure 2.2: Example of deterministic (panel (a)) and probabilistic (panel (b)) pruning of the action set when $m = 2$. Depicted action choices at the node $\boldsymbol{x}_1$ in Figure 2.1(b), with transition probabilities $\mathbb{P}(\boldsymbol{x}_1 \to \boldsymbol{x}_5 \mid \boldsymbol{a}) = p(\boldsymbol{a})$ and $\mathbb{P}(\boldsymbol{x}_1 \to \boldsymbol{x}_2 \mid \boldsymbol{a}) = 1 - p(\boldsymbol{a})$. The specific cost function $K(p(\boldsymbol{a}))$ is chosen for the sake of illustration only. Many application-motivated examples of cost functions are considered in §2.3 and §2.4. Panel (a): The transition probabilities associated with each action in $A_1 = [\underline{\boldsymbol{a}}, \bar{\boldsymbol{a}}] \cup \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_4\}$ are indicated in green on the p-axis. During the deterministic pruning process, all actions $\boldsymbol{a} \in A_1$ that are transition-equivalent to another action $\tilde{\boldsymbol{a}} \in A_1$ with $K(p(\boldsymbol{a})) \geq K(p(\tilde{\boldsymbol{a}}))$ are removed along with the corresponding portion of the transition cost curve (crossed in red). Panel (b): Following the deterministic pruning, we obtain $\check{K}(p)$ by taking the lower convex envelope of K (solid and dashed purple curve), and the resulting A_1^{co} is indicated in orange above the p-axis. Replaceable actions $\boldsymbol{a}'$ and $\boldsymbol{a}_1$ are also removed at this stage. The transition probabilities associated with the remaining useful pure actions A_1^u are indicated on the p-axis in solid purple, and their corresponding transition cost values are also marked in solid purple along $\check{K}$.

2.1.3 Opportunistically Stochastic Shortest Path Problems

We now focus on a specific subclass of SSPs, in which every stochastically realizable path is also realizable using only deterministic actions (i.e., any pure action $\boldsymbol{a} \in A_i$ for which $\mathcal{I}(\boldsymbol{a})$ is a singleton). Later, we will demonstrate their usefulness

in approximating solutions of continuous optimal control problems (§2.3) and in routing of autonomous vehicles (§2.4).

Definition 7 (OSSP)**.** *We will refer to an SSP as* Opportunistically Stochastic *(OSSP) if*

$$\exists \boldsymbol{a} \in A_i \text{ s.t. } p_{ij}(\boldsymbol{a}) > 0 \qquad \Longrightarrow \qquad \exists \tilde{\boldsymbol{a}} \in A_i \text{ s.t. } p_{ij}(\tilde{\boldsymbol{a}}) = 1 \qquad \textit{holds for all } i \textit{ and } j. \tag{2.9}$$

If the set of potential successor nodes $\mathcal{N}(\boldsymbol{x}_i)$ has m elements, then the post-pruning A_i^u can be identified with a subset of Ξ_m, which in an OSSP will include all vertices of this probability simplex, encoding deterministic transitions. Correspondingly, the randomized-policy version of this problem (rOSSP), would have the action set $A_i^{co} = \Xi_m$.

We note that the OSSPs are also closely related to Multimode Stochastic Shortest Path (MSSP) problems introduced in [105]. MSSPs are based on a more stringent requirement that, whenever some nodes are possible successors of $\boldsymbol{x}_i$ under any specific action, then *every* probability distribution over those same successor nodes must be realizable. To be more precise, the *modes* in MSSPs are defined as subsets of $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{t}\}$, describing possible successor nodes under each particular class of actions. Each non-terminal node $\boldsymbol{x} \in X \backslash \{\boldsymbol{t}\}$ has its own set of such modes, denoted $\mathcal{S}(\boldsymbol{x})$, and each mode $s \in \mathcal{S}(\boldsymbol{x})$ can be specified by enumerating the nodes in it; i.e., $s = \{\boldsymbol{z}_1^s, \ldots, \boldsymbol{z}_{|s|}^s\} \subset \mathcal{N}(\boldsymbol{x}) \subset X \setminus \{\boldsymbol{x}\}$. MSSPs are built on two critical assumptions:

(M1) any available action has all of its successor nodes in one of these modes; i.e.,

$$\forall \boldsymbol{x}_i \in X, \boldsymbol{a} \in A(\boldsymbol{x}_i) \quad \exists s \in \mathcal{S}(\boldsymbol{x}_i) \text{ s.t. } \mathcal{I}(\boldsymbol{a}) \subset s;$$

(M2) every probability distribution over the set of successor nodes in any mode is achievable via some action; i.e.,

$$\forall \boldsymbol{x}_i \in X\backslash\{\boldsymbol{t}\},\ s \in \mathcal{S}(\boldsymbol{x}_i),\ \text{and}\ \boldsymbol{\xi} \in \Xi_{|s|} \quad \exists \boldsymbol{a} \in A(\boldsymbol{x}_i)\ \text{s.t.}\ p(\boldsymbol{x}_i, \boldsymbol{a}, \boldsymbol{z}_r) = \xi_r, \quad \forall \boldsymbol{z}_r \in s,$$

where $\Xi_{|s|}$ is the $(|s|-1)$-dimensional probability simplex.

Thus, a decision at each stage of an MSSP is twofold: a deterministic choice of a mode and a choice of a probability distribution over the possible successor nodes in that mode. This makes it natural to represent the actions as $\boldsymbol{a} = (s, \boldsymbol{\xi}) \in A(\boldsymbol{x})$, where $\boldsymbol{\xi} \in \Xi_{|s|}$, and then re-write the optimality equation (2.3) as

$$\begin{aligned} V^s(\boldsymbol{x}) &= \min_{\boldsymbol{\xi} \in \Xi_{|s|}} \left\{ C^s(\boldsymbol{x}, \boldsymbol{\xi}) + \sum_{r=1}^{|s|} \xi_r U(\boldsymbol{z}_r^s) \right\}, \\ U(\boldsymbol{x}) &= \min_{s \in \mathcal{S}(\boldsymbol{x})} V^s(\boldsymbol{x}), \qquad\qquad \forall \boldsymbol{x} \in X\backslash\{\boldsymbol{t}\}, \\ U(\boldsymbol{t}) &= 0. \end{aligned} \tag{2.10}$$

OSSP is a generalization of MSSP since the assumption **(M2)** implies the availability of deterministic transitions from $\boldsymbol{x}_i$ to each $\boldsymbol{z}_r \in s \in \mathcal{S}(\boldsymbol{x}_i)$. Unless an MSSP is fully deterministic, it must also include a continuum of actions spanning entire probability simplexes corresponding to each mode, making all modes "perfect" in this sense. This clearly does not have to be the case in general OSSPs, for which each A_i^u might be a proper (or even finite) subset of the probability simplex.

On the other hand, for any given OSSP, it is easy to construct an MSSP with the same value function; e.g., by using the convexified cost $\check{C}$ instead of the original C. To have a single mode at every node, one can define $\mathcal{S}(\boldsymbol{x}) = \{s\}$ with $s = \mathcal{N}(\boldsymbol{x}) = \bigcup_{\boldsymbol{a} \in A(\boldsymbol{x})} \mathcal{I}(\boldsymbol{a})$, and then use the convexified cost $\check{C}$ instead of the

original C. The action set $A(\boldsymbol{x})$ can be then extended by adding all relaxed actions to satisfy **(M2)**.

It is worth noting that in the definition of $\check{C}$, the result will not change if the relaxed actions are formed by only using pure actions $\tilde{\boldsymbol{a}}$ such that $\mathcal{I}(\tilde{\boldsymbol{a}}) \subset \mathcal{I}(\boldsymbol{a})$. (In all other cases, $\boldsymbol{a}_\lambda \notin \Lambda(\boldsymbol{a})$.) This suggests a natural approach to introduce a minimal set of "imperfect modes" in OSSPs. We will say that $s \subset \mathcal{N}(\boldsymbol{x})$ is an imperfect mode if $\exists \boldsymbol{a} \in A(\boldsymbol{x})$ s.t. $s = \mathcal{I}(\boldsymbol{a})$ but $\nexists \tilde{\boldsymbol{a}} \in A(\boldsymbol{x})$ s.t. s is a proper subset of $\mathcal{I}(\tilde{\boldsymbol{a}})$. (This ensures that a mode is never a subset of another mode.) An action $\boldsymbol{a} \in A(\boldsymbol{x})$ is associated with a mode $s \in \mathcal{S}(\boldsymbol{x})$ if $\mathcal{I}(\boldsymbol{a}) \subset s$. Thus, every action is associated with at least one mode and might well be associated with multiple modes simultaneously.

To provide a concrete example, we focus on the node $\boldsymbol{x}_6$ in Figure 2.1(b) and consider $A(\boldsymbol{x}_6) = \{\boldsymbol{a}_1, ..., \boldsymbol{a}_{10}\}$, where the first five actions are deterministic ($\mathcal{I}(\boldsymbol{a}_j) = \{\boldsymbol{x}_j\}$, $j = 1, ..., 5$) while the rest of them are not:

$$\mathcal{I}(\boldsymbol{a}_6) = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}, \quad \mathcal{I}(\boldsymbol{a}_7) = \{\boldsymbol{x}_2, \boldsymbol{x}_3\}, \quad \mathcal{I}(\boldsymbol{a}_8) = \{\boldsymbol{x}_3, \boldsymbol{x}_4\},$$
$$\mathcal{I}(\boldsymbol{a}_9) = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, \quad \mathcal{I}(\boldsymbol{a}_{10}) = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4\}.$$

Based on the above definition, $\mathcal{N}(\boldsymbol{x}_6) = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_5\}$ and $\mathcal{S}(\boldsymbol{x}) = \{s_1, ..., s_4\}$ with $s_1 = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ (associated actions: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \boldsymbol{a}_6, \boldsymbol{a}_7, \boldsymbol{a}_9$), $s_2 = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4\}$ (associated actions: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_4, \boldsymbol{a}_6, \boldsymbol{a}_{10}$), $s_3 = \{\boldsymbol{x}_3, \boldsymbol{x}_4\}$ (associated actions: $\boldsymbol{a}_3, \boldsymbol{a}_4, \boldsymbol{a}_8$), and $s_4 = \{\boldsymbol{x}_5\}$ (associated action: $\boldsymbol{a}_5$). Compared to MSSPs, these modes are "imperfect" since **(M2)** is violated. But each mode can be "perfected" without changing the value function by including relaxed actions and using $\check{C}$ to define their cost.

2.2 Label-Setting and Monotone Causality in OSSPs

Since the value function $\boldsymbol{U} = [U_1, \ldots, U_n]^T$ is a fixed point of the operator $\mathcal{T}$ defined in (2.4), the simplest approach for computing $\boldsymbol{U}$ is through *value iterations* (VI), in which one starts with an initial guess $\boldsymbol{W}^0 \in \mathbb{R}^n$ and updates it iteratively by taking $\boldsymbol{W}^{k+1} = \mathcal{T}\boldsymbol{W}^k$. The operator $\mathcal{T}$ is generally not a contraction unless all stationary policies are known to be proper [18]. But Tsitsiklis and Bertsekas have shown that assumptions (**A1**)-(**A5**) guarantee that this fixed point is unique and $\boldsymbol{W}^k \to \boldsymbol{U}$ as $k \to \infty$ regardless of $\boldsymbol{W}^0$ [18]. Unfortunately, for a general SSP, this does not necessarily occur after any finite number of iterations. The convergence can be slow and the VI algorithm is often impractical for large problems. Finding computationally efficient alternatives to these basic value iterations has been an active research area in the last several decades. One obvious direction is to use a Gauss-Seidel relaxation (GSVI), where the components of $\boldsymbol{W}^{k+1}$ are computed sequentially and the previously computed components are immediately used in computing the remaining ones; i.e.,

$$\boldsymbol{W}_i^{k+1} \;=\; \min_{\boldsymbol{a} \in A_i} \mathcal{F}_i \left(\boldsymbol{a},\, [\boldsymbol{W}_1^{k+1}, \ldots, \boldsymbol{W}_{i-1}^{k+1}, \boldsymbol{W}_i^k, \ldots, \boldsymbol{W}_n^k]^T \right).$$

But the efficiency of this approach is heavily dependent on the ordering imposed on the nodes/states in X even if the convergence is achieved in finitely many steps. E.g., for the very simple example in Figure 2.1(a), GSVI will require 3 iterations to converge with the default node ordering though only 1 iteration would be needed if the ordering were reversed. More generally, this difference in the number of needed iterations can be as high as $O(n)$. To address this, some implementations of GSVI alternate through several orderings that are likely to be efficient [20]. In others, the so called *label-correcting* methods, the next node/state to be updated is determined dynamically, based on the current vector of tentative labels (or values)

$\boldsymbol{W}$ and the history of updates up till that point [11, 78, 15, 17, 53]. Some of these methods were originally developed for the deterministic shortest path setting, but have since been adapted to SSPs as well, particularly when used to discretize Hamilton-Jacobi equations; e.g., [81, 19, 6].

Other notable approaches include *topological value iterations* (in which the topological structure imposed on X by A is taken into account to attempt splitting an SSP into a sequence of causally ordered subproblems) [33, 34], policy iterations (in which the goal is to produce an improving sequence of stationary policies (μ_k), with $\boldsymbol{U}$ recovered as a limit of their respective policy value vectors) [59], and hybrid policy-value iterations [104].

Here we are primarily interested in a subclass of SSPs for which VI actually does converge in at most n iterations regardless of $\boldsymbol{W}^0$. (Typically, this happens when at least one element of $\boldsymbol{W}^k$ becomes newly converged in each iteration.) But even assuming the minimization in (2.4) can be performed in $O(1)$ operations, the computational cost of a single value iteration is $O(n)$, which yields the overall cost of $O(n^2)$ up to convergence. In such SSPs, the same worst-case $O(n^2)$ computational cost also holds for all VI variants mentioned above, while the policy iterations would still require an infinite number of steps for full convergence if the action sets A_i are infinite. In contrast, our goal is to obtain $\boldsymbol{U}$ in $O(n \log n)$ or $O(n)$ operations, bounding the number of "approximate U_i" updates based on that node's stochastic outdegree (i.e., $|\mathcal{N}(\boldsymbol{x}_i)|$) rather than on the overall number of non-terminal nodes $n \gg \max_{\boldsymbol{x}_i} |\mathcal{N}(\boldsymbol{x}_i)|$. For the deterministic shortest path (SP) problems, this cost reduction is accomplished by the classical *label-setting* methods reviewed in §2.2.1. We then describe the prior (implicit) conditions for their general SSP-applicability in §2.2.2, and derive new (explicit) conditions for

their OSSP-applicability in §2.2.3.

2.2.1 Label-Setting Methods for Deterministic SP Problems

Classical shortest/cheapest path problems on directed graphs can be interpreted as a subclass of SSPs in which all actions yield deterministic transitions. In that setting, $C_{ij} = C(\boldsymbol{x}_i, \boldsymbol{x}_j)$ encodes the cost of a direct $(\boldsymbol{x}_i \to \boldsymbol{x}_j)$ transition, with the set of potential successor nodes denoted $N_i = N(\boldsymbol{x}_i) = \{\boldsymbol{x}_j \in X \mid C_{ij} < \infty\}$. We will further assume that $C_{ij} \geq \delta \geq 0$. The value function satisfies the following dynamic programming equations:

$$U_i = \min_{\boldsymbol{x}_j \in N(\boldsymbol{x}_i)} \{C_{ij} + U_j\}, \quad \forall \boldsymbol{x}_i \in X \setminus \{\boldsymbol{t}\}, \tag{2.11}$$

with $U(\boldsymbol{t}) = 0$. Efficient algorithms for solving (2.11) are well-known and covered in standard references (e.g., [2, 14]), but we provide a quick overview for the sake of completeness.

The key idea of label-setting methods is to re-order iterations so that the tentative value of each node $\boldsymbol{x}_i$ is updated at most $|N_i|$ times. This yields a significant performance advantage over value iterations, particularly when the outdegree of nodes is bounded and relatively small, $\kappa = \max_i |N_i| \ll n$. Since the number of updates per node is bounded by κ, such methods are also often considered *noniterative*. The non-negativity of transition costs implies a *monotone causality* property:

$$U_i \text{ may depend on } U_j \text{ only if } \; U_i \geq U_j. \tag{2.12}$$

If all $C_{ij} > 0$ and nodes could be ordered monotonically based on U_i's, GSVI would

converge in a single iteration. But since such an ordering is not known in advance, it has to be obtained at run-time. Dijkstra's algorithm [37] accomplishes this by recomputing tentative labels V_i while maintaining two lists of nodes: the list of known/converged nodes $K = \{\boldsymbol{x}_i \mid V_i = U_i \text{ is confirmed}\}$ and the list of tentative nodes $L = X \setminus K$ for which $V_i \geq U_i$. The basic version of this algorithm starts with $V(\boldsymbol{t}) = U(\boldsymbol{t}) = 0$, $K = \{\boldsymbol{t}\}$, and $V_i = +\infty$ for all $\boldsymbol{x}_i \in L = X \setminus \{\boldsymbol{t}\}$. At each stage of Dijkstra's method, the node $\bar{\boldsymbol{x}}$ with the current smallest finite value in L is moved to K, and other tentative nodes adjacent to it (i.e., all $\boldsymbol{x}_i \in L$ s.t. $\bar{\boldsymbol{x}} \in N_i$) are updated by setting $V_i := \min\limits_{\boldsymbol{x}_j \in N_i \cap K} \{C_{ij} + V_j\}$, or, more efficiently, by using $V_i := \min\left(V_i,\, C(\boldsymbol{x}_i, \bar{\boldsymbol{x}}) + V(\bar{\boldsymbol{x}})\right)$. The process terminates once $L = \emptyset$ or once all values remaining in L are infinite. (In the latter case, the nodes remaining in L are not path-connected to $\boldsymbol{t}$.) The method terminates after at most κn updates and yields the correct U_i values even if $\delta = 0$. To identify $\bar{\boldsymbol{x}}$ efficiently, L is typically implemented using heap-sort data structures, resulting in an overall computational cost of $O(n \log n)$.

If $\delta > 0$, the nodes whose tentative values are less than δ apart cannot depend on each other. This makes it possible to use another label-setting method due to Dial [36], in which the tentative values are not sorted but instead placed into "buckets" of width δ. At each stage of the algorithm, all tentative nodes in the current smallest bucket are moved to K simultaneously, and all tentative labels of nodes adjacent to them are updated, switching these adjacent nodes to new buckets if necessary. Since inserting to and deleting from a bucket can be performed in $O(1)$ time, the overall computational complexity of Dial's method is $O(n)$. Unlike Dijkstra's algorithm, Dial's method is also naturally parallelizable. Nevertheless, which of them is more efficient in practice depends on the properties of the SP problem.

2.2.2 Applicability of Noniterative Methods to general SSPs

Avoiding iterative methods in general SSPs is usually harder. Given any stationary policy μ, one can define the corresponding dependency digraph G_μ by starting with the nodes X and adding arcs wherever direct transitions are possible under μ. (I.e., adding an arc from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ whenever $p_{ij}\left(\mu(\boldsymbol{x}_i)\right) > 0$.) We will call an SSP *causal* if there exists an optimal stationary policy μ^* whose dependency digraph G_{μ^*} is acyclic. As noted in [14, Volume 2, §2.3.3], value iterations on any such causal SSP will converge to $\boldsymbol{U}$ after at most n iterations. If one were to use a reverse topological ordering of G_{μ^*} to sort X, the GSVI algorithm would converge in a single iteration. This property is trivially satisfied by special *explicitly causal* SSPs, in which the dependency digraph of *every* stationary policy is acyclic (and thus the transition digraph G is as well), making it easy to obtain the full $\boldsymbol{U}$ with only n node value updates regardless of the properties of C. (This mirrors the fact that Dijkstra's and Dial's methods are not really needed to find deterministic shortest paths on acyclic graphs.)

Turning to a broader class of SSPs that are not explicitly causal, a natural question to consider is whether the label-setting methods might be applicable. Following Bertsekas [14], we will say that an optimal stationary policy μ^* is *consistently improving* if, for all $\boldsymbol{x}_i \neq \boldsymbol{t}$,

$$p_{ij}(\mu^*(\boldsymbol{x}_i)) > 0 \implies U_i > U_j. \tag{2.13}$$

This property is a stochastic analogue of (2.12). If such μ^* is known to exist, (2.13) guarantees that an SSP-version of Dijkstra's method will correctly produce $\boldsymbol{U}$ as

its output[3]. In terms of implementation, the only difference from the deterministic SP case described in §2.2.1, is that, once $\bar{\boldsymbol{x}}$ is moved to K, we would need to update all $\boldsymbol{x}_i$'s such that $\bar{\boldsymbol{x}} \in \mathcal{N}(\boldsymbol{x}_i)$ by using

$$V_i \;:=\; \min\left(V_i,\; \min_{\substack{\boldsymbol{a}\in A_i \text{ s.t.} \\ \bar{\boldsymbol{x}}\in\mathcal{I}(\boldsymbol{a})\subset K}} \mathcal{F}_i(\boldsymbol{a}, \boldsymbol{V})\right). \tag{2.14}$$

Continuing this approach, we will say that an optimal stationary policy μ^* is *consistently δ-improving* if, for some $\delta > 0$ and all $\boldsymbol{x}_i \neq \boldsymbol{t}$,

$$p_{ij}(\mu^*(\boldsymbol{x}_i)) > 0 \implies U_i \geq U_j + \delta. \tag{2.15}$$

The existence of such a μ^* similarly guarantees the applicability of Dial's method[4] with buckets of width δ. If we assume that the minimization in (2.14) can be performed in $O(1)$ operations and stochastic outdegrees $|\mathcal{N}(\boldsymbol{x}_i)|$ are bounded by a constant κ, then the overall computational cost will scale the same way as in the deterministic case: $O(n \log n)$ for Dijkstra's and $O(n)$ for Dial's method.

Definition 8. *We will say that an SSP is* monotone causal *(MC) if at least one of its optimal stationary policies is consistently improving. We will refer to an SSP as* monotone δ-causal *(δ-MC) if that optimal stationary policy is consistently δ-improving.*

Unfortunately, the above criteria based on (2.13) and (2.15) are implicit and hard to apply in practice since none of the optimal policies are known before $\boldsymbol{U}$ is computed. Thus, it is natural to search for sufficient (δ-)MC criteria that

[3]It might seem natural to pose a "$U_i \geq U_j$" condition in (2.13), to fully mirror (2.12). But for non-deterministic actions, this condition turns out to be too weak and may result in Dijkstra's failing to converge to $\boldsymbol{U}$. See [105, Figure 1] for a simple example.

[4]Consistently δ-improving policies were first defined in [105] with a strict inequality in (2.15) and $\delta \geq 0$. Since the buckets are non-overlapping and have a positive width δ, we know that $(U_j + \delta)$ is never in the same bucket as U_j. Thus, the above definition is more suitable to guarantee the applicability of Dial's method.

can be verified a priori and locally (e.g., based on the cost function properties at each node / state) without considering the global structure of the SSP. The first two such SSP examples were found by Tsitsiklis [101, 102], who used them to apply Dijkstra's and Dial's methods to regular-grid semi-Lagrangian discretizations of isotropic optimal control problems. Related and more general methods were also developed by others (see the detailed discussion in §2.3), and the Multimode SSPs were then introduced in [105] – both to provide an overall structure for discussing the label-setting in PDE discretizations and to show that the monotone (δ-)causality can be guaranteed even for some SSPs unrelated to optimal control problems. The idea was to pose the sufficient conditions based on the cost C only, and have them verified on a mode-per-mode, node-per-node basis. Given the close relationship between the OSSPs and MSSPs, it is worth highlighting the main difference between the criteria in [105] and those developed below. In addition to a restrictive condition **(M2)** described in §2.1.3, the criteria developed in [105] have either required C to be concave or posed conditions on its derivatives, while we allow C to be merely lower semi-continuous.

2.2.3 Guaranteeing Monotone Causality in OSSPs

Consider an OSSP satisfying assumptions **(A1)-(A4)**, **(A5')**, **(A6)**, and (2.9). Suppose an action $\boldsymbol{a} \in A_i$ results in a list of possible successor nodes $\mathcal{I}(\boldsymbol{a}) = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_m\} \subset X \setminus \{\boldsymbol{x}_i\}$, with the respective transition probabilities[5] $p(\boldsymbol{x}_i, \boldsymbol{a}, \boldsymbol{z}_j) = \xi_j > 0$ for $j = 1, \ldots, m$ and $\boldsymbol{\xi} = (\xi_1, ..., \xi_m) \in \text{int}(\Xi_m)$. Assuming this $\boldsymbol{a} \in A_i$ is not deterministic (i.e., $m > 1$) and choosing any specific $r \in \{1, ..., m\}$, we define

[5] Of course, these m, $\boldsymbol{z}_j$'s, and ξ_j's are always action-dependent. But for the sake of readability, we do not indicate this in the notation whenever a specific action is clear from the context.

$\boldsymbol{\gamma}_r = (\gamma_{r,1}, ..., \gamma_{r,m})$ to be an oblique (proportional) projection of $\boldsymbol{\xi}$ as follows

$$\gamma_{r,j} \;=\; \begin{cases} 0, & \text{if } j = r; \\ \xi_j/(1-\xi_r), & \text{otherwise.} \end{cases}$$

We note that once we omit the r-th zero component of $\boldsymbol{\gamma}_r$, it can be thought of as a point in int $\left(\Xi_{(m-1)}\right)$. The set $\Lambda(\boldsymbol{\gamma}_r)$ of relaxed actions transition-equivalent to $\boldsymbol{\gamma}_r$ is nonempty, and the convexified cost of $\boldsymbol{\gamma}_r$ can be defined as $\check{C}(\boldsymbol{\gamma}_r) = \min_{\boldsymbol{a}_\lambda \in \Lambda(\boldsymbol{\gamma}_r)} C(\boldsymbol{a}_\lambda)$.

Theorem 1 (Sufficient Monotone δ-Causality Condition in OSSPs)**.**
Suppose there exists a $\delta \geq 0$ such that, for all $\boldsymbol{x}_i \in X \backslash \{\boldsymbol{t}\}$ and $\boldsymbol{a} \in A_i^u$,

- *if $\boldsymbol{a}$ is deterministic, then $C(\boldsymbol{x}_i, \boldsymbol{a}) \geq \delta$;*
- *if $\boldsymbol{a}$ is not deterministic, then*

$$C(\boldsymbol{x}_i, \boldsymbol{a}) \geq (1-\xi_r)\check{C}(\boldsymbol{\gamma}_r) + \xi_r \delta, \qquad \forall r \in \{1, ..., |\mathcal{I}(\boldsymbol{a})|\}. \tag{2.16}$$

If these conditions are satisfied, this OSSP is monotone causal and Dijkstra's method is applicable. If $\delta > 0$, the OSSP is monotone δ-causal and Dial's method with buckets of width δ is also applicable.

Proof. Define the set of optimal actions $A_i^* = \underset{\boldsymbol{a} \in A_i}{\operatorname{argmin}} \left\{ C(\boldsymbol{x}_i, \boldsymbol{a}) + \sum_{j=1}^{n+1} p_{ij}(\boldsymbol{a}) U_j \right\}$, and suppose that $\boldsymbol{a}^* \in A_i^*$ yields the minimal number of successor nodes among them. (I.e., $|\mathcal{I}(\boldsymbol{a}^\sharp)| \geq |\mathcal{I}(\boldsymbol{a}^*)|$ for all $\boldsymbol{a}^\sharp \in A_i^*$.) If $\boldsymbol{a}^*$ is deterministic, then $\mathcal{I}(\boldsymbol{a}^*) = \{\boldsymbol{z}_1\}$ and U_i depends only on $U(\boldsymbol{z}_1)$ while $U_i = C(\boldsymbol{x}_i, \boldsymbol{a}^*) + U(\boldsymbol{z}_1) \geq \delta + U(\boldsymbol{z}_1)$. For $\delta = 0$ this inequality is strict due to (**A5'**).

If $\boldsymbol{a}^*$ is not deterministic, its use leads to one of the successor nodes $\{\boldsymbol{z}_1, ..., \boldsymbol{z}_m\}$ with respective probabilities $\xi_r > 0$, and U_i depends on all of the $U(\boldsymbol{z}_r)$ with

$r \in \{1, ..., m\}$. Choosing any specific r and combining the optimality of $\boldsymbol{a}^*$ with (2.16), we see that

$$U_i \;=\; C(\boldsymbol{x}_i, \boldsymbol{a}^*) + \sum_{j=1}^{m} \xi_j U(\boldsymbol{z}_j) \;\geq\; (1-\xi_r)\check{C}(\boldsymbol{\gamma}_r) + \xi_r \delta + \sum_{j=1}^{m} \xi_j U(\boldsymbol{z}_j). \tag{2.17}$$

On the other hand, using $\boldsymbol{a}^*$ must be strictly better that using any relaxed action $\boldsymbol{a}_\lambda \in \Lambda(\boldsymbol{\gamma}_r)$. Otherwise, the best pure action $\tilde{\boldsymbol{a}}$ among those potentially selected by $\boldsymbol{a}_\lambda$ would have to be at least as good as $\boldsymbol{a}^*$, leading to $\tilde{\boldsymbol{a}} \in A_i^*$ with $|\mathcal{I}(\tilde{\boldsymbol{a}})| \leq m-1$, which is a contradiction. Thus, we have $U_i < \mathcal{F}_i(\boldsymbol{a}_\lambda, \boldsymbol{U})$ for all $\boldsymbol{a}_\lambda \in \Lambda(\boldsymbol{\gamma}_r)$, which implies

$$U_i \;<\; \check{C}(\boldsymbol{\gamma}_r) + \sum_{j \neq r} \gamma_{r,j} U(\boldsymbol{z}_j) \;=\; \check{C}(\boldsymbol{\gamma}_r) + \frac{1}{(1-\xi_r)} \sum_{j \neq r} \xi_j U(\boldsymbol{z}_j). \tag{2.18}$$

Multiplying both sides by $(1-\xi_r)$ and rearranging,

$$U_i \;<\; (1-\xi_r)\check{C}(\boldsymbol{\gamma}_r) + \sum_{j \neq r} \xi_j U(\boldsymbol{z}_j) + \xi_r U_i. \tag{2.19}$$

Combining (2.17) and (2.19), canceling the terms present on both sides of the inequality, and dividing by $\xi_r > 0$, we obtain $U_i \;>\; \delta + U(\boldsymbol{z}_r)$, which proves the monotone (δ-)causality of this OSSP. □

The above condition is explicit and sharp in the sense outlined in Theorem 3, but it requires checking the convexified cost of multiple oblique projections for each non-deterministic action. We now provide another sufficient condition, which is more restrictive but easier to verify.

Theorem 2 (Simplified Monotone δ-Causality Condition in OSSPs)**.**
Suppose there exists a $\delta \geq 0$ such that, for all $\boldsymbol{x}_i \in X \backslash \{\boldsymbol{t}\}$, $\boldsymbol{a} \in A_i^u$, and every $r \in \{1, ..., |\mathcal{I}(\boldsymbol{a})|\}$,

$$C(\boldsymbol{x}_i, \boldsymbol{a}) \;\geq\; \sum_{j=1, j \neq r}^{m} \xi_j C_j + \xi_r \delta, \tag{2.20}$$

where $\mathcal{I}(\boldsymbol{a}) = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_m\}$, $p(\boldsymbol{x}_i, \boldsymbol{a}, \boldsymbol{z}_j) = \xi_j > 0$, *and each* C_j *encodes the cost of a deterministic action* $\boldsymbol{e}_j \in A_i$ *transitioning to* $\boldsymbol{z}_j$. *If these conditions are satisfied, this OSSP is monotone (*δ*-)causal.*

Proof. If $\boldsymbol{a}$ is deterministic and leads to $\boldsymbol{z}_r$, then (2.20) yields $C_r = C(\boldsymbol{x}_i, \boldsymbol{a}) \geq \delta$. If $\boldsymbol{a}$ is not deterministic, then all of these $\boldsymbol{e}_j$'s exist by the definition of OSSPs, and a relaxed action $\boldsymbol{a}_\lambda$ selecting each $\boldsymbol{e}_j$ with probability $\xi_j/(1-\xi_r)$ is transition-equivalent to $\boldsymbol{\gamma}_r$. Thus, $C(\boldsymbol{a}_\lambda) = \frac{1}{(1-\xi_r)} \sum_{j=1, j\neq r}^{m} \xi_j C_j \geq \check{C}(\boldsymbol{\gamma}_r)$ and (2.20) implies (2.16).

Since it is simple and instructive, we also include an alternative/direct proof that does not use oblique projections. Suppose $\boldsymbol{a} \in A_i$ is optimal. If it is deterministic and leads to $\boldsymbol{z}_r$, then $\xi_r = 1$ and (2.20) implies $U(\boldsymbol{x}_i) = C_r + U(\boldsymbol{z}_r) \geq \delta + U(\boldsymbol{z}_r)$.

If that $\boldsymbol{a}$ is not deterministic and (2.20) holds with $\xi_r \in (0, 1)$,

$$U(\boldsymbol{x}_i) \;=\; C(\boldsymbol{x}_i, \boldsymbol{a}) + \sum_{j=1}^{m} \xi_j U(\boldsymbol{z}_j) \;\geq\; \sum_{j=1, j\neq r}^{m} \xi_j \left(C_j + U(\boldsymbol{z}_j)\right) + \xi_r \left(\delta + U(\boldsymbol{z}_r)\right). \tag{2.21}$$

From the definition of the value function, $(C_j + U(\boldsymbol{z}_j)) \geq U(\boldsymbol{x}_i)$ for all j. Combining with (2.21),

$$U(\boldsymbol{x}_i) \geq \sum_{j=1, j\neq r}^{m} \xi_j U(\boldsymbol{x}_i) + \xi_r \left(\delta + U(\boldsymbol{z}_r)\right) = (1-\xi_r) U(\boldsymbol{x}_i) + \xi_r \left(\delta + U(\boldsymbol{z}_r)\right) \quad \Longrightarrow \quad U(\boldsymbol{x}_i) \geq \delta + U(\boldsymbol{z}_r).$$

□

Remark 1. *Note that the two criteria* (2.16) *and* (2.20) *are equivalent when* $m = 2$. *The latter also has a simple geometric interpretation illustrated in Figure 2.3 with* $\boldsymbol{\xi} = (1-p, p) \in \Xi_2$ *and the assumption that all actions* $\boldsymbol{a} \in A_i$ *lead to the same two successor states* $\boldsymbol{z}_1$ *and* $\boldsymbol{z}_2$. *The oblique projections of any* $\boldsymbol{\xi} \in \text{int}(\Xi_2)$

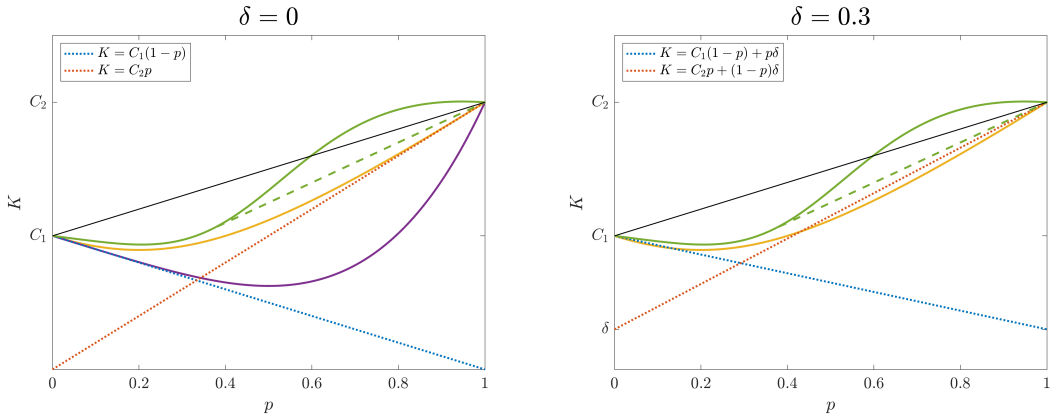


Figure 2.3: The geometric interpretation of condition (2.20) for $m = 2$ with $\delta = 0$ (panel (a)) and $\delta = 0.3$ (panel (b)) for several examples of transition cost functions. The solid purple and solid gold curves are smooth and convex. The solid green curve is smooth and nonconvex, and the green-dashed curve is its convexified version. In all cases, $K(p)$ is monotone (δ-)causal provided that the curve stays entirely above the two restriction lines on the interval $[0, 1]$. In (a), the purple curve violates the condition (2.20) for $r = 1$ even with $\delta = 0$; so, it cannot be monotone causal. While the other two curves satisfy (2.20) with $\delta = 0$, only the nonconvex curve (along with its convexified version) is monotone δ-causal for $\delta \leq 0.3$. The smooth gold convex curve can only be monotone causal (with $\delta = 0$), as the restriction lines imposed by (2.20) coincide with its tangent lines at $p = 0$ and $p = 1$.

will correspond to deterministic transitions to $\boldsymbol{z}_1$ *and* $\boldsymbol{z}_2$*. This OSSP is monotone (δ-)causal if the graph of* $K(p) = C(1-p, p)$ *does not venture below the two straight lines* $K = (1-p)C_1 + \delta p$ *and* $K = pC_2 + \delta(1-p)$*.*

For $m > 2$*, the two criteria are only equivalent if* $\sum_{j=1, j\neq r}^{m} \xi_j C_j = (1-\xi_r)\check{C}(\boldsymbol{\gamma}_r)$ *for each* r*, which is only true if* $\check{C}$ *is linear on* $\partial\Xi_m$ *(or, equivalently, if* $A_i^u \cap \partial\Xi_m$ *contains only deterministic actions). Unfortunately, this is often not the case for OSSPs arising in practical applications. Consider, for example, the case where* $m = 3$*,* $A_i = \Xi_3$*, and the cost function is* $C(\boldsymbol{\xi}) = \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2}$*. Theorem 1 guarantees its monotone causality: if* $\xi_r > 0$ *then* $\check{C}(\boldsymbol{\gamma}_r) = C(\boldsymbol{\gamma}_r) = (1-\xi_r)^{-1}\sqrt{\sum_{j\neq r} \xi_j^2}$*. Thus,*

$(C(\boldsymbol{\xi}))^2 - ((1-\xi_r)C(\boldsymbol{\gamma}_r))^2 = \xi_r^2 > 0$, *which verifies that* (2.16) *holds for* $\delta = 0$. *On the other hand, the simpler criterion from Theorem 2 is not satisfied by this* C *since* (2.20) *requires* $\sqrt{\xi_1^2+\xi_2^2+\xi_3^2} \geq \sum_{j\neq r}\xi_j$, *which is violated on a large part of* Ξ_3, *including at* $\boldsymbol{\xi} = (\frac{1}{3},\frac{1}{3},\frac{1}{3})$.

