The Dissertation Committee for Carlin Liao
certifies that this is the approved version of the following dissertation:

Modular autonomous intersection management simulation for stochastic and priority auction paradigms

Committee:

Stephen D. Boyles, Supervisor

Christian Claudel

Krishna Kumar

Peter Stone

Modular autonomous intersection management simulation for stochastic and priority auction paradigms

by

Carlin Liao, B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2021

For 奶奶.

Acknowledgments

I'd like to extend my gratitude to Dr. Boyles and the many other members of our humble research group past and present for their guidance throughout my doctoral studies. In particular, thank you to Priyadarshan Patil, Cesar Yahia, and William Alexander for their mentorship and camaraderie as senior doctoral students (even if most of them are graduating after me), as well as Rishabh Thakkar and Karthik Velayutham for their contributions to bringing my AIM conceptualization into reality.

Thanks to my friends Aurik Sarker and Grace Yao for helping me work through solutions to particular sticking points in the development of my AIM framework, as well as to Kevin Tran, Andre Ponce, and Alyssa So for their recommendations for and experience in software development.

And thank you to my family for supporting me as I worked through the most challenging parts of my research just as we together worked through the most challenging times of the century.

Modular autonomous intersection management simulation for stochastic and priority auction paradigms

Publication No. ____________________

Carlin Liao, Ph.D.
The University of Texas at Austin, 2021

Supervisor: Stephen D. Boyles

Automated intersections, when combined with the proliferation of autonomous vehicles (AVs), allow for more precise and innovative methods to control traffic at these integral choke points in the road system. In this dissertation, I develop a refined, modular framework for autonomous intersection management (AIM) simulation and implement it as a software library with robust documentation and testing to support present and future research in this field. Demonstrating this framework's efficacy, I apply it to study two topic areas in the AIM space: stochastic movement and priority auctions.

Stochastic AIM is introduced as an extension of traditional AIM that permits probabilistic reservations of space and time in an intersection. Its use case is motivated by the integration of human-driven vehicles into AIM using augmented reality guidance to behave more accurately to AV movement, while still making some stochastic deviations from AV-identical trajectories.

These deviations are quantified using experimental data from human drivers in a driving simulator merged into a stochastic vehicle movement model. Experimental results suggest that, with this paradigm, AIM can decrease delay significantly, even at low AV penetration levels (less than 20%).

Finally, I conceptualize intersection priority auctions into the newly developed AIM framework as itself a modular framework that supports the dispatch of multiple vehicles simultaneously from either separate lanes or a single lane without relying on preset signal phases. This auction framework further supports three payment formulas for the winner of the priority auction: first-price, second-price, or a novel externality payment mechanism. Using experiments implemented in the novel AIM simulator, my results demonstrate significant reduction in value-weighted delay using the multiple dispatch configuration and novel payment mechanism compared to other configurations, with the novel formula incentivizing truthful reporting of valuations more than its alternatives.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 What is an automated intersection?

Automated intersections, sometimes known as "autonomous" or "smart" intersections, are computerized traffic intersections that can communicate with vehicles in real time.

Autonomous (or automated) intersection management (AIM) is the process of controlling these intersections, taking advantage of their ability to process and respond to requests from vehicles to supplement or replace existing stop signs or singalized intersections, which may account for up to 295 million hours of delay in the United States alone [13]. First introduced by Kurt Dresner and Peter Stone in 2004, AIM has been the next step in a rich history of intersection control optimizations [20, 47, 48]. In virtual experiments, it's been shown to reduce delay by orders of magnitude compared to present-day traffic signals by sharing information between the intersection and connected, autonomous vehicles, allowing AIM to sequence movements together through the intersection with minimal clearance, even if they'd be incompatible in a standard signalized intersection [15]. The tight tolerances connected and automated vehicles are capable of allow AIM to exhibit vehicle flow comparable to interlocking, flowing streams at high speed.

AIM (or at least the original variant pioneered by Dresner and Stone) achieves this by using reservation-based controls, where vehicles approaching the intersection transmit their intended path to the intersection manager, requesting permission to enter. Should the intersection find no conflict between the requested trajectory and any already confirmed, AIM accepts the vehicle's request and saves its reservation of the time and space in the reservation conflict area to memory, ready to check it against the next vehicle's request. This process allows most vehicles to proceed through the intersection without stopping or even slowing down, leading to the the improvements against signalized control schemes where a majority of approaching vehicles need to brake to a stop and wait for their phase of the turn signal.

AIM and its promise of full two-way communication between the control mechanism and its participants enable the use and study of novel approaches to traffic management, but we must first address its greatest limitation: human drivers.

1.2 Accommodating human drivers

AIM promises substantial reductions in delay at intersections, but with one major caveat: these gains only manifest when participating vehicles are mostly or totally automated [14]. Although this fully automated future is on the horizon, there's still a long time before it comes to pass, requiring us to either mothball the AIM concept for several decades or adjust it to better accommodate human drivers [33]. In the latter case, human drivers are often included in AIM simulations by conceptually separating them from automated vehicles (AVs), barring them from participating in the tightly-sequenced reservation framework and only allowing them to move during traffic signal phases that AVs need not abide by [38].

The difficulty of incorporating human-driven vehicles into the AIM system comes from the assumption that they're unable to follow the same instructions as are issued to automated vehicles, and that they're incapable of communication with the intersection computer anyway. But both are negotiable: although we'll never be able to assume that human drivers can follow directions as precisely as a computerized vehicle, they can be taught to mimic them so long as we allow for some error, and the latter we may be able to ameliorate.

Presently, many drivers already have access to guidance through intersections in the form of navigation assistance on their smartphones or the vehicle itself. A natural extension of this would be to use that connectivity to communicate instructions from the driver and their vehicle to the intersection and back. While a present-day mobile device or onboard computer might not be capable of full level 5 autonomous driving, using an advanced user interface like a heads-up display (HUD) or augmented reality (AR) eyewear, they can provide trajectory and acceleration guidance to a human driver. This support can help drivers approach the precision necessary to participate in AIM like an AV with a larger margin of error.

Then again, AIM's performance also deteriorates dramatically as buffer size increases. AVs can reasonably be expected to use close to all of the space and time it reserves with its small buffer, but a human-driven vehicle and its imprecision, even with trajectory guidance, will carve out a much larger spread, much of which it won't personally use despite making it more difficult for other vehicles to pass through the intersection.

Enter stochastic reservations: instead of making spacetime reservations all-or-nothing, as is the case in standard AIM frameworks, stochastic AIM allows for partial reservation of spacetime units, so long as the probability that multiple vehicles use the same unit (a.k.a. a crash) is lower than some preset threshold derived from a user-set tolerable crash incidence rate. This allows us to reclaim at least some of the efficiency lost to the larger reservations required by human-driven, guided-trajectory vehicles while still ensuring low-to-zero probability of collisions depending on the threshold set.

1.3 A note on collisions

Understandably, transportation agencies and government officials in charge of funding research into AIM and potentially implementing it in the real world have been reluctant to support research into stochastic vehicle behavior because it implies a nonzero possibility of a collision, even if those probabilities are deep into the tails of their distributions. Instead, agencies implicitly prefer frameworks that are specifically engineered so that crashes are impossible in simulation. However, the probability of a collision, even when engineered to be impossible, is never zero simply due to the nature of an intersection being the crossing of two opposing streams of traffic. Current AIM models that deny the possibility of a collision simply aren't being honest about this potential outcome. With stochastic AIM, we'll now have the ability to not only simulate and examine crashes, but also quantify the probability that they happen so we can more accurately represent a potential future with AVs and human drivers cooperating under AIM.

1.4 Exploring alternative measures of priority

The basic concept of connected, computerized intersections allows for interesting and novel approaches to traffic management. No longer limited by lack of dialogue between the control mechanism and its participants, AIM makes possible priority schemas other than those imposed by the physical limitation of vehicles blocking other vehicles from entering. Intersections can now take into account participants' *value* of time instead of treating every participant's time as equal.

Let's motivate this use case with the example shown in Figure 1.1.

Suppose vehicle A, which has just reached this intersection, has an urgent trip to make. Perhaps they're rushing to the hospital, or late for their Ph.D. thesis defense. Either way, they have a strong desire to leave first and are willing to pay for it. Under present-day traffic control schemes, this wouldn't be possible as it would either have to wait for its light to turn green or all four earlier arrivals move in the case of a stop sign.

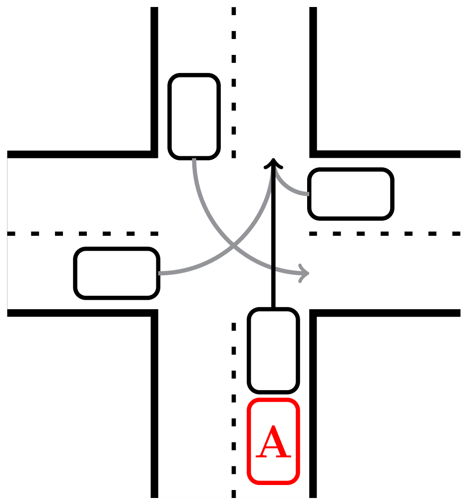

Figure 1.1: Several vehicles at an intersection, with their desired trajectories marked.

If the intersection controller were value-aware, they would be able to receive a signal and payment from vehicle A and prioritize the progress of that vehicle and the one before it, getting it through the intersection as fast as physically possible without ramping over the vehicle in front of it or driving into oncoming traffic. Although this might seem far-fetched, we already have a very limited form of value-aware intersections if you think of A as an emergency vehicle with an effectively unlimited budget!

That edge case aside, this motivating scenario is complicated when the other vehicles in the intersection have their own nonzero valuations as well. So long as they're able to efficiently communicate them as smart intersections with AVs do, a robust value-aware priority mechanism would be able to fully distinguish between their different values. Doing so would preclude the use of tolls, which can only make a binary distinction between vehicles willing to pay and those that aren't, leaving unbreakable ties within the two categories.

Enter auctions. Unlike fixed or even dynamic-value tolls, auctions can be fully discriminatory because participants are reporting their valuations themselves in the form of bids, making them the number one solution for value-aware AIM with 100% connected vehicles. That said, there's no one way to run an auction, and the simplest mechanisms have many non-obvious drawbacks. That makes the implementation of the intersection priority auction an open question, and worth exploring in this dissertation.

1.4.1 A note on equity

Changing intersections to consume value instead of simply allowing free usage does raise significant ethical concerns. This would allow wealthy drivers and passengers to pay their way through intersections using only a small fraction of their total wealth, while forcing low income drivers to incur major delays because they can't allocate as much money to pay their way through auctions. To some extent, this further incentivize carpooling as a method of travel with low environmental impact and more wealth than each member had individually, but transportation agencies and the general public may regard intersection auctions with real money to be unpalatable due to this inequality (although some surveys dispute this [5]).

One solution is for auctions to avoid using dollars, but instead consume an auction-specific, non-transferable currency issued by the state to individual drivers that refreshes periodically. This currency would be completely removed from a driver's personal wealth, allowing it to be equitable for all citizens. Granted, this would require completely new infrastructure and general mindset with how cities approach traffic, but so would implementing intersection priority auctions in the first place.

That said, the intent of this dissertation topic is not necessarily to propose a fully workable and realizable mechanism for converting intersections to value prioritization instead of arrival time. My intent is only to refine and extend prior research on intersection priority auctions to better inform (but not dictate) the likely long path to what smart intersections will look like in the future. In doing so, this allows me to further explore the novel application of auction theory to this unique problem and generate fresh ideas that can be used even outside of traffic research.

1.5 A preview of this text

Now that the principles of this dissertation text have been sufficiently motivated, I'll summarize what you, the reader, can expect in the following chapters.

Chapter 2 will detail the framework by which I define AIM, as well as the implementation of the framework as the simulator that I use to run my experiments.

Chapter 3 covers stochastic vehicle movement and stochastic reservations (real and simulated) under the AIM framework as well as the experiments, real-world (to some extent) and digital, that validate this new framework extension.

Chapter 4 goes into the history of the research on intersection auctions before laying out three auction mechanisms, the last of which is newly developed for this dissertation, and the experiments conducted to compare them.

Chapter 5 closes by summarizing the key contributions of this dissertation and detailing directions for future work.

Now let's dive in.