For an OSSP already known to be monotone causal, another natural question to ask is about the largest $\delta \geq 0$ for which it is monotone δ-causal. (If that maximal $\delta_\star$ is zero, the problem can be only treated by Dijkstra's method. Otherwise, $\delta_\star > 0$ specifies the the largest bucket-width usable in Dial's method.) With $m = 2$, the answer has a simple geometric interpretation.

Proposition 1 (Maximum Allowable δ when $m = 2$)**.** *Suppose that all non-deterministic actions in* A_i *lead to* $\{\boldsymbol{z}_1, \boldsymbol{z}_2\}$ *and satisfy* (2.20) *for* $\delta = 0$. *Letting* $\boldsymbol{e}_1, \boldsymbol{e}_2 \in A_i$ *be the deterministic actions such that* $p(\boldsymbol{x}_i, \boldsymbol{e}_j, \boldsymbol{z}_j) = 1$, *we will use the notation* $C_j = C(\boldsymbol{x}_i, \boldsymbol{e}_j)$.
The largest δ *for which the MC conditions are satisfied is then* $\delta_\star = \min(\delta_1, \delta_2)$, *where*

$$\delta_1 = \min_{\boldsymbol{a}\in A_i\setminus\{e_2\}} \frac{C(\boldsymbol{x}_i,\boldsymbol{a}) - p(\boldsymbol{x}_i,\boldsymbol{a},\boldsymbol{z}_2)C_2}{p(\boldsymbol{x}_i,\boldsymbol{a},\boldsymbol{z}_1)} \qquad \textit{and} \qquad \delta_2 = \min_{\boldsymbol{a}\in A_i\setminus\{e_1\}} \frac{C(\boldsymbol{x}_i,\boldsymbol{a}) - p(\boldsymbol{x}_i,\boldsymbol{a},\boldsymbol{z}_1)C_1}{p(\boldsymbol{x}_i,\boldsymbol{a},\boldsymbol{z}_2)} \tag{2.22}$$

Furthermore, if $A_i = \Xi_2$ *while* $C(\boldsymbol{\xi}) = C(\boldsymbol{x}_i, \boldsymbol{\xi})$ *is convex on* Ξ_2 *and differentiable at* $\boldsymbol{e}_1 = (1,0)$ *and* $\boldsymbol{e}_2 = (0,1)$, *then*

$$\delta_1 = C(\boldsymbol{e}_2) + \frac{\partial C}{\partial \xi_1}(\boldsymbol{e}_2) - \frac{\partial C}{\partial \xi_2}(\boldsymbol{e}_2) \qquad \textit{and} \qquad \delta_2 = C(\boldsymbol{e}_1) - \frac{\partial C}{\partial \xi_1}(\boldsymbol{e}_1) + \frac{\partial C}{\partial \xi_2}(\boldsymbol{e}_1) \tag{2.23}$$

Proof. Since (2.20) already holds with $\delta = 0$ for all non-deterministic $\boldsymbol{a} \in A_i$, the ratios minimized in 2.22 are always non-negative. Based on assumptions **(A1)**-**(A3)**, **(A5')**, the minima are attained and guarantee that (2.20) is satisfied for

each $r = 1, 2$ with $\delta = \delta_r$. Figure 2.4(a) illustrates this idea and the restriction lines corresponding to $\delta_\star$ for an example of a monotone causal OSSP with $|A_i| = 6$.

If $A_i = \Xi_2$, each action can be again identified with its probability distribution $\boldsymbol{\xi} = (1-p, p)$ for $p \in [0, 1]$. If $C(\boldsymbol{\xi})$ is convex then so is $K(p) = C(1-p, p)$. So, the inequality (2.20) for $r = 1$ can be rewritten as $K(p) \geq \delta + p(K(1) - \delta)$, which is equivalent to $K'(1) \leq K(1) - \delta$ due to convexity. (I.e., this is the requirement for the tangent at $p = 1$ to lie no lower than the δ-dependent restriction line; see the solid green curve in Figure 2.4(b).) Thus, the largest δ that can guarantee this is $\delta_1 = K(1) - K'(1)$. The same argument for $r = 2$ results in $\delta_2 = K(0) + K'(0)$, and the chain rule yields (2.23). □

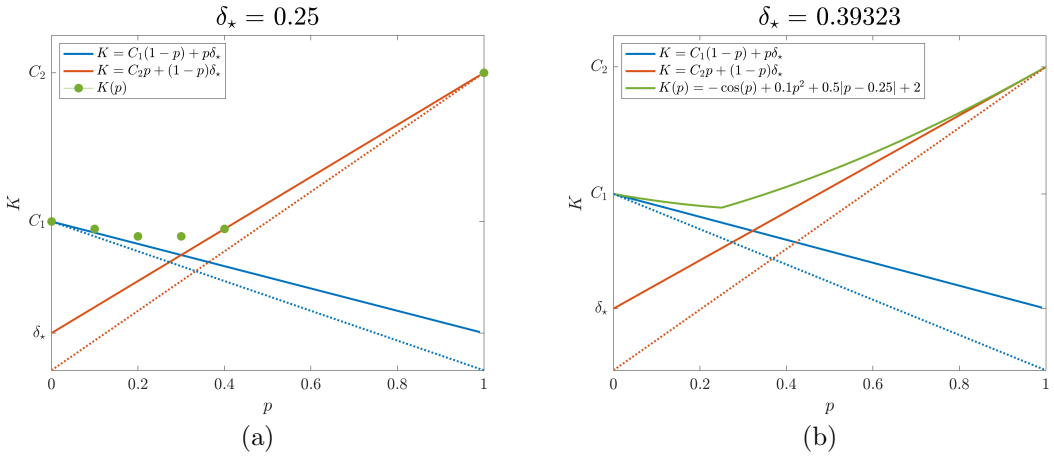


Figure 2.4: Illustration of the geometric interpretation of $\delta_\star$ for monotone δ-causality when $m = 2$ for panel (a): a monotone-causal OSSP where $|A_i|$ is finite and the transition cost function is piecewise-continuous and convex, and panel (b): a monotone-causal OSSP where $A_i = \Xi_2$ and the transition cost function is convex and continuous, yet not everywhere differentiable. In panel (a), $|A_i| = 6$ and $\delta_\star$ is easily computed via an application of (2.22). In panel (b), while $K(p)$ is not differentiable at $p = 0.25$, it is differentiable at $p = 0$ and $p = 1$, so the conditions of Proposition 1 are still satisfied. Thus, $\delta_\star$ follows immediately from (2.23).

The (δ-)MC criterion in Theorem 1 is local (i.e., stated using the cost function

and actions at each node separately) and does not require any information about the global topological structure of OSSP. As a result, it is sufficient but not necessary: for some OSSPs, a label-setting method might compute the value function correctly even if the conditions of Theorem 1 are violated. But these conditions are "sharp" in the sense that, whenever they are violated, there will exist a (possibly different) OSSP with the same local structure, for which label-setting methods would compute the value function incorrectly.

Theorem 3 (Sharpness of MC Condition)**.** *Suppose that the condition* (2.16) *of Theorem 1 is violated for some specific* $\delta \geq 0$, $\boldsymbol{x}_i \neq \boldsymbol{t}$, *non-deterministic action* $\boldsymbol{a}$, *and* $r \in \{1, ..., m = |\mathcal{I}(\boldsymbol{a})|\}$.

Then there exists a non-monotone δ*-causal OSSP in which*

1. $\boldsymbol{a} \in A_i$, *with* $\mathcal{I}(\boldsymbol{a}) = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_m\} \in X \setminus \{\boldsymbol{x}_i\}$ *and* $p(\boldsymbol{x}_i, \boldsymbol{a}, \boldsymbol{z}_j) = \xi_j > 0$ *for* $j = 1, ..., m$;
2. $\exists \tilde{\boldsymbol{a}} \in A_i$, *with* $\mathcal{I}(\tilde{\boldsymbol{a}}) = \mathcal{I}(\boldsymbol{a}) \setminus \{\boldsymbol{z}_r\}$, $C(\boldsymbol{x}_i, \tilde{\boldsymbol{a}}) = \check{C}(\boldsymbol{\gamma}_r)$, *and*
$$p(\boldsymbol{x}_i, \tilde{\boldsymbol{a}}, \boldsymbol{z}_j) = \gamma_{r,j} = \begin{cases} 0, & \text{if } j = r \\ \xi_j/(1-\xi_r), & \text{otherwise} \end{cases} \qquad \text{for } j = 1, ..., m;$$
3. $$C(\boldsymbol{x}_i, \boldsymbol{a}) \ < \ (1-\xi_r)C(\boldsymbol{x}_i, \tilde{\boldsymbol{a}}) + \xi_r \delta; \tag{2.24}$$
4. $\boldsymbol{a}$ *is the unique optimal action at* $\boldsymbol{x}_i$*;*
5. $U(\boldsymbol{x}_i) \ < \ U(\boldsymbol{z}_r) + \delta$.

For $\delta = 0$, *this implies that any label-setting method will produce a wrong answer for this OSSP. (*U_i *will be computed incorrectly since* $\boldsymbol{z}_r$ *will be still in* L *by the time* V_i *is last updated.) For* $\delta > 0$, *this means that a wrong answer would be produced by Dial's method with the bucket-width* δ.

Proof. To construct the simplest such OSSP example, we will assume that there is only one action $\hat{\boldsymbol{a}}$ available at every $\boldsymbol{x} \notin \{\boldsymbol{x}_i, \boldsymbol{t}\}$, and that $\hat{\boldsymbol{a}}$ leads to an immediate termination deterministically; i.e., $p(\boldsymbol{x}, \hat{\boldsymbol{a}}, \boldsymbol{t}) = 1$. This essentially allows us to prescribe any values to $\boldsymbol{x} \in X \backslash \{\boldsymbol{x}_i, \boldsymbol{t}\}$ since $U(\boldsymbol{x}) = C(\boldsymbol{x}, \hat{\boldsymbol{a}})$. (The resulting OSSP will be thus explicitly causal, but this does not imply the monotone causality.) At $\boldsymbol{x}_i$, we will consider the minimal relevant set of actions $A_i = \{\boldsymbol{a}, \tilde{\boldsymbol{a}}, \boldsymbol{e}_1, ..., \boldsymbol{e}_m\}$, where $p(\boldsymbol{x}_i, \boldsymbol{e}_j, \boldsymbol{z}_j) = 1$, with costs $C(\boldsymbol{x}_i, \boldsymbol{a})$ and $C(\boldsymbol{x}_i, \tilde{\boldsymbol{a}})$ already prescribed above and satisfying (2.24), while all $C(\boldsymbol{x}_i, \boldsymbol{e}_j)$ costs will be prescribed later.

The key quantity of interest in this proof is $B = U(\boldsymbol{z}_r) - \sum_{j=1, j\neq r}^{m} \gamma_{r,j} U(\boldsymbol{z}_j)$. To ensure that $\boldsymbol{a}$ is optimal and $\tilde{\boldsymbol{a}}$ is not, we must have

$$U(\boldsymbol{x}_i) = C(\boldsymbol{x}_i, \boldsymbol{a}) + \sum_{j=1}^{m} \xi_j U(\boldsymbol{z}_j) \; < \; C(\boldsymbol{x}_i, \tilde{\boldsymbol{a}}) + \sum_{j=1, j\neq r}^{m} \gamma_{r,j} U(\boldsymbol{z}_j),$$

which is equivalent to $B < \overline{B} = \frac{C(\boldsymbol{x}_i, \tilde{\boldsymbol{a}}) - C(\boldsymbol{x}_i, \boldsymbol{a})}{\xi_r}$. The condition $U(\boldsymbol{x}_i) \; < \; U(\boldsymbol{z}_r) + \delta$ is equivalent to $B > \underline{B} = \frac{C(\boldsymbol{x}_i, \boldsymbol{a}) - \delta}{1 - \xi_r}$. These bounds on B can be satisfied simultaneously since (2.24) is equivalent to $\underline{B} < \overline{B}$. E.g., we can choose $U(\boldsymbol{z}_r) = C(\boldsymbol{z}_r, \hat{\boldsymbol{a}}) = 2\delta + (\underline{B} + \overline{B})/2$ and $U(\boldsymbol{z}_j) = C(\boldsymbol{z}_j, \hat{\boldsymbol{a}}) = 2\delta$ for all $j \neq r$. Finally, the non-optimality of all $\boldsymbol{e}_j$'s (i.e., $U(\boldsymbol{x}_i) \; < \; C(\boldsymbol{x}_j, \boldsymbol{e}_j) + U(\boldsymbol{z}_j)$, $\forall j \neq r$) is easy to ensure by selecting $C(\boldsymbol{x}_j, \boldsymbol{e}_j)$ to be sufficiently large once all U values are already selected. □

To illustrate the connection to causality criteria previously derived for MSSPs, we now turn to homogeneous transition cost functions, which often arise naturally in applications. Recall that $H : \mathbb{R}^m \mapsto \mathbb{R}$ is *homogeneous of degree* d if $H(\beta \boldsymbol{q}) = |\beta|^d H(\boldsymbol{q})$ holds for all $\beta \in \mathbb{R}, \boldsymbol{q} = (q_1, ..., q_m) \in \mathbb{R}^m$. Further, when $H(\boldsymbol{q})$ is smooth, Euler's Homogeneous Function Theorem applies and $H(\boldsymbol{q}) = \frac{1}{d} \sum_{j=1}^{m} \; q_j \frac{\partial H}{\partial q_j}(\boldsymbol{q})$, $\forall \boldsymbol{q} \in \mathbb{R}^m$.

Focusing on a single node $\boldsymbol{x}$ and one of its modes $s = \{\boldsymbol{z}_1^s, \ldots, \boldsymbol{z}_{|s|}^s\} \subset X \backslash \{\boldsymbol{x}\}$, in

MSSPs one can choose any $\boldsymbol{\xi} \in \Xi_{|s|}$ for the desired probability distribution over s. The corresponding cost $C^s(\boldsymbol{x}, \boldsymbol{\xi})$ can be separately defined for each $\boldsymbol{x}$ and $s \in \mathcal{S}(\boldsymbol{x})$; when they are clear from context, it is convenient to abbreviate the notation to $C(\boldsymbol{\xi})$ and $m = |s|$. As shown in [105, Theorem 3.2], an MSSP is guaranteed to be monotone δ-causal if, in its every mode, $C(\boldsymbol{\xi})$ is continuously differentiable, homogeneous of degree d and satisfies

$$\xi_j > 0 \quad \Longrightarrow \quad \frac{\partial C}{\partial \xi_j}(\boldsymbol{\xi}) - (d-1)C(\boldsymbol{\xi}) \; > \; \delta \geq 0, \qquad \forall \boldsymbol{\xi} \in \Xi_m, \; j \in \{1, ..., m\}. \tag{2.25}$$

Note that this covers the example $C(\boldsymbol{\xi}) = \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2}$ over Ξ_3 already considered in Remark 1. This $C(\boldsymbol{\xi})$ is homogeneous of degree one with $\frac{\partial C}{\partial \xi_i}(\boldsymbol{\xi}) = \xi_i / \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2} > 0$ whenever $\xi_i > 0$. Thus, by [105, Theorem 3.2], this C is monotone causal (with $\delta = 0$), which can be also concluded from our Theorem 1.

Below we show that, for $m = 2$, the condition (2.25) implies our (2.20).

Proposition 2 (Monotone δ-Causality for Homogeneous $C(\boldsymbol{\xi})$ when $m = 2$)**.** *Suppose $m = 2$, and consider a transition cost function $C(\boldsymbol{\xi})$ which is convex, continuously differentiable, homogeneous of degree d, and satisfying* (2.25)*. Then it also satisfies*

$$C(\boldsymbol{\xi}) \; \geq \; \max(C_1\xi_1 + \delta\xi_2, \, \delta\xi_1 + C_2\xi_2) \quad \forall \boldsymbol{\xi} \in \Xi_2, \tag{2.26}$$

where C_1 and C_2 are the shorthands for the cost of deterministic actions $\boldsymbol{e}_1 = (1, 0)$ and $\boldsymbol{e}_2 = (0, 1)$.

Proof. Define $K : [0, 1] \mapsto \mathbb{R}$ by setting $K(p) = C(1 - p, p)$, as in Figure 2.3. Arguing by contradiction, assume that, for some $\bar{\boldsymbol{\xi}} = (1 - \bar{p}, \, \bar{p})$, we have $C(\bar{\boldsymbol{\xi}}) < \delta\bar{\xi}_1 + C_2\bar{\xi}_2$ or, equivalently, $K(\bar{p}) < \delta + \bar{p}(K(1) - \delta)$. By the convexity of K, this inequality must also hold for all $p \in [\bar{p}, 1)$ and implies $K'(1) > K(1) - \delta$. (Illustrated

by the solid purple curve in Figure 2.3(a).) From the chain rule, this is equivalent to

$$-\frac{\partial C}{\partial \xi_1}(\boldsymbol{e}_2) + \frac{\partial C}{\partial \xi_2}(\boldsymbol{e}_2) \ > \ C(\boldsymbol{e}_2) - \delta.$$

By Euler's Homogeneous Function Theorem, $dC(\boldsymbol{e}_2) = \frac{\partial C}{\partial \xi_2}(\boldsymbol{e}_2)$; thus,

$$\frac{\partial C}{\partial \xi_1}(\boldsymbol{e}_2) \ < \ (d-1)C(\boldsymbol{e}_2) + \delta.$$

Since C is continuously differentiable, this inequality also holds for $\boldsymbol{\xi} \in \Xi_2$ sufficiently close to $\boldsymbol{e}_2$, which contradicts (2.25).

If we instead assume $C(\bar{\boldsymbol{\xi}}) \ < \ C_1\bar{\xi}_1 + \delta\bar{\xi}_2$ (or, equivalently, $K(\bar{p}) \ < \ K(0) + \bar{p}(\delta - K(0))$), a similar argument leads to $K'(0) < \delta - K(0)$, yielding the same contradiction for $\boldsymbol{\xi} \in \Xi_2$ near $\boldsymbol{e}_1$. □

2.3 OSSPs and Continuous Optimal Control Problems

> "This world, concrete flowers grow."
>
> "luther (with sza)" by Kendrick Lamar, SZA

2.3.1 Background, prior work, and notation

SSPs are also quite useful in approximating solutions of stationary Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDEs), which arise naturally in areas as diverse as robotics, anomaly detection, optimal control, computational geometry, image registration, photolithography, and exploration geophysics [88].

We will focus on a simple example of a time-optimal control problem in an open domain $\Omega \subset \mathbb{R}^m$. With $\boldsymbol{y}(t)$ encoding the position at the time t, the dynamics is prescribed by $\boldsymbol{y}'(t) = f(\boldsymbol{y}(t), \boldsymbol{a}(t))\boldsymbol{a}(t)$, where $\boldsymbol{a}(t) \in S_{m-1} = \{\boldsymbol{a} \in \mathbb{R}^m \mid |\boldsymbol{a}| = 1\}$ is a unit vector describing our chosen direction of motion, while the speed $f : \Omega \times S_{m-1} \mapsto \mathbb{R}_+$ generally depends both on the position and the direction. We will generally assume that f is Lipschitz continuous and will leverage the properties of the corresponding *speed profile* $\mathcal{V}(\boldsymbol{x}) = \{f(\boldsymbol{x}, \boldsymbol{a})\boldsymbol{a} \mid \boldsymbol{a} \in S_{m-1}\}$, which describes all velocities achievable at $\boldsymbol{x}$. It is contained in a spherical shell with the inner and outer radii $F_1(\boldsymbol{x}) = \min_{\boldsymbol{a}} f(\boldsymbol{x}, \boldsymbol{a})$ and $F_2(\boldsymbol{x}) = \max_{\boldsymbol{a}} f(\boldsymbol{x}, \boldsymbol{a})$. We will further assume that $\exists F_1, F_2 \in \mathbb{R}$ such that $0 < F_1 \leq F_1(\boldsymbol{x}) \leq F_2(\boldsymbol{x}) < F_2$ holds for all $\boldsymbol{x} \in \Omega$. In addition to the $\boldsymbol{a}(\cdot)$-dependent time-to-$\partial\Omega$, we also incur a Lipschitz continuous exit-penalty $q : \partial\Omega \mapsto \mathbb{R}_{+,0}$. The value function $u(\boldsymbol{x})$ encodes the optimal overall time to exit starting from $\boldsymbol{x}$, formally defined by taking the infimum over all measurable controls $\boldsymbol{a}(\cdot)$. Our assumptions guarantee that u will be Lipschitz-continuous on Ω; moreover, if it is also smooth, a standard argument shows that u must also satisfy the HJB equation

$$\begin{aligned} H(\nabla u(\boldsymbol{x}), \boldsymbol{x}) = \min_{\boldsymbol{a} \in S_{m-1}} \{(\nabla u(\boldsymbol{x}) \cdot \boldsymbol{a}) f(\boldsymbol{x}, \boldsymbol{a})\} + 1 = 0, \quad \boldsymbol{x} \in \Omega \\ u(\boldsymbol{x}) = q(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega. \end{aligned} \tag{2.27}$$

More generally, even if u is not smooth, it can be recovered as a unique *viscosity solution* of this PDE [8]. If we further assume that $\mathcal{V}(\boldsymbol{x})$ is convex for all $\boldsymbol{x} \in \Omega$, this guarantees the existence of time-optimal trajectories, which are encoded by the characteristic curves of (2.27).

A frequently encountered subclass of problems has isotropic dynamics (i.e., with $f(\boldsymbol{x}, \boldsymbol{a}) = f(\boldsymbol{x})$) and allows for further simplification. In this case, $\mathcal{V}(\boldsymbol{x})$ is always just a ball, the optimal direction is known to be $\boldsymbol{a}_* = -\nabla u / |\nabla u|$, the optimal trajectories coincide with the gradient lines of u, and (2.27) reduces to an

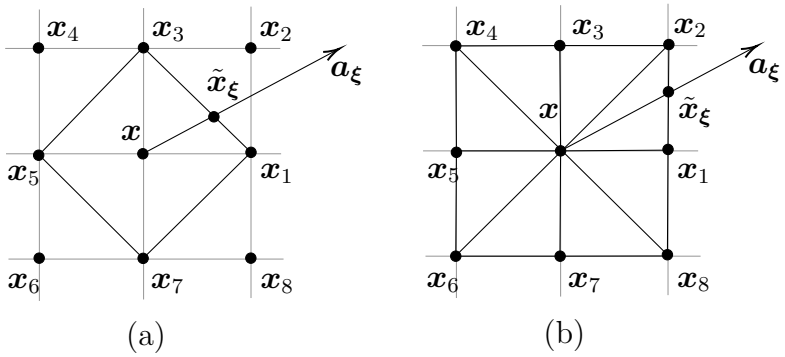


Figure 2.5: Two computational stencils based on a uniform Cartesian grid in $\mathbb{R}^2$. The value function is computed at $\boldsymbol{x}$, and $\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}$ is the new position after traveling in a straight line along the chosen direction of motion $\boldsymbol{a}_{\boldsymbol{\xi}}$ with speed $f(\boldsymbol{x}, \boldsymbol{a}_{\boldsymbol{\xi}})$. In panel (a), the modes are essentially quadrants: $\mathcal{S}(\boldsymbol{x}) = \{(\boldsymbol{x}_1, \boldsymbol{x}_3), (\boldsymbol{x}_3, \boldsymbol{x}_5), (\boldsymbol{x}_5, \boldsymbol{x}_7), (\boldsymbol{x}_7, \boldsymbol{x}_1)\}$. In panel (b), each quadrant is split into two modes: $\mathcal{S}(\boldsymbol{x}) = \{(\boldsymbol{x}_1, \boldsymbol{x}_2), (\boldsymbol{x}_2, \boldsymbol{x}_3), (\boldsymbol{x}_3, \boldsymbol{x}_4), (\boldsymbol{x}_4, \boldsymbol{x}_5), (\boldsymbol{x}_5, \boldsymbol{x}_6), (\boldsymbol{x}_6, \boldsymbol{x}_7), (\boldsymbol{x}_7, \boldsymbol{x}_8), (\boldsymbol{x}_8, \boldsymbol{x}_1)\}$.

Eikonal equation $|\nabla u(\boldsymbol{x})| f(\boldsymbol{x}) = 1$.

Efficient numerical methods for both the Eikonal equation and the general (anisotropic) HJB (2.27) have been an active area of research in the last 30 years. Many of such methods start with a semi-Lagrangian discretization [55, 43] on a grid or simplicial mesh X over $\bar{\Omega}$. Suppose $\mathcal{S}(\boldsymbol{x})$ is the set of simplexes used to build a computational stencil at $\boldsymbol{x}$, with each $s \in \mathcal{S}(\boldsymbol{x})$ having vertices $\boldsymbol{x}, \boldsymbol{z}_1^s, ..., \boldsymbol{z}_m^s$. (For $m = 2$ and a Cartesian grid, two examples of such stencils are shown in Figure 2.5.) For a chosen s, any $\boldsymbol{\xi} \in \Xi_m$ defines a "waypoint" $\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}} = \sum_j \xi_j \boldsymbol{z}_j^s$ on the opposite face of the simplex and the corresponding direction of motion $\boldsymbol{a}_{\boldsymbol{\xi}} = (\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}} - \boldsymbol{x})/|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}} - \boldsymbol{x}|$. Assuming we choose that direction and follow it from $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}$, the time it takes will be approximately $|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}} - \boldsymbol{x}|/f(\boldsymbol{x}, \boldsymbol{a}_{\boldsymbol{\xi}})$ and the remaining minimal time from there on can be approximated by linear interpolation as $u(\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}) \approx \sum_j \xi_j u(\boldsymbol{z}_j^s)$. Using $U(\boldsymbol{x})$ for the function approximating the true solution u at meshpoints/gridpoints, this

yields the following first-order accurate semi-Lagrangian discretization:

$$U(\boldsymbol{x}) = \min_{s\in\mathcal{S}} \min_{\boldsymbol{\xi}\in\Xi_m} \left\{ \frac{|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}} - \boldsymbol{x}|}{f(\boldsymbol{x}, \boldsymbol{a}_{\boldsymbol{\xi}})} + \sum_{j=1}^{m} \xi_j U(\boldsymbol{z}_j^s) \right\}, \qquad \forall \boldsymbol{x} \in X \cap \Omega; \tag{2.28}$$

$$U(\boldsymbol{x}) = q(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in X \cap \partial\Omega.$$

Assuming that $\partial\Omega$ is well resolved by a family of computational grids $\{X^h\}$, it is well-known that $U^h \to u$ uniformly under grid refinement [i.e., as $|X^h| \to +\infty$ with $\left(\max_{\boldsymbol{x}\in X^h} \max_{s\in\mathcal{S}(\boldsymbol{x})} \max_j |\boldsymbol{z}_j^s - \boldsymbol{x}|\right) \to 0$]. In the approach pioneered by Kushner and Dupuis [67], the same system of equations can be obtained by choosing a suitable controlled Markov process on X. Indeed, taking $C^s(\boldsymbol{x}, \boldsymbol{\xi}) = |\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}} - \boldsymbol{x}|/f(\boldsymbol{x}, \boldsymbol{a}_{\boldsymbol{\xi}})$, it is easy to see that the above is actually an MSSP defined in (2.10) with modes $s = \{\boldsymbol{z}_1^s, ..., \boldsymbol{z}_m^s\}$. The nodes $\boldsymbol{x} \in \partial\Omega$ are assumed to have a single deterministic action only, leading to $\boldsymbol{t}$ at the cost $q(\boldsymbol{x})$. In a yet another approach [93], the same system can be also derived through finite differences, approximating directional derivatives of u for each $(\boldsymbol{z}_j^s - \boldsymbol{x})$. Since the discretized system (2.28) is nonlinear and coupled, it has been traditionally treated with value iterations, but there was always a significant interest in speeding them up, often mirroring the label-correcting and label-setting methods on graphs. Here we focus on the latter, referring to [24, 25] for a review of the former, the hybrid versions, and parallelization.

The key label-setting ideas were first developed in the isotropic case. If $C^s(\boldsymbol{x}, \boldsymbol{a}_{\boldsymbol{\xi}})$ is monotone (δ-)causal for each s, the system (2.28) may be solved using noniterative label-setting methods. This was first done by Tsitsiklis [101, 102], who showed that any Eikonal discretized in a semi-Lagrangian way on the stencil in Figure 2.5(a) (or its m-dimensional version) can be solved by Dijkstra's method. Sethian independently derived the same result in an Eulerian (finite-difference) framework. Working in the context of isotropic front propagation problems, he

introduced a Dijkstra-like Fast Marching Method [89], which was soon generalized to handle higher-order accurate discretizations [90] and arbitrary simplicial meshes both in $\mathbb{R}^m$ and on manifolds [66, 91]. Tsitsiklis also showed that Dial's method with bucket width $\delta = \frac{h}{F_2\sqrt{2}}$ is applicable to an Eikonal discretized on the stencil in Figure 2.5(b) while its 3D version can be treated with the bucket width $\delta = \frac{h}{F_2\sqrt{3}}$. Another Dial-like approach was defined in [65]. In all these cases, the proofs have heavily relied on specific discretization approaches and stencil choices. The MSSPs introduced in [105] have later provided a unifying framework, in which it is easy to prove all related δ-MC conditions based on the properties of $C^s(\boldsymbol{x}, \boldsymbol{\xi})$.

Building label-setting methods to correctly solve the system (2.28) in the anisotropic setting is significantly more challenging since characteristics of the HJB PDE no longer coincide with the gradient lines of u. Aside from the small set of problems with "grid-aligned anisotropy" [77, 3], the causality is generally not guaranteed on usual stencils of the type shown in Figure 2.5. Sethian and Vladimirsky have handled this problem by introducing Dijkstra-like Ordered Upwind Methods (OUMs), in which the stencil is extended dynamically (at run-time) just enough to ensure the monotone causality [92]. In OUMs, the range of stencil extension is governed by an anisotropy coefficient[6] $\Upsilon(\boldsymbol{x}) = F_2(\boldsymbol{x})/F_1(\boldsymbol{x})$, which is equal to one (no stencil extension) in the isotropic case. The resulting algorithms are indifferent to the exact speed profile, using the same dynamic stencil extension for all $\mathcal{V}(\boldsymbol{x})$ that share the same $F_1(\boldsymbol{x})$ and $F_2(\boldsymbol{x})$. This yields the computational cost of $O(\Upsilon^{m-1} n \log n)$, where $n = |X|$ and $\Upsilon = \max_{\boldsymbol{x}} \Upsilon(\boldsymbol{x})$. Subsequently, a number of papers have aimed to reduce this stencil extension and pre-build a static causal stencil for each $\boldsymbol{x}$ based on the finer properties of $\mathcal{V}(\boldsymbol{x})$. The first general

[6]The anisotropy coefficient can be used to bound from above the angle between $-\nabla u(\boldsymbol{x})$ and the optimal direction of motion $\boldsymbol{a}_*(\boldsymbol{x})$. This is why the stencil extension proportional to $\Upsilon(\boldsymbol{x})$ is sufficient to ensure the monotone causality. However, later OUMs were also extended to approximate the solutions of HJB PDEs in which both $F_2(\boldsymbol{x})$ and $\Upsilon(\boldsymbol{x})$ might be infinite [22].

anisotropic δ-MC criterion was derived in [105] and then extended to non-smooth $\mathcal{V}(\boldsymbol{x})$ in [4]. Starting with a uniform Cartesian grid, Mirebeau showed how to build MC-stencils for elliptic and ellipsoidal speed profiles for $m = 2, 3$ with the uniform bound on stencil cardinality $|\mathcal{N}(\boldsymbol{x})|$ independent of Υ [71]. A related approach was also developed to build reduced-cardinality MC-stencils for general (smooth and convex) speed profiles in 2D [72] and 3D [35]. But in all of these cases, the MC criteria were based on the properties of the Hamiltonian H, on implicit conditions for the derivatives of f, or on analytic properties of a function embedding $\mathcal{V}$ as a level set. Below we show that the criteria for OSSPs developed in §2.2.3 yield surprisingly simple δ-MC conditions for 2D and 3D stencils, using only the basic geometric properties of $\mathcal{V}$.

Simplified Notation: Throughout the remainder of this section, we make use of the following notational conventions. Without loss of generality, we assume that the coordinate system is centered at $\boldsymbol{x} = \mathbf{0}$ and will mostly focus on a single simplex $s \in \mathcal{S}(\boldsymbol{x})$ with vertices $\mathbf{0}, \boldsymbol{z}_1, ..., \boldsymbol{z}_m$. The speed, velocity, and cost for the direction $\boldsymbol{a}_{\boldsymbol{\xi}} = \tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}/|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}|$ will be denoted by $f(\boldsymbol{\xi}) = f(\boldsymbol{x}, \boldsymbol{a}_{\boldsymbol{\xi}})$, $\boldsymbol{v}(\boldsymbol{\xi}) = f(\boldsymbol{\xi})\boldsymbol{a}_{\boldsymbol{\xi}}$, and $C(\boldsymbol{\xi}) = C^s(\boldsymbol{\xi}) = |\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}|/f(\boldsymbol{\xi})$ respectively. A similar notation ($f_j, \boldsymbol{v}_j$, and $C_j = |\boldsymbol{z}_j|/f_j$) will also be used for the speed, velocity, and cost for each direction $\boldsymbol{z}_j/|\boldsymbol{z}_j|$ toward the corresponding vertex. We will use $\mathcal{V}_s = \{\boldsymbol{v} \in \mathcal{V} \mid \boldsymbol{v} = \sum_{j=1}^m \theta_j \boldsymbol{v}_j, \text{ all } \theta_j \geq 0\}$ to denote the part of $\mathcal{V}$ falling into this simplex.

Observation 2. *If $\mathcal{V}(\boldsymbol{x})$ is convex then the function $C(\boldsymbol{\xi})$ is convex in every simplex $s \in \mathcal{S}(\boldsymbol{x})$.*

Proof. Suppose $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2 \in \Xi_m$ and $\boldsymbol{\phi} = \xi_1\boldsymbol{\eta}_1 + \xi_2\boldsymbol{\eta}_2$ for some $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \Xi_2$. The corresponding waypoints on the simplex $(\boldsymbol{z}_1, ..., \boldsymbol{z}_m)$ are $\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_i} = \sum_{j=1}^m \eta_{i,j}\boldsymbol{z}_j$ for $i = 1, 2$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{\phi}} = \xi_1\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_1} + \xi_2\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_2}$. The respective directions are $\boldsymbol{a}_{\boldsymbol{\eta}_i} = \tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_i}/|\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_i}|$ and

$\boldsymbol{a}_{\boldsymbol{\phi}} = \tilde{\boldsymbol{x}}_{\boldsymbol{\phi}}/|\tilde{\boldsymbol{x}}_{\boldsymbol{\phi}}|$. The velocity corresponding to $\boldsymbol{a}_{\boldsymbol{\phi}}$ is

$$\boldsymbol{v}(\boldsymbol{\phi}) \;=\; f(\boldsymbol{\phi})\frac{\tilde{\boldsymbol{x}}_{\boldsymbol{\phi}}}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\phi}}|} \;=\; \frac{f(\boldsymbol{\phi})}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\phi}}|}\left(\xi_1\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_1} + \xi_2\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_2}\right) \;=\; \theta_1\boldsymbol{v}(\boldsymbol{\eta}_1) + \theta_2\boldsymbol{v}(\boldsymbol{\eta}_2),$$

where $\theta_1, \theta_2 \geq 0$ since $\boldsymbol{v}(\boldsymbol{\phi})$ belongs to the angular sector formed by $\boldsymbol{v}(\boldsymbol{\eta}_1)$ and $\boldsymbol{v}(\boldsymbol{\eta}_2)$.

Since $\boldsymbol{v}(\boldsymbol{\eta}_i) = f(\boldsymbol{\eta}_i)\frac{\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_i}}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_i}|}$, we see that $\frac{f(\boldsymbol{\phi})\xi_i}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\phi}}|} = \theta_i\frac{f(\boldsymbol{\eta}_i)}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_i}|}$ for $i = 1, 2$. By definition of C, this is equivalent to

$$\theta_1 = \frac{\xi_1 C(\boldsymbol{\eta}_1)}{C(\boldsymbol{\phi})} \quad \text{and} \quad \theta_2 = \frac{\xi_2 C(\boldsymbol{\eta}_2)}{C(\boldsymbol{\phi})}. \tag{2.29}$$

The convexity of $\mathcal{V}(\boldsymbol{x})$ implies $\theta_1 + \theta_2 \geq 1$ or, equivalently, $\xi_1 C(\boldsymbol{\eta}_1) + \xi_2 C(\boldsymbol{\eta}_2) \geq C(\boldsymbol{\phi})$. □

We also note that, if $f(\boldsymbol{\xi})$ is written as a function of direction (i.e., homogeneous of degree zero) then $C(\boldsymbol{\xi}) = |\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}|/f(\boldsymbol{\xi})$ is homogeneous of degree one, making MC conditions such as (2.25) applicable. But we will see that Theorems 1 and 2 are more useful in relating MC to geometry.

Remark 2. *In the case of a convex $C(\boldsymbol{\xi})$ homogeneous of degree one, the δ-MC criterion* (2.16) *from Theorem 1 simplifies to*

$$C(\boldsymbol{\xi}) \geq C\left(\boldsymbol{\xi} - \xi_r\boldsymbol{e}_r\right) + \xi_r\delta, \qquad \forall r \in \{1, ..., m\}, \tag{2.30}$$

where $\boldsymbol{e}_r$'s are the canonical basis vectors. While $(\boldsymbol{\xi} - \xi_r\boldsymbol{e}_r)$ is not a probability distribution on successor nodes, this formula is still very suggestive. For the specific case of $m = 2$ and $\delta = 0$, a similar relation is the principle behind the notion of "F-acuteness" introduced in [72].

2.3.2 Monotone Causality in $\mathbb{R}^2$

Theorem 4 (Sufficient Causality Condition on $\mathcal{V}$ in $\mathbb{R}^2$)**.** *$C(\boldsymbol{\xi})$ is monotone causal on the simplex formed by $(\boldsymbol{x} = \mathbf{0}, \boldsymbol{z}_1, \boldsymbol{z}_2)$ if $\mathcal{V}_s(\boldsymbol{x})$ is fully contained in a parallelogram with vertices at $\mathbf{0}, \boldsymbol{v}_1, \boldsymbol{v}_2$, and $\boldsymbol{v}_1 + \boldsymbol{v}_2$.*

If the above property holds for every $s \in \mathcal{S}(\boldsymbol{x})$ (i.e., if the entire $\mathcal{V}(\boldsymbol{x})$ is contained in the union of such parallelograms), then the entire stencil is causal at $\boldsymbol{x}$. If this is the case $\forall \boldsymbol{x} \in X$, then Dijkstra's method is applicable.

Proof. Using the same argument as in Observation 2 but with $m = 2$ and deterministic $\boldsymbol{\eta}_i = \boldsymbol{e}_i$ for $i = 1, 2$, we see that $\boldsymbol{\phi} = \boldsymbol{\xi}$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{\eta}_i} = \boldsymbol{z}_i$. Thus, (2.29) reduces to

$$\theta_1 = \frac{\xi_1 C_1}{C(\boldsymbol{\xi})} \quad \text{and} \quad \theta_2 = \frac{\xi_2 C_2}{C(\boldsymbol{\xi})}. \tag{2.31}$$

If $\boldsymbol{v}(\boldsymbol{\xi}) = \theta_1 \boldsymbol{v}_1 + \theta_2 \boldsymbol{v}_2$ lies in that parallelogram, this means that $\theta_i \leq 1$ for $i = 1, 2$. Combined with (2.31), this is equivalent to (2.20) with $\delta = 0$ and Theorem 2 applies. □

By Theorem 4, the standard 8-point stencil is guaranteed to be monotone causal at $\boldsymbol{x}$ for all examples of speed profiles in Figure 2.6(a-d). The non-convex example in Figure 2.6(c) might seem paradoxical. It is well-known that replacing $\mathcal{V}$ with its convex hull does not change the value function since the velocities which are not on the convex hull boundary are never optimal for any ∇u. Thus, these eight "deterministic directions" (corresponding to transitions to eight closest gridpoints) will never be optimal in that continuous path-planning problem, but might be still optimal in a monotone causal OSSP that discretizes it. This is an artifact of our simplex-by-simplex approach of enforcing MC – the non-convexity (and the

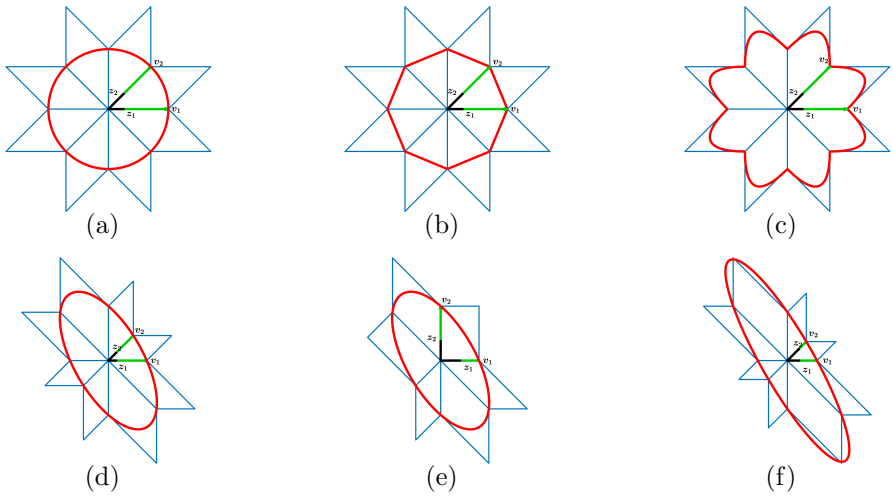


Figure 2.6: Bounding parallelograms for several speed profiles on the 8-point stencil shown in Figure 2.5(b). The speed profile is depicted in red. The vectors along directions $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are shown in black, and the velocities $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ along their respective directions $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are shown in green. Panel (a): Isotropic circular speed profile. Panel (b): Octagonal speed profile. Panel (c): A non-convex speed profile. Panels (d) and (e): Tilted elliptical speed profile. Panel (e) illustrates the result of reducing the standard 8-point stencil used in the previous examples to a 6-point stencil by removing any superfluous nodes $\boldsymbol{z}_j$ which are not crucial to the monotone-causality of the stencil. The resulting MC stencil is not symmetric relative to $\boldsymbol{x}$ despite the symmetry of the speed profile $\mathcal{V}(\boldsymbol{x})$. Panel (f): Tilted elliptical speed profile with an aspect ratio twice that of the ellipse in panels (d) and (e). For this aspect ratio, monotone-causality no longer holds on the standard 8-point stencil. The condition in Theorem 4 is violated on two simplexes in $\mathcal{S}(\boldsymbol{x})$.

irrelevance of a large number of directions) is not known if we can only consider the portion of $\mathcal{V}$ falling between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Figure 2.6(e) illustrates MC-stencil reduction by merging two pairs of simplexes from panel (d). As shown in [71], an MC-stencil based on only 6 Cartesian grid points can be found for any elliptic $\mathcal{V}$, but for ellipses with larger aspect ratio and generic orientation such causal stencils become increasingly non-local. Figure 2.6(f) shows that a local/standard 8-point stencil is not MC when the ellipse's aspect ratio is sufficiently large.

When $\mathcal{V}$ is known to be convex, checking the causality condition (2.31) becomes a lot simpler, since it can be replaced by a pair of inequalities verified at $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. The following is an alternative geometric proof of the already known criterion [72, Lemma 2.2].

Proposition 3 (Causality and Tangent Information in $\mathbb{R}^2$)**.**
Consider a simplex formed by $\boldsymbol{x} = \mathbf{0}, \boldsymbol{z}_1, \boldsymbol{z}_2$. Suppose $\mathcal{V}$ is convex and smooth at $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, with $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$ denoting the respective outward normal vectors to $\mathcal{V}$. Then $C(\boldsymbol{\xi})$ is guaranteed to be monotone causal within this simplex if

$$\boldsymbol{v}_1 \cdot \hat{\boldsymbol{n}}_2 \geq 0 \quad \textit{and} \quad \boldsymbol{v}_2 \cdot \hat{\boldsymbol{n}}_1 \geq 0. \tag{2.32}$$

Proof. By convexity, the angular sector of $\mathcal{V}$ is contained in a quadrilateral formed by $\boldsymbol{v}_1, \boldsymbol{v}_2$, and two respective tangent lines T_1 and T_2. The inequalities (2.32) ensure that this quadrilateral is fully contained in the parallelogram with vertices $(\mathbf{0}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_1 + \boldsymbol{v}_2)$; see Figure 2.8(a). □

In the Eikonal case, it is well known that the stencil is MC at $\boldsymbol{x}$ if and only if each angle between $(\boldsymbol{z}_i - \boldsymbol{x})$ and $(\boldsymbol{z}_j - \boldsymbol{x})$ is non-obtuse in each $s \in \mathcal{S}(\boldsymbol{x})$ [91]. Since the isotropic $\mathcal{V}$ is just a ball, in 2D this is a simple consequence of (2.32). For smooth convex anisotropic speed profiles in 2D, the above criterion also provides an easy to verify alternative to the "negative-gradient-acuteness" condition developed in [4].

2.3.3 Monotone Causality in $\mathbb{R}^3$

It might seem intuitive that the 3D version of the MC condition in Theorem 4 should require the portion of $\mathcal{V}$ falling into the simplex $(\boldsymbol{x} = \mathbf{0}, \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3)$ to lie

within a parallelepiped $\Pi = \{\sum_{j=1}^{3} \theta_j \boldsymbol{v}_j \mid \theta_j \in [0,1] \text{ for } j = 1,2,3\}$. However, this condition turns out to be insufficient. As a motivating (counter-)example, consider a simplex s corresponding to a positive orthant, based on unit vectors $\boldsymbol{z}_j = \boldsymbol{e}_j$ with $C_j = 1$ for $j = 1,2,3$. We will assume that in this simplex there are only 4 principal directions of motion ($\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$, and along $\hat{\boldsymbol{\xi}} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$), and for the other directions in s the velocity is obtained by taking a convex combination of those four. In other words, the relevant part of the speed profile $\mathcal{V}_s$ is a union of three triangles with respective vertices $\{\boldsymbol{v}(\hat{\boldsymbol{\xi}}), \boldsymbol{v}_1, \boldsymbol{v}_2\}$, $\{\boldsymbol{v}(\hat{\boldsymbol{\xi}}), \boldsymbol{v}_2, \boldsymbol{v}_3\}$, and $\{\boldsymbol{v}(\hat{\boldsymbol{\xi}}), \boldsymbol{v}_1, \boldsymbol{v}_3\}$. In this case, Π is just a unit cube and selecting $f(\hat{\boldsymbol{\xi}}) = \sqrt{3}$ ensures $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}(\hat{\boldsymbol{\xi}})\} \subset \mathcal{V}_s \subset \Pi$. To show that this simplex is not MC for the described $\mathcal{V}$, it is enough to find some $U(\boldsymbol{z}_j)$ values such that $\hat{\boldsymbol{\xi}}$ is uniquely optimal but $U(\boldsymbol{x}) < \max_j U(\boldsymbol{z}_j)$. It is easy to check that this is the case with $U(\boldsymbol{z}_1) = 0$, $U(\boldsymbol{z}_2) = \frac{1}{4}$, $U(\boldsymbol{z}_3) = \frac{3}{2}$. Since $|\tilde{\boldsymbol{x}}_{\hat{\boldsymbol{\xi}}} - \boldsymbol{x}| = |\hat{\boldsymbol{\xi}}| = \frac{1}{\sqrt{3}}$, we have $C^s(\hat{\boldsymbol{\xi}}) = |\tilde{\boldsymbol{x}}_{\hat{\boldsymbol{\xi}}} - \boldsymbol{x}|/f(\hat{\boldsymbol{\xi}}) = \frac{1}{3}$. Thus, $U(\boldsymbol{x}) = \frac{1}{3} + \frac{1}{3}U(\boldsymbol{z}_1) + \frac{1}{3}U(\boldsymbol{z}_2) + \frac{1}{3}U(\boldsymbol{z}_3) = \frac{11}{12} < \min_j\{C_j + U_j\} = 1$, and so $\hat{\boldsymbol{\xi}}$ is uniquely optimal. On the other hand, $U(\boldsymbol{x}) = \frac{11}{12} < U(\boldsymbol{z}_3)$, with the latter not available in time if we tried to compute $U(\boldsymbol{x})$ using Dijkstra's method.

The correct sufficient condition is more subtle and uses oblique projections. Before stating it, we need to introduce some additional notation. If (i,j,k) is any permutation of $(1,2,3)$, we define the parallelograms $\Pi^k = \{\theta_1 \boldsymbol{v}_i + \theta_2 \boldsymbol{v}_j \mid \theta_1, \theta_2 \in [0,1]\}$ and the parts of $\mathcal{V}_s$ in those same planes $\mathcal{V}^k = \{(\theta_1 \boldsymbol{v}_i + \theta_2 \boldsymbol{v}_j) \in \mathcal{V} \mid \theta_1, \theta_2 \geq 0\}$. Finally, we also define the corresponding generalized semi-infinite cylinder as $\Psi^k = \{\theta_1 \boldsymbol{v}_k + \theta_2 \boldsymbol{v} \mid \boldsymbol{v} \in \mathcal{V}^k, \theta_1 \geq 0, \theta_2 \in [0,1]\}$.

Theorem 5 (Causality Condition on $\mathcal{V}$ in $\mathbb{R}^3$)**.** *A simplex s is guaranteed to be monotone causal if $\mathcal{V}$ is convex, $\mathcal{V}_s \subset \left(\Psi^1 \cap \Psi^2 \cap \Psi^3\right)$, and $\mathcal{V}^r \subset \Pi^r$ for $r = 1,2,3$.*

Proof. We will prove that the resulting C satisfies the conditions of Theorem 1

with $\delta = 0$. For deterministic actions $\boldsymbol{e}_i$ this follows directly from **(A5')**.

If $\boldsymbol{v}(\boldsymbol{\xi}) \in \mathcal{V}^1$, then $\mathcal{I}(\boldsymbol{\xi}) = 2$ and both of its oblique projections are deterministic: $\boldsymbol{\gamma}_2 = \boldsymbol{e}_3$ and $\boldsymbol{\gamma}_3 = \boldsymbol{e}_2$. So, (2.16) is equivalent to $C(\boldsymbol{\xi}) \geq \xi_2 C_2$ and $C(\boldsymbol{\xi}) \geq \xi_3 C_3$. But both of these are guaranteed by $\mathcal{V}^1 \in \Pi^1$; see Theorem 4. The same argument also covers the cases $\boldsymbol{v}(\boldsymbol{\xi}) \in \mathcal{V}^2$ and $\boldsymbol{v}(\boldsymbol{\xi}) \in \mathcal{V}^3$.

If $\boldsymbol{\xi} \in \text{int}(\Xi_3)$, then $\boldsymbol{v}(\boldsymbol{\xi}) = \sum_{j=1}^{3} \theta_j \boldsymbol{v}_j$. A similar derivation to the one in Observation 2 and Theorem 4 shows

$$\theta_j \;=\; \frac{\xi_j C_j}{C(\boldsymbol{\xi})} \;=\; \xi_j \frac{f(\boldsymbol{\xi})}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}|} \frac{|\boldsymbol{z}_j|}{f_j} \;>\; 0 \qquad \forall j. \tag{2.33}$$

So, for any $r \in \{1, 2, 3\}$, we have the oblique projection $\boldsymbol{\gamma}_r$, the corresponding waypoint $\tilde{\boldsymbol{x}}_{\boldsymbol{\gamma}_r} = \sum_{j \neq r} \gamma_{r,j} \boldsymbol{z}_j$, and velocity $\boldsymbol{v}(\boldsymbol{\gamma}_r) = \sum_{j \neq r} \frac{\gamma_{r,j} \boldsymbol{z}_j}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\gamma}_r}|} f(\boldsymbol{\gamma}_r)$. Thus,

$$\begin{aligned}
\boldsymbol{v}(\boldsymbol{\xi}) &= \theta_r \boldsymbol{v}_r + \sum_{j \neq r} \theta_j \boldsymbol{v}_j \;=\; \theta_r \boldsymbol{v}_r + \frac{f(\boldsymbol{\xi})}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}|} \sum_{j \neq r} \xi_j \frac{|\boldsymbol{z}_j|}{f_j} \boldsymbol{v}_j \;=\; \theta_r \boldsymbol{v}_r + \frac{f(\boldsymbol{\xi})}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}|} \sum_{j \neq r} \xi_j \boldsymbol{z}_j \\
&= \theta_r \boldsymbol{v}_r + (1 - \xi_r) \frac{f(\boldsymbol{\xi})}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}|} \sum_{j \neq r} \gamma_{r,j} \boldsymbol{z}_j \;=\; \theta_r \boldsymbol{v}_r + \left[(1 - \xi_r) \frac{f(\boldsymbol{\xi})}{|\tilde{\boldsymbol{x}}_{\boldsymbol{\xi}}|} \frac{|\tilde{\boldsymbol{x}}_{\boldsymbol{\gamma}_r}|}{f(\boldsymbol{\gamma}_r)} \right] \boldsymbol{v}(\boldsymbol{\gamma}_r), \\
&= \theta_r \boldsymbol{v}_r + \left[(1 - \xi_r) \frac{C(\boldsymbol{\gamma}_r)}{C(\boldsymbol{\xi})} \right] \boldsymbol{v}(\boldsymbol{\gamma}_r),
\end{aligned}$$

where the sequence of equalities uses (2.33), the definition of $\boldsymbol{v}_j$, the definition of $\boldsymbol{\gamma}_r$, the representation of $\boldsymbol{v}(\boldsymbol{\gamma}_r)$, and the formula for C. Since $\mathcal{V}_s \subset \Psi^r$, the coefficient in front of $\boldsymbol{v}(\boldsymbol{\gamma}_r)$ has to be less than or equal to 1. By Observation 2, the convexity of $\mathcal{V}_s$ guarantees $\check{C}(\boldsymbol{\gamma}_r) = C(\boldsymbol{\gamma}_r)$, and thus $C(\boldsymbol{\xi}) \geq (1 - \xi_r) \check{C}(\boldsymbol{\gamma}_r)$. $\square$

Another interesting question is the amount of anisotropic distortion compatible with the monotone causality of a standard Cartesian stencil in $\mathbb{R}^m$. For $m = 2$ and the stencil in Figure 2.5(a), $\mathcal{V}(\boldsymbol{x})$ simply needs to lie in a rectangle whose vertices are at $(-f_{\text{W}}, f_{\text{N}})$, $(f_{\text{E}}, f_{\text{N}})$, $(f_{\text{E}}, -f_{\text{S}})$, $(-f_{\text{W}}, -f_{\text{S}})$. Here the subscripts indicate directions of motion and the coordinate system is centered at $\boldsymbol{x}$.

But as we already saw, with $m = 3$ and the standard 6 point stencil, the answer is more complicated and depends on $f(\boldsymbol{a})$ in all directions parallel to coordinate planes (comprising $\mathcal{V}^k$'s). For the example at the start of this subsection, $\Big(\Psi^1 \cap \Psi^2 \cap \Psi^3\Big) = \{(v_1, v_2, v_3) \in \mathbb{R}^3_{+,0} \mid \max(v_1+v_2, v_2+v_3, v_1+v_3) \le 1\}$. Thus, Theorem 5 guarantees that this simplex is MC only if $f(\hat{\boldsymbol{\xi}}) \le \sqrt{3}/2$. For another interesting 3D example, consider a speed profile that is known to be isotropic (i.e., $f(\boldsymbol{a}) = F$) but only for $\boldsymbol{a}$'s that are parallel to coordinate planes (x_1, x_2), (x_2, x_3), and (x_1, x_3). Theorem 5 guarantees that the standard 6-point stencil will be monotone causal if the full $\mathcal{V}(\boldsymbol{x})$ lies within a *tricylinder* $\Big(\Psi^1 \cap \Psi^2 \cap \Psi^3\Big) = \{(v_1, v_2, v_3) \in \mathbb{R}^3 \mid \max(v_1^2 + v_2^2, v_2^2 + v_3^2, v_1^2 + v_3^2) \le F^2\}$.

We also note that in principle the argument in Theorem 5 can be recursively extended to any $m > 3$. (The second half of the proof can be adopted for any m, but the conditions on $\partial\Xi_m$ will be more complicated. E.g., for $m = 4$, the current theorem provides conditions for all 3-dimensional faces of Ξ_4.) We do not pursue this extension because the criteria would not be as geometrically suggestive and because grid discretizations of (2.27) are most commonly used when $m = 2$ or 3.

Similarly to the 2D case, for a smooth convex speed profile, the conditions of Theorem 5 can be replaced by criteria verified on the facets of the simplex only. The following Proposition is a 3D analog of Proposition 3. It also provides an explicit and easier to verify version of the 3D causality condition derived in [35, Section 2.3].

Let $\hat{\boldsymbol{n}}(\tilde{\boldsymbol{v}})$ denote an outward pointing normal vector to $\mathcal{V}$ at any $\tilde{\boldsymbol{v}} \in \mathcal{V}^r$. The support hyperplane to $\mathcal{V}$ at $\tilde{\boldsymbol{v}}$ is specified by $(\boldsymbol{v} - \tilde{\boldsymbol{v}}) \cdot \hat{\boldsymbol{n}}(\tilde{\boldsymbol{v}}) = 0$. We define the r-th "tangential envelope" of $\mathcal{V}_s$ as

$$\Phi^r = \{\boldsymbol{v} = \theta_1\boldsymbol{v}_1 + \theta_2\boldsymbol{v}_2 + \theta_3\boldsymbol{v}_3 \mid \theta_i \ge 0 \text{ for } i = 1, 2, 3 \text{ and } (\boldsymbol{v} - \tilde{\boldsymbol{v}}) \cdot \hat{\boldsymbol{n}}(\tilde{\boldsymbol{v}}) \le 0 \text{ for all } \tilde{\boldsymbol{v}} \in \mathcal{V}^r\}\,.$$

Proposition 4 (Normal MC verification in 3D.)**.** *Suppose that the speed profile* $\mathcal{V}_s$ *is convex and smooth. A simplex* $s \in \mathcal{S}(\boldsymbol{x})$ *with vertices* $\boldsymbol{x} = \mathbf{0}, \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3$ *and corresponding velocities* $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ *is guaranteed to be monotone causal if*

1. $\boldsymbol{v}_i \cdot \hat{\boldsymbol{n}}(\boldsymbol{v}_j) \geq 0, \quad \forall i, j \in \{1, 2, 3\};$
2. $\boldsymbol{v}_r \cdot \hat{\boldsymbol{n}}(\tilde{\boldsymbol{v}}) \geq 0, \quad \forall r \in \{1, 2, 3\},\ \tilde{\boldsymbol{v}} \in \mathcal{V}^r.$

Proof. By Proposition 3, the first condition implies that $\mathcal{V}^r \subset \Pi^r$ for all $r \in \{1, 2, 3\}$.

We note that $\mathcal{V}_s \subset \Phi^r$ due to convexity of $\mathcal{V}$ and that $\partial\Phi^r$ is tangential to $\mathcal{V}$ along the $\mathcal{V}^r \subset \partial\Psi^r$. Thus, $\mathcal{V}_s \subset \Psi^r$ if and only if $\Phi^r \subset \Psi^r$, which is ensured by the second condition above. This shows that s satisfies the criteria listed in Theorem 5. □

2.3.4 Monotone δ-Causality in $\mathbb{R}^2$

The MC conditions derived above are fully based on $\mathcal{V}$ and the set of directions represented in a particular simplex; i.e., the distances $|\boldsymbol{z}_i - \boldsymbol{x}|$ were irrelevant in specifying the MC-constrained set for $\mathcal{V}_s$. This is not the case for monotone δ-causality as we show below for $m = 2$.

Theorem 6 (δ-Causality Condition on $\mathcal{V}$ in $\mathbb{R}^2$)**.** *Suppose that a simplex* $s \in \mathcal{S}(\boldsymbol{x})$ *formed by vertices* $\boldsymbol{x} = \mathbf{0}$, $\boldsymbol{z}_1$, *and* $\boldsymbol{z}_2$ *is monotone causal for a specific* $\mathcal{V}(\boldsymbol{x})$ *and that* $0 < \delta \leq \min(C_1 = \frac{|\boldsymbol{z}_1|}{f_1}, C_2 = \frac{|\boldsymbol{z}_2|}{f_2})$. *This simplex is monotone* δ*-causal if* $\mathcal{V}_s$ *is contained in a quadrilateral with vertices at* $\boldsymbol{x} = \mathbf{0}$, $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, *and* $\boldsymbol{w}(\delta) =$

$\theta_1^{\#}(\delta)\boldsymbol{v}_1 + \theta_2^{\#}(\delta)\boldsymbol{v}_2$, *where*

$$\theta_1^{\#}(\delta) = \frac{C_1(C_2 - \delta)}{C_1C_2 - \delta^2} \quad \textit{and} \quad \theta_2^{\#}(\delta) = \frac{C_2(C_1 - \delta)}{C_1C_2 - \delta^2}. \tag{2.34}$$

Proof. For every $\boldsymbol{\xi} \in \Xi_2$, the corresponding velocity is $\boldsymbol{v}(\boldsymbol{\xi}) = \theta_1\boldsymbol{v}_1 + \theta_2\boldsymbol{v}_2$, where $\theta_1, \theta_2 \geq 0$. By Theorem 2, monotone δ-causality of $C(\boldsymbol{\xi})$ is guaranteed if

$$C(\boldsymbol{\xi}) \geq \xi_1 C_1 + \xi_2\delta \quad \text{and} \quad C(\boldsymbol{\xi}) \geq \xi_1\delta + \xi_2 C_2.$$

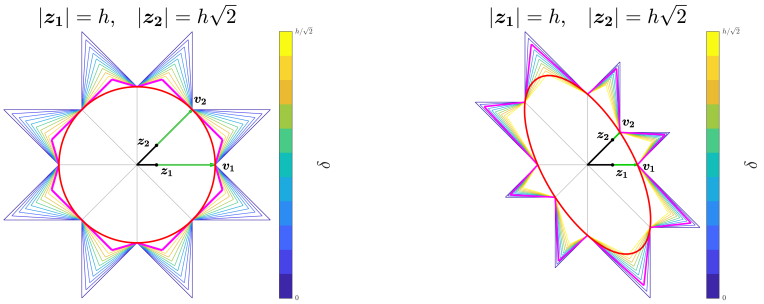
Dividing through by $C(\boldsymbol{\xi})$ and using (2.31), we see that the equivalent conditions are

$$1 \geq \theta_1 + \frac{\theta_2\delta}{C_2} \quad \text{and} \quad 1 \geq \theta_2 + \frac{\theta_1\delta}{C_1}.$$

Each of these inequalities specifies a half-plane where $\boldsymbol{v}(\boldsymbol{\xi})$ is allowed to lie, with two restriction lines L_1 (passing through the points $\boldsymbol{v}_1$ and $\frac{C_2}{\delta}\boldsymbol{v}_2 = \frac{1}{\delta}\boldsymbol{z}_2$) and L_2 (passing through the points $\boldsymbol{v}_2$ and $\frac{C_1}{\delta}\boldsymbol{v}_1 = \frac{1}{\delta}\boldsymbol{z}_1$). Solving for the intersection point $\boldsymbol{w}(\delta)$, we obtain (2.34). □

It is easy to see that this quadrilateral is fully contained in the MC parallelogram Π and that $\boldsymbol{w}(\delta) \to (\boldsymbol{v}_1 + \boldsymbol{v}_2)$ as $\delta \to 0$. If $\delta = \min(C_1, C_2)$, one of the restriction lines coincides with the straight line connecting $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, and the δ-MC region becomes a triangle $(\boldsymbol{0}, \boldsymbol{v}_1, \boldsymbol{v}_2)$, ensuring that either $\frac{\boldsymbol{z}_1}{|\boldsymbol{z}_1|}$ or $\frac{\boldsymbol{z}_2}{|\boldsymbol{z}_2|}$ will be always optimal.

Similarly to the MC setting, if $C^s(\boldsymbol{\xi})$ is δ-causal for all $\boldsymbol{s} \in \mathcal{S}(\boldsymbol{x})$, then the entire stencil is δ-MC at $\boldsymbol{x}$ and the speed profile $\mathcal{V}(\boldsymbol{x})$ is inscribed within the union of bounding δ-MC quadrilaterals which encompass all possible directions of motion. Figure 2.7 displays this result for two speed profiles on the 8-point stencil depicted in Figure 2.5(b).



(a) (b)

Figure 2.7: Bounding δ-MC "sunflowers" for the 8-point stencil (grid spacing $h = 1$ for $|\boldsymbol{z}_1| = h$, $|\boldsymbol{z}_2| = h\sqrt{2}$) found by fixing $f_i(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{a}_{\boldsymbol{e}_i})$ for 8 directions $(\boldsymbol{x}_i - \boldsymbol{x})$ and varying δ. The f_i are chosen to match an isotropic speed profile in panel (a) and an anisotropic elliptical speed profile in panel (b); shown in red. The direction vectors corresponding to $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are shown in black, while the velocity vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are shown in green. The bounding sunflower corresponding to the maximum $\Delta(\boldsymbol{x})$ applicable for this ($\mathcal{V}$, stencil) combination is shown in magenta.

The next obvious question is on the largest δ such that a specific simplex is δ-MC. The answer can be found from Proposition 1, but the geometric derivation provided below is more natural.

Proposition 5 (Max δ for Smooth, Strictly Convex $\mathcal{V}$ in $\mathbb{R}^2$)**.** *Suppose that $m = 2$ and $\mathcal{V}(\boldsymbol{x})$ is smooth and strictly convex. Suppose also that a simplex $s \in \mathcal{S}(\boldsymbol{x})$ formed by $\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{z}_1$, and $\boldsymbol{z}_2$ is MC for this $\mathcal{V}(\boldsymbol{x})$. The maximal δ for which this simplex is δ-MC can be computed as $\Delta(\boldsymbol{x}, s) = \min\{\delta_1, \delta_2\}$, where*

$$\delta_1 = C_2 \frac{\boldsymbol{v}_2 \cdot \hat{\boldsymbol{n}}_1}{\boldsymbol{v}_1 \cdot \hat{\boldsymbol{n}}_1} \quad \text{and} \quad \delta_2 = C_1 \frac{\boldsymbol{v}_1 \cdot \hat{\boldsymbol{n}}_2}{\boldsymbol{v}_2 \cdot \hat{\boldsymbol{n}}_2}, \tag{2.35}$$

and $\hat{\boldsymbol{n}}_i$'s denote normal vectors to $\mathcal{V}(\boldsymbol{x})$ at respective $\boldsymbol{v}_i$'s.

Proof. Starting with the MC-parallelogram, gradually increasing δ moves the lines L_1, L_2 (described in Theorem 6) and shrinks the δ-MC quadrilateral. For a convex and smooth speed profile, this process can be continued until one of the L_i's becomes tangential to $\mathcal{V}(\boldsymbol{x})$ at $\boldsymbol{v}_i$. Suppose this happens with $i = 1$ first, when $\delta = \delta_1$.

The line L_1 passes through the points $\boldsymbol{v}_1$ and $\frac{C_2}{\delta_1}\boldsymbol{v}_2$. Tangentiality of L_1 means that $\left(\frac{C_2}{\delta_1}\boldsymbol{v}_2 - \boldsymbol{v}_1\right) \cdot \hat{\boldsymbol{n}}_1 = 0$, which is equivalent to the expression in (2.35). The case where L_2 becomes tangential first similarly yields the expression for δ_2. $\square$

If either of these lines is already tangential to $\mathcal{V}(\boldsymbol{x})$ when $\delta = 0$, this means that the simplex is MC but not δ-MC for any $\delta > 0$; see Figure 2.8(b). To find the largest δ such that the entire stencil is δ-MC at $\boldsymbol{x}$, we define $\Delta(\boldsymbol{x}) = \min_{s \in \mathcal{S}(\boldsymbol{x})} \Delta(\boldsymbol{x}, s)$. Similarly, let $\Delta(X) = \min_{\boldsymbol{x} \in X} \Delta(\boldsymbol{x})$. For an MC discretization, if $\Delta(X) > 0$ then Dial's method with the bucket width $\Delta(X)$ is also applicable. We note that, since $C_i = \frac{|\boldsymbol{z}_i|}{f_i}$, for any fixed speed function $f(\boldsymbol{x}, \boldsymbol{a})$, the allowed bucket width $\Delta(X)$ will be shrinking under grid refinement.

Proposition 5 also provides a simpler derivation for some of the results previously found for specific 2D-stencils. E.g., in the isotropic case $\mathcal{V}$ is just a ball; so, $\boldsymbol{v}_i$ and $\hat{\boldsymbol{n}}_i$ are parallel. If β is an angle between $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, (2.35) reduces to $\delta_1 = C_2 \cos\beta = \frac{|\boldsymbol{z}_2|}{f(\boldsymbol{x})} \cos\beta$ and $\delta_2 = C_1 \cos\beta = \frac{|\boldsymbol{z}_1|}{f(\boldsymbol{x})} \cos\beta$. In the 8-point stencil of Figure 2.5(b), $|\boldsymbol{z}_1| = h$, $|\boldsymbol{z}_2| = h\sqrt{2}$, and $\beta = \pi/4$. Thus, $\delta_1 = \frac{h}{f(\boldsymbol{x})}$, $\delta_2 = \frac{h}{\sqrt{2} f(\boldsymbol{x})}$, and $\Delta(\boldsymbol{x}) = \delta_2$. As a result, $\Delta(X) = \frac{h}{F_2\sqrt{2}}$, which matches the bucket width derived in [102]. General 2D stencils and triangulated meshes can be treated similarly, using upper bounds on $|\boldsymbol{z}_i|$'s and β.