Chapter 2

AIM Simulation

In this chapter I'll detail and defend the framework and key assumptions of AIM as I define it for the purposes of this dissertation. Furthermore, I will describe the implementation of a new AIM simulation software, argue for its *raison d'etre* compared to its predecessors, and explain how it will serve experiments made for this dissertation and beyond.

2.1 Framework

2.1.1 Vehicles

I'll begin by defining the vehicles and drivers participating in AIM, which are treated as equivalent.

Vehicles are defined as rectangles with length l and width w that can vary across individual vehicles. They spawn on a set approach lane with a set movement they intend to make through the intersection (or an intended destination endpoint, in the case of multiple intersection experiments). Their position $\vec{x}$, heading θ, speed v, and acceleration a will change in real time as they progress through the intersection.

Although vehicles can physically have variable maximum acceleration and deceleration capabilities, they are required to be able to achieve a minimum acceleration rate $a_{\min}$ and minimum braking rate $b_{\min}$ (defined to be negative) set by the roads and intersections in order to participate. We assume all vehicles in the experiment are able to achieve these rates. Furthermore, both roads and intersections require that the vehicles use only $a_{\min}$ and $b_{\min}$ rates of acceleration and braking in the experiment, which all vehicles are able to do precisely on roads and do *to the best of their ability* in the intersection.

Vehicles also have as characteristics means and standard deviations that characterize their lateral (μ_τ, σ_τ) and throttle (μ_h, σ_h) tracking abilities. Their lateral tracking score characterizes their mean and likely breadth of deviation perpendicular to the provided trajectories through the intersection at the midpoint of the trajectory. The throttle tracking score does the same for their ability to accelerate precisely so that the vehicle can exit the intersection at the time it's instructed to. Automated vehicles are assumed to be perfectly precise and have all their μ and σ values set to 0; only human-driven, AR-assisted vehicles will have non-zero lateral and throttle tracking scores.

For experiments requiring valuations (i.e., auctions), each individual vehicle has a fixed value of time ν, but this ν can vary across vehicles. Vehicles are assumed to be wholly self-interested, behaving greedily moment-to-moment without regard for prior or future interactions.

As for their behavior, vehicles are assumed to always be accelerating to the road or intersection's speed limit $v_{\max}$, unless:

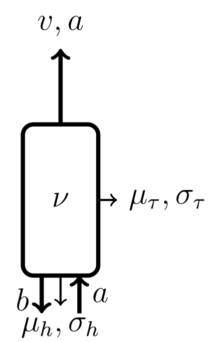

Figure 2.1: A vehicle

1. the vehicle needs to brake to maintain its buffer with the preceding vehicle,
2. the vehicle needs to brake to stop before the intersection line of an intersection it does not have permission to enter, or
3. the vehicle is already at $v_{\max}$ and must neither brake nor accelerate to maintain speed.

Braking and accelerating will be set to precisely $a_{\min}$ and $b_{\min}$ on roads for all vehicles, but they can be varied according to its throttle score when any part of the vehicle is an intersection.

2.1.2 Trajectories

Trajectories are a continuous curve segment between two points $f(p)$. Progress along the curve is defined proportionally by $p \in [0, 1]$, and each point on the curve must return both a physical 2D position in real coordinates $\vec{x}$

and a direction ("heading") θ so vehicles along the trajectory can be mapped to both a location and a direction of travel.

2.1.3 Lanes

Lanes are the basic building block of this AIM framework. A lane is defined by a trajectory and holds an ordered list of vehicles currently traversing itself. Vehicles are divided into three sections that are treated as individual points: front, center, and rear, with the center tracking the reference position of the vehicle and front and rear tracking the front and rear edges of the vehicle by length scaled up by a buffer factor of β identical for all vehicles in the simulation. (As an example, if $\beta = .1$, "front" would be .6 vehicle lengths ahead of the center of the vehicle, and vice versa for "rear"). The progress of each section along the lane is tracked individually but updated using the same vehicle v and a for all three sections.

As vehicles transition between key parts of the AIM simulation, e.g., from a road into the next intersection, they may occupy more than one lane at a time, but each section of the vehicle may only occupy one facility at any given moment.

2.1.4 Roads

Roads are groups of parallel lanes with identical width and length that connect intersections to each other, vehicle spawners, and removers (more on these in the next section). Although they're parallel, these lanes' start and

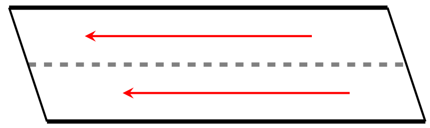

Figure 2.2: Example road

endpoints can be offset from each other at an angle as in Figure 2.2.

Roads are divided into three regions with borders defined at certain progressions across the road:

1. the entrance region (i.e., entering from the last spawner or intersection)
2. the lane-changing region
3. the approach region (i.e., approaching the next intersection or remover)

The entrance region prevents collisions during vehicle transitions by dictating per lane whether the road can accept new vehicles in time from the object upstream from it and checking if there's a vehicle currently present in itself.

The upstream object is either a vehicle spawner or an intersection. If a spawner, it would prevent a spawned vehicle from entering the road (simulating backed-up traffic). If an intersection, that same presence would prevent the intersection from allowing vehicles to secure a reservation ending in this lane until the entrance region of that road lane is cleared. Both of these cases make it vital for the entrance region (and thus each road) to be substantially longer than the longest vehicle permitted in the experiment.

As the name might indicate, the lane-changing region follows the approach region and allows for vehicles to switch road lanes before approaching the next intersection. Lane-changing has API hooks in the implementation of this AIM framework (to be discussed later), but will not be supported as it's not considered in scope for this dissertation, so I won't discuss it further here.

Finally, the approach region is the last road section ahead of the intersection, and could be considered part of the intersection. Lane-changing is forbidden in this section, and only vehicles in the approach region may communicate to the intersection to try and secure a reservation.

2.1.5 Vehicle spawners

Vehicle spawners decide when to create new vehicles, what characteristics they should have, and where to place them into the simulation on road the spawner is attached to. Vehicle spawners are provided at initialization with an expected rate of vehicle spawns λ and a pool of vehicle characteristics to draw from randomly including length, destination/intended movement through the intersection, and so on. Vehicles are spawned according to a Poisson process such that the probability of spawning in each timestep is equivalent to λ. Vehicles may still spawn even if there is no room for them on the road the spawner is connected to; when this is the case, spawned vehicles are held in a virtual queue (simulating backed-up traffic) to be spawned in order on the first timestep with available space. Priority is dictated by how many timesteps back the vehicle spawned, and only one vehicle may enter the road per timestep.

2.1.6 Vehicle removers

As the name implies, vehicle removers remove vehicles from the simulation and log their characteristics and traversal time for later study.

2.1.7 Intersection

An intersection connects a set of incoming road lanes with a set of outgoing road lanes, using and managing the area between the incoming and outgoing lanes that we call the "conflict area". Vehicles enter from their incoming lane and must cross the conflict area to reach their desired outgoing lane.

Incoming and outgoing lanes are assumed to be static across their entire length, e.g., lanes don't split or combine in the intersection approach or exiting an intersection. This makes left and right turn pockets ineligible for AIM modeling under this framework unless the intersection's approach area is defined only starting at the beginning of the shortest turn pocket.

Incoming road lanes are connected to outgoing lanes by intersection lanes, which are predefined trajectories through the conflict area. Incoming lanes may be connected to multiple outgoing lanes, and vice versa, (e.g., right turns and through movements from the same lane), but not all incoming lanes are connected to every outgoing lane. Exactly which ones are connected is an intersection parameter.

To control the movement through the conflict area, each intersection

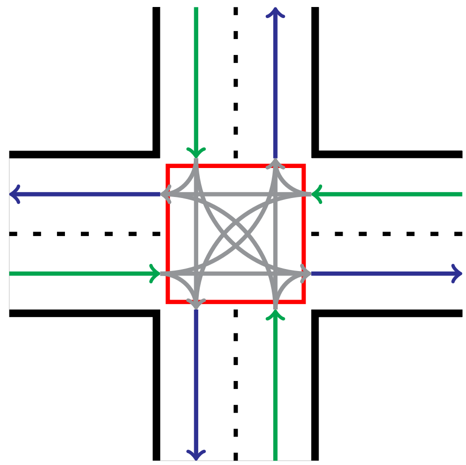

Figure 2.3: An example intersection

has its own *tiling* and *manager*. The tiling divides the conflict area into discrete spacetime units that can be reserved in part or in full, and the manager controls which vehicles will be able to attain reservations through the intersection. More on these below:

2.1.7.1 Tilings

The tiling dictates how movements through the intersection are interpreted into discrete timespace tiles that can individually be reserved by one or more vehicles. Tilings also handle the storage and updating of this reservation data.

Tilings hold layers of tiles, with each layer mapping to a future timestep and each tile mapping to an area or point in in the intersection. Each tile tracks the reservation(s) confirmed on it. Proposed reservations, or "requests", are checked against each tile; if any one tile finds the proposed trajectory incompatible with those already confirmed on it, the entire request is rejected as incompatible.

The request trajectory checking procedure that translates a proposed vehicle movement into tiles used is worth elaborating on. When a reservation request is submitted to the tiling for checking, it starts a nested simulation from the time of proposing vehicle's soonest entrance time from road to intersection at its projected velocity. The entrance time and velocity are knowable and deterministic because vehicle behavior, as defined in this framework, is itself deterministic as described in Section 2.1.1. (See A for the derivation of the soonest entrance/exit time and velocity (or, at least, the interesting parts).

Starting from the soonest exit (i.e., the exit of the front section of the vehicle), a copy of the requesting vehicle or vehicles is placed with its front bumper just entering the edge of the intersection. From here, the tiling runs the nested simulation of the vehicle's path through the intersection lane, marking any and all tiles that the vehicle uses according to its physical dimensions, heading, and buffer size. (Both stochastic reservations and auctions involve additional calculations for accommodating probabilistic tile usage and grouped requests, respectively. These will be detailed in their own Chapters 3 and 4.)

The tiling adds one time unit of temporal padding to the reservation at the beginning of the movement, where the vehicle enters the intersection, in order to accommodate rounding errors in the soonest exit calculation. The tiling also adds a variable time length buffer of tiles when and where the vehicle is scheduled to exit the intersection in order to prevent collisions from happening just outside of the intersection where it no longer has visibility, based on Diana Toader's patch to a Dresner et al. AIM variant [41]. So long as any tile the tiling associates with a request rejects the request as incompatible with its confirmed reservations (if any), the entire request is discarded as invalid. Combined with the requirement that vehicles aren't allowed to change lanes or wait for the next intersection in the entrance region of a road, the post-movement buffer ensures that there will be sufficient spacing between the last vehicle that exited the intersection and this one.

If none of the tiles used by the reservation request are disqualified, the request is eligible for approval by the intersection manager, but it need not do so. More on this in Chapter 4.

The tiling's record of confirmed reservations only applies to future steps of the simulation. As time elapses in the simulation, the tiling deletes the layer of tiles associated with the timestep that's just passed (if there is one) to avoid a memory leak. New layers of the tiling are added as vehicles make requests, so the tiling could be very deep if a faraway vehicle makes an exceptionally slow request, or as shallow as 0 if there are no vehicles in the simulation.

Presently, the only tiling used in this dissertation is the square tiling, which divides the conflict area into grid squares of equal size determined by an input parameter. The grids are extended to cover the entire area of the intersection as defined by its minimum and maximum x and y values, regardless of if the conflict area is rectangular or oblong.

Alternative tilings could include hexagonal tiles, or tiles placed only at or around points of conflict between intersection trajectories instead of covering the entire area.

2.1.7.2 The manager

The manager is the logical component of the intersection, essentially acting as the automation in AIM. To that end, you could call this the *policy* the intersection implements to decide which vehicles can request and secure movements through the intersection.

Policies implemented for this dissertation include:

Stop signs. All vehicles must come to a complete stop before entering the intersection. Only a single vehicle is allowed in the conflict area at a times. Which vehicle receiving priority through the intersection is decided based on when they arrived.

First come first served (FCFS). As vehicles approach the intersection, they call forward to the intersection with the soonest time and associated velocity at which they can reach the intersection, along with their desired movement. The intersection simulates their movement, checking if it's incompatible with already confirmed reservations. Vehicles can make requests as soon as they enter the approach region of the incoming lane, and can continue to do so continuously until it receives a reservation (thus the FCFS).

Auctions. These will be detailed in Chapter 4.

2.1.7.3 The reservation process

The reservation process works as follows:

1. The manager determines which lanes are eligible to request a reservation. Note that the manager queries *lanes* instead of *vehicles* because leading vehicles in each lane securing a reservation is a necessary prerequisite for all following vehicles to secure their own reservations.

2. The manager polls eligible lanes for their reservation requests and submits them to the tiling for validation. These reservation requests come from the first vehicle (or vehicles, depending on the policy) in the lane that do not yet have permission to enter the intersection.

3. The tiling does a nested simulation of each request to check if they're in conflict with already confirmed reservations (or each other, again depending on the policy). See Section 2.1.7.1 for more information.

4. The manager chooses some subset of compatible requests and confirms them with the tiling, which issues the vehicles permission to enter the intersection.

5. Depending on the policy, the process either ends here and waits for a certain condition before running again (e.g., the next timestep for FCFS or when the conflict area clears for auctions, more on this later) or continues to loop back to the beginning until no lanes are eligible.

There are certain classes of managers and policies that work a little differently from the process described above should they be implemented (e.g., traffic signals), but the general pattern is very similar.

2.2 Introducing NAAIMS

Not Another Autonomous Intersection Management Simulator, or NAAIMS, is a modular library for automated intersection management simulation implementing the framework described in the prior section, so named due to the propensity for every researcher studying in this field to implement their own simulator. This section details the implementation of the discrete timestep simulation loop, and contrasts NAAIMS with both its predecessor, Dresner's AIM [14, 15], and alternative optimization-based AIM simulators to justify its creation.

NAAIMS is open-source, licensed under GPL-3.0, and freely available for cloning and modification on GitHub at `https://github.com/spartalab/naaims`.

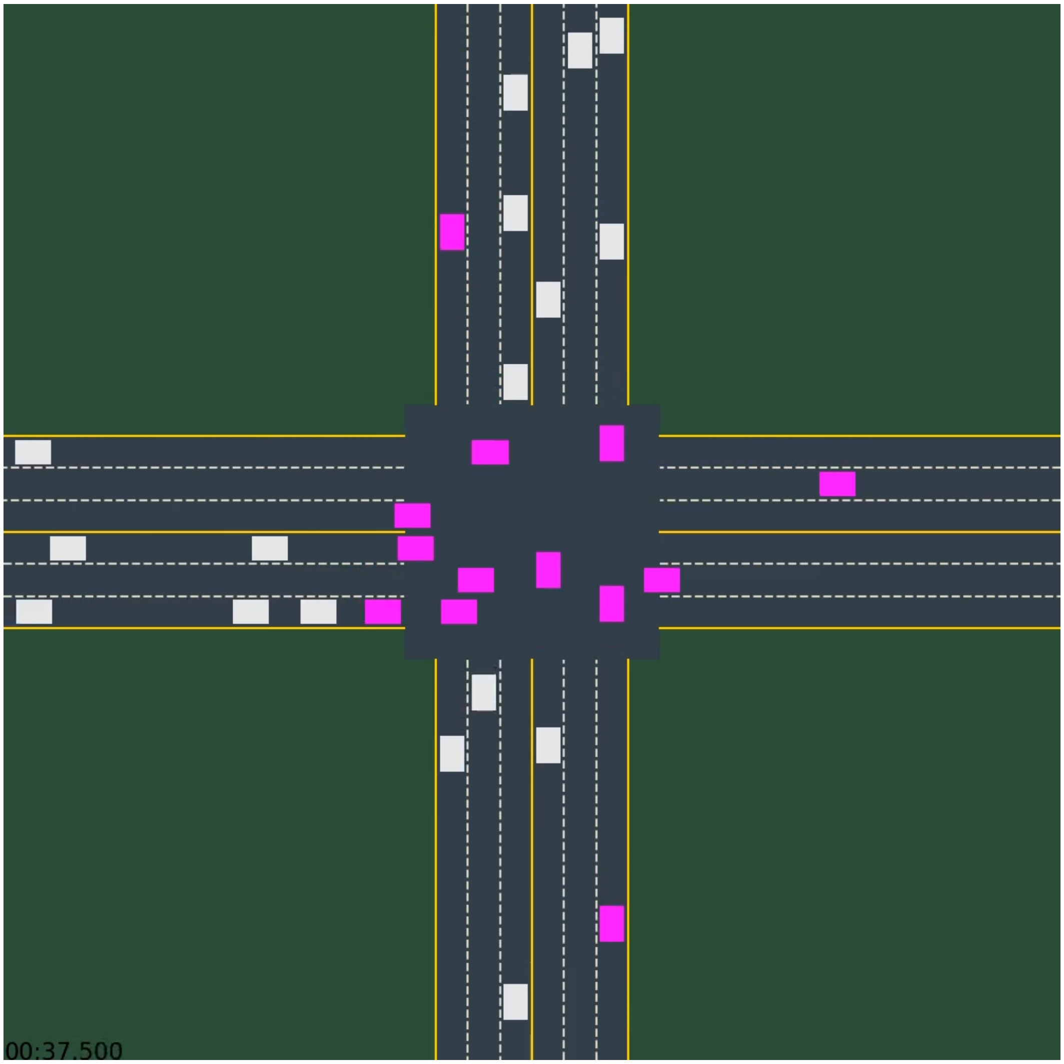

Figure 2.4: A screenshot from a NAAIMS instance

2.2.1 Simulation loop

Following the initiation of all road features as described in the previous section, NAAIMS iterates through timesteps until a preset end time. By default timesteps are set to 60 per simulated second, but this can be reduced for more efficiency.

Each timestep of the simulation proceeds as follows:

- Update vehicle acceleration, then speed
- Update vehicle positions
- Transfer exiting vehicle sections between key objects (roads, intersections, spawners, removers)
- Update intersection logic and schedule new reservations

2.2.2 Differences from Dresner's AIM simulator

NAAIMS most directly inherits from Dresner's AIM simulator and its inheritors designed by Stone, Au, Sharon, and many others [3, 14, 15, 38], and shares the same general approach to discrete time simulation of automated intersections. If a reader is familiar with any one of Dresner's AIM or its descendants like AIM4 and H-AIM, most of their knowledge of their operation, strengths, and weaknesses will apply similarly to NAAIMS. Because of this, it's worthwhile for us to focus specifically on where NAAIMS differs from Dresner's AIM.

The first and most obvious distinction is that NAAIMS is a completely new implementation written in Python 3 with the goal of being a modern, approachable, and extendable discrete time AIM simulation platform that will be easily usable for research even outside of that of the author. To this end, there is a strong focus on the development of documentation and tests alongside development of the base simulation software.

Furthermore, there are a few significant differences in the way NAAIMS and the AIM simulation framework it implements set up the problem and handles the actual process of reservation requests and vehicle movement.

Intersections have perfect knowledge of all vehicle characteristics, in particular, their position and speed at all times. Consequently, the intersection need only test reservations for the first vehicle in a lane instead of every vehicle. This is in contrast to Dresner's AIM which asks for vehicles to self-report their best guess time at which they'll reach the intersection from their current lane. These self-reported values may be faulty due to drivers' incomplete knowledge of what vehicles lie ahead of them and their future plans, which the intersection has access to in the form of reservations.

Given the present day proliferation of high-quality cameras and object identification algorithms, the assumption of vehicle connectivity, sensing technology, as well as the pace of progress up until the time when automated intersections will reach physical implementation, I'm confident that this is not as much of a reach in assumptions as it might have been in 2006. And, in general, Dresner's framework targeted potential real-world application of AIM, e.g., in this experiment where a physical vehicle was controlled by a simulated automated intersection manager [35], requiring the simulator to be generous when estimating sensing error. In contrast, NAAIMS takes approaches this question from more of a traffic engineering approach compared to an electrical engineering one, still adhering to physical behavior as much as possible but abstracting away sensing errors and non-compliant behavior where convenient to simplify the pursuit of examining new traffic control mechanisms.

Reservation requests only check maximum acceleration trajectories when predicting a vehicle's potential trajectory through an intersection. Dresner's AIM also considered fixed velocity trajectories, which NAAIMS does not, and we can imagine an ideal simulator checking every possible acceleration profile to make a request possible. This is a reasonable assumption to the author, as it enforces an easily defensible preference for spending as little time in the intersection conflict area as possible over potentially getting a reservation more quickly.

General approach. Dresner's AIM targeted real-world applications of AIM, e.g., with this mixed reality simulation of what autonomous intersections, but the implementation of the NAAIMS framework is also somewhat different from Dresner's.

Vehicle movement is modeled by the lane it's in, not individual driver agents. Vehicles no longer need to "think for themselves" anymore, centralizing computation to the road and intersection and reducing the number of calculations necessary. This is primarily an implementation decision for smoother simulation with less computational resources, as in principle of course vehicles are controlled by themselves and not the intersection, but in general the framework implemented by NAAIMS assumes a greater degree of direct control over vehicles by the intersection than Dresner's AIM does, reflecting a prioritization of simulating ideal traffic compared to the original's focus on encouraging compliance through mechanism design.

Abstraction of intersection management policies. While ad hoc modifications to Dresner's AIM made comparing intersection policies outside of traffic signals and FCFS possible, NAAIMS and its framework have this modularization of intersection management policies built in from the beginning, making it easier than ever to compare different policies like auctions to FCFS and traffic signals in an apples-to-apples way, without caveats for different implementation frameworks and assumptions.

Intersection geometry isn't limited to just right angles. Road and intersection geometry have been abstracted into connected incoming and outgoing lanes instead of being rigidly locked to a grid, allowing us to potentially simulate real road networks outside of midtown Manhattan.

Abstraction of intersection conflict area discretization. Although both Dresner's AIM and the current version of NAAIMS only implement square grid tiles in the conflict area, NAAIMS can easily swap between different discretization methods such as conflict points, hexagonal tiles, and so on.

There are also several features NAAIMS adds to the feature list to support the studies relevant to this dissertation

- stochastic tile reservations
- auction management policies (basic, Carlino, sequenced, etc.)
- multi-vehicle reservations (for sequenced auctions)

These will be elaborated on in the following chapters.

Finally, there are several features at least partially supported by the API but that will not be completed for this dissertation as they are not in scope for its topics of study, including:

- Trajectories other than Bezier curves (e.g., circular arcs, more efficient straight lines)
- Lane-changing on roads between intersections
- Multi-intersection simulations and support structures (e.g., routing)
- Non-square intersection tilings
- Traffic signal intersection manager

2.2.3 Direct simulation vs. optimization functions

To the best of my knowledge, Dresner's AIM, its derivatives, and NAAIMS are the only discrete time AIM simulators in use in the automated intersection study community [9]. Most research in this field utilizes optimization frameworks to study traffic through intersections, setting up systems of equations and constraints reflecting certain interpretations of what intersection traffic is like.

Optimization methods need to balance the demand of optimizing over a large enough time horizon to be useful, while still constraining the problem enough to be computationally tractable in real time [27, 28]. That said, discrete time methods can also be criticized for having weaknesses as well. The standard policy used by discrete time AIM, FCFS, can be demonstrably sub-optimal compared to traffic signals [26] and in general under heavy traffic [50].

Broadening the umbrella, related research characterizes these intersection control mechanisms as either "signalized", using methods reminiscent of modern-day traffic signals but with more reactivity [16, 17, 19, 23, 30, 49], or "signal-free", which analyze the trajectories of vehicles directly without the abstraction of signal phases [10]. "Signal-free" includes both rule and reservation-based control schemes like AIM FCFS as well as optimization methods [2], in one case even drawing from similar problems in the study of aircraft routing [28]. These control mechanisms aren't limited to a single intersection, as one example, Lin et al., have extended their concept of rhythm control, which blends cyclic phases with responsive optimization, to networks of intersections [31].

I believe that there is room for discrete time simulations in this field of research. Broadly, AIM experiments based on optimization functions, even time-windowed ones, rely on much stronger assumptions on intersection configurations and vehicle behavior than discrete time simulators do. Even slight changes in the numerical setup dramatically changing their results [50], which is significant given the quantity of analyses formulating their problems as mixed integer or something similar [27, 28, 32].

In contrast, the basics of discrete time AIM simulation require even fewer assumptions than what I've implemented in this dissertation. Provided the user is alright with observing crashes, even our strict vehicle behavior model can be loosened to reflect broader traffic patterns since there isn't a rigidly defined optimization problem that has to have a valid solution. Given how little concrete information we have on what real automated intersections will look like, I think that a policy-based framework like this discrete time simulator makes more sense when it comes to exploring the possibilities available to AIM, and it requires fewer real-time computational resources than constantly updating optimization time windows (generally speaking; many optimization studies of AIM do intentionally design their optimization approach to be quick enough for real time).