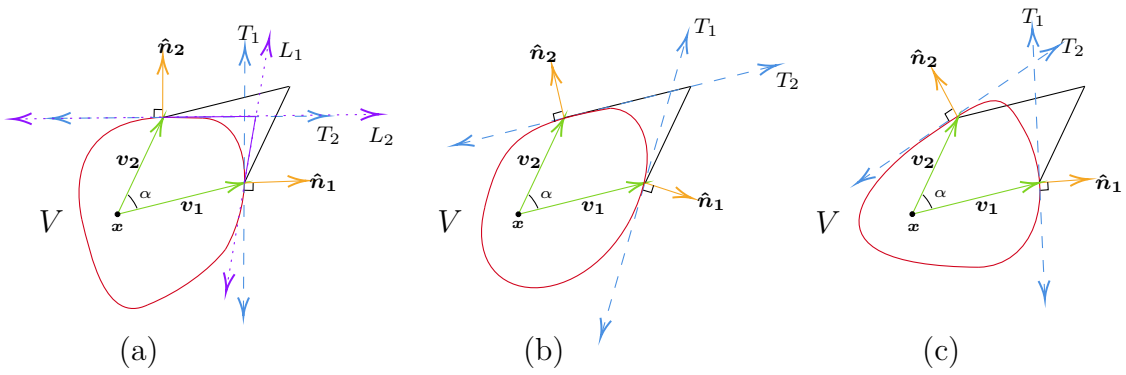


Figure 2.8: Simplex s and causality conditions on $\mathcal{V}_s$ for three different convex speed profiles. Tangent lines to $\mathcal{V}$ at $(\boldsymbol{x}+\boldsymbol{v}_1)$ and $(\boldsymbol{x}+\boldsymbol{v}_2)$ are labeled T_1 and T_2 respectively and plotted in dashed blue. $\boldsymbol{v}_1, \boldsymbol{v}_2$, and α are equivalent in all three examples. $\mathcal{V}_s$ in panel (a) satisfies the monotone causality conditions posed in Theorem 4 and the monotone δ-causality conditions posed in Theorem 5. The corresponding sides of the resulting bounding quadrilateral are shown in solid purple. In this case, L_2 (shown in dotted purple) is tangential to $\mathcal{V}$, and $C(\boldsymbol{\xi})$ is monotone δ-causal in this simplex with $\delta = \delta_2$. In panel (b), $\mathcal{V}_s$ only satisfies the conditions of Theorem 4 and $C(\boldsymbol{\xi})$ cannot be monotone δ-causal, as the side of the parallelogram opposite to $\boldsymbol{v}_1$ is tangential to $\mathcal{V}$. Finally in panel (c), $\mathcal{V}_s$ violates the $\boldsymbol{v}_1 \cdot \hat{\boldsymbol{n}}_2 \geq 0$ condition in Proposition 3, and $C(\boldsymbol{\xi})$ cannot be monotone causal on this simplex.

2.4 OSSPs and Autonomous Vehicle Routing

"Your fate decides,

the roads you're going to find."

"Amor Fati" by Washed Out

Routing modules in autonomous vehicles (AVs) have multiple interacting levels of planning and control which work together to bring the vehicle from its starting destination $\boldsymbol{s}$ to a predetermined destination $\boldsymbol{t}$ [70]. Typically, the routing

module first produces a Strategic Plan (SP), which is a set of deterministic, cost[7]-minimizing, turn-by-turn directions from $\boldsymbol{s}$ to $\boldsymbol{t}$. SP is usually precomputed based on traffic and weather conditions using some label-setting method to identify the deterministic cheapest path on a graph representing the road network. After that, the system may determine a Tactical Plan (TP) outlining the planned timing and location of lane switch maneuvers (LSMs) necessary to execute the SP. Finally, the SP and TP are communicated to the Operational Control (OC) module which is responsible for planning the vehicle's continuous trajectory, steering, and dynamically accelerating / decelerating in order to follow the planned route. OC occurs in real time and is heavily constrained by safety overrides based on the actual observed vehicles. If a planned LSM fails, this often makes it necessary to follow a suboptimal path and repeatedly recompute both SP and TP.

The overall performance can be clearly improved by directly modeling the probabilistic nature of LSMs and minimizing the expected cost to target. This is the essence of the new approach developed by Jones, Haas-Heger, and van den Berg [62], which can be characterized as computing a combined stochastic Strategic / Tactical Plan (STP). They cast the routing problem as an SSP in which the states correspond to nodes within a lane-level road network such as the examples in Figure 2.9(a) and Figure 2.12(b). Using our terminology, their model is an example of an OSSP in which $|s| \leq 2$ for all $s \in \mathcal{S}(\boldsymbol{x})$, $\boldsymbol{x} \in X$. Deterministic modes (with $|s| = 1$) are used to model the actions that normally don't fail (e.g., moving forward wherever lane changes are not allowed or available, or turning at an intersection without any lane changes). On the other hand, a possible switch to each specific adjacent lane is encoded using a separate mode with $|s| = 2$. In [62], each such mode is associated with exactly three LSM actions: the deterministic stay-

[7]E.g., time, fuel consumption, toll charges, passengers' comfort or a combination of these factors.

in-lane, the deterministic "forced" LSMs, and one tentative LSM which succeeds with probability $\tilde{p} \in (0, 1)$. Figure 2.9(b) shows an example of the LSMs available to the vehicle at $\boldsymbol{x}$ in mode $s_1 = \{\boldsymbol{x}_j, \boldsymbol{x}_k\}$. The vehicle may deterministically transition to $\boldsymbol{x}_j$, deterministically transition to $\boldsymbol{x}_k$, or use a stochastic transition leading to $\boldsymbol{x}_k$ or $\boldsymbol{x}_j$ with respective probabilities $\tilde{p}$ and $(1 - \tilde{p})$.

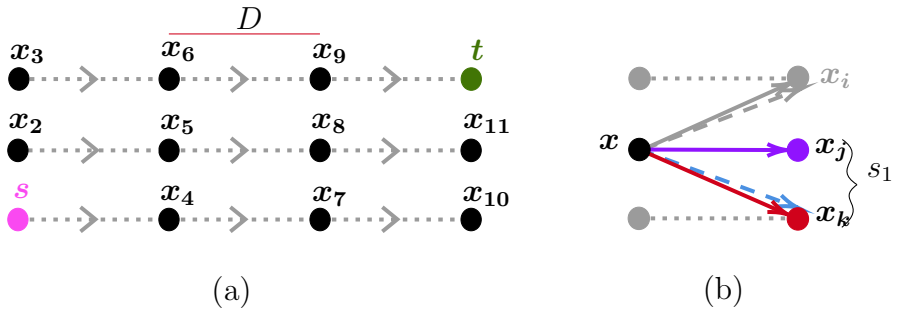


Figure 2.9: Panel (a): Example lane-level road network representation of a three-lane highway. Lanes are discretized into cells of length D meters, and each node marks the center of a cell. The vehicle travels from the starting point $\boldsymbol{s}$ to the destination $\boldsymbol{t}$ via a series of planned LSMs. Panel (b): Actions available at node $\boldsymbol{x}$ in mode $s_1 = \{\boldsymbol{x}_j, \boldsymbol{x}_k\} \in \mathcal{S}(\boldsymbol{x})$. The vehicle may continue driving in the current lane and directly transition to $\boldsymbol{x}_j$ (solid purple arrow), forcefully switch lanes and directly transition to $\boldsymbol{x}_k$ (solid red arrow), or attempt a tentative lane change (dashed blue arrow). The other mode available at $\boldsymbol{x}$ is $s_2 = \{\boldsymbol{x}_j, \boldsymbol{x}_i\}$, encoding a possible switch to another lane.

The vehicle's transitions are penalized as

$$K(0) = g(\boldsymbol{x}), \quad K(\tilde{p}) = g(\boldsymbol{x}) + \tilde{p}g_1, \quad \text{and} \quad K(1) = g(\boldsymbol{x}) + g_1 + (1 - \tilde{p})g_2 \tag{2.36}$$

where $g(\boldsymbol{x})$ is the cost of traveling to the next node in the same lane, $g_1 > 0$ is the additional cost incurred if a tentative LSM is successful, and $g_2 > g_1$ is a large penalty incurred for forcing the lane switch. For an unforced/tentative LSM, the transition probability is modeled as $\tilde{p} = 1 - e^{-\alpha D}$ where D is the distance to the next node in the same lane, and α is the success rate determined from local

traffic data. The cost $K(p)$ in (2.36) is monotonically increasing in p based on the assumption that the instantaneous cost of changing lanes will always be higher than staying in the current one.

Jones et al. have proved in [62] that a Dijkstra-like method is applicable to this SSP provided

$$g(\boldsymbol{x}) \;\geq\; \alpha D g_2. \tag{2.37}$$

The same result also follows as a direct application of our Theorem 2, which for $m = 2$ is equivalent to Theorem 1. Since $K(p)$ is monotonically increasing on $[0, 1]$, we only need to check the inequality (2.20) for $r = 1$, which requires

$$K(\tilde{p}) = g(\boldsymbol{x}) + \tilde{p} g_1 \;\geq\; \tilde{p}\,[g(\boldsymbol{x}) + g_1 + (1 - \tilde{p}) g_2]\,, \tag{2.38}$$

or, equivalently, $g(\boldsymbol{x}) \geq \tilde{p} g_2$. By the convexity of exponentials, $\alpha D \geq \tilde{p}$, and so (2.37) implies (2.38). We note that, outside of MSSPs [105], this is the first known example of a monotone causal OSSP. Moreover, if (2.37) holds, it is easy to use our Proposition 1 to show that this OSSP is also monotone δ-causal with $\Delta = \min_{\boldsymbol{x}} g(\boldsymbol{x}) - \tilde{p} g_2$.

The theory developed in §2.1 and §2.2 significantly widens the scope of STP models that can be treated with label-setting methods[8] . E.g., we can now similarly account for a larger number of tentative LSM actions and different cost models with monotonically increasing and convex $K(p)$. The OSSP framework also allows for a useful reinterpretation of the lane change success probability in terms of the *lane change urgency*[9], which reflects the degree of controller's willingness to alter the

[8]The following material is subject to Provisional US Patents 10471-01-US and 10471-02-US.

[9]It might seem that the urgency of an LSM should be fully determined by the vehicle's distance to its next preplanned turn, which has to be executed from the lane we are trying to switch to. But if similarly cheap alternate routes are available, missing a planned turn or highway exit may only have a very minor effect on the vehicle's total time (or other relevant expenditures) up to $\boldsymbol{t}$. Thus, an accurate assessment of urgency should take into account the global road network structure and traffic patterns.

vehicle's velocity to ensure a successful lane change [52]. In reality, drivers *gradually* increase or decrease LSM urgency in response to the local traffic conditions or nearby infrastructure [1]. Thus, urgency exists on a continuous spectrum in which the stay-in-lane maneuver corresponds to no urgency ($p = 0$), the forced lane change maneuver corresponds to full urgency ($p = 1$), and all LSMs with stochastic outcomes correspond to intermediate urgency levels ($p \in (0, 1)$).

The simplest generalization of the model in [62] is to allow for more than one (but finitely many) intermediate LSMs, which might be executed *progressively* at $\boldsymbol{x}$. The cost of a forced LSM in (2.37) reveals that upon selecting the forced lane change maneuver, the vehicle attempts to switch lanes at the intermediate level first, and then only forces the LSM should that initial attempt fail. As the LSM urgency increases between the first attempt and second attempt, the LSMs themselves in this framework may also be described as *escalating*.

To be more precise, suppose there are $L + 1$ available LSMs with associated success probabilities $p_\ell \in \mathcal{P}$ such that $0 = p_0 < \cdots < p_\ell < \cdots < p_L = 1$. The stay-in-lane cost is $K(p_0) = g(\boldsymbol{x})$, and maneuvers with $p_\ell > 0$ are subject to additional penalties $Y_\ell > 0$ which are monotonically increasing in ℓ and incurred upon a successful lane change at the corresponding urgency level. If these actions are executed in escalating manner, their expected cost has the form

$$K(p_\ell) = p_{\ell-1}K(p_{\ell-1}) + (1-p_{\ell-1})\left[K(p_{\ell-1}) + Y_\ell\right] = K(p_{\ell-1}) + (1-p_{\ell-1})Y_\ell, \qquad \ell = 1, \ldots, L. \tag{2.39}$$

The convexity of K is ensured when $\frac{K(p_{\ell+1})-K(p_\ell)}{p_{\ell+1}-p_\ell} \geq \frac{K(p_\ell)-K(p_{\ell-1})}{p_\ell-p_{\ell-1}}$ holds for all $\ell = 1, \ldots L-1$, which implies that all of these actions are actually "useful" in the sense of Observation 1. If the above inequality does not hold, some of these LSM actions can be safely removed (in pre-processing) to reduce the computational cost

of label-setting methods without affecting the value function.

By direct application of Theorem 2, the OSSP will be monotone (δ-)causal when

$$K(p_{\ell-1})+(1-p_{\ell-1})Y_\ell \geq p_\ell\,[K(p_{L-1})+(1-p_{L-1})Y_L]+(1-p_\ell)\delta, \qquad \ell = 1,\ldots,L-1.$$

We note that the exact model used in [62] can be recovered by taking $L = 2$, $p_1 = \tilde{p}$, $Y_1 = \tilde{p}g_1$, and $Y_2 = g_2 + g_1$.

Whether or not the above progressive escalation framework is realistic is debatable: a lane change may take six or more seconds to complete [45] and attempting multiple consecutive LSMs before reaching the successor node in the same lane might be impossible. But our framework can be similarly used for any increasing sequence of costs $K_\ell = K(p_\ell)$. Due to the monotonicity of K, the inequality (2.20) has to be enforced for $r = 1$ only and will hold automatically for $r = 2$. The lane switch mode will be monotone (δ-)causal as long as

$$K(p) \;\geq\; pK(1) + (1-p)\delta \tag{2.40}$$

holds for every attainable $p < 1$. This criterion works with models of finitely many urgency levels ($p \in \{p_0, ..., p_L\}$) and also for models with a continuous spectrum of urgency levels ($p \in [0, 1]$); see also Figure 2.3. The maximum allowable δ can be similarly computed using Proposition 1.

We also mention two other LSM-cost models illustrated by the numerical experiments covered in the next subsection. For the continuous urgency spectrum, one of the simplest (monotone, convex) cost models is quadratic

$$K(p) = \beta_s(\boldsymbol{x})p^2 + \gamma_s(\boldsymbol{x}) \tag{2.41}$$

where $\beta_s(\boldsymbol{x}), \gamma_s(\boldsymbol{x}) > 0$ are constants which may reflect traffic conditions near $\boldsymbol{x}$ or learned driver behavior. Using this $K(p)$ in (2.40) and simplifying, we see that this

lane switch mode will be δ-monotone causal as long as $\beta_s(\boldsymbol{x}) + \delta \leq \gamma_s(\boldsymbol{x})$. Thus, Dijkstra's method will be applicable if $\beta_s(\boldsymbol{x}) \leq \gamma_s(\boldsymbol{x})$ for all $\boldsymbol{x} \in X$, $s \in \mathcal{S}(\boldsymbol{x})$. If this inequality is strict, we can also use Dial's method with buckets of width $\Delta(X) = \min_{\boldsymbol{x} \in X} \min_{s \in \mathcal{S}(\boldsymbol{x})} (\gamma_s(\boldsymbol{x}) - \beta_s(\boldsymbol{x}))$.

Alternatively, we can start with an (increasing, convex) sequence of urgency levels and associated costs (p_ℓ, K_ℓ) learned from local traffic data or from the vehicle's performance analytics, and then extend it to a continuous urgency spectrum $p \in [0, 1]$ by defining $K(p)$ through a suitable interpolation.

For example, when $L = 2$, a natural choice of interpolant is a quadratic Rational Bézier Curve (RBC) with suitable control points and weights chosen to ensure that the resulting smooth approximation to $K(p)$ is monotone, convex, and monotone (δ-)causal by Theorem 2. The RBC is described parametrically as $\boldsymbol{B}(t) = (p(t), K(t))$, and the entire curve sits within the convex hull of a given set of $L + 1$ control points. The curve's shape is governed by those control points and a set of corresponding weights $\{\omega_0, \ldots, \omega_L\}$ which determine how much the curve bends toward each control point [80]. Assuming that the original three-point (p_ℓ, K_ℓ) sequence is δ-MC, we can take the first control point to be $(0, K(0))$ and the last control point to be $(1, K(1))$, with the remaining control point chosen to lie at the intersection of the lines $K = K_0$ and $K = pK_2 + (1 - p)\delta$ to ensure that the resulting curve is both monotonically increasing and monotone (δ-)causal by Theorem 2. Since RBCs are guaranteed to pass through their first and last control points, we can set $\omega_0 = \omega_2 = 1$, and we construct a system of two linear equations (one for $K(t) = K_1$ and one for $p(t) = p_1$) to compute t and the value of ω_1 which ensures that the RBC also passes through (p_1, K_1) as required. We note that, for this choice of control points, Theorem 2 guarantees that this OSSP will

be monotone causal though not δ-MC for any $\delta > 0$.

Evaluating the usefulness (and the optimal urgency) of merging to an adjacent lane is the essential part of computing the value function. E.g., starting from the point $\boldsymbol{x}$ in Figure 2.9(b), we need to find p that minimizes $(K(p) + (1-p)U_j + pU_k)$. For a continuous spectrum of p values, the availability of $K'(p)$ can be used to perform this minimization either analytically (e.g., with the quadratic K model) or semi-analytically (e.g., with the RBC-based K model described above) to ensure the computational efficiency.

2.4.1 Numerical Examples

We use a Dijkstra-like method to determine the optimal STP for a vehicle traveling through several different lane-level road networks. In all examples, $D = 10$ meters, and the direction of traffic flow between successor nodes is indicated by arrows. When visualizing optimal STPs, we use solid arrows to indicate when a deterministic LSM is optimal and dashed arrows to indicate when the optimal LSM is deterministic. The color of such arrows indicates the optimal p^*, whose value (rounded for the sake of readbility) is also shown as a label.

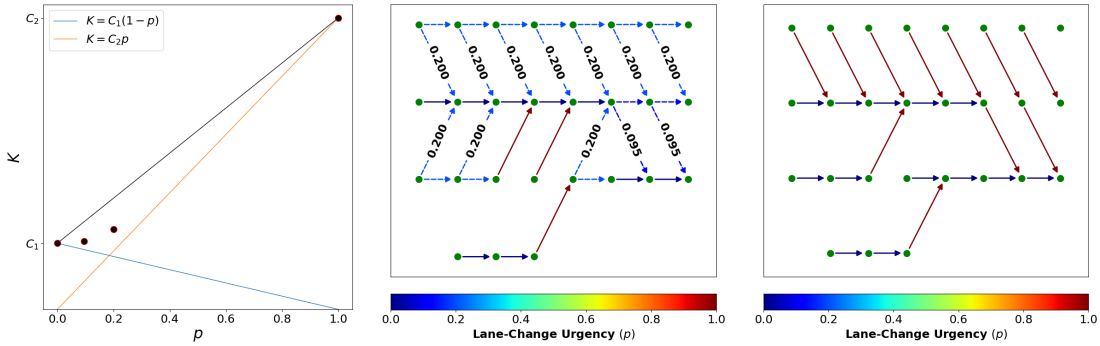
Example 1: a 3-lane highway; 4 urgency levels.
We first determine the STP for an autonomous vehicle traveling on a 1500 meter section of a three-lane highway. The target offramp $\boldsymbol{t}$ is located in the left lane, but the vehicle is incentivized to move in the right lane as much as possible. This is a common preference in routing autonomous trucks; following [62], we accomplish this by defining the cost of moving in the current lane as $g(\boldsymbol{x}) = D\sigma_i$ where $\sigma_i = 1 + i\varepsilon$, the lanes are enumerated from right to left ($i = 0, 1, 2$), and $\varepsilon > 0$ is a

fixed penalty. We also model an onramp, with additional traffic entering the right lane from a merge lane at node $\boldsymbol{x}_{\#}$, exactly $1km$ from the target. The right-lane nodes within 10 meters of $\boldsymbol{x}_{\#}$ experience an additional cost $\mu = 35$ in their $g(\boldsymbol{x})$ to account for moderate congestion due to the traffic merging in from the onramp. Nodes in the left and right lanes have only one mode, while nodes in the middle lane have two (switch to the left or to the right). In each mode, the system may choose between four available LSMs of increasing urgency with associated success probabilities $0 = p_0 < p_1 < p_2 < p_3 = 1$. $K_\ell = K(p_\ell)$ is given by (2.39) with $L = 3$, $p_1 = \tilde{p}$, $Y_1 = \tilde{p}g_1$, $p_2 = 0.2$, $Y_2 = 2$, and $Y_3 = 40$. The expression for $\tilde{p}$ mirrors the definition in [62], and we set $\alpha = 0.01$, $\varepsilon = 0.1$, and $g_1 = 3$. A plot of $K(p_\ell)$ for vehicles driving in the right lane is displayed in Figure 4.7(a). The resulting $K(p_\ell)$ is monotone δ-causal by Theorem 2 with $\delta_\star \approx 4.095$.

We compute the STP on this stretch of a highway and show the optimal policy for a smaller segment (between 970 meters and 1040 meters away from $\boldsymbol{t}$) in Figure 4.7(b). It illustrates how additional intermediate LSMs allow the vehicle to dynamically adjust the urgency of its attempts in response to anticipated locations of higher cost. E.g., vehicles in the right lane prefer to temporarily switch left as they approach the onramp with an increasing LSM urgency as they get closer. At $\boldsymbol{x}_{\#}$ this urgency decreases again (we have already suffered through most of the delay), and beyond $\boldsymbol{x}_{\#}$ those traveling on the right will prefer to continue in their current lane until they are much closer to the target. The vehicles traveling in the center lane will only start trying to switch right after passing the onramp. However, even after $\boldsymbol{x}_{\#}$ their urgency level to switch right remains low – this reflects the fact that the target offramp is already not too far and on the left.

We also show the deterministic optimal SP on the same highway segment in

Figure 4.7(c). Following these instructions would result in much more aggressive lane switching from the central and left lanes and, not surprisingly, would yield a higher overall cost to target. Comparing SP and STP over all nodes in this network, STP results in the median, average, and maximum cost reduction of 5.23%, 5.49%, and 15.65% respectively.



(a) (b) (c)

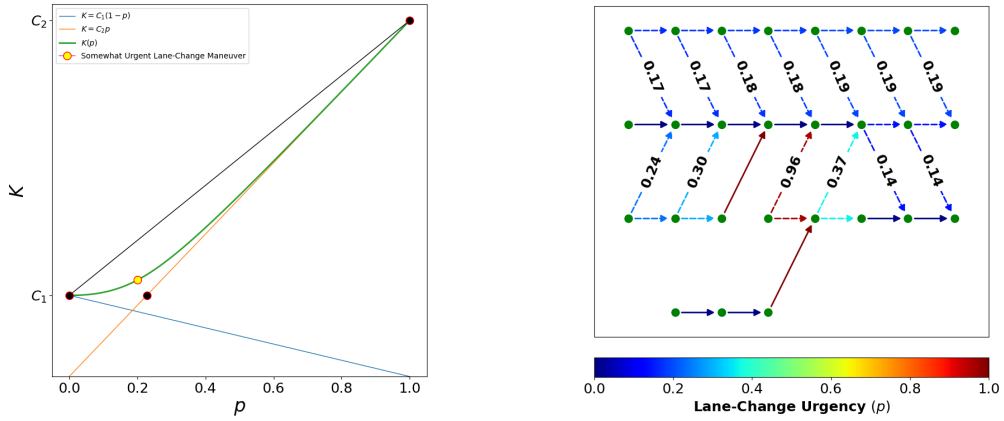
Figure 2.10: Lane-level planning on a three-lane highway between 970 meters and 1040 meters away from $\boldsymbol{t}$. Panel (a): Finite cost function $K(p)$ for vehicles traveling in the right lane (except within 10 meters from $\boldsymbol{x}_{\#}$ on either side) for the $L = 3$ escalating LSMs framework described above. C_1 is the stay-in-lane cost, $K(0)$, and C_2 is the forced LSM cost, $K(1)$. Panel (b): STP when there are four available LSMs per mode. The edges are colored corresponding to the success probability associated with the optimal LSM urgency level at $\boldsymbol{x}$. Panel (c): The deterministic SP when the stay-in-lane cost is $g(\boldsymbol{x})$ and the lane change cost is equal to $K(1)$.

Example 2: a 3-lane highway; continuous urgency spectrum.

Extending the previous example, we now use Rational Bézier Curves to construct a continuous cost function through the points $(p_0 = 0, K_0), (p_2, K_2)$, and $(p_3 = 1, K_3)$. These three points are defined for each lane in Example 1, and we know this action set is monotone causal by Theorem 2; see Figure 4.7(a).

For this stretch of a three-lane highway, there are *four* RBCs to consider –

one RBC for each lane (since $g(\boldsymbol{x})$ is lane-dependent) and one RBC for the slow-down (onramp merge) zone, which includes the right-lane nodes within 10 meters of $\boldsymbol{x}_{\#}$. Each RBC has control points $\left\{(p_0, K_0), \left(\frac{K_0}{K_1}, K_0\right), (p_3, K_3)\right\}$ and weights $\{\omega_0 = 1, \omega_1, \omega_2 = 1\}$, with ω_1 chosen to make the RBC satisfy $K(p_2) = K_2$. The resulting cost function for vehicles traveling in the right lane (except at the locations within 10 meters of $\boldsymbol{x}_{\#}$) is displayed in Figure 4.9(a).



(a) (b)

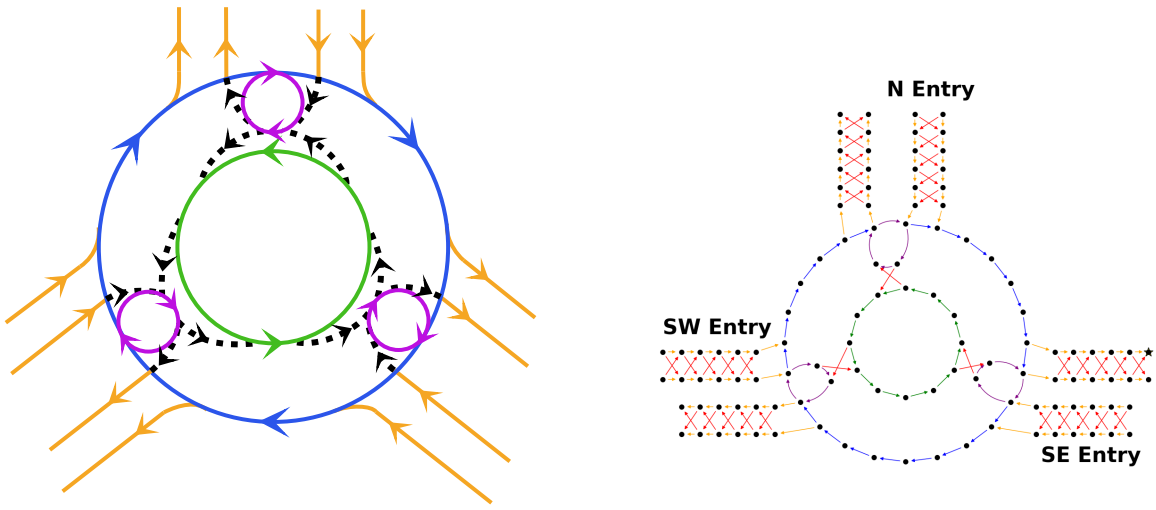
Figure 2.11: Lane-level planning on a three-lane highway between 970 meters and 1040 meters away from $\boldsymbol{t}$ in the left lane. Panel (a): Rational Bézier cost function through three cost estimate datapoints for vehicles traveling in the right lane (except 10 meters from the merge location, $\boldsymbol{x}_{\#}$). $C_1 = K(0), C_2 = K(1)$, and the point corresponding to the intermediate LSM from the dataset is plotted in yellow. Tangency of $K(p) = C_2 p$ to the cost curve prohibits monotone δ-causality by Theorem 2. Panel (b): Resulting STP under the RBC cost structure between 970 meters and 1040 meters away from $\boldsymbol{t}$.

Since $K(p_1)$ under the RBC model is fairly close to K_1 defined in the previous subsection, it is instructive to compare the RBC-based optimal STP shown in Figure 4.9(b) with the STP based on just 4 discrete LSMs in Figure 4.7(b). While the schematic representation of these policies is similar, the optimal behavior is significantly more nuanced in the case of RBCs, with gradual urgency build-ups

highlighting the advantages of continuous spectrum models.

Example 3: multiple roundabouts; continuous urgency spectrum.
We note that in both previous examples the dependency digraph G_μ is acyclic for every stationary policy. This is due to the simplicity of the underlying road network (all arrows pointing rightward), which represented a stretch of a single highway. As a result, the OSSPs are explicitly causal and (2.3) could be solved efficiently in a single Gauss-Seidel iteration (sweeping through the nodes from right to left) regardless of the properties of $K(p)$. But most road networks contain cycles and thus are not explicitly causal, making the MC properties (and applicability of label-setting methods) far more important. Non-trivial cycles are ubiquitous due to numerous road intersections on a large map, but can also be found in smaller examples with ring roads and roundabouts. The "magic" (or interconnected) roundabouts used in several cities of the United Kingdom present drivers with particularly interesting strategic choices when parts of the roundabout are clogged with traffic.

We consider one such example based on a network (shown in Figure 2.12) of five interconnected roundabouts – one outer (blue road segments, moving clockwise), one inner (green, moving counterclockwise), and three miniature roundabouts (purple) which allow vehicles to switch between the outer and the inner roundabouts. To avoid terminological confusion, we will further refer to all of them as "rings," reserving the term "Roundabout" for their interconnected combination. We assume there are three roads running into/out of this Roundabout and that the target is in the left lane of the southeastern exit, identified with a black star in Figure 2.12(b). For a driver approaching the Roundabout, the main strategic decision is whether it is worth changing lanes before the entry point and, if the answer is



(a) (b)

Figure 2.12: A magic roundabout example. Panel (a): Schematic of roundabout network with five connected traffic circles. Arrows indicate the direction of traffic flow and dashed edges indicate available transitions between roundabouts. We assume that the vehicle drives on the left-hand side of the road as is customary in the United Kingdom. Panel (b): Lane-level road network representation of the roundabout network with all possible transitions from each node shown. The arrows show available node-to-node transitions, with red arrows indicating stochastic lane changes while all other transitions are deterministic. The destination $\boldsymbol{t}$ is along the southeastern exit and identified by the black star.

yes, how the urgency of LSMs should vary as we get closer. (Entering from the left lane leads to a clockwise trajectory; entering on the right is worthwhile if you want to transition to the inner roundabout and travel counterclockwise at least at first.) The basic tradeoff is usually between following the most direct route to $\boldsymbol{t}$ and limiting exposure to heavy congestion.

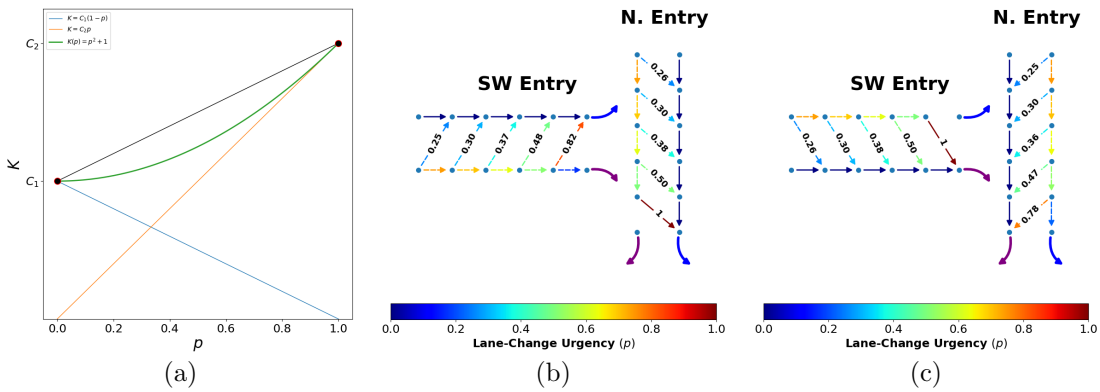
As in the previous example, wherever a lane change is possible (as indicated by red arrows in Figure 2.12(b)), the vehicle has access to a complete urgency

spectrum of LSMs, but here our transition cost is quadratic in p:

$$K(p) = \beta(\boldsymbol{x})p^2 + \gamma(\boldsymbol{x}) + [f] \tag{2.42}$$

where $\beta(\boldsymbol{x}), \gamma(\boldsymbol{x}) > 0$ reflect the congestion present within the road segment (e.g., entryways, outer ring, inner ring, mini-ring) at $\boldsymbol{x}$. The positive constant f is only added at the entrance-nodes from each of the roads (into the mini and outer rings respectively). These constants are chosen to reflect any waiting that the vehicle must endure as a result of current traffic conditions before being able to enter the roundabout. The causality condition established in Theorem 2 is satisfied when $\beta(\boldsymbol{x}) \leq \gamma(\boldsymbol{x}) + [f]$ at all $\boldsymbol{x} \in X$. Our goal is to highlight the effect of traffic congestion on the outer and inner rings; so, for the sake of simplicity, we assume that $\beta(\boldsymbol{x}) = \gamma(\boldsymbol{x}) = 1$ on all two-lane roads leading to/from the Roundabout and on the three purple mini-rings. The resulting $K(p)$ is shown in Figure 4.11(a). For the Roundabout entry nodes, we also set $f = 5$ in all experiments.

For all deterministic transitions (shown by orange, blue, purple, and green arcs), we assume the stay-in-lane cost $K = \gamma(\boldsymbol{x})$ based on the local level of congestion. We focus on its effect on the optimal policy for the cars approaching along the northern and southwestern roads. In the first experiment, the inner ring is much more congested ($K = \gamma(\boldsymbol{x}) = 6.8$ on green arcs) than the outer ring ($K = 3$ on blue arcs). As shown in Figure 4.11(b), it is optimal for the approaching cars to enter the Roundabout from the left lane and travel along the outer ring. Those cars that approach the Roundabout from the Southwest in the right lane will attempt LSMs to the left with an increasing urgency as they get closer. In the second experiment, the situation is essentially reversed with $K = \gamma(\boldsymbol{x}) = 3$ on the inner ring and $K = 5.2$ on the outer. As shown in Figure 4.11(c), the approaching cars now have a clear preference to enter through the right lane; this is largely true even for cars coming from the North, for which the counter-clockwise path (through the



(a) (b) (c)

Figure 2.13: Optimal lane-change policy when approaching a magic roundabout. Panel (a): $K(p)$ along the two-lane entryways to the roundabout network. Here, $\beta(\boldsymbol{x}) = \gamma(\boldsymbol{x}) = 1$, and $K(p)$ is only monotone causal by Theorem 2. Panels (b) and (c): optimal STPs at the southwestern and northern entryways when the inner ring has the highest congestion (panel (b)), and the outer ring has the highest congestion (panel (c)). Thick blue arrows indicate a direct entry to the outer ring while thick purple arrows indicate a direct entry into a mini ring. In both cases, vehicles tackle the tradeoff between taking a more direct route to $\boldsymbol{t}$ and the amount of congestion they are willing to encounter.

inner ring) is preferred despite being longer.

2.5 Conclusions

We have introduced a class of Opportunistically Stochastic Shortest Path (OSSP) problems and proved several sufficient "monotone causality" conditions to guarantee the applicability of efficient label-setting methods. The approach has important applications both in discrete and continuous optimization. For example, given an anisotropic inhomogeneous speed function f for motion in a continuous domain, the deterministic time-to-target minimized over all feasible paths can be found as a viscosity solution of the corresponding stationary Hamilton-Jacobi-Bellman

PDE. A first-order accurate semi-Lagrangian discretization of that PDE can be re-interpreted as an OSSP, and our MC criteria can be then used to check which discretization stencils are compatible with Dijkstra's method (in 2D and 3D) and with Dial's method (in 2D only). Importantly, the conditions we developed are expressed in terms of simple geometric properties of the anisotropic speed profile $\mathcal{V}$. However, our current analysis does not provide any guidance for finding the best MC-causal stencil for each $\mathcal{V}$ (since causal properties need to be balanced against a possible increase in local truncation errors). Restricted versions of this problem are considered in [71, 72, 35], but many aspects still remain open for general speed profiles. This balancing act is even more delicate with Dial's method, where increasing the angular resolution of a stencil decreases that stencil's spatial locality but allows using a larger bin width Δ.

In the discrete setting, we have demonstrated the usefulness of OSSPs in optimizing the lane change level routing of autonomous vehicles. Extending the prior work [62], we showed that Dijkstra's and Dial's methods are applicable to a much broader class of vehicle routing models that include multiple intermediate urgency levels (or even a continuous urgency spectrum) of lane change maneuvers and a variety of cost functions. We note that the same approach is also useful in a semi-autonomous context; our Strategic/Tactical Plan could be also used by human drivers and the recommended lane change urgency levels could be communicated (e.g., as indicator bars of varying length or color) through assistive navigation hardware or software.

OSSP models capture the inherent uncertainty of lane change maneuvers, but if the traffic conditions significantly change from those used to formulate the problem, the entire optimal policy has to be recomputed on the fly. The usability

of label-setting methods makes such occasional online replanning possible. But both Dijkstra's and Dial's method compute the optimal policy for reaching the specific target $\boldsymbol{t}$ from *each* starting node of the road network. However, most of that network is probably irrelevant for a vehicle starting at a specific location $\boldsymbol{s}$. In deterministic path planning, such single source / single target problems are often solved by an even more efficient A* method [56], which uses a "consistent heuristic" to restrict the computations to a smaller (implicitly defined) neighborhood of the optimal path from $\boldsymbol{s}$ to $\boldsymbol{t}$. While such a consistent heuristic is unavailable for general SSPs, it can be constructed in OSSPs with positive $\underline{C} = \min_{\boldsymbol{x}_i \in X} \min_{\boldsymbol{a} \in A(\boldsymbol{x}_i)} C(\boldsymbol{x}_i, \boldsymbol{a})$. This is essentially the approach explored in [63]. But since realistic road networks include short lane segments, this $\underline{C}$ is usually quite small, making the consistent heuristic very conservative and yielding little computational savings. We believe that a more promising approach is to explore the use of inconsistent heuristics (ideally, with a bound on suboptimality of resulting routing policies). It might be possible to use the "asymptotically causal" domain restriction techniques developed for discretizations of HJB equations [28], with the true optimality of trajectories recovered only in the limit (under the grid refinement).

Another interesting direction for future work is to find criteria for applicability of label-setting to "stochastic shortest path games," in which the probability distribution over the successor nodes at each stage depends on the actions chosen by two antagonistic players [79]. We note that some results on the applicability of Dijkstra's method to *deterministic* games on graphs have been developed in the last ten years [9, 16], but the case of general stochastic games on graphs remains open.

CHAPTER 3

OPTIMAL BRAKING UNDER TRAFFIC SIGNAL UNCERTAINTY

Determining optimal driving behaviors under traffic signal uncertainty is of crucial importance in modern transportation applications. Recent technological advances aim to provide real-time "signal phase and timing" (SPaT) information to smartphones and on-board vehicle systems, with the goal of improving safety, fuel efficiency, and driver's comfort. While this technology will yield likely decreases in crash rates and carbon emissions ([26]), the full transition to vehicles communicating directly with traffic signals will take several decades ([75]), and some level of uncertainty in SPaT information will remain even then due to occasional network failures and the presence of actuated or semi-actuated traffic signals ([39]).

Modeling how drivers do (or should) utilize SPaT knowledge is a popular research area; e.g., [97], [96], [69], [68], [110]. But typical prior approaches are based on either full SPaT information or ad hoc models of SPaT-robustness, with the emphasis placed on detailed vehicle dynamics, road models with multiple signalized intersections, traffic effects, and route-planning choices to come up with trip plans that optimize the fuel efficiency. In contrast, we consider the simplest scenario (one car with simplified dynamics on an empty road with a single traffic signal – as described in Section 3.1), focusing instead on a detailed model of SPaT-uncertainty due to the probability distribution over possible durations of the green light (Section 3.2). The driver knows the yellow light and red light durations, so they can deterministically plan the rest of their strategy once the initial green light turns yellow. They must also obey the traffic laws (red light and speed limit) and balance expected costs related to fuel usage, aggressive acceleration / deceleration

behavior, and total time to target.

We address this optimal control problem in the dynamic programming (DP) framework, as was recently done for a related *red phase duration uncertainty* problem in ([103]). However, the dilemma in the green phase (i.e., should we try to beat the red light?) is far more affected by the available SPaT information and the duration of the transitional yellow phase. Unlike in ([103]), where the discrete nature of DP was essential and the timestep could not be refined, our continuous DP yields a sequence of Hamilton-Jacobi-Bellman (HJB) equations, which are then solved numerically (Section 3.3) to determine the optimal acceleration and braking policy in feedback form. Our numerical experiments in Section 3.4 highlight the role of conflicting optimization goals and SPaT-uncertainty patterns in shaping drivers' behavior.

3.1 Problem Formulation

Throughout this paper, we focus on the problem of a single driver aiming to optimize their route to a target through an intersection with one traffic light. We assume the vehicle is just a point mass traveling right-to-left on a flat single-lane road (an interval $[d^*, \bar{d}]$) with no other traffic: from its starting position at $d \leq \bar{d}$ with initial speed v, through the intersection at $d_\ell < d$ and towards its target at $d^* < d_\ell$. The driver varies the car's signed acceleration and must stay under the speed limit $\bar{v}$ on the way to d^*. The car may not enter the intersection (crossing d_ℓ) while the light is red, but may do so while the light is yellow[1]. We further assume that the driver knows the yellow and red signal durations $(D_{\text{Y}}, D_{\text{R}})$.

[1]This corresponds to *permissive* yellow light laws ([10]), where the car does not have to clear the intersection by the time the light turns red. This makes the actual intersection width and the physical car length irrelevant.

Time Horizon. The entire planning process is split into (1) an initial green light phase I starting at $t = 0$ followed by (2) a yellow phase Y starting at $t = T_{\text{Y}}$, (3) a red phase R starting at $T_{\text{R}} = T_{\text{Y}} + D_{\text{Y}}$, and (4) an indefinite green phase G starting at $T_{\text{G}} = T_{\text{R}} + D_{\text{R}}$ and terminating when we reach the target at d^*. In many realistic situations, the traffic signal uncertainty will primarily rest in phase I. The limited knowledge scenario discussed in Section 3.2 will define a probability distribution on the *random turning yellow time* $\widetilde{T}_{\text{Y}}$, thereby leading to uncertainty in the finite time horizon of phase I.

Vehicle Dynamics. Let $\Omega = \{(d, v, r) : d \in [d^*, \bar{d}], v \in [0, \bar{v}], r \geq 0\}$. We will use $\boldsymbol{x} = \begin{bmatrix} d & v \end{bmatrix}^T$ to refer to the vehicle's initial configuration or a generic point in (d, v) space, switching to $\boldsymbol{y}(r) = \begin{bmatrix} d(r) & v(r) \end{bmatrix}^T$ to encode how position and velocity change with time. Starting from an initial configuration $\boldsymbol{y}(t) = \boldsymbol{x}$ and assuming that external forces acting on the car are negligible, the vehicle's dynamics are

$$\dot{\boldsymbol{y}}(r) \;=\; \boldsymbol{f}\left(\boldsymbol{y}(r), a(r)\right) \;=\; \begin{bmatrix} -v(r) \\ a(r) \end{bmatrix}, \qquad \text{for } r \geq t \tag{3.1}$$

where the vehicle's acceleration $a(\cdot)$ is a measurable control function, $a : \mathbb{R} \to A$, and the set of available (signed) acceleration values is $A = [-\alpha, \beta]$.

Cost Function. The driver's goal on the trip from d to d^* is to select $a(\cdot)$ to address several optimization objectives: fuel consumption[2], discomfort from rapid acceleration/deceleration, and time to target. We aim to minimize an integral of the *running cost*

$$K(\boldsymbol{y}(r), a) = c_1[a]_+ + c_2 a^2 + c_3 \tag{3.2}$$

[2]We approximate fuel consumption as the amount of fuel let into the engine. We assume that fuel enters the engine at a rate proportional to the car's positive acceleration (i.e., how far down the driver is pressing the gas pedal) ([13]).

where $[a]_+ = \max\{a, 0\}$. The non-negative objective weights c_i's reflect individual preferences. In principle, these can also be learned from trends in driver behavior data, as in ([21]).

Deterministic Control Problem. We begin by discussing the fully deterministic problem that starts after the light turns yellow (i.e., the optimal driving for $t \geq T_{\text{Y}}$) . In the usual style of dynamic programming, we consider the remaining phases of the traffic light backwards in time.

We first consider the last green phase of unlimited duration, G, which is best described as an *exit-time problem*, terminating when the vehicle reaches its target at the time $T^* = \min\{r \geq T_{\text{G}} \mid \boldsymbol{y}(r) = d^*\}$. The cost-to-go depends on our initial configuration and the chosen control $\boldsymbol{a}(\cdot)$, which together determine the remaining time to target $(T^* - t)$. We thus define the control-specific cost

$$\mathcal{J}_{\text{G}}(\boldsymbol{x},\, a(\cdot)) \;=\; \int_t^{T^*} K\left(\boldsymbol{y}(r), a(r)\right)\, dr, \tag{3.3}$$

starting from $\boldsymbol{x} = \boldsymbol{y}(t)$ with $t \geq T_{\text{G}}$. By a standard argument ([7]), the value function defined as

$$q(\boldsymbol{x}) = \inf_{a(\cdot)}\{\mathcal{J}_{\text{G}}(\boldsymbol{x}, a(\cdot))\} \tag{3.4}$$

can be recovered as a viscosity solution of a stationary Hamilton-Jacobi-Bellman (HJB) equation

$$0 = \min_{a \in A}\{K(\boldsymbol{x}, a) + \nabla q \cdot \boldsymbol{f}(\boldsymbol{x}, a)\} \tag{3.5}$$

with boundary conditions $q(d^*, v) = 0$, $\forall v \in [0, \bar{v}]$.

During the other two deterministic stages (Y and R), we define the control-specific cost as

$$\mathcal{J}(\boldsymbol{x}, t, a(\cdot)) = \int_t^{T_{\text{G}}} K\left(\boldsymbol{y}(r), a(r)\right) dr + q(\boldsymbol{y}(T_{\text{G}})), \tag{3.6}$$

starting from $\boldsymbol{x}$ at the time $t \in [T_{\text{Y}}, T_{\text{G}})$. The value function

$$u(\boldsymbol{x}, t) = \inf_{a(\cdot)} \{\mathcal{J}(\boldsymbol{x}, t, a(\cdot))\} \tag{3.7}$$

is now time-dependent to account for the fixed time-horizon. The state constraints detailed below can often make this value function discontinuous during the Y phase. Nevertheless, u can still be recovered as a discontinuous viscosity solution [7, Chapter 5] of the time-dependent HJB equation

$$0 = u_t + \min_{a \in A} \{K(\boldsymbol{x}, a) + \nabla u \cdot \boldsymbol{f}(\boldsymbol{x}, a)\} \tag{3.8}$$

subject to the terminal condition $u(\boldsymbol{x}, T_{\text{G}}) = q(\boldsymbol{x})$.

State Constraints. The car should never enter the intersection during the R phase[3]. This makes it natural to consider a space-time obstacle $\mathcal{I} = \{(d, v, t) \mid d = d_\ell,\, 0 \le v \le \bar{v},\, T_{\text{R}} \le t < T_{\text{G}}\}$, but we go further and forbid the car from taking any (d, v) from which it is impossible to come to a complete stop before reaching d_ℓ. (This guarantees that the car will not violate the rules even if the signal fails to switch to green light at the usual time T_{G}.) The minimal stopping distance depends on the current velocity v and the maximum deceleration rate α, yielding the curve of critical positions (the "parabola of last resort") $d_\alpha(v) = d_\ell + \frac{v^2}{2\alpha}$ and the disallowed region

$$\mathcal{I}_{\text{R}}(t) = \{(d, v) \mid d_\ell \le d < d_\alpha(v),\, 0 \le v \le \bar{v}\}$$

enforced for $t \in [T_{\text{R}}, T_{\text{G}})$. Since we are forced to use the maximum deceleration $a = -\alpha$ on the parabola of last resort, this makes it easy to compute the value function on it. Let $s_{\text{G}} = T_{\text{G}} - t$. It would take the vehicle $s_\alpha = v/\alpha$ to come to

[3]For the sake of brevity, we omit detailed algebraic derivations in favor of geometric discussion emphasizing the physical intuition and main results. Further details of the numerical implementation are discussed in Section 3.3.

a full stop, but the actual braking time might be shorter, $\hat{s}_\alpha = \min\{s_\alpha, s_{\text{G}}\}$. The value function along $(d_\alpha(v), v, t)$ is

$$C_\alpha = \hat{s}_\alpha c_2 \alpha^2 + c_3 s_{\text{G}} + q\left(\boldsymbol{y}(T_{\text{G}})\right). \tag{3.9}$$

During the Y phase, the car is allowed to have $d \le d_\alpha(v)$ as long as it has enough time to enter the intersection by the time T_{R}. The *largest* d starting from which this is possible can be found by assuming that we use the maximum allowable acceleration (but without violating the speed limit) until reaching d_ℓ. Denote the remaining yellow time as $s_{\text{R}} = T_{\text{R}} - t$, the time that it would take to reach the speed limit as $s_\beta = \frac{\bar{v}-v}{\beta}$, and the lowest starting speed from which the speed limit would be reached before T_{R} as $v_c = \bar{v} - \beta s$. For fixed v and $t \in [T_{\text{Y}}, T_{\text{R}})$, the largest allowed d is then

$$d_\beta(v,t) = \begin{cases} d_\ell + v s_{\text{R}} + \frac{\beta}{2} s_{\text{R}}^2 & v \le v_c; \\ d_\ell + v s_\beta + \frac{\beta}{2} {s_\beta}^2 + \bar{v}(s_{\text{R}} - s_\beta), & v \ge v_c. \end{cases} \tag{3.10}$$

The new disallowed region

$$\mathcal{I}_{\text{Y}}(t) \;=\; \left\{(d,v) \mid d_\beta(v,t) \le d \le d_\alpha(v),\, v \in [0, \bar{v}]\right\}$$

is enforced for $t \in [T_{\text{Y}}, T_{\text{R}})$ and illustrated in Fig. 3.1. When $d_\beta(v,t) < d_\alpha(v)$, the value function along $(d_\beta(v,t), v, t)$ is

$$C_\beta = (c_1\beta + c_2\beta^2)\hat{s}_\beta + c_3 s_{\text{R}} + u(\boldsymbol{y}(T_{\text{R}}), T_{\text{R}}) \tag{3.11}$$

where $\hat{s}_\beta = \min\{s_\beta, s_{\text{R}}\}$. We assume that the yellow phase duration D_{Y} is long enough so that, starting from every (d,v), it is possible to either enter the intersection before T_{R} or come to a full stop before the intersection; i.e., $\mathcal{I}_{\text{Y}}(T_{\text{Y}}) = \emptyset$. Whenever $d_\alpha(v) \le d_\beta(v,t)$, the latter starting position allows either option, and if speeding toward the intersection yields a lower cost, then the value function will be discontinuous at $\big(d_\beta(v,t),\, v, t\big)$.

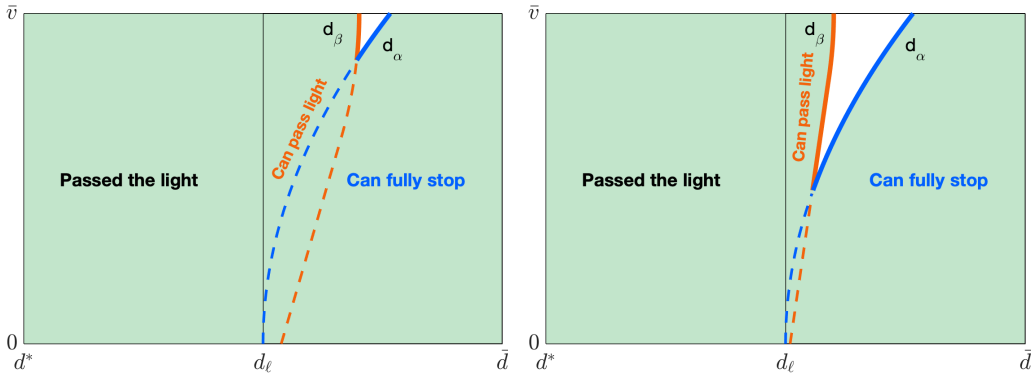


Figure 3.1: Two snapshots of the time-dependent piecewise boundary during phase Y. Allowed (d, v) configurations are shown in green, and $\mathcal{I}_{\text{Y}}(t)$ is shown in white. For the starting configurations $(d_\beta(v,t), v)$, if speeding through the light before $t = T_{\text{R}}$ yields a lower cost than waiting for $T = T_{\text{G}}$, then the value function u is discontinuous along the orange-dashed lines. As $t \to T_{\text{R}}$, $d_\beta \to d_\ell$, and $\mathcal{I}_{\text{Y}}(t) \to \mathcal{I}_{\text{R}}$.

3.2 Optimal Driving Under T_{Y} Uncertainty

"With thought of your future,

With one foot in the past,

Now just how long will it last?"

"Head Over Heels" by Tears for Fears

Optimal planning in the initial green phase is quite similar if the duration $(D > 0)$ and the starting time of that phase are fully known. For a car that has been driving toward the intersection ever since this light turned green ξ seconds before the current time $t = 0$, the natural planning horizon is $T = D - \xi$. The light will then turn yellow, and the remaining cost-to-go will be $\delta(\boldsymbol{x}) = u(\boldsymbol{x}, T_{\text{Y}})$ already computed in Section 3.1. The new value function $w(\boldsymbol{x}, t)$ can be obtained by solving (3.8) with $t \in [0, T]$ and the terminal condition $w(\boldsymbol{x}, T) = \delta(\boldsymbol{x})$. But in

this section we focus on scenarios where D is specified only probabilistically while the durations of the subsequent phases (D_{Y} and D_{R}) are known and fixed. For a simple example, consider a city where all traffic light signals have green phase durations of either D_1 or $D_2 > D_1$ seconds, and the fractions of signals in these respective categories are p_1 and $p_2 = 1 - p_1$. Even though the driver is aware of the city-wide statistics, they may not know the category of this particular signal. So, the remaining green time is a random variable $\widetilde{T}_{\text{Y}}$ with possible values $T_i = D_i - \xi$ and the corresponding probabilities p_i, $(i = 1, 2)$. We are thus selecting a control $\boldsymbol{a}(\cdot)$ to be used until the light turns yellow with the goal of minimizing

$$\mathcal{C}(\boldsymbol{x}, t, a(\cdot)) = \mathbb{E}\left\{\int_t^{\widetilde{T}_{\text{Y}}} K\left(\boldsymbol{y}(r), a(r)\right) dr + \delta(\boldsymbol{y}(\widetilde{T}_{\text{Y}}))\right\}. \tag{3.12}$$

This is an example of "initial uncertainty" ([82]), with the problem becoming fully deterministic at the time T_1: the light will either turn yellow then or we will immediately know that $\widetilde{T}_{\text{Y}} = T_2$. Thus, starting from any $t < T_1$,

$$\begin{aligned}\mathcal{C}(\boldsymbol{x}, t, a(\cdot)) &= \sum_{i=1}^{2} p_i \left(\int_t^{T_i} K\left(\boldsymbol{y}(r), a(r)\right)\, dr \;+\; \delta\left(\boldsymbol{y}(T_i)\right)\right) \\ &= \int_t^{T_1} K\left(\boldsymbol{y}(r), a(r)\right)\, dr \;+\; p_1 \delta\left(\boldsymbol{y}(T_1)\right) \;+ \\ &\quad (1 - p_1)\left(\int_{T_1}^{T_2} K\left(\boldsymbol{y}(r), a(r)\right)\, dr \;+\; \delta\left(\boldsymbol{y}(T_2)\right)\right).\end{aligned}$$

This makes it natural to treat the planning on $[0, T_1)$ and $[T_1, T_2)$ as separate optimization problems, defining *a pair* of value functions $w^1(\boldsymbol{x}, t)$ and $w^2(\boldsymbol{x}, t)$ for the respective time intervals. Each of these will solve the same PDE (3.8), but with different terminal conditions

$$w^1(x, T_1) \;=\; p_1 \delta(\boldsymbol{x}) + (1 - p_1) w^2(\boldsymbol{x}, T_1); \quad w^2(\boldsymbol{x}, T_2) = \delta(\boldsymbol{x}).$$

Generalizing this to n possible green light durations $D_1 < \ldots < D_n$ with possible remaining green time intervals $T_i = D_i - \xi$ and $\mathbb{P}\left(\widetilde{T}_{\text{Y}} = T_i\right) = p_i$ for

$i = 1, \ldots, n$, we introduce n different value functions $w^i(\boldsymbol{x}, t)$, each defined on its own time interval $[T_{i-1}, T_i)$, where for notational convenience we take $T_0 = 0$. All of these satisfy (3.8) with the terminal conditions

$$w^i(\boldsymbol{x}, T_i) \;=\; \hat{p}_i \delta(\boldsymbol{x}) + (1 - \hat{p}_i) w^{i+1}(\boldsymbol{x}, T_i)\,, \tag{3.13}$$

where $\hat{p}_i$ is the conditional probability

$$\hat{p}_i \;=\; \mathbb{P}\left(\widetilde{T}_{\mathrm{Y}} = T_i \mid \widetilde{T}_{\mathrm{Y}} > T_{i-1}\right) \;=\; p_i \,/\, \left(\sum_{j=i}^{n} p_j\right).$$

Since $\hat{p}_n = 1$, the problem is deterministic on $[T_{n-1}, T_n)$. For $i < n$, each w^i depends only on the next w^{i+1}, and all value functions can be found in a single sweep backward in time from $t = T_n$ to $t = T_0$.

3.3 Numerical Implementation

3.3.1 Numerical Methods for HJB Equation

> "Nobody said it was easy. No one ever said it would be this hard."
>
> "The Scientist" by Coldplay

Overall, our numerical approach to the "uncertain green phase duration" problem consists of the following three stages:

1. Solve the stationary HJB (3.5) for the last phase G .
2. Solve HJB (3.8) for the deterministic phases R and Y.

3. Solve a sequence of HJBs (3.8) for the uncertain phase I.

The numerical results from each stage are used in determining the terminal conditions for the following stage. Our algorithms are based on a semi-Lagrangian (SL) discretization ([43]) on a Cartesian grid over a (d, v, t) domain with (N_d+1), (N_v+1), and (N_t+1) gridpionts along the respective dimensions. For a pre-specifed N_v, we select the discretization parameters

$$\Delta v = \frac{\bar{v}}{N_v}, \quad \Delta t = \frac{\Delta v}{\max(\alpha, \beta)}, \quad \Delta d = \bar{v}\Delta t. \tag{3.14}$$

The physical position, velocity, and time at a node (i, j, k) are then $(\boldsymbol{x}_{ij}, t_k) = (d_i, v_j, t_k) = (d^* + i\Delta d,\, j\Delta v,\, k\Delta t)$ for $i = 0, \ldots, N_d,\ j = 0, \ldots, N_v$, and $k = 0, \ldots, N_t$. The usual first-order SL discretization is based on assuming that a fixed control a is used starting from $\boldsymbol{x}_{ij}$ for a small time τ. The resulting new state $\tilde{\boldsymbol{x}}^a_{ij} = (\tilde{d}^a_i, \tilde{v}^a_j)$ is usually approximated numerically (e.g., as $\tilde{\boldsymbol{x}}^a_{ij} \approx \boldsymbol{x}_{ij} + \tau \boldsymbol{f}(\boldsymbol{x}_{ij}, a)$), but our simplified dynamics (3.1) allows computing $\tilde{\boldsymbol{x}}^a_{ij}$ analytically. Moreover, (3.14) guarantees that $\tilde{d}^a_i \in [d_{i-1}, d_i]$ and $|\tilde{v}^a_j - v_j| \le \Delta v$ for any $\tau \le \Delta t$, which makes it easier to interpolate the value function at $\tilde{\boldsymbol{x}}^a_{ij}$. We also use the Golden Section Search (GSS) algorithm whenever we need to find the optimal a numerically.

Stage 1: Stationary HJB Solve

In phase G, there is never any incentive to decelerate and we only consider control values $a \in [0, \beta]$. The PDE (3.5) is stationary, and the solution $q(\boldsymbol{x}_{ij})$ is approxi-

mated by a grid function Q_{ij} satisfying

$$\begin{aligned} Q_{ij} &= \min_{a\in[0,\beta]} \left\{\tau K(\boldsymbol{x}_{ij}, a) + Q(\tilde{\boldsymbol{x}}^a_{ij})\right\}, \quad i > 0,\, j < N_v; \\ Q_{iN_v} &= \tau K(\boldsymbol{x}_{iN_v}, 0) + Q(\tilde{\boldsymbol{x}}^0_{iN_v}), \qquad \forall i; \\ Q_{0j} &= 0, \qquad \forall j; \end{aligned} \tag{3.15}$$

where the last two lines encode the need to stay under the speed limit and the boundary conditions respectively. Since $\tilde{\boldsymbol{x}}^a_{ij}$ is usually not a gridpoint, its value Q is recovered via bilinear interpolation. I.e.,

$$Q(\tilde{\boldsymbol{x}}^a_{ij}) = \gamma_1(a)Q_{\mathrm{i,\,j}} + \gamma_2(a)Q_{\mathrm{i,\,j+1}} + \gamma_3(a)Q_{\mathrm{i\text{-}1,\,j+1}} + \gamma_4(a)Q_{\mathrm{i\text{-}1,\,j}},$$

with the nonnegative bilinear coefficients adding up to 1 and $\gamma_1(a) < 1$ for all a values. The coupled system (3.15) can be solved through *value iterations*, but the dependence of the right hand side on Q_{ij} slows down the convergence considerably even when using a Gauss-Seidel relaxation. Instead, we replace the first equation in (3.15) with an equivalent

$$Q_{ij} = \min_{a\in[0,\beta]} \left\{ \frac{1}{1-\gamma_1(a)} \Big(\tau K(\boldsymbol{x}_{ij}, a) + \gamma_2(a)Q_{i,j+1} + \gamma_3(a)Q_{i-1,j+1} + \gamma_4(a)Q_{i-1,j}\Big) \right\}. \tag{3.16}$$

This effectively decouples the system, and we can now solve it in one sweep, looping through $i = 1, \ldots, N_d;\ j = N_v, ..., 0$.

Stage 2: Deterministic Y-R Phase HJB Solve

The SL discretization for the time-dependent PDE (3.8) is similar, but with several subtleties, which we describe below only briefly due to the space constraints. The grid-approximation of the value function must satisfy

$$U^k_{ij} = \min_{a\in A_{ij}} \left\{K(\boldsymbol{x}_{ij}, a)\tau + U\left(\tilde{\boldsymbol{x}}^a_{ij}, t+\tau\right)\right\} \tag{3.17}$$

for all i, j and all $k < N_t$, which can be solved backwards in time from the terminal conditions $U_{ij}^{N_t} = Q_{ij}$. The control set is based on the speed constraints: $A_{i0} = [0, \beta]$, $A_{iN_v} = [-\alpha, 0]$, and $A_{ij} = [-\alpha, \beta]$ for all other j values. On most of the domain we use $\tau = \Delta t$, and the value at $(\tilde{\boldsymbol{x}}_{ij}^a,\, t + \tau)$ is obtained via bilinear interpolation in (d, v). But we reduce τ wherever it is needed to respect additional state constraints due to the traffic light, in which case we employ *cut-cells* in (d, v) and additional linear interpolation in t.

In the red phase (with $t \in [T_{\mathrm{R}}, T_{\mathrm{G}})$), the car has a disallowed region $\mathcal{I}_{\mathrm{R}}$. For $\boldsymbol{x}_{ij}$ just right of the line $d = d_\alpha(v)$, our τ is decreased adaptively, to ensure that $\tilde{d}_{ij}^a \geq d_\alpha\left(\tilde{v}_{ij}^a\right)$. All grid cells intersected by this line are treated as cut-cells, with values of U on the parabolic boundary computed by formula (3.9).

Extending the solution into the yellow phase (for $t \in [T_{\mathrm{Y}}, T_{\mathrm{R}})$) presents two additional complications: the time-dependent disallowed region $\mathcal{I}_{\mathrm{Y}}(t)$ (see Fig. 3.1) and a possible discontinuity of $u(\boldsymbol{x}, t)$ when $d = d_\beta(v, t)$. We adopt the following two-pass procedure to handle these challenges separately. First, we solve (3.17) for all $d \geq d_\alpha(v)$ exactly as we did in phase R; i.e., treating $d = d_\alpha(v)$ as a boundary with the boundary conditions specified by (3.9). This computes the best cost attainable by waiting out the red light. But for some starting configurations, it is also possible to accelerate enough and cross the intersection before the light turns red. So, we then re-solve (3.17) for all $d \leq d_\beta(v, t)$ without treating $d = d_\alpha(v)$ as a boundary, with grid values updated only if they are smaller than those obtained in the first pass. To avoid interpolating across the discontinuity, in this second pass we use cut-cells just left of $d = d_\beta(v, t)$ with the boundary conditions specified by (3.11).

Stage 3: Uncertain Initial Phase I

Given a set of possible times when the light could turn yellow $T_1, \ldots, T_n$, each conditional value function $w^i(\boldsymbol{x}, t)$ satisfies PDE (3.8) on $t \in [T_{i-1}, T_i)$. Thus, we solve them sequentially (from w^n to w^1) using the same SL-discretization (3.17) but with the terminal conditions (3.13). We note that the latter uses $\delta(\boldsymbol{x}) = u\left(\boldsymbol{x}, T_{\text{Y}}\right)$ obtained at the end of Stage 2. Once this function is computed, Stage 3 can be repeated as needed (for different probability distributions of $\widetilde{T}_{\text{Y}}$) without repeating the computations in Stages 1 and 2. We also note that the timestep in stage I is always $\tau = \Delta t$ and cut-cells are not needed since $\mathcal{I}_{\text{Y}}(T_{\text{Y}}) = \emptyset$.

3.3.2 Optimal Trajectory Tracing

Once the value function is computed, it is easy to find an approximately optimal trajectory for every starting configuration (d, v, t). For $t \in [T_{\text{Y}}, T_{\text{G}})$, the optimal control value $a_\star$ is stored at every gridpoint while solving (3.17); at all other points, we approximate $a_\star(\boldsymbol{x}, t)$ via tri-linear interpolation. When tracing a trajectory, we use this approximately optimal feedback control to solve (3.1) until the time T_{G}. At that point, we switch to the approximately optimal feedback control obtained from (3.15) and continue until reaching the target d^*.

If starting in the initial green phase of uncertain duration from some $t \in [T_{i-1}, T_i)$, we again follow the feedback control obtained when solving for w^i. If the light turns yellow at the time T_i, we switch to a deterministic setting with the feedback control based on u. Otherwise, (i.e., if it stays green at T_i), we continue with w^{i+1}.

3.3.3 Approximating the $\mathcal{J}_3$-Constrained Pareto Front

Our running cost K defined in (3.2) can be viewed as a linear combination of constituent running costs describing separately the fuel usage, discomfort from rapid speed changes, and time to target; i.e., $K_1 = \gamma[a]_+$, $K_2 = a^2$, and $K_3 = 1$, with the coefficients (c_1, c_2, c_3) reflecting the relative importance of these to the driver. Cumulative constituent costs $(\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3)$ can be computed by integrating the corresponding K_i's along the trajectory. Rational tradeoffs among these objectives can be explored by varying ratios of c_i's. To evaluate the fuel-discomfort tradeoffs among trajectories reaching the target in at most $T^\dagger$ seconds, we fix a ratio $\frac{c_1}{c_2}$ and find c_3 such that $\mathcal{J}_3 = T^\dagger$ for the K-optimal trajectory recovered from the HJB PDE. Repeating this process for a range of $\frac{c_1}{c_2}$ values, we obtain a J_3-constrained *Pareto Front* for $(\mathcal{J}_1, \mathcal{J}_2)$.

3.4 Numerical Experiments

"I don't wanna wait in vain."

"Waiting in Vain" by Bob Marley & The Wailers

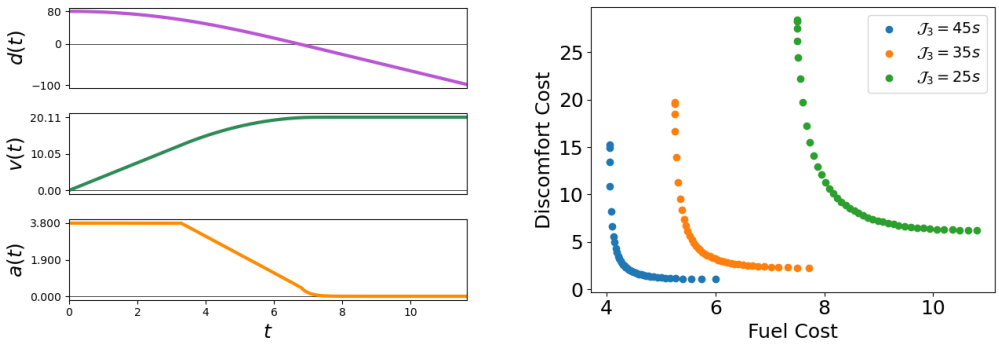
For all experiments[4], we use realistic parameter values $\bar{v} = 45\,mph \approx 20.12\,m/s$, $-\alpha = -3.8\,m/s^2$, $\beta = 3.8\,m/s^2$, $D_{\text{Y}} = 3\,s$, $D_{\text{R}} = 60\,s$. We conduct all planning on a 200-meter-long road segment with the traffic light in the middle; i.e., $\bar{d} = 100\,m$, $d_\ell = 0$, and $d^* = -100\,m$. We use a 345×181 grid in (d, v) space,

[4]In the interest of computational reproducibility, we provide the full source code, additional figures, and movies for all examples at `https://eikonal-equation.github.io/Traffic_Light_Uncertainty/`

with Δt specified by (3.14).

Mirroring the previous sections, we start with the planning in the later (deterministic) phases before considering the effects of uncertainty in the initial phase I.

Example 1: Green phase G only. The planning in this final green phase is particularly easy to interpret since there are no constraints other than the speed limit. With our simplified dynamics, the method of characteristics used on PDE (3.5) shows that any optimal trajectory will have at most three stages: (1) using the maximum acceleration $a = \beta$, (2) linearly decreasing acceleration until $a = 0$, and (3) coasting until we reach the target. We illustrate this for a specific starting $(d, v) = (80, 0)$ in Fig. 3.2A. Starting from the same (d, v), Fig. 3.2B shows the rational (fuel usage / acceleration discomfort) tradeoffs for 3 different constraint levels on the time-to-target.

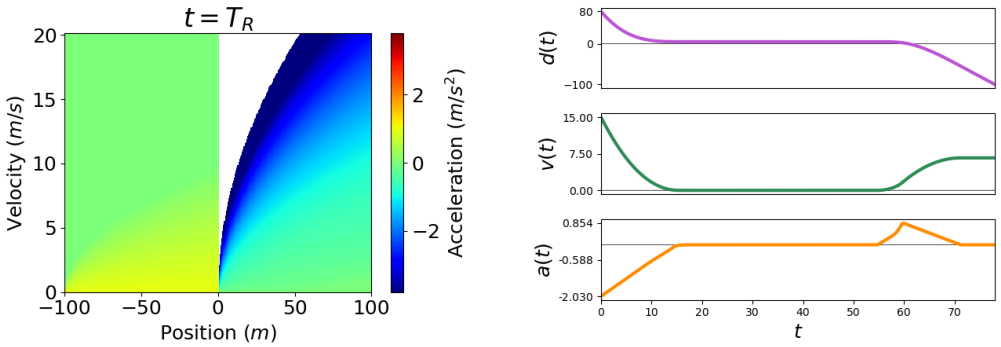


(A) (B)

Figure 3.2: Example 1 optimal trajectory and fuel-discomfort tradeoffs starting from $(d, v) = (80, 0)$. (A): Optimal $d(t)$, $v(t)$, and $a(t)$ when $(c_1, c_2, c_3) = (0.025, 0.025, 0.95)$. The individual costs along the trajectory are $\mathcal{J}_1 = 20.11$, $\mathcal{J}_2 = 66.82$ and $\mathcal{J}_3 = 11.73$s. (B): $\mathcal{J}_3$-constrained Pareto fronts for $(\mathcal{J}_1, \mathcal{J}_2)$ with $\mathcal{J}_3 = 25s$, $35s$, and $45s$.

Example 2: Phases R and G. Figure 3.3 shows what happens when the driver

starts planning as the light turns red at $t = T_{\text{R}}$. Beyond the traffic light, for $d < d_\ell$, the optimal policy is exactly the same as in phase G. But on the other side, the car cannot cross the parabolic boundary $d_\alpha(v) = d$ until $T_{\text{G}} = T_{\text{R}} + 60s$. This requires aggressive braking to the right of this parabola, though the need for harsh braking diminishes as $t \to T_{\text{G}}$ and $d \to \bar{d}$ since it is optimal to reach $d_\alpha(v) = d$ with non-zero speed just as the light turns green.