I hope that NAAIMS's intended accessibility will make it easier and more common for frameworks to be built on it and allow for more clear comparisons between the variety of approaches we've seen in this field of research.

2.2.4 A note on acceleration parameters

As defined by the framework, accelerations in NAAIMS are effectively ternary: vehicles are either accelerating at a fixed value a, decelerating at a fixed value b, or holding constant with 0 acceleration.

Exactly what these a and b values should be is a potential point of contention. Low values ought to be more tolerable to the median passenger but there is a distinct negative relationship between more comfortable acceleration values and intersection throughput.

Dresner's AIM used as default a value of around $4g$, which meant very short stopping distances but relatively high g-forces; this is unlikely to be comfortable for most passengers and would probably be higher than if a global acceleration were enforced.

There are surprisingly few papers available on comfortable accelerations, and what little there is emphasizes that it's not necessarily the acceleration that causes passenger discomfort but *jerk*, or the rate of change in acceleration. That said, modeling the fourth derivative of distance is more detail than I intend to explore with NAAIMS, so I'll rely on a survey written for the Department of Transportation in 1976 [21]. In it, it explains that $0.3g$ or $0.266g$ on the high end of comfortable if all passengers are seated properly and facing forward or backward and the source in the survey. If we accommodate passengers sitting sideways as well, this might be as low as $0.16g$.

By default NAAIMS sets a to 3 m/s^2 or about $0.3g$ and b to -2.6 m/s^2

or $0.266g$ based on the above findings. Of course, changing these values requires only a one-line alteration to the configuration file, so exploring the impact of higher or lower acceleration values is simple.

2.2.5 Default experimental setup

Chapters 3 and 2 rely on experiments in NAAIMS to illustrate the efficacy (or lack thereof) of different concepts they introduce. This subsection describes those experiments' default parameters; any deviations from these parameters are noted in the experiment descriptions in that section.

The intersection configuration used is a symmetric 4-way, 3-lane intersection with approaches $50m$ long. Lanes are specified to be 4 meters wide, and the square tiling is also set to 4 meter tiles to align with the lane widths for a total of 8 tiles in each direction.

By default, vehicles are automated (with no stochastic movement) and sized to be 3 m wide and 4.5 m long. In-lane, vehicles are treated as being 0.1x longer than they actually are to avoid collisions, and when calculating the tiles they are incident with, their vehicle outlines are extended outward by 0.1 m in each direction. As stated in the prior subsection, the global acceleration rate is set to 3 m/s^2, the braking rate to -2.6 m/s^2, and the speed limit to 15 m/s.

Vehicles spawn at a rate of 10 vehicles per minute per approach, equivalent to an hourly demand of 2,600 vehicles across the entire intersection. Turning rates are identical across all four approaches, with 10% each turning right and left and the remaining 80% making through movements.

2.3 Future work

Given that NAAIMS is purpose-built as a modular simulation package, it'd be strange if there weren't several extensions to the model already in the planning stages. Broadly, they can be divided into two categories: features to bring forward from Dresner et al.'s AIM4 implementation, and new tools to support novel studies into AIM's potential like the modules implemented for Chapters 3 and 4.

Although nearly all of the basic functionality of Dresner et al.'s AIM4 is in NAAIMS, the main missing feature is support for multiple connected intersections. This was an intentional decision in order to focus the analysis on comparing intersection control mechanisms; allowing multiple intersection requires a suite of supporting features like routing or lane-changing between intersections that could draw scarce development time away from implementing alternative control schemes. That said, illustrating how managers work in a network of intersections would give us a richer picture of AIM's efficacy. Support for multiple intersections is already planned for within the NAAIMS framework and simulation classes, pending only the implementation of the routing and lane-changing support features before we can start running experiments.

Another feature planned for inheritance from the Dresner line of simulators is traffic signal and FCFS-signal managers [14, 38]. These would be a drop-in replacement for FCFS or auctions using the abstract manager class in NAAIMS, but the challenge here is in how differently traffic signals operate compared to FCFS-like managers because they rely on temporal phases instead of reservations.

As for new features, the abstraction of the tiling space allows for replacing square tiles with other options, such as triangular or hexagonal tiles or even inconsistently shaped tiles like conflict points. Given that tiles in a square tiling are used with uneven frequency, placing tiles only where trajectories intersect or run closely together may be much more efficient than the status quo.

Chapter 3

Stochastic AIM

Most AIM models to date have assumed that vehicles follow their trajectories precisely with no adaptive variance values, accounting for the margin of error by simply drawing out a wider swath than the actual footprint of the vehicle to account for margins of error, and by default NAAIMS is no exception. On roads, this isn't as much of an issue because there are no conflicts, but being able to model stochasticity in intersections is much more important for realistically representing the caution vehicles need to execute to avoid collisions with others whose ability to follow precise directions they may not be as sure of.

Recall from section 1.2 that the original motivation for stochastic reservations is to incorporate human drivers using AR guidance into AIM like AVs but with inconsistent movement. Allowing multiple vehicles to use portions of same tile without their requests getting rejected will allow us to get tighter tolerances and more flow with the same vehicle assumptions despite imprecise driver behavior. This analysis is also a two-way street: using stochastic AIM, we can examine how much the AV to human guided vehicle split deteriorates performance–or doesn't–and give recommendations for how human drivers should be informed to make the most out of automated intersections. Observing how well simulated human drivers perform in an automated intersection using parameters derived from real human driving simulator experiments will give us insight into whether prioritizing turning radius or throttle control in the AR system will be most important for maximizing throughput and safety in stochastic AIM.

Furthermore, although this model is originally intended for to human drivers with guidance systems, extending stochastic reservations to automated vehicles will in effect accommodate the margin of error sensor systems can have, allowing us to be more honest about the true, never-nonzero likelihood of crashes in AIM systems.

To the best of my knowledge, this dissertation is the first time stochastic reservations have been applied to autonomous intersection management, at least those with discrete time tiling-based reservation systems.

In this chapter, I'll illustrate how stochastic reservations work, detail the stochastic movement model used for experiments in this chapter and how the human driving simulator experiments by our collaborators at the University of Washington inform it, and close by recounting the results of the virtual experiments in NAAIMS using the stochastic movement model and reservations.

3.1 Probabilistic reservations

Given a probability p that a reservation request uses any specific timespace tile, that tile must decide whether it can accept or reject the reservation. (How to find p is a question for the next section.)

In the deterministic setting, this decision would be simple: if there is already a confirmed reservation registered to this tile, reject the request so long as $p > 0$. (If $p = 0$ the tile shouldn't have been asked in the first place.) This is, in fact, exactly what stochastic tiles do so long as there have not yet been any reservations registered to the tile, i.e., the first request made to the tile is automatically eligible to confirm, pending the confirmation of all the other tiles used in the reservation request.

The behavior differs when the tile already has confirmed reservations. While the deterministic tile would reject all other reservations outright, the stochastic tile calculates the probability that more than one reservation, including the incoming request, will use the new tile. If this probability exceeds some predefined threshold $\overline{p}$ (i.e., the tile has tolerance for $\overline{p}$ probability of a collision), reject, otherwise approve the request as eligible to accept. To do so, the framework assumes that the probability that each reservation uses a tile is independent, which is justifiable as, in simulation, the probability distributions the realized deviations are drawn from are independent, and, in a real-world approximation, different vehicles are controlled by different agents who are not communicating with each other, only the intersection.

More formally, define for a specific stochastic tile

- $\overline{p}$ as the maximum acceptable likelihood of a conflict
- n as the number of reservations already confirmed with $p_i,\ i \in 1, \ldots, n$
- p_{n+1} as the probability that an incoming reservation request uses the tile

Given a request with probability p_{n+1} that it will use a specific tile

1. if no reservations have been confirmed yet, return eligible
2. else if the probability that more than one reservation (including the new request) uses the tile is less than $\overline{p}$, return eligible
$$1 - \prod_{i=1}^{n+1}(1 - p_i) - \sum_{i=1}^{n+1} p_i \prod_{\substack{j=1 \\ j \neq i}}^{n}(1 - p_j) \leq \overline{p}$$
3. else reject

3.1.1 Setting a crash incidence threshold

$\overline{p}$ can be related to a real-world value as it approximates the likelihood of a crash over a small amount of time and area. In traffic engineering, crash incidence is measured in crashes per million entering vehicles (MEV); we can use a reference value and divide it across the tiles in an intersection to find a reasonable $\overline{p}$. Surveying the field, the 0.58 crashes per MEV observed by the New York State Department of Transportation on signalized 4-legged intersections is one of the lowest and suitable to adapt for this purpose, and can be further lowered by a safety factor to set the stochastic intersection to be safer than a normal signalized intersection [34].

To find $\overline{p}$, first convert crashed per MEV to crashes per time unit using the intersection's expected demand in vehicles per time unit. This is roughly equivalent to the probability of a crash per time unit, i.e., across an entire tile layer, $\overline{\rho}$, and needs to be divided across all tiles in the layer. Given n_{tiles} tiles of equal size and distributing probability evenly and independently,

$$\overline{p} = 1 - (1 - \overline{\rho})^{\frac{1}{n_{\text{tiles}}}}$$

For sufficiently small $\overline{\rho}$ or sufficiently large n_{tiles}, a first order Taylor series approximation is very accurate, more numerically stable, and simply

$$\overline{p} = \frac{\overline{\rho}}{n_{\text{tiles}}}$$

3.1.2 Soft- and hard-edged stochastic reservations

At the beginning of this section, I detailed a method to allow for stochastic reservations with variable probability of usage, drawing the distinction between a deterministic AIM paradigm where vehicles fully reserve every tile they use so long as $p > 0$. But, given a probability threshold $\overline{p}$, we can implement a new variant of deterministic reservations compatible with stochastic movement such that $p > \overline{p}$ is cast to a full 100% probability of usage and $p < \overline{p}$ is cast to 0. In effect, this makes the "soft-edged" reservations at the start of this section that record mostly infinitesimal probabilities of usage at every tile in the intersection hard-edged and, in theory if $\overline{p}$ is implemented as intended, do so with a similar crash rate and by using less memory.

At the very least, this should provide a competent counterpoint to determine if the additional memory consumption of soft-edged reservations justifies their larger memory footprint.

3.2 Stochastic vehicle movement

To demonstrate the utility of stochastic reservations, given a vehicle, we need to be able to provide a reasonable probability of usage for each tile. In this section, I'll describe a relatively simple model of stochastic vehicle movement primarily to illustrate the effectiveness of stochastic AIM.

For this stochastic vehicle movement model, we can divide it into two components: the *lateral* deviation and the *throttle* deviation, illustrated in Figure 3.1. The throttle deviation describes the difference between the vehicle's ideal/instructed movement along the axis of the intersection lane trajectory it's following, whereas the lateral deviation describes the difference perpendicular to that axis.

The distributions for lateral and throttle distributions may be defined separately as independent normal distributions $N(\mu_l, \sigma_l)$ and $N(\mu_h, \sigma_h)$ respectively, or together as a multivariate Gaussian, although this will require finding correlation values between lateral and throttle tracking, or assuming that the correlation is 0. For the purposes of this simplified model the two are

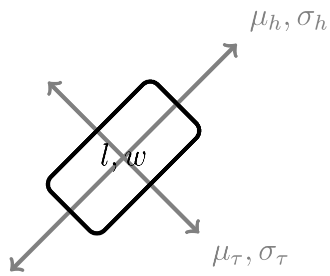

Figure 3.1: Throttle and tracking scores

treated as independent, but you can imagine that they would have significant correlation in practice.

3.2.1 Virtual reality experiments with human drivers

To examine how well human drivers would be able to perform in an automated intersection given route guidance and inform how we quantified stochastic vehicle movement by human drivers, we collaborated with behavioral experts at the University of Washington, Professor Linda Boyle and Jundi Liu, on the design and conduct of virtual reality driving simulator experiments based on their prior research with human drivers [22, 24, 29, 40].

In these experiments, we surveyed humans from a variety of demographics about their driving experience before sitting them into a virtual reality driving simulator emulating an urban environment. After being given a few minutes to familiarize themselves with the virtual reality environment and driving controls, they were instructed to begin the experiments by following a lead vehicle in the simulation by making left and right turns as well as through movements over a series of intersections. These included situations

Figure 3.2: The driving simulator

- with the lead vehicle moving at 30 and 40 mph, and
- with and without potentially conflicting traffic in the intersection.