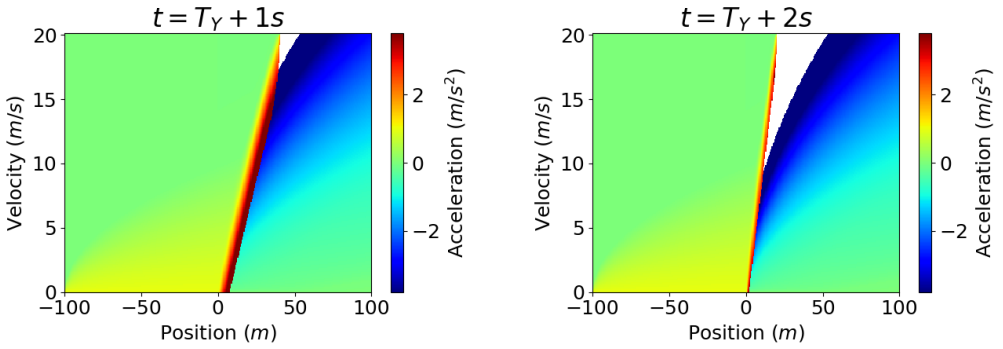
(A) (B)

Figure 3.3: Example 2 feedback controls and optimal trajectory when $(c_1, c_2, c_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. (A): Feedback controls at $t = T_{\text{R}}$ with $\mathcal{I}_{\text{R}}$ shown in white. (B): Optimal $d(t), v(t)$ and $a(t)$ for starting $(d, v, t) = (80, 15, T_{\text{R}})$. It is optimal for the driver to brake, stop, and wait until they can accelerate to arrive at $d_\alpha(v)$ at T_{G} with $v > 0$.

Example 3: Deterministic phases Y, R, and G. In the yellow phase, some of the starting configurations (with $d \leq d_\beta(v, t)$) allow beating the red light if the driver is ready to accelerate aggressively. This leads to a time-dependent disallowed set $\mathcal{I}_Y(t)$ (see Fig. 3.4) and a possible discontinuity in the value function when $d = d_\beta(v, t)$. Fig. 3.5 shows that even a slight shift in a starting position over this discontinuity will result in a drastically different optimal trajectory and cumulative cost.

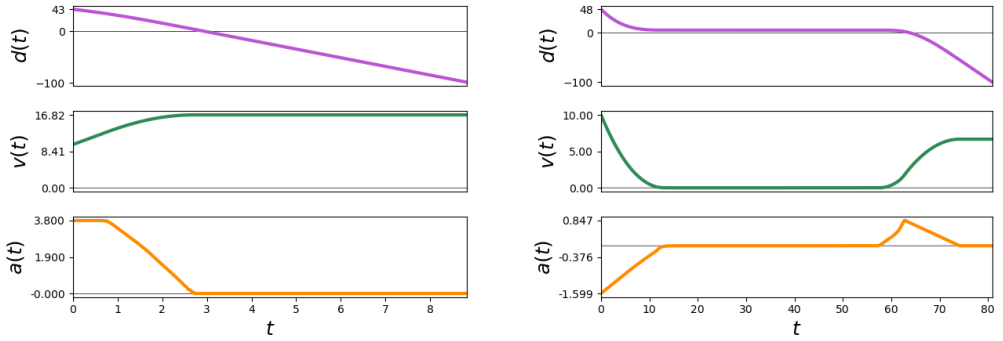
Example 4: Uncertain yellow change, $\widetilde{T}_{\text{Y}} \in \{T_1, T_2\}$.

We now start planning at the time $t = 0$ in the initial green phase I, with the



(A) (B)

Figure 3.4: Example 3 feedback controls for $(c_1, c_2, c_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ at (A): $t = T_Y + 1\,s$ and (B): $t = T_Y + 2\,s$. Both plots are the control-heatmap equivalents to the piecewise boundary schematics shown in Fig. 3.1.



(A) (B)

Figure 3.5: Example 3 optimal trajectories starting from (A): $(d, v, t) = (43, 10, T_Y)$ and (B): $(d, v, t) = (48, 10, T_Y)$. The former yields $(\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3) = (6.82, 21.32, 8.9)$ and the overall cost $\mathcal{J} = 12.35$. The latter results in $(\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3) = (6.65, 14.16, 80.95)$ and a much larger $\mathcal{J} = 33.94$.

remaining time until yellow $\widetilde{T}_Y$ taking values $T_1 = 2\,s$ or $T_2 = 6\,s$ with equal probability ($p_1 = p_2 = 1/2$). We first assume that the driver values all three objectives equally ($c_1 = c_2 = c_3 = \frac{1}{3}$); the resulting optimal feedback control for $t = 0$ is shown in Fig. 3.6A. We focus on the starting configuration $(d, v) = (94, 0.85)$, from which it is possible to beat the traffic light if $\widetilde{T}_Y = T_2$ but not if $\widetilde{T}_Y = T_1$. The optimal trajectory (shown in Fig. 3.6B) branches at the time $t = T_1$, when

we find out the true value of $\widetilde{T}_{\text{Y}}$. We note that the "optimal under uncertainty" control used for $t \in [0, T_1)$ would not be optimal for either deterministic scenario: we would accelerate far less (if at all) with $T_{\text{Y}} = T_1$ and far more with $T_{\text{Y}} = T_2$.

But just because one can beat the traffic light with $\widetilde{T}_{\text{Y}} = T_2$, it does not mean that it is always optimal to do so. Indeed, if this light duration is sufficiently unlikely, the initial acceleration on $[0, T_1)$ to preserve both options is no longer worthwhile. We demonstrate this for $(p_1, p_2) = (0.95, 0.05)$ in Fig. 3.7. (Note that there is still branching at $t = T_1$ since at that point we discover the time $(\widetilde{T}_{\text{Y}} + D_{\text{Y}} + D_{\text{R}})$ by which we need to reach the parabolic boundary as the light turns green again.) Not surprisingly, a similar decision not to rush can also result from a difference in driver's priorities; e.g., a high enough c_2 will make rapid acceleration unattractive even if T_2 is fairly likely. In Fig. 3.8 we demonstrate this for $(p_1, p_2) = (0.5, 0.5)$ and $(c_1, c_2, c_3) = (0.15, 0.75, 0.1)$.

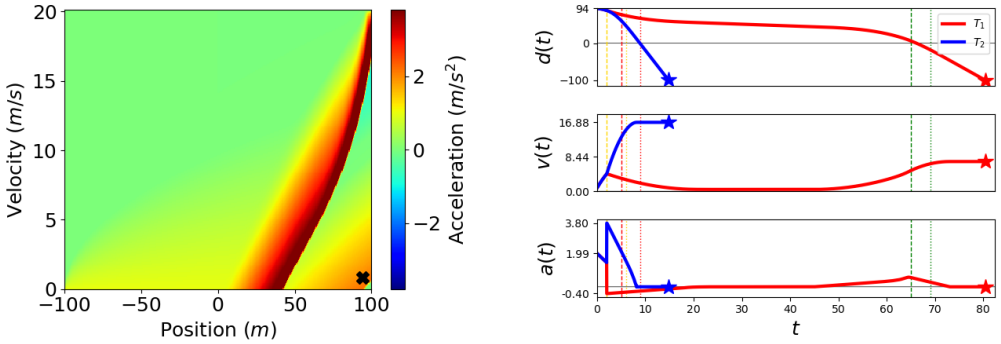
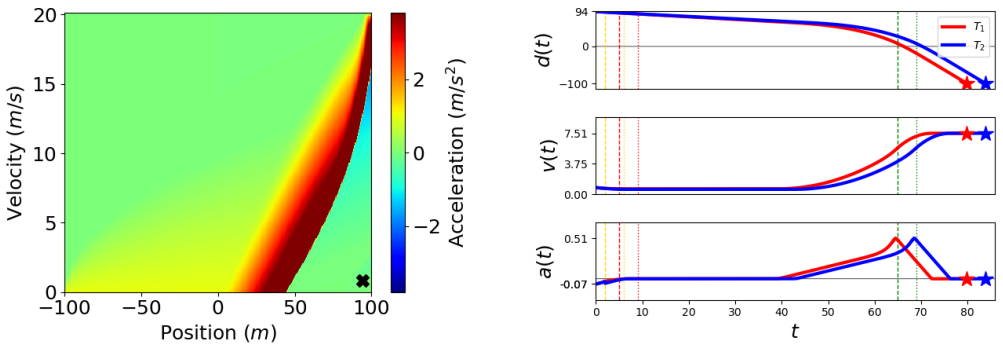


Figure 3.6: Example 4 feedback controls and trajectories for starting $(d, v) = (94, 0.85)$ when $(p_1, p_2) = (0.5, 0.5)$ and $(c_1, c_2, c_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. (A): Feedback controls at $t = 0$. The "X" indicates the vehicle's starting point. (B): Optimal $d(t)$, $v(t)$ and $a(t)$. Vertical lines corresponding to the possible turning yellow, red, and green times are shown in their respective colors.

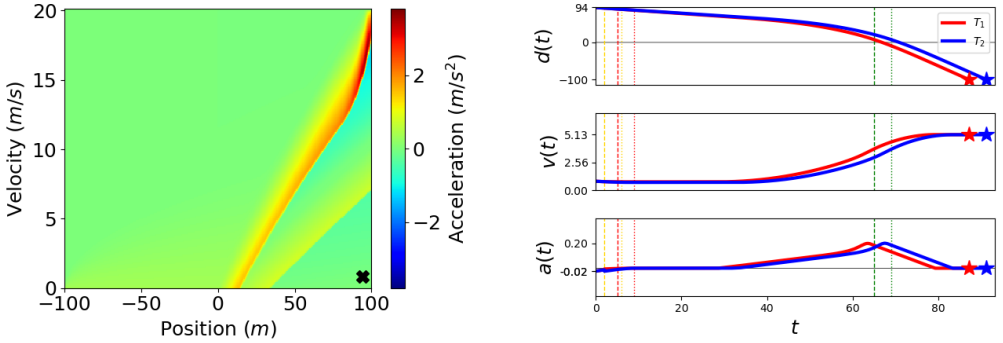
Example 5: Uncertain yellow change, $\widetilde{T}_{Y} \in \{T_1, T_2, T_3\}$.

In our final example, we use $(T_1, T_2, T_3) = (2\,s, 4\,s, 6\,s)$ with $(p_1, p_2, p_3) =$



(A) (B)

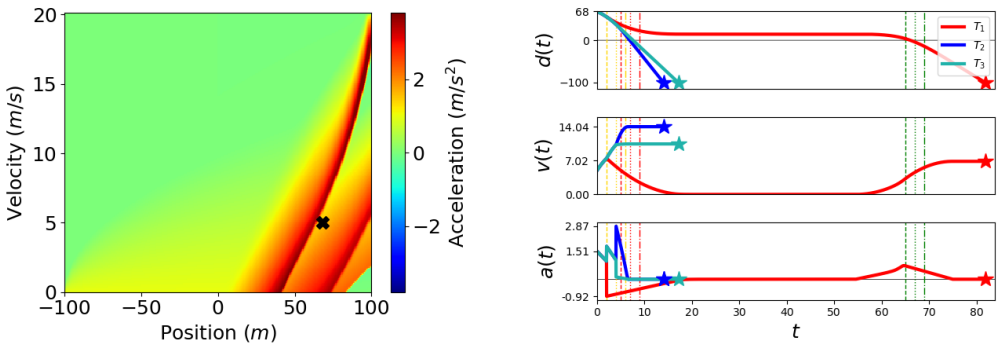
Figure 3.7: Example 4 feedback controls and trajectories for starting $(d, v) = (94, 0.85)$ when $(p_1, p_2) = (0.95, 0.05)$, and $(c_1, c_2, c_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. (A): Feedback controls at $t = 0$. The "X" indicates the vehicle's starting point. (B): Optimal $d(t)$, $v(t)$ and $a(t)$. Vertical lines corresponding to the possible turning yellow, red, and green times are shown in their respective colors.



(A) (B)

Figure 3.8: Example 4 feedback controls and trajectories for starting $(d, v) = (94, 0.85)$ when $(p_1, p_2) = (0.5, 0.5)$, and $(c_1, c_2, c_3) = (0.15, 0.75, 0.1)$. (A): Feedback controls at $t = 0$. The "X" indicates the vehicle's starting point. (B): Optimal $d(t)$, $v(t)$ and $a(t)$. Vertical lines corresponding to the possible turning yellow, red, and green times are shown in their respective colors.

$(0.25, 0.25, 0.5)$ and $c_1 = c_2 = c_3 = \frac{1}{3}$ starting from $(d, v) = (68, 5)$. The optimal control (shown in Fig. 3.9) has three branches: if the light stays green at $t = T_1$, it becomes clear that we can beat the red light, but how much acceleration will be needed to do so will be revealed at $t = T_2$.



(A) (B)

Figure 3.9: Example 5 feedback controls and trajectories for starting $(d, v) = (68, 5)$ when $(p_1, p_2, p_3) = (0.25, 0.25, 0.5)$, and $(c_1, c_2, c_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. (A): Feedback controls at $t = 0$. The "X" indicates the vehicle's starting point. (B): Optimal $d(t)$, $v(t)$, and $a(t)$. The (T_1, T_2, T_3) and their corresponding T_{R} and T_{G} are marked by the yellow, red, and green vertical lines respectively.

3.5 Conclusions

We provide a framework for determining optimal driving strategies in the face of initial traffic light uncertainty and competing optimization objectives. While our basic setup is intentionally simple, the proposed optimization under uncertainty approach is much broader. We hope that it will be useful in modifying prior models with detailed vehicle dynamics ([96]), route selection on complex road networks with many signalized intersections ([69]), and game-theoretic traffic effects due to independent decision making of multiple drivers ([60]). A similar approach will also be useful in treating other uncertainty models with more general (e.g., continuous) phase-duration distributions as well as pedestrian-actuated signal timing changes ([39]).

CHAPTER 4

OPTIMALITY OF MOTION CAMOUFLAGE UNDER ESCAPE UNCERTAINTY

"Moving in the shadows, sneaking past the warning sign."

"Dark Days (Feat. Sylvan Esso)" by Local Natives

Camouflaging is often used in nature as a concealment mechanism by pursuers hoping to sneak up on a potential evader. Although organisms across the animal kingdom utilize a variety of such tactics, the active motion camouflage (MC) displayed by male hover flies during mating [29] and dragonflies settling territorial disputes [74] is of particular interest in both biological and engineering applications since it is induced by the pursuer's *movement* rather than by a set physical characteristic such as fur patterns or body texture. A pursuer engaging in stationary-point MC chooses their trajectory in attempt to trick the evader's visual system into believing that the pursuer is a part of their perceived background optical flow [32]. Figure 4.1 illustrates the MC motion constraints geometrically. When the pursuer's trajectory stays on the line between the evader and the object at the evader's focal point ($\boldsymbol{z}_{\#}$), the evader can only detect changes in the pursuer's size and not any relative motion.

Modeling MC trajectories in a control-theoretic setting has been of interest to biologists and engineers alike since Srinivasan and Davey first proposed a mathematical description of this behavior in [94]. Prior studies in the literature which address these tactics in an optimal control framework primarily aim to determine

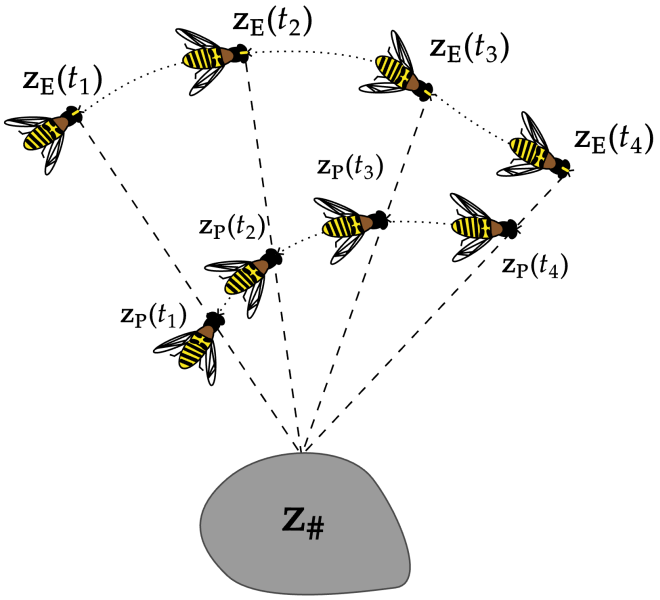


Figure 4.1: Illustration of stationary point motion camouflage with respect to the boulder at $z_\#$. The evader and pursuer at four snapshots in time are identified by $z_E(\cdot)$ and $z_P(\cdot)$ respectively.

optimal trajectories along which the pursuer always uses MC [107, 84, 64, 23]. Such trajectories are generally obtained by optimizing a performance index in an open-loop framework under the MC motion constraints. These approaches do not address the fundamental question of when it may be beneficial for the pursuer to utilize MC, and they also rely on the realism-limiting assumption that the evader will never feel threatened and attempt to escape prior to capture.

The main contribution of this chapter is the development of a continuous-time dynamic programming framework that can be used to determine situations in which it is optimal for a pursuer to utilize MC amidst uncertainty in the evader's escape attempt time. Our framework is centered around solving a sequence of Hamilton-Jacobi-Bellman (HJB) equations from which we can recover the pursuer's optimal trajectories. MC-optimality from a given starting location is then determined by identifying a fraction of the optimal trajectory which is (approxi-

mately) MC.

Although our approach may be adapted to analyze benefits of MC usage in a variety of biological and vehicular systems (see [95, 61, 98] for examples), we illustrate our framework through the model problem of an energy-optimizing hover fly (pursuer) trying to capture another hover fly (evader) that will, at some uncertain time, feel threatened and attempt to escape. The evader's escape attempt is modeled as the result of a non-homoegeneous Poisson point process with rate function λ that controls the rate of switching from an initial stalk phase (in which the pursuer may utilize MC) to a direct chase phase (in which the pursuer rapidly darts after the escaping evader).

The rest of this chapter is organized as follows: Section 4.1 describes the model problem, biological assumptions, and the two pursuit phases. Section 4.2 presents the main optimal control framework and the HJB PDEs. Section 4.3 outlines the numerical methods used to solve the PDEs. Section 4.4 shows four examples which illustrate a sampling of the pursuer's energy-optimal flight trajectories for different λ. We also report statistics on the percentage of stalk phase flight spent in MC from over 200,000 starting points to illustrate how properties of E's visual system affect the optimality of MC tactics. Conclusions and future directions are discussed in Section 4.5. We also note here that while this chapter is based heavily on the work presented in [47], further modifications were made to the original model since its publication.

4.1 Model Problem Formulation

We motivate the development of our optimal control framework by focusing on the following biological model problem:

Model Problem: *When is it optimal for an energy-optimizing hover fly to engage in stationary point motion camouflage while pursuing another hover fly who will, at some unknown time, feel threatened and attempt to escape?*

4.1.1 Notation and Assumptions:

Notation: Throughout the chapter, we will refer to the pursuer hover fly as "P" and the evader hover fly as "E." We let $\boldsymbol{z}_P = [x_P, y_P]^T$ and $\boldsymbol{z}_E = [x_E, y_E]^T$ refer to points in $\mathbb{R}^2$ which specify P and E's initial positions, while $\boldsymbol{z} = [x, y]^T$ refers to a generic point in $\mathbb{R}^2$. The vectors $\boldsymbol{z}_P(t) = [x_P(t), y_P(t)]^T$ and $\boldsymbol{z}_E(t) = [x_E(t), y_E(t)]^T$ reflect the time-dependent nature of P and E's positions.

Biological Assumptions: Following experimental data presented in [100, 30], Table 4.1 summarizes our assumptions about P's physiological characteristics and their associated parameter values. These characteristics include the hover fly's average weight, torso length, wing length, flight speed capabilities, and general visual radius. Since male hover flies are known to patrol resource-rich locations for mates [29], we assume that P pursues E while they are en route to feed at the only food source in the region - a flower located at $\boldsymbol{z}_\star = (x_\star, y_\star)$. Male hover flies can predict a female's trajectory reasonably well [30], so we suppose that P has complete knowledge of E's route $\boldsymbol{z}_E(t)$ to $\boldsymbol{z}_\star$ at the start of the planning process.

With compound-eyes, hover flies have $\approx 360^\circ$ vision, but we assume that their

Table 4.1: Summary of considered hover fly physiological traits and their assumed parameter values

Hover Fly Physiological Traits & Biological Parameter Values			
Trait	Symbol	Value	Units
Approx. torso length	l	7.75	mm
Approx. wing length	w	5.625	mm
Approx. avg. weight	$-$	12	mg
Radius within which vision is strongest	D	0.75	m
Drag Coefficient	C	2.4	n/a
Density of air	ρ	1.23	kg/m^3
Cross sectional area	A	1.19×10^{-4}	m^2
P's maximum stalk flight speed	F_P	4	m/s
P's maximum chase flight speed	G_P	11.11	m/s
E's flight speed while being chased	G_E	10	m/s

vision is *strongest* within a D meter radius [29]. Beyond this radius, E's visual detection abilities continuously wane.

4.1.2 Pursuit Phases and Planning Horizons

Based on behavioral observations [106], the planning process consists of two phases:

1. *Stalk Phase*: E has not yet felt threatened by P and has not initiated an escape attempt. P may use MC tactics in this stage.
2. *Direct Chase Phase*: E feels threatened by P and immediately attempts to escape by directly fleeing along a straight line trajectory. P chases directly

after E along their escape trajectory.

Stalk Phase Time Horizon & Quantifying MC Deviation: The stalk phase begins at $t = 0$ and lasts until $T_s = \min\left\{\hat{T}_1, T_1\right\}$. $\hat{T}_1$ is the uncertain time at which E initiates their escape, and $T_1 = \min\{t \geq 0 \,|\, \boldsymbol{z}_P(t) \in \mathcal{T}\}$, where $\mathcal{T} := \{\boldsymbol{z} \in \mathbb{R}^2 \,|\, |\boldsymbol{z} - \boldsymbol{z}_E(t)| \leq \epsilon\}$. E's escape attempt can either occur while traveling to $\boldsymbol{z}_\star$ or while sitting stationary at $\boldsymbol{z}_\star$ during feeding. The escape attempt time is a non-homogeneous exponentially distributed random variable with pointwise rate function λ that depends on P's trajectory at time t and properties of E's visual system. This rate is primarily influenced by two factors:

- How attuned E is to P's angular deviations in the background optical flow.
- How tolerant E is to P's proximity (mutual distance) within their visual field.

At time t, the angular displacement between E's focal line to $\boldsymbol{z}_\#$ and P's current position $\boldsymbol{z}$ is given by

$$\theta(\boldsymbol{z}, t) := \cos^{-1}\left(\frac{\boldsymbol{r}_\#(t) \cdot \boldsymbol{r}(t)}{|\boldsymbol{r}_\#(t)||\boldsymbol{r}(t)|}\right), \tag{4.1}$$

where $\boldsymbol{r}(t) := \boldsymbol{z} - \boldsymbol{z}_E(t)$, and $\boldsymbol{r}_\#(t) := \boldsymbol{z}_\# - \boldsymbol{z}_E(t)$. A diagram of these vectors in relation to $\boldsymbol{z}_E$, $\boldsymbol{z}_P$, and $\boldsymbol{z}_\#$ is displayed in Figure 4.2. We also note that θ is irrelevant when $|\boldsymbol{r}(t)| = 0$ since capture will have already occurred. We say that E is in "perfect MC" when $\theta = 0$ and $\boldsymbol{r}(t)$ is exactly aligned with $\boldsymbol{r}_\#(t)$.

While θ is an exact quantification of P's instantaneous MC violation as discussed in [85], it alone is not a suitable measure of how E perceives and responds to P's MC behavior. After a certain threshold violation angle $\tilde{\theta} \in [0, \frac{\pi}{2}]$, P's MC violation is too large to have any meaningful impact on whether or not E will perceive them as a threat. The only influencing factor on E's threat perception when

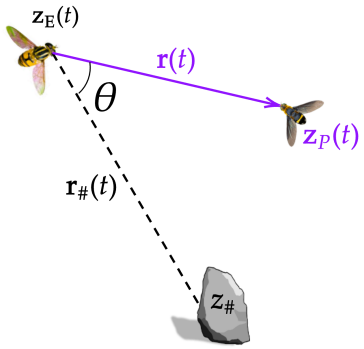


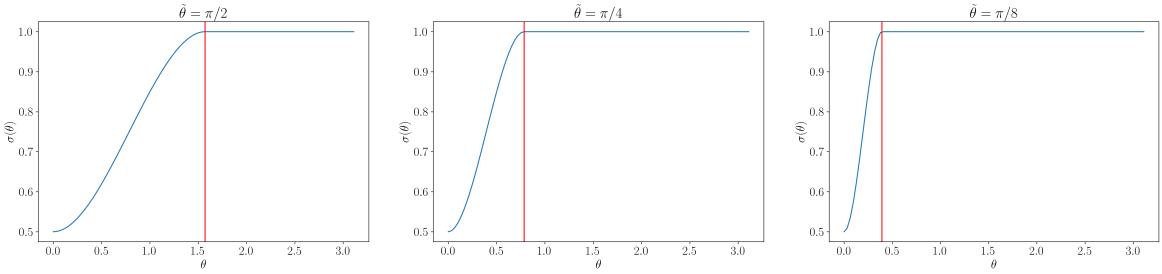
Figure 4.2: Diagram of $\boldsymbol{r}_{\#}(t)$ and $\boldsymbol{r}(t)$. The exact MC violation θ is related to the dot product of those two vectors.

$\theta \geq \tilde{\theta}$ is the mutual distance between the two hover flies. To this end, E is *more* attuned to angular deviations when $\tilde{\theta}$ is *lower*, so P must remain closer to perfect MC if they want to successfully trick E using this tactic. On the other hand E is *less* attuned to the angular deviations when $\tilde{\theta}$ is *higher*, and P can stray a bit farther from perfect MC without dramatically increasing the chances of spooking E.

We can quantify these effects on E's threat perception by defining the following smoothstep *response function*,

$$\sigma(\theta) = \begin{cases} B & \theta = 0 \\ -2(1-B)\tilde{\theta}^{-3}\theta^3 + 3(1-B)\tilde{\theta}^{-2}\theta^2 + B & 0 \leq \theta \leq \tilde{\theta} \\ 1 & \theta \geq \tilde{\theta}, \end{cases} \tag{4.2}$$

where $B \in (0, 1)$ reflects how well-duped E is by MC behaviors. Lower values of B indicate that E is strongly tricked by MC tactics overall, while higher values of B indicate that E is not as responsive to MC. E's response increases from B to 1 as a function of θ, and $\tilde{\theta}$ controls how rapid this increase is. We provide plots of $\sigma(\theta)$ for three different values of $\tilde{\theta}$ and $B = 1/2$ in Figure 4.3.



(A) (B) (C)

Figure 4.3: Plots of $\sigma(\theta)$ for: (A) $\tilde{\theta} = \pi/2$, (B) $\tilde{\theta} = \pi/4$, and (C) $\tilde{\theta} = \pi/8$. In each plot, $\tilde{\theta}$ is indicated by the red vertical line.

We now utilize the mutual distance between the two hover flies and $\sigma(\theta)$ to define the threat detection rate. Taking inspiration from a two-dimensional Gaussian, the pointwise threat detection rate is

$$\lambda(\boldsymbol{z}, t) := S \exp\left(-\frac{M(|\boldsymbol{r}(t)| - \epsilon)^2}{\sigma(\theta)}\right) \tag{4.3}$$

where S is a positive constant set to reflect the overall strength of E's detection abilities, $M > 0$ is a positive constant that reflects how tolerant E is to P's proximity. We note that when $t \geq T_\star$, the t-dependence in λ can be suppressed since E's location is fixed at $\boldsymbol{z}_\star$. In light of the above, the probability that E does *not* attempt an escape within the time interval $[t, t + \tau]$ is

$$p(\boldsymbol{z}(t), t) = e^{-\int_t^{t+\tau} \lambda(\boldsymbol{z}(s), s) ds}. \tag{4.4}$$

Direct Chase Phase Time Horizon: At T_s, the direct chase begins, and it ends with E's capture at $T_2 = \min\{t \geq T_s \mid |\boldsymbol{z}_P(t) - \boldsymbol{z}_E(t)| \leq \gamma\}$. A table summarizing the phases, termination times, and termination criteria presented in the discussion above is available in Table 4.2.

Energy Expenditure Quantification: Metabolic processes in biological systems are inherently complex. To keep the discussion straightforward and retain the

Table 4.2: Summary of pursuit stages and time horizons

Pursuit Stages & Their Termination Criteria		
Phase	Termination Time	Termination Criteria / Explanation
Stalk	$T_s = \min\{\hat{T}_1, T_1\}$	Ends when E is spooked at either $\hat{T}_1$ (uncertain time) or T_1 (deterministic, time when P is first within ϵ m of E).
Direct Chase	T_2	When P is first within γ m of E.

focus on the mathematical framework being developed, we focus on quantifying the following:

W: The amount of baseline operational power that P needs to exert to at least stay afloat in air via hovering.

K: The amount of power that P needs to expend to overcome the drag force during flight.

We model the hovering power W as a constant, and following the experimental results published in [76], we take the hover fly's average mass-specific hover power to be $\approx$ 42.5 watts per kilogram. This yields an approximate hovering power $W \approx 0.00051$ watts for a 12 milligram hover fly.

To determine the power K required to overcome the drag force at time t, we first compute the drag force $\boldsymbol{F}_D(t)$ on the hover fly. $\boldsymbol{F}_D(t)$ *opposes* P's direction of motion at time t, and it is heavily dependent on the hover fly's Reynold's number (Re) and drag coefficient (C).

Although Re is often determined experimentally, its mathematical formula can

be derived from the Navier-Stokes equations and is given by

$$Re = \frac{\rho L|\boldsymbol{v}|}{\eta} \tag{4.5}$$

where ρ is the density of the fluid in kg/m^3, L is the characteristic length of the object, and η is the viscosity of the fluid. We take $\rho \approx 1.23\,kg/m^3$, $L \approx 0.00775\,m$ (the length of the hover fly's torso), and $\eta \approx 1.81 \times 10^{-5}\,kg/ms$ (the viscosity of air at approximately 60° Fahrenheit). Thus, our approximation for the hover fly's Re satisfies

$$Re \approx 509.67|\boldsymbol{v}|. \tag{4.6}$$

Based on our assumptions about the hover fly's flight speed capabilities outlined in Table 4.1, the hover fly's Reynolds number satisfies the following upper bound through both the stalk phase and the direct chase phase,

$$Re < 5662.43. \tag{4.7}$$

This upper bound on Re suggests that the hover fly will never enter the turbulent flow regime that occurs when $Re > 2.5 \times 10^5$.

We now use the insight into the hover fly's Reynolds number to determine an appropriate formula for the magnitude of the drag force. Although the relationship between C and Re is usually determined experimentally, several approximation formulas have been developed for $Re < 2 \times 10^5$ [57]. These approximations have the general form

$$C = \frac{24}{Re} + \Upsilon, \tag{4.8}$$

where Υ is a sum of situation-specific correction terms. For $Re < 2 \times 10^5$, a common approximation formula takes $\Upsilon = \frac{6}{1+\sqrt{Re}} + 0.4$. While we acknowledge that including the terms in Υ would lead to greater physical realism in the computations, we opt to ignore them for the sake of simplicity. Using $C \approx \frac{24}{Re}$ and the fact that

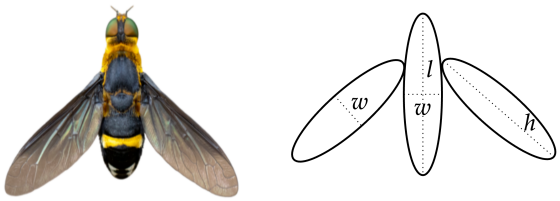


Figure 4.4: Diagram of the hover fly's cross sectional area approximation. The cross sectional area A is approximated by computing the sum of the areas of two ellipses for the wings and the area of one ellipse for the torso (including the head). The two wings are identical and are assumed to have major radius $h/2$ and minor radius $w/2$. The torso has major radius $l/2$ and minor radius $w/2$. Values for l, h, and w are specified in Table 4.1.

the drag coefficient is related to the *magnitude* of the drag force by $C = \frac{2F_D}{\rho|\boldsymbol{v}|^2L^2}$, we find that

$$F_D \approx 12\eta L|\boldsymbol{v}|. \tag{4.9}$$

This suggests that in the relevant Reynolds number regime, the magnitude of the drag force is *linearly* proportional to the hover fly's speed, $|\boldsymbol{v}(t)|$. Thus,

$$\boldsymbol{F}_D(t) = \left(\frac{1}{2}C\rho A|\boldsymbol{v}(t)|\right)\frac{-\boldsymbol{v}(t)}{|\boldsymbol{v}(t)|} \tag{4.10}$$

where A is the approximation of the hover fly's cross sectional area. A diagram of the approximation approach is provided in Figure 4.4. Thus, the force required of P to overcome drag at time t is $\boldsymbol{F}_P(t) = -\frac{1}{2}C\rho A\boldsymbol{v}(t)$, and using the definition of power yields

$$P(t) = \frac{1}{2}C\rho A\boldsymbol{v}(t)\cdot\boldsymbol{v} = \frac{1}{2}C\rho A|\boldsymbol{v}(t)|^2. \tag{4.11}$$

As such, the running cost function that captures P's *total* power expenditure at time t is

$$K(\boldsymbol{v}(t)) = \frac{1}{2}C\rho A|\boldsymbol{v}(t)|^2 + W. \tag{4.12}$$

By the Work-Energy Theorem, the integral of $K(\boldsymbol{v}(t))$ over the time interval $[t, t+\tau]$ gives P's cumulative energy loss during that period.

4.2 Optimal Control Framework

We now present the full optimal control framework that is used to address the Model Problem introduced in Section 4.1. We begin with an overview of the deterministic optimal control problem that models the direct chase phase interaction and then transition into a discussion of the randomly terminated optimal control problem characterizing the stalk phase interaction.

4.2.1 Direct Chase Phase

> "I should be flying straight with no delay, 'Cause time takes from everyone."
>
> "Patience" by Tame Impala

In the deterministic direct chase phase ($t \geq T_s$), P optimizes for energy loss while darting after E in response to their escape attempt. Experimental results suggest that hover flies generally follow escape trajectories that correspond with direct fleeing *away* from the perceived threat [38, 99]. In mathematical terms, E flies along a straight-line escape trajectory in the direction *opposite* of $\boldsymbol{r}(T_s)$ with constant speed G_E. P then darts after E along the same line with speed $v \in S = [G_E + \alpha, G_P]$, where α is a positive constant smaller than $G_P - G_E$. Since P's minimum direct chase speed is larger than G_E, P will eventually capture E when their mutual distance falls below γ meters.

P and E's coinciding straight line trajectory and the capture guarantee posed by their relative speeds allows us to reduce the spatial dimension of the problem

from two to one. For notational purposes, we let $z(t) = |\boldsymbol{z}_P(t) - \boldsymbol{z}_E(t)|$,. The system dynamics here is simply the relative velocity $\dot{z}(s) = G_E - v(s)$. Thus, P's optimal direct chase energy expenditure can be determined by solving the following deterministic optimal control problem:

$$q(z) = \inf_{v(\cdot)\in\mathcal{S}} \left\{ \int_t^{T_2} \frac{1}{2} C\rho A (v(s))^2 + W ds \right\} \tag{4.13}$$

where $\mathcal{S}$ is the set of admissible control functions. An application of dynamic programming to (4.13) and appropriately Taylor expanding shows that the value function $q(z)$ is the solution to the following ordinary differential equation (ODE)

$$G_E \frac{dq}{dz} + \min_{v\in S} \left\{ -v\frac{dq}{dz} + \frac{1}{2} C\rho A v^2 \right\} + W = 0 \tag{4.14}$$

with boundary condition $q(z) = 0$ at $z = \gamma$. Since the direct chase is *radially symmetric*, the solution to Equation (4.14) is the same along every direction around the circle surrounding E.

It is straightforward to analytically determine P's optimal speed of motion v_* during the direct chase. Assume first that P is allowed to move with unbounded speed. Taking the derivative of Equation (4.13) with respect to v and solving for the minimizer as function of $\frac{dq}{dz}$ gives

$$v_* = \frac{1}{C\rho A}\frac{dq}{dz}. \tag{4.15}$$

Substituting the optimal speed given in (4.15) back into the ODE yields the following quadratic for $\frac{dq}{dz}$,

$$-\frac{1}{2C\rho A}\left(\frac{dq}{dz}\right)^2 + G_E\frac{dq}{dz} + W = 0. \tag{4.16}$$

The roots are $\frac{dq}{dz} = C\rho A G_E \mp C\rho A\sqrt{G_E^2 + \frac{2W}{C\rho A}}$. Taking the positive root and plugging in the numerical values for G_E, G_P, C, ρ, and A as given in Table 4.1 reveals that $v_* \approx 20.14\, m/s$. Since the Hamiltonian is *convex* in v and the optimum

value falls *above* the admissible range of speed values for P, P's optimal velocity is $v_* = G_P = 11.11\, m/s$.