The position, heading, and speed of the lead vehicle (i.e., the ideal trajectory the human driver was instructed to follow) and the driver's vehicle (i.e., their actual exhibited trajectory) were collected in the experiments for a sample of about 30 drivers, with selected results, aggregated from both speed limit scenarios but without traffic.

From these virtual reality experimental results, I converted the timestamped location for each human-driven vehicle and automated vehicle, normalized them to the time and location of the human-driven vehicle's entry into the intersection, and examined the lateral and throttle deviation distri-

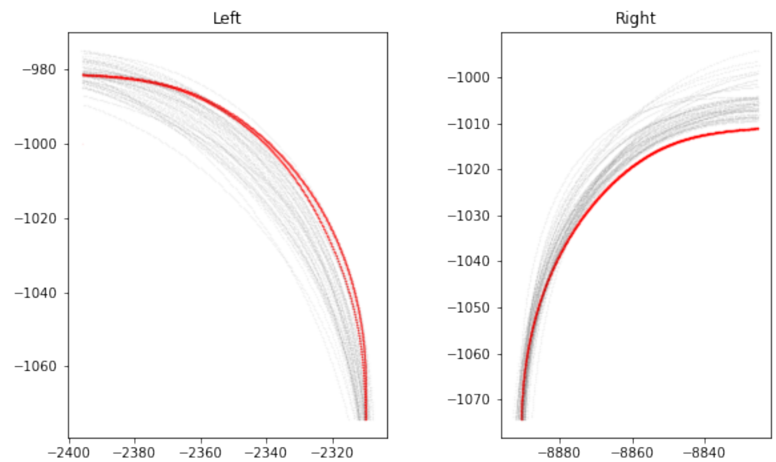

Figure 3.3: Left and right turn trajectories for human drivers (gray) and the provided reference trajectories (red)

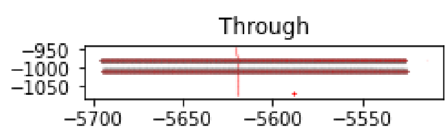

Figure 3.4: Through trajectories for human drivers (gray) and the provided reference trajectories (red)

butions. This requires the tacit assumption that human-driven vehicles always enter the intersection at precisely the right time, which wasn't reflected in the data we collected, but this assumption matches the behavior of the NAAIMS simulator and can likely be engendered in drivers using a more sophisticated trajectory information model. The simple one of a "ghost" pace vehicle used in the driving simulator was far from ideal for our use case, but it was necessary due to the limitations in the driving simulator software.

As seen in Figure 3.3, when it comes to lateral deviation, human drivers tended to oversteer on left turns, which are longer, and understeer on right turns, which are shorter. They're most precise as they're entering an intersection, but their lateral deviation blooms as they move through the intersection. Figure 3.4 shows the same isn't true of through movements, as human drivers were able to follow the given trajectory so precisely that it's difficult to display in a figure with a true-to-life aspect ratio.

Throttle deviations aren't quite as easy to depict in a graph, but the general trend comparing human driver timestamps to their automated counterpart is that human driven vehicles tend to move through the intersection about 8% faster than instructed, albeit with a wide distribution that has a small fraction of drivers moving slower than instructed.

Using the normalized trajectories, for each vehicle I approximated the distribution of

1. the lateral deviation at the exact center of the trajectory normalized by the length of the trajectory, and
2. the throttle deviation by way of time elapsed between intersection entry and exit relative to the entry and exit time of the pace vehicle

(More on why these values in the next subsection.) Using these observations, I approximated the distribution of a typical driver's lateral and throttle deviation as a normal distribution, which allowed me to give my stochastic AIM experiments firmer empirical footing.

There is a caveat here in that due to the limitations of the driving simulator software, participants could not be reasonably instructed to exactly follow the lead vehicle's trajectory, but rather to stay within 200 to 400 feet because humans aren't capable of significantly higher degrees of precision. Based on their judgement of participants, it seemed to have not been a concern.

3.2.2 Incorporating the stochastic movement model into NAAIMS

3.2.2.1 Projection and realization

In NAAIMS, the lateral and throttle deviation models are each used at least twice per vehicle: once to project their movement and again to realize their movement.

The former occurs when the tiling is doing its nested simulation of a vehicle's reservation request to identify which tiles its reservation will use and to what extent. Their deviation values are based on their lateral and throttle tracking scores as converted into a normal distribution, μ_τ, σ_τ, μ_h, and σ_h as defined in Section 2.1.1. Distributions are used to find the probability p that a vehicle's reservation uses a specific tile when checking its request. Since we're defining the throttle and tracking distributions as independent for the purposes of this model, p is just the product of the individual p_τ and p_h from the throttle and tracking distributions, respectively.

The realized values come into play when the probabilistic movement of the vehicle collapses into a realized value when it enters the intersection according to its probabilistic reservation.

3.2.2.2 Lateral deviation

The lateral deviation of the movement model is constrained by one of the physical assumptions of NAAIMS that has some basis in reality. In NAAIMS, autonomous or not, vehicles are assumed to behave precisely and accurately between intersections, never deviating from the centerline laterally (except during lane changes) or from the exact vehicle behavior specified in section 2.1.1.

This constraint, combined with the results observed from our human driving simulator experiments, informs the stochastic vehicle movement model used by NAAIMS. Because vehicles must enter and exit the intersection at precise lateral points (but not necessarily times), that limits the variety of lateral deviation profiles vehicles can exhibit in the simulator. As such, a vehicle's lateral deviation is modeled based on its lateral deviation at the center of the intersection lane it's following through the intersection (by length). This value is made unitless by normalizing by the length of the trajectory, so the same lateral deviation score will represent a smaller actual deviation on a short right turn than on a longer left turn. By convention, a deviation toward the left of the trajectory is considered negative and right considered positive.

From the driving simulator experiments, I observed the normalized lateral deviation at center distribution to have a mean μ_τ of -0.0888 with a standard deviation σ_τ of 0.0631, for turns only. In NAAIMS, new sample values are pulled from a Gaussian distribution with these settings unless the movement is a through movement, in which case the simulator assumes that the vehicle will follow the centerline with perfect accuracy.

During reservation projection, tiles' p_τ are found by first calculating the component of tiles' distance to the center of the vehicle perpendicular to the vehicle's heading, i.e., its lateral deviation. Given this distance, the width of the vehicle, and the length of the tile along the axis perpendicular to the vehicle, the probability of usage of this tile is the area under the normal distribution defined by $N(\mu_\tau, \sigma_\tau)$ overlapped by the width of the vehicle plus the incidence length of the tile at that distance.

When realizing a vehicle's lateral deviation after it's secured a reservation, as soon as the vehicle breaks the intersection boundary, the intersection lane simulates its realized stochastic movement by drawing a value from the $N(\mu_\tau, \sigma_\tau)$ distribution. This represents the vehicle's maximum lateral deviation, which as modeled increases linearly from 0 at entrance to the full amount at the center of the trajectory before decreasing back to 0 as it reaches the end of the intersection.

3.2.2.3 Throttle deviation

Unlike with lateral deviation, NAAIMS has no requirement for vehicles to exit the intersection at any specific time. The primary constraint on throttle deviation is the requirement that vehicles do not exceed the given speed limit, which makes it so that the stochastic vehicle tile distribution is most evident in congested scenarios where the acceleration profile can vary.

The throttle deviation is measured in relative time units, with a more complex recipe than for lateral deviation. First, divide the observed or realized time to exit for the human driven vehicle by the reference time to exit of the intended acceleration profile to make it unitless. Given this, take the difference between this ratio and 1 such that positive values indicate a tendency to over-accelerate and exit the intersection faster than intended, and negative values, which under-accelerate and take too long to exit.

From the driving simulator experiments, this came out to an average μ_h of 0.0752 with a standard deviation σ_h of 0.1402.

Unlike the lateral deviation, which was measured in distance units, the throttle deviation being measured in time units makes it more challenging to convert into the distribution of lateral deviation *distances* necessary to find the probability of usage of a tile according to the throttle p_h. To do so, NAAIMS runs a small Monte Carlo simulation with 30 trials for each vehicle being projected, and defines the throttle distribution at each timestep as the distribution $N(\mu'_h, \sigma'_h)$ of the component of the distance between the vehicle's stochastic-projected location and actual-projected location along the vehicle's heading. This is converted into a probability of usage p_h for each tile using a similar process to the lateral deviation p_τ using the vehicle's length and the tile's incident length along the vehicle's heading.

3.3 Experimental results

To demonstrate the efficacy of stochastic reservations (or lack thereof), I ran a series of experiments to examine the throughput of stochastic AIM

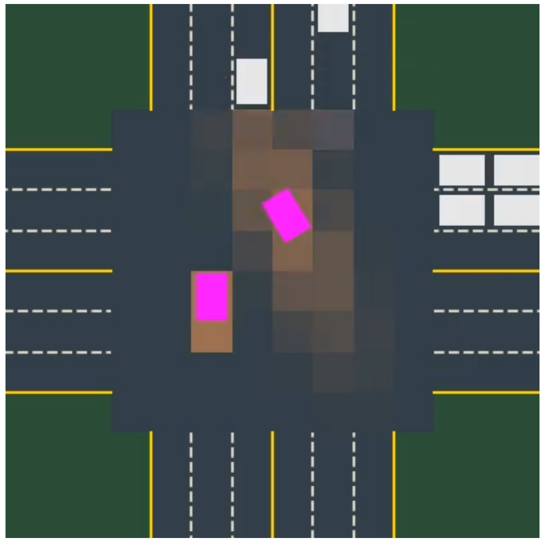

Figure 3.5: An example stochastic reservation in NAAIMS

compared to other options. The experiments designed in this section focus on the use case of stochastic reservations as applied to human-driven, guided-trajectory vehicles in the AIM environment. Automated vehicles are modeled as having perfect accuracy, with only the simulated "human"-driven vehicles exhibiting any kind of deviation, lateral or throttle.

For reference, the default experimental settings can be found in subsection 2.2.5, with only deviations from this setup remarked upon in this section.

3.3.1 Stochastic AIM performance as demand increases

Hypothesis: Guided "human" drivers with stochastic reservations perform better than both traditional signalized traffic managers and simply carving out large buffered reservations for human drivers.

Experiment: To test this, I compared 30 trials each of three different demand levels and four manager variants: soft-edged stochastic reservations with stochastic vehicles, hard-edged deterministic reservations (i.e., deterministic reservations with nonzero probability of multiple vehicles using the same tile under a low tolerance) with stochastic movement, deterministic reservations with deterministic vehicle movement, and a traditional singalized intersection timed using 4 cycles (one per approach) and Webster's formula [46].

Results: As seen in Figure 3.6, both FCFS variants with stochastic movement perform much more akin to fully autonomous FCFS than a traditional signalized intersection. Furthermore, soft-edged, stochastic reservations appears perform slightly better than large, hard-edged, deterministic reservations, especially at high demand.

3.3.2 Average delay as a function of AV penetration

Prior research shows that FCFS only starts to show significant performance increases as AVs approach 100% of the population. One of the main motivations for stochastic AIM is to decrease the percentage at which FCFS begins to take effect by tightening the spread of human-driven, guided vehicles.