From the analysis above, it is straightforward to produce an analytical formula for P's optimal direct chase energy expenditure, taking into account the constant nature of $\frac{dq}{dz}$ and the result that $v_* = 11.11\, m/s$ for the given set of model parameters. Starting from position $\boldsymbol{z}$ at time $t \in [0, T_2]$, the optimal amount of energy that P will spend chasing E to capture is given by,

$$q(\boldsymbol{z}, t) = \frac{|\boldsymbol{z} - \boldsymbol{z}_E(t)|}{(v_* - G_E)} \left(\frac{1}{2} C \rho v_*^2 + W \right). \tag{4.17}$$

4.2.2 Stalk Phase Optimal Control Problem

> "I was only walking by, didn't mean to catch your eye."
>
> "Prep-School Gangsters" by Vampire Weekend

During the stalk phase of uncertain duration ($t \in [0, T_s)$), P's optimal course of action is found by solving the following problem:

Stalk Phase Problem: *Solve a randomly-terminated optimal control problem to determine P's optimal velocity $\boldsymbol{v}(\cdot)$ which minimizes their expected energy loss until the direct chase begins at T_s.*

Since E is stationary on the flower until being spooked or until P gets within ϵ cm, we compute P's expected energy loss over two stages $t \in [0, T_\star)$ and $t \geq T_\star$. Our approach to this two-stage control problem falls within the general class of randomly terminated optimal control problems developed in [5] and similar to

multistage examples discussed in [83]. While these two stages can be combined and treated with one HJB PDE, we choose to separate them in our presentation here for the sake of clarity.

Dynamics: P moves with the simple dynamics

$$\dot{\boldsymbol{z}}_P(t) = \boldsymbol{v}(t) \tag{4.18}$$

where the velocity vector $\boldsymbol{v}(t)$ is restricted to the ball $\mathcal{V} := \{\boldsymbol{v} \in \mathbb{R}^2 \mid 0 \leq |\boldsymbol{v}| \leq F_P\}$ for all $t \in [0, T_s)$. F_P is assumed to be $4\,m/s$ as given in Table 4.1. P can predict E's stalk phase trajectory, so E's position $\boldsymbol{z}_E(t)$ is given by a known function $\boldsymbol{f}(t)$, $\boldsymbol{f} : [0, T_\star] \to \mathbb{R}^2$. We also impose the restriction on E's trajectory that $|\boldsymbol{f}'(t)| \leq F_P$.

Indefinite Horizon ($t \geq T_\star$): If E arrives at $\boldsymbol{z}_\star$ *without* being spooked, P's expected energy loss after $T_\star$ is characterized by an indefinite horizon problem which terminates at $\hat{T}_\star = \min\{\hat{T}_1, T_1\}$. The expected cost is

$$\mathcal{C}_S(\boldsymbol{z}, \boldsymbol{v}(\cdot)) = \mathbb{E}_{\hat{T}_\star}\left\{\int_t^{\hat{T}_\star} K(\boldsymbol{v}(s))\, ds + q(\boldsymbol{z}(\hat{T}_\star))\right\}. \tag{4.19}$$

where $t \geq T_\star$, and the time dependence in (4.17) is suppressed since E is stationary at $\boldsymbol{z}_\star$. The value function is

$$w(\boldsymbol{z}) = \inf_{\boldsymbol{v}(\cdot)} \{\mathcal{C}_S(\boldsymbol{z}, \boldsymbol{v}(\cdot))\}, \tag{4.20}$$

and by standard arguments in [8], $w(\boldsymbol{z})$ is a viscosity solution of the stationary (time-independent) HJB PDE

$$0 = \min_{\boldsymbol{v} \in \mathcal{V}} \{K(\boldsymbol{v}) + \nabla w(\boldsymbol{z}) \cdot \boldsymbol{v}\} + \lambda(\boldsymbol{z})(q(\boldsymbol{z}(\hat{T}_\star)) - w(\boldsymbol{z})) \tag{4.21}$$

with the boundary condition $w(\boldsymbol{z}) = q(\boldsymbol{z})$ for all $\boldsymbol{z} \in \mathcal{T}$.

Finite Horizon ($t \in [0, T_\star)$): Prior to E's arrival at $\boldsymbol{z}_\star$, the expected energy loss is described by a finite horizon problem which terminates at $\hat{T}_F = \min\{\hat{T}_1, T_\star, T_1\}$.

It is given by

$$\mathcal{C}_F(\boldsymbol{z}, t, \boldsymbol{v}(\cdot)) = \mathbb{E}_{\hat{T}_F}\left\{\int_t^{\hat{T}_F} K(\boldsymbol{v}(s))\, ds + \tilde{\delta}(\boldsymbol{z}_P(\hat{T}_F), \hat{T}_F)\right\}. \tag{4.22}$$

where $\tilde{\delta}(\boldsymbol{z}_P(\hat{T}_F), \hat{T}_F) = q(\boldsymbol{z}(\hat{T}_F), \hat{T}_F)$ when $\hat{T}_F < T_\star$, and $\tilde{\delta}(\boldsymbol{z}_P(\hat{T}_F), \hat{T}_F) = w(\boldsymbol{z}(T_\star))$ when $\hat{T}_F = T_\star$. The value function for $t \in [0, T_\star)$ is defined as

$$u(\boldsymbol{z}, t) = \inf_{\boldsymbol{v}(\cdot)} \left\{\mathcal{C}_F(\boldsymbol{z}, t, \boldsymbol{v}(\cdot))\right\}, \tag{4.23}$$

and arguments in [8] show that $u(\boldsymbol{z}, t)$ is a viscosity solution of the following time-dependent HJB PDE,

$$0 = u_t(\boldsymbol{z}, t) + \min_{\boldsymbol{v}\in\mathcal{V}} \left\{K(\boldsymbol{v}) + \nabla u(\boldsymbol{z}, t) \cdot \boldsymbol{v}\right\} + \lambda(\boldsymbol{z}, t)(q(\boldsymbol{z}, t) - u(\boldsymbol{z}, t)) \tag{4.24}$$

with the terminal condition $u(\boldsymbol{z}, T_\star) = w(\boldsymbol{z})$ and the boundary condition on a time-dependent neighborhood of E, $u(\boldsymbol{z}, t) = q(\boldsymbol{z}, t)$ for all $\boldsymbol{z} \in |\boldsymbol{z} - \boldsymbol{f}(t)| \leq \epsilon$.

Since the cost function $K(\boldsymbol{v})$ is *isotropic*, the optimal velocity vector $\boldsymbol{v}_* = (v_x, v_y)$ in Equations (4.21) and (4.24) will always *oppose* the direction of the gradient. As such, it is straightforward to determine an analytical expression for $\boldsymbol{v}_*$ as a function of ∇u. Assuming that P's speed is *unconstrained*, the first order optimality conditions require the gradient of the control Hamiltonian $H(\boldsymbol{v}) = K(\boldsymbol{v}) + \nabla u(\boldsymbol{z}, t) \cdot \boldsymbol{v}$ to satisfy

$$\frac{\partial H}{\partial v_x} = 0 \quad \text{and} \quad \frac{\partial H}{\partial v_y} = 0. \tag{4.25}$$

This yields

$$v_x = -\frac{1}{C\rho A}\frac{\partial u}{\partial x}, \quad v_y = -\frac{1}{C\rho A}\frac{\partial u}{\partial y}, \quad \text{and} \quad |\boldsymbol{v}_*| = \frac{|\nabla u|}{C\rho A}. \tag{4.26}$$

Since $K(\boldsymbol{v})$ is *convex*, the resulting $\boldsymbol{v}_*$ given by the formulas in (4.26) is the global, speed-unconstrained minimum of H. However, when $|\boldsymbol{v}_*|$ *exceeds* F_P, P's speed-constrained optimal velocity becomes

$$\boldsymbol{v}_* = -F_P \frac{\nabla u}{|\nabla u|}. \tag{4.27}$$

4.3 Numerical Implementation

The numerical method used to solve the stalk phase HJB PDEs is based on a first order semi-Lagrangian (SL) discretization [42] of (4.21) and (4.24) on a Cartesian grid Z over the domain $\overline{\Omega} := [0, \bar{d}] \times [0, \bar{d}] \times [0, T_\star]$. Z consists of $N_d + 1$, $N_d + 1$, and $N_t + 1$ nodes in the x, y, and t dimensions respectively. For fixed N_d, the grid spacings are given by

$$\Delta x = \Delta y = \frac{\bar{d}}{N_d}, \;\; \Delta t \leq \frac{\min(\Delta x, \Delta y)}{F_P}. \tag{4.28}$$

Thus, P's position and time at node (i, j, k) is $(\boldsymbol{z}_{ij}, t_k) = (i\Delta x, j\Delta y, k\Delta t)$ for $i, j = 0, \ldots, N_d$ and $k = 0, \ldots, N_t$. After τ seconds of applying control $\boldsymbol{v}$ starting from $\boldsymbol{z}_{ij}$, P ends up at position $\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}} = \boldsymbol{z}_{ij} + \tau \boldsymbol{v}$. We take $\tau = \Delta t$, and wherever the probability that E will not be threatened is used, we compute it as $p(\boldsymbol{z}_{ij}, t_k) \approx \tau\lambda(\boldsymbol{z}_{ij}, t_k)$. The value function at points falling within $|\boldsymbol{z}_{ij} - \boldsymbol{z}_E(t_k)| \leq \gamma$ is set to the analytical solution $q(\boldsymbol{z}_{ij}, t_k)$.

4.3.1 Numerical Method for the Stationary HJB

The solution $w(\boldsymbol{z}_{ij})$ to Equation (4.21) is approximated by a grid function W_{ij}. For all $i, j < N_d + 1$, W_{ij} satisfies,

$$W_{ij} = \min_{\boldsymbol{v} \in \mathcal{V}} \left\{ \tau K(\boldsymbol{v}(s)) + p(\boldsymbol{z}_{ij}, t_k) W(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}}) + (1 - p(\boldsymbol{z}_{ij}, t_k)) q(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}}) \right\}. \tag{4.29}$$

The value for $W(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}})$ is computed via bilinear interpolation, and $q(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}})$ is computed via the formula given in (4.17). We solve the system of equations using *value iterations*. We execute a variant of the Fast Sweeping method described in [109] which utilizes Gauss-Seidel (GS) iterations with alternating directions to produce

the approximation to w at each node. At each iteration, the domain is swept using one of the following orderings:

(1): $i = 0, \ldots, N_d$; $j = 0, \ldots, N_d$

(2): $i = N_d, \ldots, 0$; $j = 0, \ldots, N_d$

(3): $i = N_d, \ldots, 0$; $j = N_d, \ldots, 0$

(4): $i = 0, \ldots, N_d$; $j = N_d, \ldots, 0$

The update candidate at node (i, j) and iteration $k + 1$ is generally given by

$$W_{ij}^{k+1} = \min_{\boldsymbol{v} \in \mathcal{V}} \left\{ \tau K(\boldsymbol{v}(s)) + p(\boldsymbol{z}_{ij}, t_k) \widetilde{W}^{k,k+1}(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}}) + (1 - p(\boldsymbol{z}_{ij}, t_k)) q(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}}) \right\}. \quad (4.30)$$

$\widetilde{W}_{i,j}^{k,k+1}$ is the approximation to the value function at $\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}}$ computed via bilinear interpolation. Since the value at some of the grid nodes used in the bilinear interpolation routine may have already been updated at iteration $k+1$, the superscript in $\widetilde{W}$ accounts for this while acknowledging that *which* of these values have already been updated depends on the sweep ordering being used at the current iteration.

Since $\widetilde{W}^{k,k+1}$ is dependent on the current value at node (i, j), we can greatly accelerate the convergence by eliminating W_{ij}^k from the right hand side of (4.30). Suppressing the $\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}}$ and t_k dependence in p for notational simplicity, we obtain the equivalent update formula,

$$\begin{aligned} W_{ij}^{k+1} = \min_{\boldsymbol{v} \in \mathcal{V}} \Bigg\{ & \frac{1}{1 - p\alpha_1(\boldsymbol{v})} (\tau K(\boldsymbol{v}) + p(\alpha_2(\boldsymbol{v}) W_2^{k,k+1} + \alpha_3(\boldsymbol{v}) W_3^{k,k+1} \\ & + \alpha_4(\boldsymbol{v}) W_4^{k,k+1}) + (1 - p) q(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}})) \Bigg\} \end{aligned} \quad (4.31)$$

where $W_2^{k,k+1}, W_3^{k,k+1}$, and $W_4^{k,k+1}$ are the remaining corners of the interpolation cell used to compute $\widetilde{W}^{k,k+1}$, and $\alpha_1(\boldsymbol{v}), \alpha_2(\boldsymbol{v}), \alpha_3(\boldsymbol{v})$, and $\alpha_4(\boldsymbol{v})$ are the control-dependent interpolation weights associated with each node.

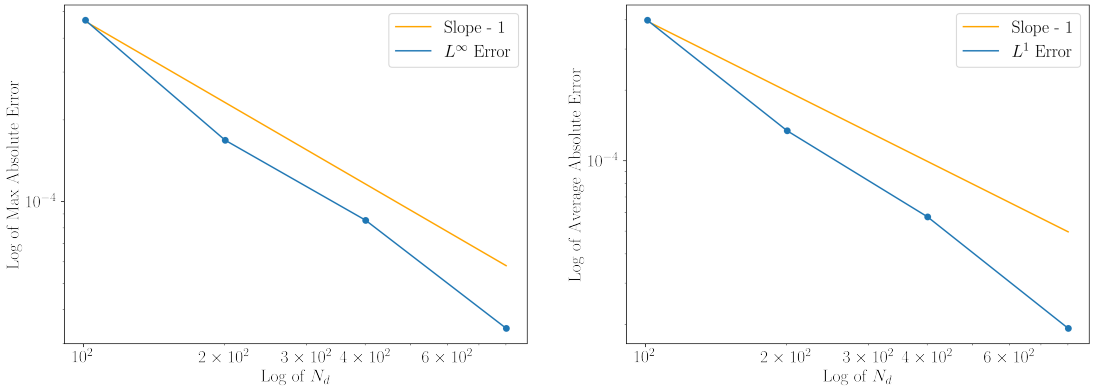
The optimal velocity $\boldsymbol{v}_*$ can be determined via a grid search over a discretized version of $\mathcal{V}$ or via an analytical expression with appropriate gradient approximation. Further details of the latter approach are discussed at the end of this subsection.

Stationary HJB Convergence Studies

We establish the stationary PDE's convergence under grid refinement by computing a "ground truth" solution $\overline{W}$ on a 1601×1601 spatial grid in the case where λ is quite anisotropic (e.g., the parameters used in Example 4.4.4 when $M = 10$ and $\tilde{\theta} = \pi/8$). We set the camouflaging point $\boldsymbol{z}_{\#}$ to be $0.6\,m$ due south of E's position at $\boldsymbol{z}_\star = (2.4, 2.1)$. We determine the infinity norm and L^1 norm of the pointwise absolute error between $\overline{W}$ and the solution W_{N_d} of the corresponding HJB PDE computed on an $N_d + 1 \times N_d + 1$ spatial grid for $N_d \in \{100, 200, 400, 800\}$. The algorithm terminates once the maximum pointwise difference between successive iterations falls below 1×10^{-6}. Log-log convergence plots for both norms are shown in Figure 4.5. A table containing the number of iterations (sweeps) needed for termination, the norms, and convergence rate estimates are provided in Table 4.3. It is evident from the convergence plots shown in Figure 4.5 and the data in Table 4.3 that our method is converging under the grid refinement for anisotropic λ.

4.3.2 Numerical Method for the Time-Dependent HJB

Equation (4.24) is treated similarly, but with slight modifications to accommodate the time dependence. The grid function U_{ij}^k that approximates the solution to



(A) (B)

Figure 4.5: Stationary stalk phase HJB PDE convergence study plots. (A) Log-log plot of the infinity norm of the absolute error when $M = 10$, $\tilde{\theta} = \pi/8$. (B) Log-log plot of the L^1 norm of the absolute error when $M = 10, \tilde{\theta} = \pi/8$.

Table 4.3: Stationary HJB Convergence Study Data

Stationary Conv. Study: $M = 10$, $\tilde{\theta} = \pi/8$					
N_d	Sweeps	L^∞ Error	$\ln\left(\frac{e_{k+1}}{e_k}\right)$	L^1 Error	$\ln\left(\frac{e_{k+1}}{e_k}\right)$
100	12	0.00046	–	0.00039	–
200	12	0.00017	1.468	0.00013	1.569
400	12	8.53×10^{-5}	0.978	5.75×10^{-5}	1.221
800	12	3.41×10^{-5}	1.321	1.92×10^{-5}	1.581

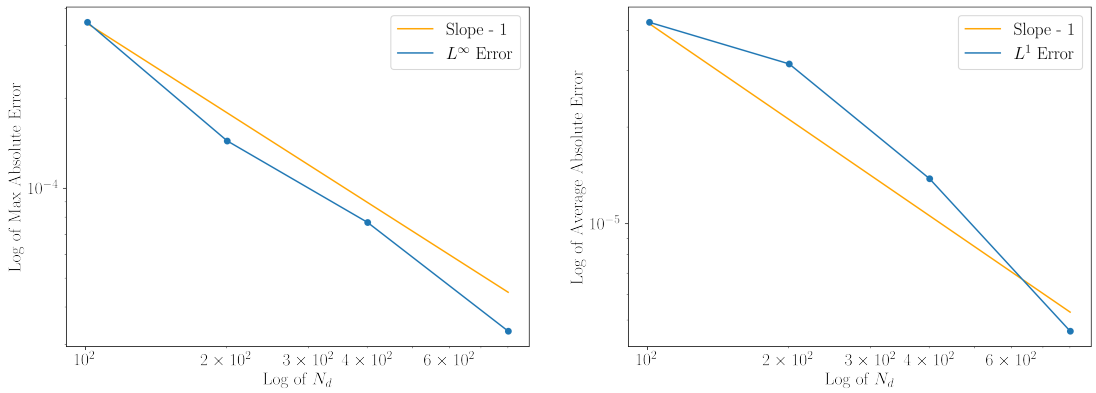
(4.24) satisfies the following formula for all $i, j < N_d + 1$ and all $k < N_t$,

$$U(\boldsymbol{z}_{ij}, t_k) = \min_{\boldsymbol{v} \in \mathcal{V}} \{\tau K(\boldsymbol{v}) + p(\boldsymbol{z}_{ij}, t_k) U(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}}, t_k + \tau) + (1 - p(\boldsymbol{z}_{ij}, t_k)) q(\tilde{\boldsymbol{z}}_{ij}^{v}, t_k)\}. \quad (4.32)$$

We solve backwards in time from $U(\boldsymbol{z}_{ij}, t_{N_t}) = W_{ij}$, and we recover $U(\tilde{\boldsymbol{z}}_{ij}^{\boldsymbol{v}}, t_k + \tau)$ via bilinear interpolation in x and y.

Time-Dependent HJB Convergence Studies

Just as was done for the stationary HJB PDE convergence study, we compute a "ground truth" solution $\overline{U}$ on a $1601 \times 1601 \times 9143$ spatio-temporal grid over the domain $[0,4] \times [0,4] \times [0,2]$ for the same parameters, camouflaging point location, and evader's stalk phase trajectory considered in Example 4.4.4. We determine both the infinity norm and the L^1 norm of the pointwise absolute error between $\overline{U}$ and the solution U_{N_d} of the HJB PDE computed on an $N_d+1 \times N_d+1 \times N_t+1$ grid for $(N_d, N_t) \in \{(100, 571), (200, 1142), (400, 2285), (800, 4572)\}$. Log-log plots of the absolute error in each norm are presented in Figure 4.6 and a table summarizing the convergence data is available in Table 4.4.



(A) (B)

Figure 4.6: Time-dependent stalk phase HJB PDE convergence study plots. (A) Log-log plot of the infinity norm of the absolute error when $M = 10, \tilde{\theta} = \pi/8$. (B) Log-log plot of the L^1 norm of the absolute error when $M = 10, \tilde{\theta} = \pi/8$.

As the grid is refined, the scheme shows evidence of convergence in L^∞ and in L^1. In L^∞, the convergence rate is slightly better than one (see Figure 4.6 (A) and the second and third columns of Table 4.4). In L^1, the rate becomes slightly better than first order after the slower decrease between $N_d = 100$ and $N_d = 200$ (see Figure 4.6 (B) and the fourth and fifth columns of Table 4.4).

Table 4.4: Time-Dependent HJB Convergence Study Data

Time-Dependent Conv. Study: $M = 10$, $\tilde{\theta} = \pi/8$				
N_d	L^∞ Error	$\ln\left(\frac{e_{k+1}}{e_k}\right)$	L^1 Error	$\ln\left(\frac{e_{k+1}}{e_k}\right)$
100	0.00036	–	4.24×10^{-5}	–
200	0.00014	1.319	3.14×10^{-5}	0.431
400	7.68×10^{-5}	0.906	1.38×10^{-5}	1.189
800	3.32×10^{-5}	1.209	4.62×10^{-6}	1.579

Using the Analytical Expression to Compute $\boldsymbol{v}_*$: To accelerate the computation, we can utilize the analytical expression for $\boldsymbol{v}_*$ given in Equation (4.26) with a suitable numerical gradient approximation in lieu of an optimization algorithm. The gradient $\nabla u = (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})$ is computed using an *upwind* finite difference discretization. The standard, first-order, finite difference operators used to compute the partial derivative in each direction are given by

$$D_x^+ U = \frac{U_{i+1,j}^{k+1} - U_{ij}^{k+1}}{\Delta x}, \quad D_x^- U = \frac{U_{ij}^{k+1} - U_{i-1,j}^{k+1}}{\Delta x}, \tag{4.33}$$

and

$$D_y^+ U = \frac{U_{i,j+1}^{k+1} - U_{ij}^{k+1}}{\Delta y}, \quad D_y^- U = \frac{U_{ij}^{k+1} - U_{i,j-1}^{k+1}}{\Delta y}, \tag{4.34}$$

where the i, j indices specify the spatial grid node, and $k+1$ refers to the timeslice from which the U values are obtained. To satisfy the upwinding condition, we select which operator to use in each direction to ensure that the characteristic is straddled by the computational stencil. That is, for the approximation of $\frac{\partial u}{\partial x}$, we select

$$D_x U = \begin{cases} 0, & U_{ij}^{k+1} \leq \min(U_{i+1,j}^{k+1}, U_{i-1,j}^{k+1}) \\ D_x^+ U, & U_{ij}^{k+1} > \min(U_{i+1,j}^{k+1}, U_{i-1,j}^{k+1}) = U_{i+1,j}^{k+1} \\ D_x^- U, & U_{ij}^{k+1} > \min(U_{i+1,j}^{k+1}, U_{i-1,j}^{k+1}) = U_{i-1,j}^{k+1}. \end{cases} \tag{4.35}$$

In the event that $U_{ij}^{k+1} > U_{i+1,j}^{k+1} = U_{i-1,j}^{k+1}$, the point $(\boldsymbol{z}_{ij}, t_k)$ lies on a shockline, and either D_x^+ or D_x^- is a valid choice. The operator used in the approximation of $\frac{\partial u}{\partial y}$ is chosen similarly, and the optimal speed is scaled to F_P when necessary.

Optimal Trajectory Tracing & MC Optimality: After computing U and $\boldsymbol{v}_*$ at each $(\boldsymbol{z}_{i,j}, t_k)$, we reconstruct an approximation to P's full optimal trajectory until E's capture starting from a specified (x, y) at $t = 0$. We utilize the results from the PDE solve and interpolate the optimal controls to reconstruct P's optimal velocity at each position along the trajectory. We use this to integrate P's stalk phase dynamics in steps of Δt, and we continue this process up until $t = T_s$. From T_s onward, we integrate P's direct chase phase dynamics along the analytically known optimal direction of motion $\boldsymbol{a}_*(T_s) = \frac{\boldsymbol{z}_E(T_s) - \boldsymbol{z}_P(T_s)}{|\boldsymbol{z}_E(T_s) - \boldsymbol{z}_P(T_s)|}$ until P captures E at $t = T_2$.

Since it is impossible to recognize perfect MC due to computational and physiological limitations, we determine the portions of P's optimal trajectory which are *approximately MC* (A-MC) by calculating θ using the formula given in Equation (4.1) at each point along the stalk phase trajectory. A-MC occurs when $\theta \leq 0.01$ radians, which is below the resolvable angle limit for many compound eye visual systems.

4.4 Numerical Examples

"Happy to be out, happy to not be seen."

"East River Swimmer" by Petite League

In all examples, E's food source is a flower located at $\boldsymbol{z}_\star = (2.4, 2.1)$, and the surroundings are homogeneous except for a boulder at $\boldsymbol{z}_\# = (1.5, 1.4)$. E's stalk phase trajectory is given by $\boldsymbol{f}(t) = [2.4 - 0.05(T_\star - t) + \cos(T_\star - t) - 1, 2.1 - 0.05(T_\star - t)^2 - \sin(T_\star - t)]^T$, with $T_\star = 2\,s$. E's maximum speed is $\approx 2.1\,m/s$, and in all plots, the evader's stalk phase trajectory is shown in magenta.

The parameters D, F_P, G_P, and G_E are set to their values posted in Table 4.1. We set the detection strength parameter S to 400, the baseline power $W \approx 0.00051$ as computed in Section 4.1, $\gamma = 0.025\,m$, and $\epsilon = 0.05\,m$. We impose a $401 \times 401 \times 2285$ grid over the domain $\overline{\Omega} = [0, 4] \times [0, 4] \times [0, 2]$, and the spatial spacings between grid nodes are computed according to the formulas in (4.28). We use the time spacing $\Delta t = 0.000875$ which satisfies the restriction given in (4.28), and we take $\tau = \Delta t$. The stationary phase Fast Sweeping routine terminates once the maximum of the absolute pointwise difference between the value function approximations at successive iterations falls below 1×10^{-6}.

Through our examples, we aim to illustrate how E's visual sensitivity to MC-deviation ($\sigma(\theta)$) and tolerance (M) for other organisms within visual range affects the optimality of MC tactics during the stalk phase. We set $B = 1/2$ in the definition of $\sigma(\theta)$ and numerically solve the multistage optimal control problem

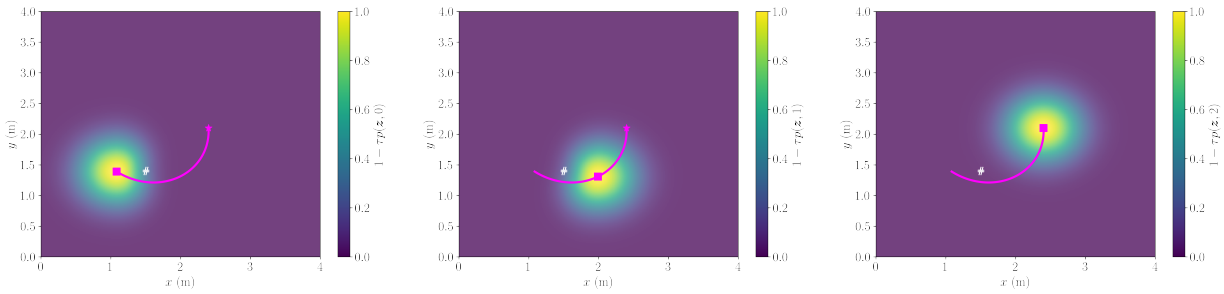
for various values of the threshold angle $\tilde{\theta}$ and proximity tolerance parameter M in (4.3). Snapshots at three different times of the threat detection probability (assuming that P and E remain motionless for a τ second period starting at time t) are displayed for each example.

To draw larger conclusions about the effects of E's visual sensitivity and tolerance on when it is optimal to engage in A-MC behavior, we also gather statistics on overall A-MC usage during the stalk phase flight while $\boldsymbol{z}_{\#}$ is visible to E. For each of the four scenarios considered, we trace the optimal trajectories starting from 200,000 unique points within a region of radius $D = 0.85\,m$, which encompasses the realm where E's visual detection abilities are strongest at $t = 0$, and we record the percentage of stalk phase flight time that P spends in A-MC for each set of initial conditions. We then generate a heatmap centered at $\boldsymbol{z}_E(0)$ where each point is colored according to the percentage of stalk phase flight time that P spends in A-MC starting from that location.

4.4.1 Ex. 1: Low Tol. ($M = 4$), High Threshold ($\tilde{\theta} = \frac{\pi}{2}$)

In our initial time-dependent example, E has low to moderate tolerance for other organisms in their visual range ($M = 4$), and the threshold angle is at a maximum ($\tilde{\theta} = \pi/2$). Snapshots of the threat detection probability within the next τ seconds at three different instances in time ($t = 0$, $t = 1$, and $t = T_\star$) are displayed in Figures 4.7 (A), (B), and (C) respectively. The threat detection probability is largely radially symmetric, with the exception of a *slight* indention of *lower* probability centered along $\boldsymbol{r}_{\#}$ (see Figures 4.7 (A) and (B)). Once E is far enough from $\boldsymbol{z}_{\#}$, only P and E's mutual distance affects the probability that E will consider P as a threat. This results in the radially symmetric threat detection probability

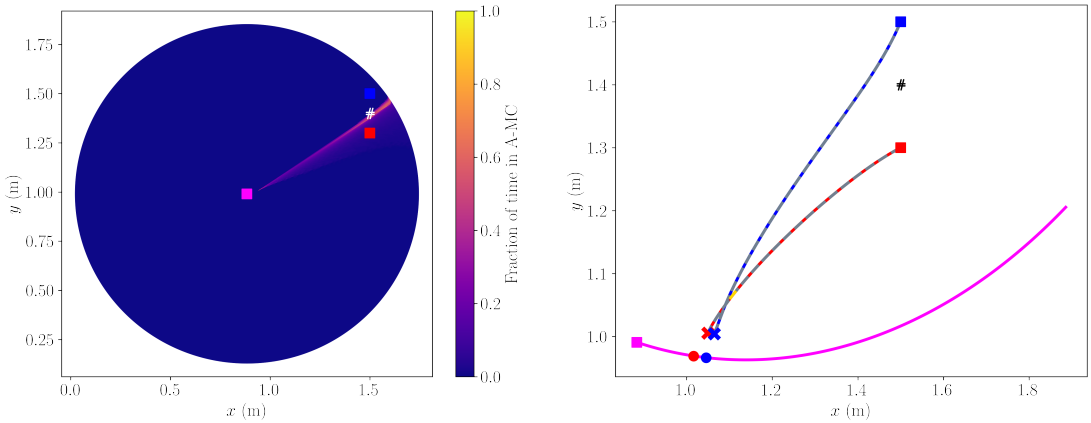
in Figure 4.7 (C).



(A) (B) (C)

Figure 4.7: Example 1: Probability that E spots P as a threat between t and $t + \tau$ if both are motionless at their respective locations during that time period when $t = 0$ (A), $t = 1$ (B), and $t = T_\star$ (C). E's current position along the trajectory is marked by the magenta square, and $\boldsymbol{z}_\#$ is shown by the white "#." The threat detection probability throughout E's flight is mostly radially symmetric, except for a small indention centered around $\boldsymbol{r}_\#(t)$ that creates a region of slightly lower threat detection probability.

From the A-MC percentage plot displayed in Figure 4.8 (A), it is clear that significant A-MC usage is only beneficial from starting locations which are nearly aligned with $\boldsymbol{r}_\#(0)$. This is reflected P's sample optimal trajectories starting from $(1.5, 1.3)$ and $(1.5, 1.5)$ in Figure 4.8 (B), as neither trajectory makes an effort to align with $\boldsymbol{r}_\#(t)$ and opts to head directly toward E instead. Assuming that E does not feel threatened earlier, the direct chase phase begins at $t = 0.135\,s$ and $t = 0.164\,s$ when P starts from $(1.5, 1.3)$ and $(1.5, 1.5)$ respectively. In both cases, P's trajectory coincides with the optimal trajectories computed for the case when *only* mutual distance affects the threat detection probability (i.e., $\sigma = 1$ throughout E's visual region) shown by the gray dashed lines in Figure 4.8 (B). This suggests that when E's MC-threshold angle $\tilde{\theta}$ is high and their tolerance is low, expending the energy to correct to an A-MC trajectory is not worth it, as E's visual system is less-attuned to the positional differences associated with angular



(A) (B)

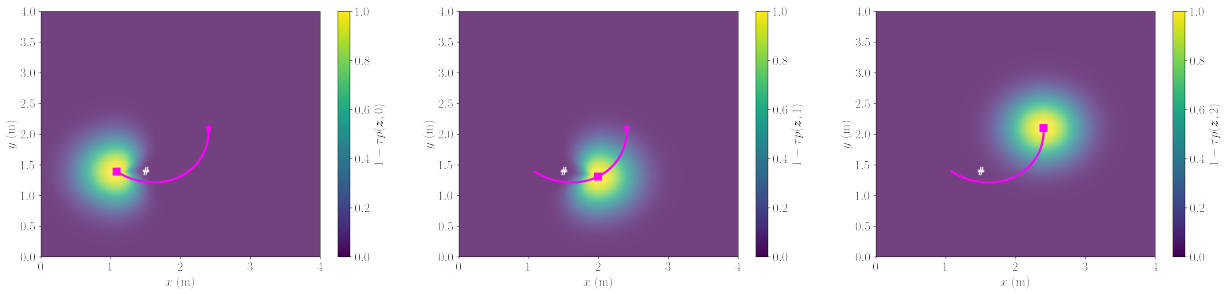
Figure 4.8: Example 1: Heatmap of the percentage of stalk phase flight time spent using A-MC from 200,000 initial positions for P (A). E's starting position is marked by the magenta square, and $\boldsymbol{z}_{\#}$ is shown by the white "#." Using A-MC is worthwhile only from starting positions very close to (or on) $\boldsymbol{r}_{\#}(0)$. Example trajectories starting from $\boldsymbol{z} = (1.5, 1.3)$ (marked by the red square in both panels) and $\boldsymbol{z} = (1.5, 1.5)$ (marked by the blue square in both panels) (B). The locations along E's trajectory at which the direct chase begins for each of P's trajectories are marked by a circle whose color corresponds to the associated pursuit trajectory. The optimal trajectories starting from $\boldsymbol{z} = (1.5, 1.3)$ and $\boldsymbol{z} = (1.5, 1.5)$ computed when A-MC is not relevant to E's threat detection probability are shown by the gray dashed lines.

deviations from the camouflaging point.

4.4.2 Ex. 2: Low Tol. ($M = 4$), Low Threshold ($\tilde{\theta} = \frac{\pi}{8}$)

Next, we consider a situation in which E's tolerance remains low, but their MC-threshold angle is significantly reduced to $\tilde{\theta} = \pi/8$. These parameters support the formation of a distinct sector of *low* threat detection probability within a period of τ seconds centered along $\boldsymbol{r}_{\#}(t)$ (Figure 4.9 (A) and (B)). This sector rotates counterclockwise as E moves along their trajectory but disappears once $\boldsymbol{z}_{\#}$ is far

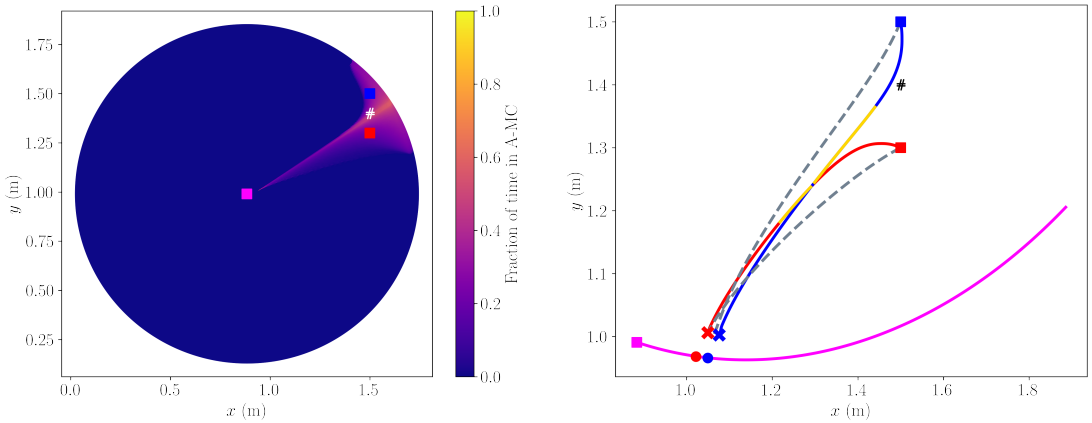
enough away from E (Figure 4.9 (C)).