Table 3.7 suggests that, not only does 100% "human" vehicles driving under stochastic AIM start at a much better delay value than under a signalized intersection, the performance improvement as the scenario's AV percentage increases is much smoother as well. Instead of performance gains

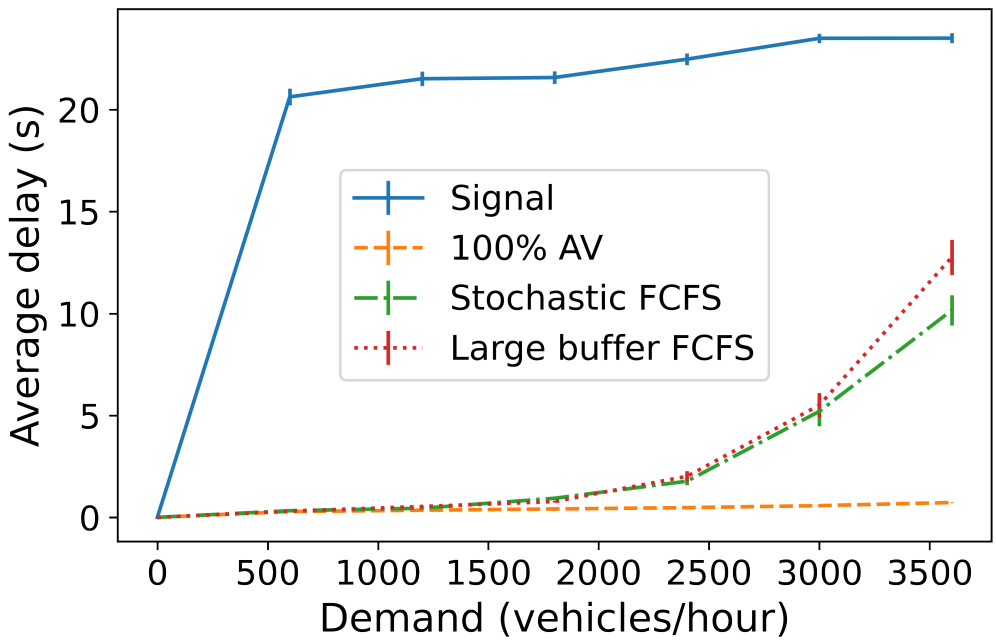

Figure 3.6: Delay as a function of demand under several FCFS scenarios

being most sharp near 100% AVs, the drop in delay is almost linear, even after accounting for the margins of error.

If these results hold, we may be able to experience most of the benefits of automated intersections much earlier as it's no longer contingent on full level 5 autonomy for all vehicles on the road, but instead only the rollout of guidance systems in connected but human-driven vehicles.

3.3.3 Impact of lateral versus longitudinal tracking

Lateral and throttle tracking abilities have disparate impacts on throughput. Consider Table 3.1, which summarizes the results of a trial with 100%

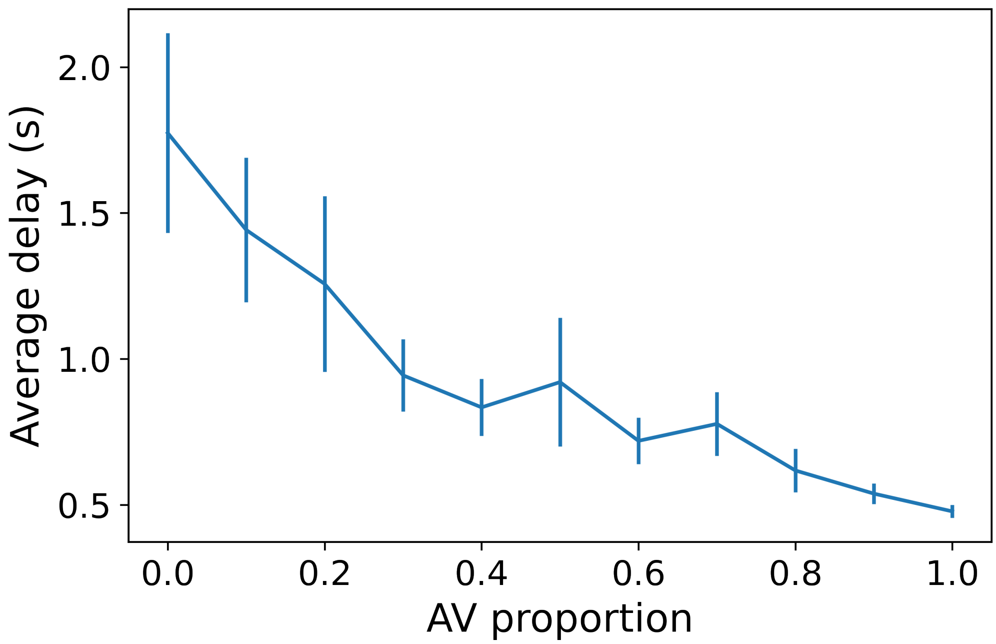

Figure 3.7: Delay as a function of AV proportion

human guided vehicles at a demand level of 3,600 vehicles per hour where throttle or lateral tracking were alternately reduced to 0. (100% human guided vehicle and 100% AV scenarios are also shown for comparison.)

Table 3.1: Lateral vs. longitudinal tracking delay (s)

	Mean	SD
Both	1.774	0.936
Throttle deviation only	1.069	0.355
Lateral deviation only	0.683	0.145
Neither	0.478	0.357

Cursory inspection and a difference in means hypothesis test with the null hypothesis of no difference in traversal time quickly demonstrate that reducing throttle deviation is likely to have a much stronger impact on intersection throughput than reducing tracking deviation.

That said, this is likely to be an emergent feature of our demand scenario's 80% through movement demand, combined with the assumption that even human-driven vehicles move with perfect lateral accuracy on a straight line. Granted, this is based on driving simulator data which I could argue further reinforces the hypothesis that teaching drivers to control their throttle more accurately will have much more return on investment than on improving turns when they make up a comparatively smaller percentage of intersection movements.

3.3.4 Varying crash tolerance and $\overline{p}$

Hypothesis: Decreasing the crash tolerance—which, in turn, decreases the $\overline{p}$ parameter in the tiling—increases average delay at the intersection.

Experiment: For this experiment, I varied the crash tolerance logarithmically above and below the default value of 0.05 crashes per MEV, which itself was lowered one order of magnitude from the incidence rate found by the New York State Department of Transportation [34]. This value, C, is related to the $\overline{p}$ through a series of unit conversions as follows:

$$\overline{p} = \frac{C \text{ crashes}}{10^6 \text{ vehicles}} \times \frac{1 \text{ vehicles}}{\text{min}} \times \frac{1 \text{ min}}{60 \text{ sec}} \times \frac{s \text{ timesteps}}{\text{sec}} \times \frac{1}{n_{\text{tiles}}}$$

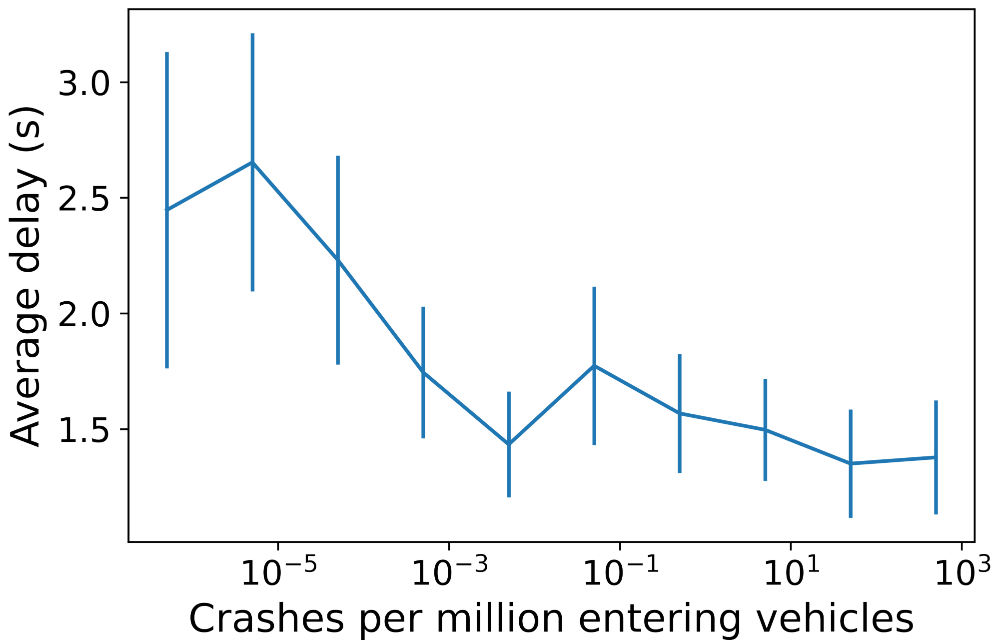

Figure 3.8: Tolerable crashes per MEV versus delay

where s is 15 timesteps per second in the default experimental parameters and n_{tiles} is $8 \times 8 = 64$.

Results: Figure 3.8 shows that the hypothesis is generally true. Although the relationship is somewhat ambiguous near the default value, tolerable crashes per MEV much greater than 0.05 does indeed appear to be associated with lower delay, and vice versa, although the effect seems to plateau at extreme values. Future real-world implementations of stochastic AIM will need to take this relationship into account when deciding on their $\overline{p}$ and tolerable crash rates.

3.4 Future work

Stochastic reservations, because they require fine calculations of usage probability per tile, likely benefit from a denser discretization of the intersection compared to the fully autonomous paradigm. I suspect tailoring this more closely to reflect stochastic reservations' imprecision will improve performance.

Furthermore, our movement model will also stand to benefit from the implementation of a more sophisticated conceptualization of stochastic vehicle movement that, among other things, quantifies the correlation between lateral and longitudinal deviation.

For additional rounds of human driving simulator experiments, my observations while preparing this report suggest that observing how drivers can follow an acceleration profile will be very valuable to refining our stochastic movement model, considering how important throttle deviation is to throughput. Likewise, given drivers' tendency to oversteer on left turns and understeer on right turns, providing drivers with a static reference of how hard or soft they should turn may markedly improve their tracking performance at minimal cost.

Chapter 4

Priority Auctions

At an intersection, individual vehicles have the self-interested goal of moving through the intersection as quickly as possible. They're prevented from doing so by the vehicles in front of them in their lane, as well as by vehicles approaching the intersection with conflicting movements. In the AIM paradigm, the intersection decides which of these vehicles gets priority, in this case by way of repeated auctions for scarce space and time in the intersection conflict area. All vehicles in the intersection are competing for use of this area as soon as possible.

This chapter will detail how other authors have approached competitions for intersection access before detailing a framework for priority auctions–including bidding and win condition, followed by the specifics of three payment mechanisms, a description of how auctions are implemented in the NAAIMS package, and experimental results before concluding with directions for future work in the priority auction space.

The key novel contributions of this chapter are

- describing a consistent framework for per-vehicle intersection auctions

- incorporating multiple dispatch (vehicles from multiple lanes) into the auction mechanism

- incorporating sequencing (multiple vehicles from a single lane) into the auction mechanism

- introducing a new, externality-based payment mechanism for priority auctions

4.1 Related work

Intersection priority auctions find their roots in Beckmann et al., who proposed that network-wide tolls can encourage drivers to minimize total travel time instead of acting in their own self interest [6].

Applying timespace auctions to AIM was first introduced by Schepperle and Böhm in 2007 and 2008 [36, 37], who found that auctioning off priority by way of intersection time-slots was able to reduce value consumed and increase driver satisfaction. Their work was expanded upon by Vasirani and Ossowski as well as Carlino et al.; both applied similar auction mechanisms to networks of intersections for an improvement in weighted delay, although Vasirani and Ossowski relied on reserve prices for routes between intersections in addition to auctions at intersections [8, 44].

Since then, there's been considerable interest in the potential of intersection auctions and their real-world applicability. As referenced in prior sections, transportation engineers like Levin et al. have worked to incorporate intersection auctions into their study of dynamic traffic flow models [25], while economists, Zakharenko in particular, has been working on refining the theory behind intersection priority auctions to create provably optimal and incentive-compatible mechanisms [51].

Some authors have even explored how receptive the general public might be to auction-based control schema [5, 7] and how to apply auctions to present-day traffic signals using micro-auctions through cell phones [4]. Yet more have considered how auctions could work with both autonomous and human-driven vehicles [1], echoing some of the sentiments in the prior chapter of this dissertation. Some members of this dissertation committee have even been contacted by the patent holder of routing bids and transfers to gauge interest in collaborating based on this and past research [43].

4.1.1 Desiderata for an ideal mechanism

Based on my review of existing literature combined with knowledge of intersections and driver priorities, an ideal intersection priority auction mechanism would have the following traits:

1. supports intersections of any configuration and any turn restrictions
2. supports vehicles with heterogeneous characteristics
3. supports realistic vehicle kinematics
4. be incentive compatible

5. be strategy-proof
6. be computationally tractable in real-time
7. be fair for all participants, which can have many meanings, but I'll define it as not advantaging specific vehicles based on the results of factors outside of cooperating vehicles' control, such as the prior auction

This said, achieving all of these with a single mechanism may be impossible, like in Arrow's possibility theorem. No prior author has achieved all of the above, although Zakharenko comes closest, but only for the most restrictive case of a zipper merge [51]. At best, an auction mechanism may only be able to address most but not all of these desired traits, which is what this chapter attempts with extensions of prior mechanisms.