(A) (B) (C)

Figure 4.9: Example 2: Probability that E spots P as a threat between t and $t + \tau$ if both are motionless at their respective locations during that time period when $t = 0$ (A), $t = 1$ (B), and $t = T_\star$ (C). E's current position along the trajectory is marked by the magenta square, and $\boldsymbol{z}_\#$ is shown by the white "#". With $\tilde{\theta} = \pi/8$, there is a distinct sector of lower threat detection probability aligned with $\boldsymbol{r}_\#(t)$ while $\boldsymbol{z}_\#$ is clear enough to E's visual system.

The A-MC percentage plot displayed in Figure 4.10 (A) shows that MC tactics are beneficial from a slightly wider sector of the considered starting locations within E's visual radius in comparison to the results shown in Figure 4.8 (A). Assuming E does not feel threatened by P until they are within ϵ meters, the duration of P's stalk phase flight starting from $(1.5, 1.3)$ and $(1.5, 1.5)$ is 0.14 seconds and 0.167 seconds respectively. From the lower starting point, P uses A-MC tactics for 0.0236 seconds (16.9% of their stalk phase flight time), while from the higher starting point, P engages in A-MC for 0.047 seconds (28.2% of their stalk phase flight time). These statistics and observations suggest that in low-tolerance settings, utilizing A-MC for at least a portion of the stalk phase trajectory might be worthwhile for P when their starting distance is far enough away from E, yet close enough to $\boldsymbol{z}_\#$ so that they can align themselves with $\boldsymbol{r}_\#(t)$ in the face of E's generally high spotting rate within the region where their visual capabilities are strongest.



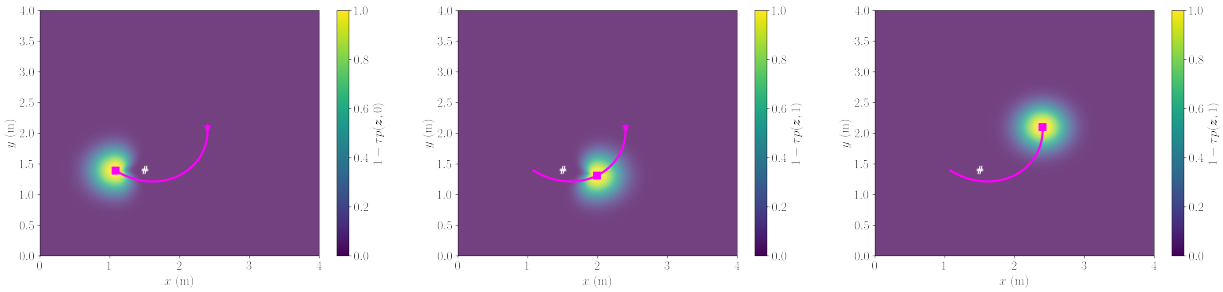
(A) (B)

Figure 4.10: Example 2: (A) Heatmap of the percentage of stalk phase flight time spent using A-MC from 200,000 initial positions for P. E's starting position is marked by the magenta square, and $\boldsymbol{z}_{\#}$ is shown by the white "#." Using A-MC is worthwhile starting from a wider sector. Initial conditions close to $\boldsymbol{r}_{\#}(0)$ and farthest from $\boldsymbol{z}_E(0)$ have the highest percentage of stalk-phase A-MC usage. Example trajectories starting from $\boldsymbol{z} = (1.5, 1.3)$ (marked by the red square in both panels) and $\boldsymbol{z} = (1.5, 1.5)$ (marked by the blue square in both panels) (B). The locations along E's trajectory at which the direct chase begins for each of P's trajectories are marked by a circle whose color corresponds to the associated pursuit trajectory. The optimal trajectories starting from $\boldsymbol{z} = (1.5, 1.3)$ and $\boldsymbol{z} = (1.5, 1.5)$ computed when A-MC is not relevant to E's threat detection probability are shown by the gray dashed lines. Starting from $(1.5, 1.3)$, P spends 16.9% of their stalk phase flight in A-MC, while starting from $(1.5, 1.5)$, P's A-MC usage percentage rises to 28.8%. The A-MC portions are highlighted in gold.

4.4.3 Ex. 3: High Tol. ($M = 10$), High Threshold ($\tilde{\theta} = \frac{\pi}{2}$)

We now transition to examining situations in which E has *high* tolerance for other organisms in view ($M = 10$). First, we consider an example where $M = 10$ and $\tilde{\theta} = \pi/2$. With such a high threshold angle and tolerance value, the probability that E will consider P a threat within the τ-second period is greatly reduced across

many locations farther than ϵ meters away from E (Figure 4.11 (C)). While $\boldsymbol{z}_{\#}$ is within the realm of E's strongest visual detection capabilities, the pointwise probability is *mostly* radially symmetric with the exception of a *slight* indention forming a region of lower probability centered along $\boldsymbol{r}_{\#}(t)$ (Figures 4.11 (A) and (B)).



(A) (B) (C)

Figure 4.11: Example 3: Probability that E spots P as a threat between t and $t+\tau$ if both are motionless at their respective locations during that time period when $t = 0$ (A), $t = 1$ (B), and $t = T_\star$ (C). E's current position along the trajectory is marked by the magenta square, and $\boldsymbol{z}_{\#}$ is shown by the white "#." With $M = 10$, the chance that P will be spotted as a threat is significantly reduced at many locations within the region where E's visual detection abilities are strongest and there is a slight indention of lower threat detection probability aligned with $\boldsymbol{r}_{\#}(t)$.

The heatmap in Figure 4.12 (A) reveals that despite E's high tolerance for other organisms within view, A-MC usage is worthwhile for P starting within a slightly wider region of positions near $\boldsymbol{z}_{\#}$ and close to $\boldsymbol{r}_{\#}(0)$ in comparison to the Example 1 high threshold results displayed in Figure 4.8 (A). The asymmetry and the somewhat annular region within which A-MC usage is slightly above 0% is likely due to properties of E's trajectory under these visual parameters. In the early stages of E's trip along this particular trajectory, P has the ability to move approximately four times faster than E. This property and the lower spotting rate

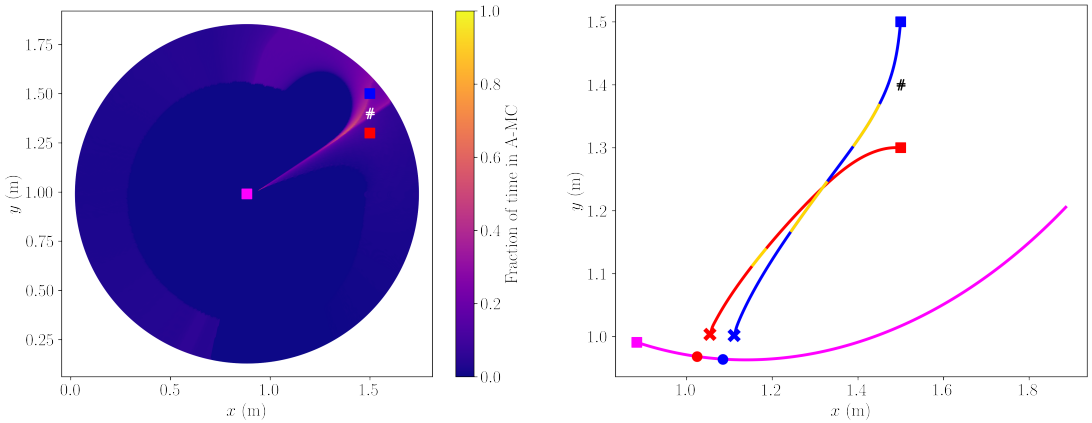
may render it feasible for P take a less-direct route which allows them to stay within less-risky regions, even from starting positions which are on the opposite side of E to the camouflaging location at $t = 0$.

The sample trajectories presented in Figure 4.12 (B) suggest that from both starting positions, P's A-MC usage is likely more coincidental than intentional. From $(1.5, 1.3)$, P spends 7.97% of their stalk phase flight time engaging in A-MC. Even though starting from $(1.5, 1.5)$, P spends approximately 26% in A-MC, the fact that its usage is not continuous over one time interval likely means that A-MC engagement from that point is also coincidental.

4.4.4 Ex. 4: High Tol. ($M = 10$), Low Threshold ($\tilde{\theta} = \frac{\pi}{8}$)

Lastly, we simulate a scenario in which $M = 10$, but E's threshold angle is much lower ($\tilde{\theta} = \pi/8$). That is, E is generally tolerant to other organisms within view, but E's visual system can resolve the angular deviations from background objects well. Here, narrowing the threshold angle makes the sector of lower threat detection more distinct, just as is the case in Example 2 where E also has $\tilde{\theta} = \pi/8$.

Figure 4.14 (A) shows that with high enough tolerance and low enough threshold angle, there is a *significant* region of initial conditions starting from which it is beneficial to utilize A-MC tactics during the stalk phase flight. This is supported by the sample trajectories presented in Figure 4.14 (B), as P starting from *both* $(1.5, 1.3)$ and $(1.5, 1.5)$ makes intentional adjustments to spend a significant portion of their stalk phase flight using A-MC. P reaches the chase region in $0.144\,s$ and $0.202\,s$ from $(1.5, 1.3)$ and $(1.5, 1.5)$ respectively and spends 0.055 seconds and 0.085 seconds in A-MC starting from those respective positions. That amounts to

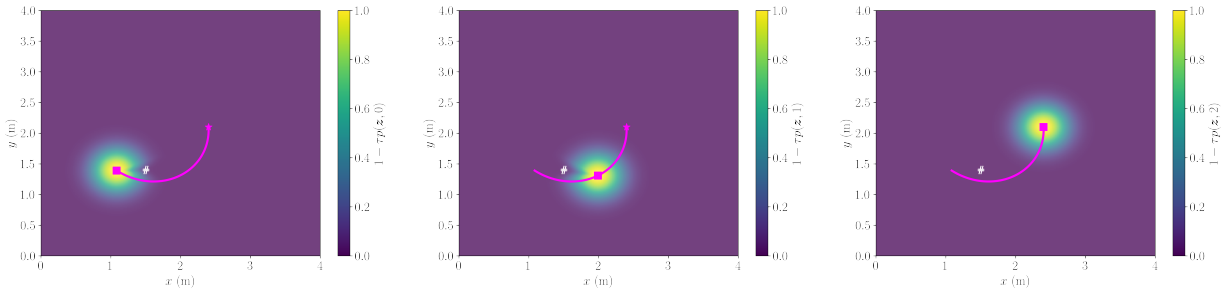


(A) (B)

Figure 4.12: Example 3: (A) Heatmap of the percentage of stalk phase flight time spent using A-MC from 200,000 initial positions for P. E's starting position is marked by the magenta square, and $\boldsymbol{z}_{\#}$ is shown by the white "#." Using A-MC is worthwhile along stalk phase trajectories starting from a small subset of positions on (or close to) $\boldsymbol{r}_{\#}(0)$ and from positions near the visual boundary, but close to $\boldsymbol{z}_{\#}$. (B) Example trajectories starting from $\boldsymbol{z} = (1.5, 1.3)$ (marked by the red square in both panels) and $\boldsymbol{z} = (1.5, 1.5)$ (marked by the blue square in both panels) (B). The locations along E's trajectory at which the direct chase begins for each of P's trajectories are marked by a circle whose color corresponds to the associated pursuit trajectory. Both trajectories get closer to $\boldsymbol{r}_{\#}(t)$, but A-MC usage is likely coincidental based on the low usage along the red trajectory (gold section after the halfway mark) and the discontinuous usage along the blue trajectory (gold sections).

38.79% of stalk phase flight in A-MC starting from $(1.5, 1.3)$ and 42.42% of stalk phase flight in A-MC starting from $(1.5, 1.5)$.

P's advantage in following an A-MC trajectory under the present conditions is starkly apparent when comparing the solid optimal trajectories in shown in Figure 4.14 (B) to the gray-dashed optimal trajectories for the isotropic-λ case (e.g., $M = 10$, $\tilde{\theta} = 0$). Following the gray-dashed trajectories, it takes P $0.135\,s$ and $0.164\,s$ to reach the chase region from $(1.5, 1.3)$ and $(1.5, 1.5)$ respectively. The



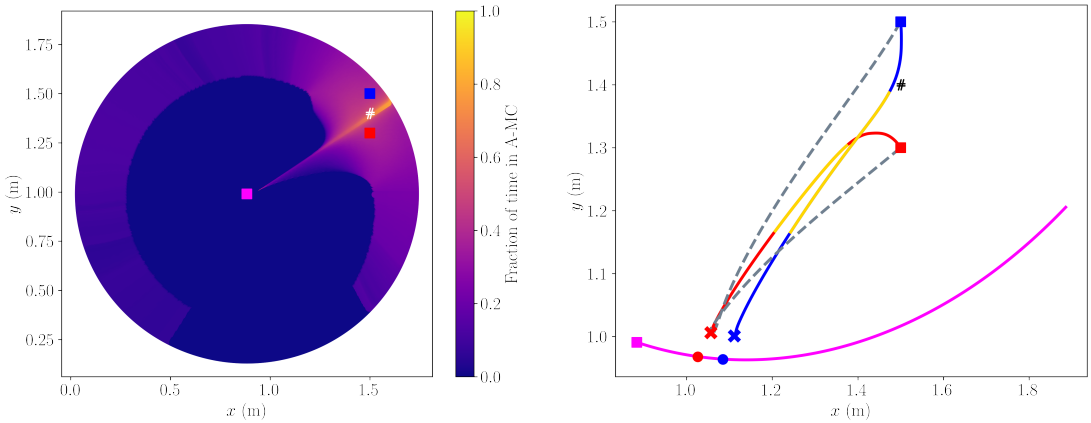
(A) (B) (C)

Figure 4.13: Example 4: Probability that E spots P as a threat between t and $t + \tau$ if both are motionless at their respective locations during that time period when $t = 0$ (A), $t = 1$ (B), and $t = T_\star$ (C). E's current position along the trajectory is marked by the magenta square, and $\boldsymbol{z}_{\#}$ is shown by the white "#." With E's lower threshold angle ($\theta = \pi/8$), there is now a distinct sector of lower threat detection probability centered along $\boldsymbol{r}_{\#}(t)$ for over half of E's flight to $\boldsymbol{z}_\star$.

stalk phase flight duration differences between the two suggests that following a *longer* path can be less risky when λ is not radially symmetric. By following the solid trajectories, P is hedging their bets. Since the longer trajectories spend a significant portion of time in a region of comparably less risk, P takes the improved chances of getting *closer* to E before E spots them and forces the start of the high-cost direct chase.

4.5 Conclusions

We presented a continuous-time dynamic programming framework to determine when it is optimal for an energy-optimizing pursuer to utilize motion camouflage amidst uncertainty in the evader's escape response. We illustrated our setup for the biological example of a hover fly pursuing another hover fly, and we showed how varying visual acuity and tolerance to P's presence in the evader's visual range



(A) (B)

Figure 4.14: Example 4: (A) Heatmap of the percentage of stalk phase flight time spent using A-MC from 200,000 initial positions for P. E's starting position is marked by the magenta square, and $\boldsymbol{z}_{\#}$ is shown by the white "#." Using A-MC is worthwhile from a wider range of starting positions around $\boldsymbol{z}_{\#}$ and along $\boldsymbol{r}_{\#}(0)$. (B) Two example trajectories starting from $\boldsymbol{z} = (1.5, 1.3)$ (marked by the red square in both panels) and $\boldsymbol{z} = (1.5, 1.5)$ (marked by the blue square in both panels). The locations along E's trajectory at which the direct chase begins for each of P's trajectories are marked by a circle whose color corresponds to the associated pursuit trajectory. The dashed lines correspond to the optimal trajectories when $\tilde{\theta} = 0$. A-MC usage is optimal for significant portions of both trajectories, and the A-MC segments are highlighted in gold.

can affect the stalk-phase optimality of MC strategies.

Our experiments suggest that MC tactics are most worthwhile in a distinct parameter regime where M is relatively large and $\tilde{\theta}$ is relatively low. Intentional motion camouflage usage seems to be beneficial for P when E's tolerance for other organisms in proximity is high enough, and E's visual system is strong enough to resolve positional differences associated with angular deviations from background objects. However, our experiments themselves are also limited in scope. None of the simulations shown consider changes in MC usage across different possible stalk

phase trajectories for E. Properties of E's trajectory (e.g., curvature, acceleration, average speed) may be largely responsible for the asymmetry in the region where A-MC usage is worthwhile, and as a next step, it will be necessary to investigate how these properties affect when and where MC is useful. Doing so will allow us to draw more informed and generalized conclusions about the benefits (or lack thereof) of MC tactics.

Despite our focus on hover fly interactions, we emphasize that this approach to analyzing motion camouflaging behaviors is applicable to a broader class of pursuer-evader interactions. One can readily modify the control, cost function, dynamics, and λ as appropriate to apply the framework to other biological or vehicular systems. Possible extensions include adding more food sources to introduce uncertainty in E's intended destination and introducing additional camouflaging points. We also hope that our approach may serve as a building block for optimization under uncertainty in more complex scenarios such as optimal pursuit in multi-pursuer / multi-evader systems [111], motion camouflage in mutual pursuit systems [73], and guiding bio-inspired robots in three spatial dimensions [87].

BIBLIOGRAPHY

[1] Kazi Iftekhar Ahmed. *Modeling Drivers' Acceleration and Lane Changing Behavior*. PhD thesis, Massachusetts Institute of Technology, 1999.

[2] RK Ahuja, TL Magnanti, and JB Orlin. *Network Flows*. Prentice Hall, Inc., Upper Saddle River, NJ, 1993.

[3] Ken Alton and Ian M Mitchell. Fast Marching Methods for Stationary Hamilton-Jacobi Equations with Axis-Aligned Anisotropy. *SIAM Journal on Numerical Analysis*, 47(1):363–385, 2009.

[4] Ken Alton and Ian M Mitchell. An Ordered Upwind Method with Precomputed Stencil and Monotone Node Acceptance for Solving Static Convex Hamilton-Jacobi Equations. *Journal of Scientific Computing*, 51(2):313–348, 2012.

[5] June Andrews and Alexander Vladimirsky. Deterministic control of randomly-terminated processes. *Interfaces and Free Boundaries*, 16(1):1–40, 2014.

[6] Stanley Bak, Joyce McLaughlin, and Daniel Renzi. Some Improvements for the Fast Sweeping Method. *SIAM Journal on Scientific Computing*, 32(5):2853–2874, 2010.

[7] Martino Bardi and Italo Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Birkhauser, 1997.

[8] Martino Bardi, Italo Capuzzo Dolcetta, et al. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, volume 12. Springer, 1997.

[9] Martino Bardi and Juan Pablo Maldonado López. A Dijkstra-type algorithm for dynamic games. *Dynamic Games and Applications*, 6(3):263–276, 2016.

[10] Jay Beeber. An explanation of Mats Jarlstrom's extended kinematic equation. *ITE Journal*, 90(3):34–38, 2020.

[11] R Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[12] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[13] S. Bennett. *Medium/Heavy Duty Truck Engines, Fuel & Computerized Management Systems*. Cengage Learning, 2016.

[14] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I and II. Athena Scientific, Boston, MA, 3rd edition, 2001.

[15] Dimitri P Bertsekas. A simple and fast label correcting algorithm for shortest paths. *Networks*, 23(8):703–709, 1993.

[16] Dimitri P Bertsekas. Robust shortest path planning and semicontractive dynamic programming. *Naval Research Logistics (NRL)*, 66(1):15–37, 2019.

[17] Dimitri P Bertsekas, Francesca Guerriero, and Roberto Musmanno. Parallel Asynchronous Label-Correcting Methods for Shortest Paths. *Journal of Optimization Theory and Applications*, 88(2):297–320, 1996.

[18] Dimitri P Bertsekas and John N Tsitsiklis. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[19] Folkmar Bornemann and Christian Rasch. Finite-element discretization of static Hamilton-Jacobi equations based on a local variational principle. *Computing and Visualization in Science*, 9:57–69, 2006.

[20] Michelle Boué and Paul Dupuis. Markov Chain Approximations for Deterministic Control Problems with Affine Dynamics and Quadratic Cost in the Control. *SIAM J. Numer. Anal*, 36(3):667–695, 1999.

[21] Vadim A Butakov and Petros Ioannou. Personalized driver assistance for signalized intersections using v2i communication. *IEEE Trans. Intell. Transp. Syst.*, 17(7):1910–1919, 2016.

[22] M.K. Cameron. Finding the quasipotential for nongradient SDEs. *Physica D: Nonlinear Phenomena*, 241(18):1532–1550, 2012.

[23] Nicole E Carey, Jason J Ford, and Javaan S Chahl. Biologically inspired guidance for motion camouflage. In *2004 5th Asian Control Conf. (IEEE Cat. No. 04EX904)*, volume 3, pages 1793–1799. IEEE, 2004.

[24] A. Chacon and A. Vladimirsky. Fast two-scale methods for eikonal equations. *SIAM Journal on Scientific Computing*, 34(2):A547–A578, 2012.

[25] A. Chacon and A. Vladimirsky. A parallel two-scale method for eikonal equations. *SIAM Journal on Scientific Computing*, 37(1):A156–A180, 2015.

[26] J Chang et al. Estimated benefits of connected vehicle applications: dynamic mobility applications, AERIS, V2I safety, and road weather management applications. Technical Report FHWA-JPO-15-255, US Dept. of Transportation, 2015.

[27] Yunxia Chen, Qing Zhao, Vikram Krishnamurthy, and Dejan Djonin. Transmission Scheduling for Optimizing Sensor Network Lifetime: A Stochastic Shortest Path Approach. *IEEE Transactions on Signal Processing*, 55(5):2294–2309, 2007.

[28] Zachary Clawson, Adam Chacon, and Alexander Vladimirsky. Causal Domain Restriction for Eikonal Equations. *SIAM Journal on Scientific Computing*, 36(5):A2478–A2505, 2014.

[29] Thomas S Collett and MF Land. Visual control of flight behaviour in the hoverfly syritta pipiens l. *J. Compar. Physiol.*, 99:1–66, 1975.

[30] Thomas S Collett and MF Land. How hoverflies compute interception courses. *J. of Comparative Physiol.*, 125:191–204, 1978.

[31] Michael G Crandall and Pierre-Louis Lions. Viscosity solutions of hamilton-jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

[32] Innes C Cuthill, Samuel R Matchette, and Nicholas E Scott-Samuel. Camouflage in a dynamic world. *Curr. Opin. in Behav. Sci.*, 30:109–115, 2019.

[33] Peng Dai and Judy Goldsmith. Topological Value Iteration Algorithm for Markov Decision Processes. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 1860–1865, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[34] Peng Dai, Daniel Weld, et al. Focused Topological Value Iteration. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 19, pages 82–89, 2009.

[35] François Desquilbet, Jian Cao, Paul Cupillard, Ludovic Métivier, and Jean-Marie Mirebeau. Single Pass Computation of First Seismic Wave Travel

Time in Three Dimensional Heterogeneous Media with General Anisotropy. *Journal of Scientific Computing*, 89:1–37, 2021.

[36] Robert B Dial. Algorithm 360: Shortest-path forest with topological ordering [h]. *Communications of the ACM*, 12(11):632–633, 1969.

[37] Edsger W Dijkstra et al. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[38] Paolo Domenici, Jonathan M Blagburn, and Jonathan P Bacon. Animal escapology ii: escape trajectory case studies. *Journal of Experimental biology*, 214(15):2474–2494, 2011.

[39] Seifeldeen Eteifa, Hesham A Rakha, and Hoda Eldardiry. Predicting Coordinated Actuated Traffic Signal Change Times using Long Short-Term Memory Neural Networks. *Transp. Res. Rec.*, page 03611981211000748, 2021.

[40] Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Society, 2022.

[41] E. A. Fainberg. On controlled finite state Markov processes with compact control sets. *Theory of Probability & Its Applications*, 20(4):856–862, 1976.

[42] Maurizio Falcone and Roberto Ferretti. *Semi-Lagrangian approximation schemes for linear and Hamilton—Jacobi equations*. SIAM, 2013.

[43] Maurizio Falcone and Roberto Ferretti. *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations*, volume 133. SIAM, 2014.

[44] EUGENE A Feinberg. A markov decision model of a search process. *Contemporary Mathematics*, 125:87–96, 1992.

[45] GM Fitch et al. Analysis of Lane-Change Crashes and Near-Crashes. *US Department of Transportation, National Highway Traffic Safety Administration*, 2009.

[46] Lester Randolph Ford. Network flow theory. *Rand Corporation Paper, Santa Monica, 1956*, 1956.

[47] Mallory E Gaspard. Optimality of motion camouflage under escape uncertainty. *IEEE Control Systems Letters*, 2024.

[48] Mallory E Gaspard and Alexander Vladimirsky. Optimal driving under traffic signal uncertainty. *IFAC-PapersOnLine*, 55(16):25–31, 2022.

[49] Mallory E Gaspard and Alexander Vladimirsky. Monotone causality in opportunistically stochastic shortest path problems. *arXiv preprint arXiv:2310.14121*, 2023.

[50] Marissa Gee, Nicolas Gonzalez-Granda, Sunay Joshi, Nagaprasad Rudrapatna, Anne Somalwar, Stephen P Ellner, and Alexander Vladimirsky. Navigating the landscape of fear. *bioRxiv*, pages 2024–08, 2024.

[51] Marissa Gee and Alexander Vladimirsky. Optimal path-planning with random breakdowns. *IEEE Control Systems Letters*, 6:1658–1663, 2021.

[52] Peter G Gipps. A Model for the Structure of Lane-Changing Decisions. *Transportation Research Part B: Methodological*, 20(5):403–414, 1986.

[53] F Glover, R Glover, and D Klingman. The threshold shortest path algorithm. *Networks*, 14(1):12–37, 1986.

[54] Andrew V Goldberg and Chris Harrelson. Computing the shortest path: A search meets graph theory. In *SODA*, volume 5, pages 156–165, 2005.

[55] R. Gonzales and E. Rofman. On deterministic Control Problems: an Approximate Procedure for the Optimal Cost, I, the Stationary Problem. *SIAM J. Control Optim.*, 23(2):242–266, 1985.

[56] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[57] Russell Herman. A first course in differential equations for scientists and engineers. *Self-Published Book*, 2017.

[58] Sungkweon Hong, Sang Uk Lee, Xin Huang, Majid Khonji, Rashid Alyassi, and Brian C Williams. An Anytime Algorithm for Chance Constrained Stochastic Shortest Path Problems and its Application to Aircraft Routing. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 475–481. IEEE, 2021.

[59] Ronald A Howard. *Dynamic programming and Markov processes.* John Wiley, 1960.

[60] Kuang Huang, Xuan Di, Qiang Du, and Xi Chen. A game-theoretic framework for autonomous vehicles velocity control: Bridging microscopic differential games and macroscopic mean field games. *arXiv preprint arXiv:1903.06053*, 2019.

[61] LI Jianqing, LI Chaoyong, and Yonghe Zhang. Guidance strategy of motion camouflage for spacecraft pursuit-evasion game. *Chin. J. Aeronaut.*, 2023.

[62] Mitchell Jones, Maximilian Haas-Heger, and Jur van den Berg. Lane-Level Route Planning for Autonomous Vehicles. In *Algorithmic Foundations of Robotics XV: Proceedings of the Fifteenth Workshop on the Algorithmic Foundations of Robotics*, pages 312–327. Springer, 2022.

[63] Mitchell Jones, Maximilian Haas-Heger, and Jur van den Berg. Lane-level route planning for autonomous vehicles. *The International Journal of Robotics Research*, 43(9):1425–1440, 2024.

[64] Eric W Justh and PS Krishnaprasad. Steering laws for motion camouflage. *Proc. Roy. Soc. A: Math., Phys., Eng. Sci.*, 462(2076):3629–3643, 2006.

[65] S. Kim. An O(N) Level Set Method for Eikonal Equations. *SIAM J. Sci. Comp.*, 22:2178–2193, 2001.

[66] Ron Kimmel. Fast Marching Methods on Triangulated Domains. *Proc. Nat. Acad. Sci.*, 95:8431–8435, 1998.

[67] H. J. Kushner and P. G. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Academic Press, New York, 1992.

[68] Grant Mahler and Ardalan Vahidi. Reducing idling at red lights based on probabilistic prediction of traffic signal timings. In *2012 American Control Conference (ACC)*, pages 6557–6562. IEEE, 2012.

[69] Grant Mahler and Ardalan Vahidi. An optimal velocity-planning scheme for vehicle energy efficiency through probabilistic prediction of traffic-signal timing. *IEEE Trans. Intell. Transp. Syst.*, 15(6):2516–2523, 2014.

[70] John A Michon. A Critical View of Driver Behavior Models: What Do We Know, What Should We Do? In *Human Behavior and Traffic Safety*, pages 485–524. Springer, 1985.

[71] Jean-Marie Mirebeau. Anisotropic Fast-Marching on Cartesian Grids Using

Lattice Basis Reduction. *SIAM Journal on Numerical Analysis*, 52(4):1573–1599, 2014.

[72] Jean-Marie Mirebeau. Efficient fast marching with Finsler metrics. *Numer. math.*, 126(3):515–557, 2014.

[73] Matteo Mischiati and PS Krishnaprasad. The dynamics of mutual motion camouflage. *Syst. & Control Lett.*, 61(9):894–903, 2012.

[74] Akiko Mizutani, Javaan S Chahl, and Mandyam V Srinivasan. Motion camouflage in dragonflies. *Nature*, 423(6940):604–604, 2003.

[75] Jim Motavalli. It's time for smart traffic lights. Autoweek, Oct. 30, 2020.

[76] Xiao Lei Mou, Yan Peng Liu, and Mao Sun. Wing motion measurement and aerodynamics of hovering true hoverflies. *Journal of Experimental Biology*, 214(17):2832–2844, 2011.

[77] Stanley Osher and Ronald P Fedkiw. Level Set Methods: An Overview and Some Recent Results. *Journal of Computational physics*, 169(2):463–502, 2001.

[78] Ulla Pape. Implementation and efficiency of Moore-algorithms for the shortest route problem. *Mathematical programming*, 7:212–222, 1974.

[79] Stephen D Patek and Dimitri P Bertsekas. Stochastic shortest path games. *SIAM Journal on Control and Optimization*, 37(3):804–824, 1999.

[80] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer Science & Business Media, 1996.

[81] L. C. Polymenakos, D. P. Bertsekas, and J. N. Tsitsiklis. Implementation of efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 43(2):278–283, 1998.

[82] Dongping Qi, Adam Dhillon, and Alexander Vladimirsky. Optimality and robustness in path-planning under initial uncertainty. *preprint:* ***https://arxiv.org/abs/2106.11405***, 2021.

[83] Dongping Qi, Adam Dhillon, and Alexander Vladimirsky. Optimality and robustness in path-planning under initial uncertainty. *Dyn. Games and Appl.*, 2024.

[84] Iñaki Rañó. An optimal control strategy for two-dimensional motion camouflage with non-holonimic constraints. *Biol. Cybern.*, 106:261–270, 2012.

[85] Inaki Rañó. Direct collocation for two dimensional motion camouflage with non-holonomic, velocity and acceleration constraints. In *2013 IEEE Int. Conf. Robotics and Biomimetics (ROBIO)*, pages 109–114. IEEE, 2013.

[86] N Sariff and Norlida Buniyamin. An overview of autonomous mobile robot path planning algorithms. In *2006 4th student conference on research and development*, pages 183–188. IEEE, 2006.

[87] Andrey V Savkin and Hailong Huang. Bioinspired bearing only motion camouflage uav guidance for covert video surveillance of a moving target. *IEEE Systems Journal*, 15(4):5379–5382, 2020.

[88] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press, 1996.

[89] James A Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[90] James A Sethian. Fast Marching Methods. *SIAM review*, 41(2):199–235, 1999.

[91] James A Sethian and Alexander Vladimirsky. Fast methods for the eikonal and related Hamilton–Jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences*, 97(11):5699–5703, 2000.

[92] James A Sethian and Alexander Vladimirsky. Ordered upwind methods for static Hamilton-Jacobi equations. *Proceedings of the National Academy of Sciences*, 98(20):11069–11074, 2001.

[93] James A Sethian and Alexander Vladimirsky. Ordered Upwind Methods for Static Hamilton–Jacobi Equations: Theory and Algorithms. *SIAM Journal on Numerical Analysis*, 41(1):325–363, 2003.

[94] Mandyam V Srinivasan and Matthew Davey. Strategies for active camouflage of motion. *Proc. Roy. Soc. B: Biol. Sci.*, 259(1354):19–25, 1995.

[95] Reuben Strydom and Mandyam V Srinivasan. Uas stealth: Target pursuit

at constant distance using a bio-inspired motion camouflage guidance law. *Bioinspiration & biomimetics*, 12(5):055002, 2017.

[96] Chao Sun, Jacopo Guanetti, Francesco Borrelli, and Scott J Moura. Optimal eco-driving control of connected and autonomous vehicles through signalized intersections. *IEEE Internet Things J.*, 7(5):3759–3773, 2020.

[97] Chao Sun, Xinwei Shen, and Scott Moura. Robust optimal eco-driving control with uncertain traffic signal timing. In *2018 Annual American Control Conference (ACC)*, pages 5548–5553. IEEE, 2018.

[98] Min Tan, Shichang Zhang, Martin Stevens, Daiqin Li, and Eunice J Tan. Antipredator defences in motion: animals reduce predation risks by concealing or misleading motion signals. *Biol. Rev.*, 99(3):778–796, 2024.

[99] Malin Thyselius, Paloma T Gonzalez-Bellido, Trevor J Wardill, and Karin Nordström. Visual approach computation in feeding hoverflies. *Journal of Experimental Biology*, 221(10):jeb177162, 2018.

[100] Malin Thyselius and Karin Nordström. Hoverfly locomotor activity is resilient to external influence and intrinsic factors. *Journal of Comparative Physiology A*, 202(1):45–54, 2016.

[101] JN Tsitsiklis. Efficient Algorithms for Globally Optimal Trajectories. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 2, pages 1368–1373. IEEE, 1994.

[102] John N Tsitsiklis. Efficient Algorithms for Globally Optimal Trajectories. *IEEE transactions on Automatic Control*, 40(9):1528–1538, 1995.

[103] Panagiotis Typaldos, Ioanna Kalogianni, Kyriakos Simon Mountakis, Ioannis Papamichail, and Markos Papageorgiou. Vehicle trajectory specification in presence of traffic lights with known or uncertain switching times. *Trans. Res. Rec.*, 2674(8):53–66, 2020.

[104] JAEE Van Nunen. A Set of Successive Approximation Methods for Discounted Markovian Decision Problems. *Zeitschrift fuer operations research*, 20:203–208, 1976.

[105] Alexander Vladimirsky. Label-Setting Methods for Multimode Stochastic Shortest Path Problems on Graphs. *Mathematics of Operations Research*, 33(4):821–838, 2008.

[106] D Weihs and Paul W Webb. Optimal avoidance and evasion tactics in predator-prey interactions. *J. Theor. Biol.*, 106(2):189–206, 1984.

[107] Yunjun Xu. Subspace optimal control and motion camouflage. In *AIAA Guidance, Navigation and Control Conf. and Exhibit*, page 6301, 2008.

[108] Zelda B Zabinsky, Robert L Smith, J Fred McDonald, H Edwin Romeijn, and David E Kaufman. Improving hit-and-run for global optimization. *J. Global Optim.*, 3:171–192, 1993.

[109] Hongkai Zhao. A fast sweeping method for eikonal equations. *Mathematics of computation*, 74(250):603–627, 2005.

[110] Shuaidong Zhao and Kuilin Zhang. Online predictive connected and automated eco-driving on signalized arterials considering traffic control devices and road geometry constraints under uncertain traffic conditions. *Transportation research part B: methodological*, 145:80–117, 2021.

[111] Yinglu Zhou, Yinya Li, Andong Sheng, Guoqing Qi, and Jinliang Cong. Optimal control strategies and target selection in multi-pursuer multi-evader differential games. *Neurocomputing*, page 127701, 2024.