4.2 The intersection priority auction framework

4.2.1 Key assumptions

For the sake of a consistent auction model, I make the following assumptions that may differ from prior research:

Auctions only occur when an intersection is completely clear of reservations. To ensure fairness for all participants, the next intersection auction only triggers once the intersection conflict area is empty and no tiles are spoken for. If this limitation wasn't set, vehicles coming from the same lane as the winner of the last auction would be unfairly advantaged because by nature of the intersection geometry they can't have an incompatible movement with the already confirmed movement.

The intersection auctioneer is benevolent and doesn't care about its own revenue. I assume that its only goal is to order vehicles to pass through the intersection in such a way as to minimize their value-weighted delay, including any payments made to the intersection auction manager.

Vehicles report their valuations truthfully. Although some of the mechanisms outlined in this chapter will include provisions to incentive truthful reporting, these considerations are not provably strategy-proof. Ultimately I'm assuming that vehicles are telling the truth about their values of time (VOTs), although I do run experiments to quantify to what extent vehicles can benefit from misreporting their VOTs.

Vehicles would always prefer for vehicles ahead of them in lane to move sooner. One of the implicit assumptions when designing this auction model is that a trailing vehicle always benefits from vehicles ahead of them in their lane moving before vehicles in competing lanes. Although this makes sense intuitively–vehicles need the vehicles in front of them to move before they can move themselves–this is not always true; more on this in section 4.6.1.

4.2.2 Sets eligible to win the auction

As a result of several of my assumptions coming together, at its simplest, only vehicles leading an incoming lane are eligible to win an auction and complete their desired movement through the intersection. We can broaden our definition of a "winner" to include all vehicles who benefit from the winning movement, i.e., all vehicles trailing the winning lane-leading vehicle in an incoming road lane. Under this paradigm, we conceptualize a winner not as an individual vehicle, but as an entire incoming road lane and the vehicles on that lane (or the null set, if no lanes contain vehicles).

We can increase the auction manager's throughput to more closely follow the free-flowing movement of FCFS by permitting multiple vehicles to move in a single auction. This can take one of two forms (or both can be combined):

- multiple dispatch, allowing vehicles from more than one incoming lane to move at a time
- sequencing, allowing more than one vehicle from a single lane to move in a single auction

Sequencing follows naturally from the idea that *lanes* not *vehicles* win an auction, so we'll leave this topic for now and expand upon it in subsection 4.2.6, but multiple dispatch adds complexity to the eligible winning sets.

Ideally, we'd let every lane with vehicles on it through together, but the point of an intersection manager is to avoid collisions, so this isn't tenable. We can use the core idea of having multiple lanes be eligible to win, and add the condition that their lane leading movements don't conflict (similar to traffic signal phases, but without fixed patterns).

To wit, under multiple dispatch, eligible winning sets consist of incoming road lanes with non-conflicting lane-leading movements.

There is significant additional complexity to multiple dispatch, as deciding which movements are compatible if the intersection configuration is arbitrary and variable vehicle approach speeds increases the computational power required to resolve auctions, but doing so will allow for freer flowing movement more akin to present day signalized intersections. By allowing multiple lanes' leaders to move at once, we can make more efficient use of the limited time available in the conflict region.

4.2.3 Calculating bids

If we're focused on the movement of a lane-leading vehicle as the potential winner of an auction, a natural first guess would be for the bid associated with it to be the value of time ν of the vehicle making the movement applied to the time the movement uses in the intersection t_m. This design, however, misses two key points:

- winning is not necessarily about wanting to use time in the intersection alone, but primarily to prevent competitors from doing so before you
- the lane-leading vehicle isn't the only one that stands to gain from its

own movement winning

I'll motivate each point using an example.

Suppose we have a vehicle A with a value of time of \$100 per second and a short, one-second movement competing with vehicle B with a much lower VOT of \$1 per second but a much longer movement of 200 seconds. A would bid \$100 to B's \$200 and lose, forcing them to wait 200 seconds at a cost of \$20,000 to itself. They would've preferred instead to bid their VOT against the time consumed by B, or the longest intersection time consumed by any lane movement eligible to win, which we can simplify to simply bidding a rate instead of a fixed value.

Second, consider the example illustrated in Figure 4.1. Let A have a VOT of \$1, C \$, and B \$100. If only lane leaders were allowed to bid, A would lose to C and leave B stuck behind A for the entire duration of C's movement, despite having more than enough value to outbid everyone in the auction combined.

Given the above points, the bid for an individual lane ℓ and its lane-leading movement is defined as

$$b_\ell = \nu_\ell = \sum_{i \in \ell} \nu_i,$$

where ν_i is the VOT of vehicle i. This method ensures that every vehicle that stands to benefit from a movement gets to contribute to the bid, and that their bid quantifies how much they value avoiding delay rather than being dependent on a specific time length.

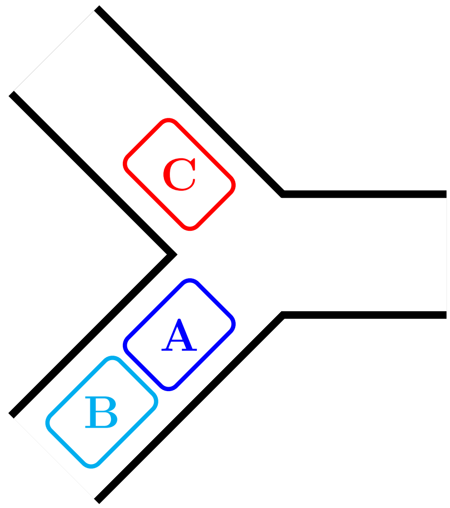

Figure 4.1: One vehicle versus two

In the multiple dispatch case, the effective bid for a set of lanes is simply the sum of the ν_ℓ of the set.

4.2.4 Deciding the winner

Quite simply, the winner of the auction is the set with the highest bid b. This can be the null set if there are no vehicles in the intersection, a single lane in the default case, and one or more lanes under multiple dispatch.

This can be made more complex when we consider where the intersection can put its thumb on the scale: some might apply subsidies for equities sake, e.g., to prevent a high-flow lane from constantly dominating a low-flow lane by sheer volume [8], or to account for expected future arrivals who would also benefit from winning this auction but who haven't yet entered the intersection area. For the purposes of this framework, the intersection manager does not intervene as the focus is on designing a clear mechanism and not necessarily equity concerns.

4.2.5 Payment calculation

After deciding the winning and losing sets, the mechanism must finally determine how much each vehicle pays. The payment may not be directly related to what they bid–in fact, decoupling these as much as possible would further disincentive vehicles from misreporting their VOT–and must be translated from a rate to a fixed value by applying the movement time of the winning set. A few key values used by some of the payment mechanisms being described:

- b_Ω, the bid of the winning set.
- t_Ω, the time consumed by the winning movement. In the multiple dispatch case, this will be defined as the time of the longest or latest-exiting movement in the set.
- b_E, the second-highest bid.
- t_E, the time of the movement of the second-highest bidder. Same rules as above in the multiple dispatch case.

Designing a good payment mechanism is tricky because this is the main way an auction mechanism incentivizes truthful reporting of VOTs. The primary contribution of this chapter, in addition to formulating this auction framework, is to introduce a new payment mechanism that improves upon prior mechanisms. I'll go into full detail on the both predecessor and the new payment mechanism in the next section.

4.2.6 Resolving sequences

Auctions with sequencing allow consecutive vehicles in a single lane to move together. This is beneficial for both the trailing vehicles in a sequence and total system delay because sequenced vehicles use less total intersection time than having a vehicle wait for the preceding vehicle to clear the intersection before they can move themselves does. The challenge of incorporating sequencing into auction mechanisms is its computational complexity outside of predetermined signal phases and it being detrimental to most other self-interested vehicles. Overcoming these to implement a sequenced auction mechanism is one of the key contributions of this chapter.

The key point to keep in mind when thinking about sequenced auctions is that allowing a n-length sequence to win is strictly worse for losing lanes because a sequence takes up strictly more time in the intersection that an $(n-1)$-length sequence.

To accommodate this, this framework requires that the $(n-1)$-length sequences win the auction before additional sequence lengths can be considered. Two consequences of this requirement are

1. sequenced auctions must be designed as separate auctions that run after

the $(n-1)$-length auction that each trailing vehicle in a sequence must win to confirm their sequence

2. leading vehicles in a sequence have no incentive to contribute to the win of trailing vehicles in their sequence

Given these premises, let's describe the framework of a sequenced auction. For the rest of this discussion on sequenced auctions, I'll restrict the conversation to a single winning lane only. Conceptually, describing sequenced auctions with multiple dispatch is difficult because of possibility the winning and first losing sets containing the same lane, complicating the clarity of the sequencing mechanism, but this simpler sequenced mechanism can be applied to the multiple dispatch case with a few adjustments.

4.2.6.1 Sets eligible to win the auction

Because sequenced auctions are only run after the original auction winner has been decided, the sequenced auction's eligible winners are either the one winning lane seeking to extend the sequence of the winning lane(s), or the null set, ending the sequence and readying for the next auction.

The vehicles that benefit from the sequence winning are the trailing vehicle(s) vying to add to the sequence, and all vehicles behind them in their lane (but not the vehicles in the sequence that are already confirmed). The vehicles that benefit from the sequence ending early are those in the highest value eligible set that doesn't include the road lane of the sequence vying for extension.

4.2.6.2 Calculating bids

The vehicles that contribute to the sequences' eligible set bids are the vehicles that stand to benefit. The vehicle vying to have its movement sequenced and the vehicles behind it bid for the sequence to win, while the vehicles in the highest bidding set without the winning lane bid for the sequence to end early.

Relative to the bid of the entire winning lane, as you progress down the lane, the ith vehicle in a sequence has a supporting VOT from trailing vehicles of

$$b_i = \sum_{i}^{n_\ell} \nu_i$$

Notice that $b_1 > b_2 > \ldots$; in other words, the longer a sequence, the lower its effective bid, making it easier for the competing set to end the sequence.

4.2.6.3 Deciding the winner

Unsurprisingly, the winner is the sequence extension if their bid is higher, or the null set ending the sequence if the second highest eligible set wins.

4.2.6.4 Payment calculation

Sequenced auctions utilize the same payment mechanisms as the original auction by lane, with one slight adjustment to the key time lengths:

- t_Ω becomes the *marginal* additional intersection time consumed by the vehicle vying to be sequenced

- t_E is the difference between when the movement of the second highest bidding lane would have ended compared to when the movement of the $(n-1)$-length winner ended, with a minimum of 0 to represent how much additional time past what's already been reserved the alternate winner needs to complete its movement and pay for itself.

These times serve to compare the winning sequence against the counterfactual of "what if the second highest bidding lane had won instead." More on payment mechanisms in the next section.

4.3 Payment mechanisms

As mentioned earlier, the payment mechanism is the primary way priority auctions have to incentivize vehicles to truthfully report their VOTs. For this chapter, I've formulated three payment mechanisms to compare which one is empirically the most equitable, effective, and truth-incentivizing.

4.3.1 First price

This mechanism design is based on the most basic payment mechanism in general, first-price auctions. (As an aside, the payment mechanisms described here are usually described as *auction* mechanisms in literature outside of this dissertation, but this framework adjusts these mechanisms to have identical steps prior to payment.) The version used in this dissertation is based on the mechanism Levin used in a study examining the throughput of auction-based control in dynamic traffic assignment [25], with some modifications to match the NAAIMS auction framework. For example, in order to conform to our fairness requirement, the next auction is only run when the last vehicle has cleared the intersection, compared with Levin's implementation where auctions can run at all times, so long as there are vehicles that are stopped at the intersection boundary.

Put simply, in the first price auction, the winning bid b_Ω is applied to the time used by the winning set t_Ω and extracted from each vehicle that bid for it.

$$p_{\text{1st}} = b_\Omega t_\Omega$$

Each vehicle i in the winning lane(s) pays its reported VOT ν_i times the time of the winning set.

$$p^i_{\text{1st}} = \nu_i t_\Omega$$

4.3.1.1 Potential flaws

Although very simple and easy to compute, all first-price auctions share a fundamental flaw: bidders are incentivized to under-report their true valuations as what they report is directly tied to what they pay if they win. Instead, they'd prefer to bid just high enough to beat the valuation of the second highest bidder, but no more, to avoid the "winner's curse" of overpaying [39].

Furthermore, this auction mechanism also fails to account for the threat

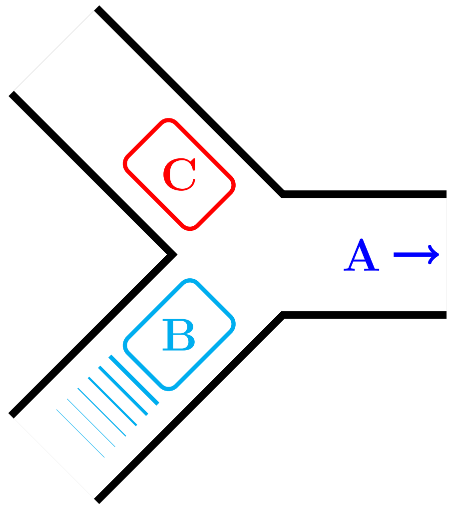

Figure 4.2: First-price example

of new arrivals in the participants' behavior. Consider the example illustrated in Figure 4.2. Reducing VOTs and bids to fixed values for clarity, let

$$\begin{aligned}\nu_C &> \nu_A, \nu_B \\ \nu_C &< \nu_A + \nu_B - \delta \\ 0 < \delta &< \min\{\nu_A, \nu_B\} \\ \nu_{\text{new}} &= 100\nu_A\end{aligned}$$

Suppose A wins this auction and as they make their movement one hundred new vehicles arrive behind B. In the next auction B and the new arrivals bid $\nu_B + \nu_{\text{new}} \gg \nu_C$. Now C has to wait for the other lane to clear up before it can leave. C would have preferred to overbid in the first auction than wait this long, illustrating the need to factor in the threat of new arrivals into the bidding process.

4.3.2 Second price

As far as this author knows, the idea of single-intersection auctions originated from Carlino's work in 2013 [8]. The mechanism proposed was a second-price auction, so used because this type of auction in general encourages bidders to reveal their true valuation when bidding instead of the low-balling behavior discussed in the prior section because they don't need to pay their bid if they win, only the amount the *second* highest bidder (or "first loser") bid.

The second-highest bid b_E is applied to the time length of the winning movement and extracted proportionally from each vehicle that bid for the winner.

$$p_{\text{2nd}} = b_E t_\Omega$$

Each vehicle i in the winning lane(s) pays its fraction of the second-highest bid proportional to how much it contributed to the winning it using its reported VOT ν_i times the time of the winning set.

$$p^i_{\text{2nd}} = \frac{\nu_i}{b_\Omega} b_E t_\Omega$$

4.3.2.1 Potential flaws

In the case of a single winning vehicle, $\frac{\nu_i}{b_\Omega} = 1$, this mechanism fully decouples their payment from how much they bid (excluding the victory condition), removing the incentive to under-report their true value. This decoupling is only partial in case of multiple vehicles, as calculating winners' payments based on the proportion of the winning total they bid introduces its own problem.

Consider again the example illustrated in Figure 4.1, reusing the same bids and valuations from the example in Section 4.3.1.1.

Suppose B and C bid truthfully, while A bids $b_A > \nu_A - \delta$. Under the second-price auction, A and B win and must together pay ν_C. If A bid ν_A, they would pay $\nu_C \frac{\nu_A}{\nu_A+\nu_B}$, but if A had lied and under-reported their valuation, bidding $\nu_A - \delta$, they would have only had to pay

$$\nu_C \frac{(\nu_A - \delta)}{(\nu_A - \delta) + \nu_B}$$

The second-price auction eliminated one reason for vehicles to under-report their true value only to replace it with another, and it's still vulnerable to the threat of new arrivals described in subsection 4.3.1.1. To avoid this, the winner's payment must not be directly dependent on their bid in any way and quantify the latter threat.

4.3.3 Externality

To determine an individual vehicle's payment for an individual auction, we approximate the externality of its victory–how much it harms other participants by winning–as in Vickrey-Clarke-Groves (VCG) auctions [11,18,45], and make that its payment for winning the auction. This serves to better avoid incentivizing unintended behavior and encourage truthful reporting from vehicles of their value of time. This payment mechanism is newly developed for this dissertation and attempts to extend Zakharenko's mechanism to arbitrary intersection configurations.

The externality of a winning vehicle i is the difference between the total value across all vehicles of the current scenario and if i wasn't in the auction. To find this, we consider 3 sets of lanes:

- the actual winners Ω;
- the first losers E, the lanes that would have won had i not been in the auction; and
- everyone else $L \setminus (E \cup \Omega)$; the lanes that lose the auction either way.

An example is illustrated in Figure 4.3.

Recall from Chapter 2 that we've assumed for each lane ℓ, new arrivals

1. enter at an average rate per time unit of λ_ℓ and
2. have an average VOT μ_ν.

By winning, i

1. forces every first loser in E to wait the winning movement time t_Ω and suffer new bidders,
2. benefits every actual winner Ω by saving them the first loser's movement time t_E and from the arrival of new bidders, and

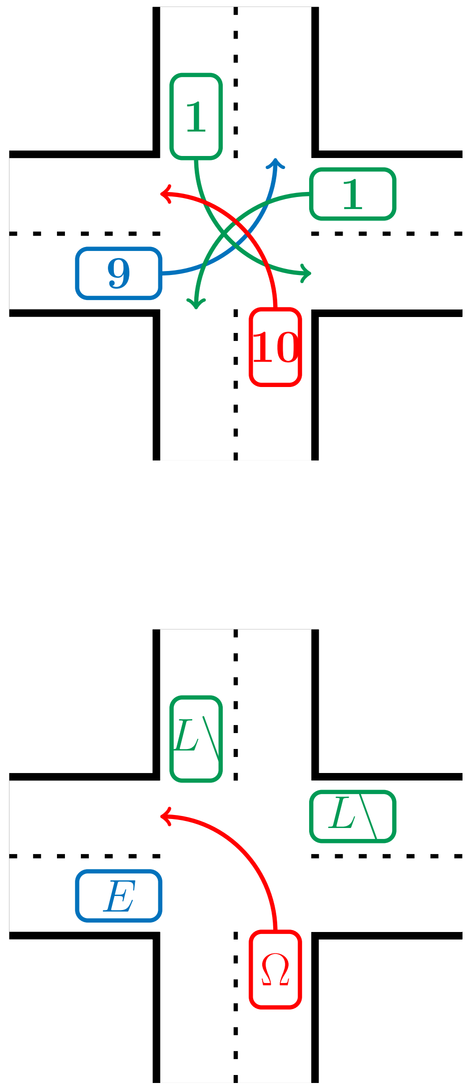

Figure 4.3: Winner (Ω), first loser (E), and everyone else ($L\backslash$) example

3. causes every other lane $L \setminus (E \cup \Omega)$ to wait the difference $t_\omega - t_E$ longer (or skip this time, if the difference is negative).

Note that this t_E is defined slightly differently from subsection 4.2.5.

Thus the estimated externality χ_i of vehicle i is

$$\begin{aligned}\chi_i = &\sum_{\epsilon \in E} \left(\nu_\epsilon + \overline{\nu}_\epsilon \overline{\mu}_\epsilon \frac{t_\Omega}{2} \right) t_\Omega - \sum_{\omega \in \Omega \setminus i} \left(\nu_\omega + \overline{\nu}_\omega \overline{\mu}_\omega \frac{t_E}{2} \right) t_E \\ &+ \sum_{\ell \in L \setminus (E \cup \Omega)} \left(\nu_\ell + \mu_\nu^\ell \lambda_\ell \frac{|t_\Omega - t_E|}{2} \right) (t_\Omega - t_E)\end{aligned}$$

Two special cases to note are

1. if removing the study vehicle doesn't change the winning set, $t_\omega - t_E = 0$ and the vehicle has 0 externality. This applies even if every vehicle in the winning set has a VOT too small to individually affect the outcome, making the total payment across all vehicles 0.

2. χ_i can be negative if $t_\Omega < t_E$ because vehicles aside from the first losers benefit if a short movement wins instead of a long one.

The second case can cause unintended consequences, as we don't want vehicles to be able to get paid by winning a priority auction or this would incentivize a lot of induced vehicle trips aiming to farm the system for payouts. On the other hand, discarding negative externalities would effectively raise the cost of longer movements in relative terms.

To approximately reconcile these competing priorities, the externality mechanism only collects payments after the vehicle has experienced every auction it'll experience in the intersection and exited. At this point, we'll sum its payments across all its auctions and then floor them so i pays

$$p_i = \max\left\{0, \sum_{\alpha\in\mathcal{A}} \chi_i^\alpha\right\}$$

where $\mathcal{A}$ is the set of all auctions i won and χ_i^α the externality of i winning auction α. This aggregation reduces the skew of flooring on the auction mechanism.

Although this mechanism borrows concepts from VCG, to be clear, this auction framework is not a proper VCG auction. The externality calculation we use is an estimate, as it requires removing a winning vehicle and rerunning the auction without it. This isn't realistic due to the physical configuration of an intersection, but it suffices for the purposes of our mechanism.

This auction mechanism is inspired by that of Zhakarenko [51]. Unlike most other studies on the topic that are mostly done by computer scientists and transportation engineers, Zhakarenko approaches this problem as an economist, deriving the most theoretically sound value-aware ordering of vehicles through a choke point. However, his findings were limited to only a two-to-one-lane zipper merge, limiting the utility of his mechanism design for modeling intersection traffic.

This payment mechanism takes the principles of Zhakarenko's mechanism and applies them to an intersection as closely as possible, discarding provable strategy-proofness for general applicability. The design of this mechanism is intended to directly address the flaws identified in the first- and second-price auctions and, while it doesn't fully and provably prevent them, the mechanism caveats here are intended to ameliorate their exploitability.

4.4 Implementing auctions in NAAIMS

Although the first- and second-price auctions are relatively simple to implement, being limited to the intersection manager, the particulars of the sequenced auction require extensions to the tiling and the way NAAIMS processes reservation requests. Normally, checking if a request is possible is dependent solely on the request being checked, since vehicles from different lanes aren't moving together. However, because sequenced auctions allow us to dispatch vehicles from multiple incoming lanes at once, the intersection also needs to check if requests competing in a sequenced auction are compatible with each other, instead of only already confirmed reservations.

This is an additional feature added to the intersection tiling in the form of the potential reservation marking system. When checking a vehicle's trajectory for a sequenced auction, the tiling is also instructed to mark tiles encountered with the potential reservation of that vehicle, like with confirmations except with the ability to register multiple potential reservation systems onto a single tile (stochastic reservations excepted). So long as requests share at least one tile with each other, then they're incompatible and cannot be included in the same winning set. Collecting these incompatibility lists from every tile used across all potential reservations in an auction will allow the intersection to determine every possible set of winning movements and calculate their bids accordingly. Optimization is possible given that if a pair of reservations is incompatible, then sets containing that pair and other lanes are also incompatible, but the general problem of identifying the highest value set of compatible reservations given pairs that are incompatible reduce to the NP-complete Boolean satisfiability problem [12].

Another modification to accommodate sequenced auctions is to allow the reservation request of one vehicle to depend on another, i.e., for a trailing vehicle's reservation request to depend on the request of the vehicle in front of it. That way, if the request of a vehicle leading or in the middle of a vehicle sequence finds that their reservation is incompatible with a confirmed reservation or, more likely, only the front of the sequence is part of a winning set, the requests middle and all the way down the sequence can be automatically rejected without disrupting the front of the sequence. It also assists with the request marking system, as with this in place only the request of the leading vehicle needs to be marked on a tile; if that's already present, it's implied that the request of trailing vehicles is also dependent on that tile and they're of course never incompatible with the request of the leading vehicle, so there's no need to mark the tile with the request of the trailing vehicle. (This comes into play if the vehicle buffer in the intersection is larger than the vehicle length buffer when traveling along lanes, and with the edge tile buffers that extend forward and backward into time further than the vehicle is actually expected to use physically.)

However, simulating conflicts is very computationally expensive, so we limit sequences to consecutive vehicles in the same lane with the same movement, so we don't need to simulate mix and matched sequences of movements with different start times.

4.5 Experiments

Several experiments are necessary to confirm the base efficacy of these auction formulations. The sequenced auction is designed and intended to be a strict improvement of the first- and second-price auctions, but confounding emergent behavior may make it perform worse than intended.

4.5.1 Comparison experiments

The most basic set of experiments are self-explanatory: create several traffic profiles using a random generator from given average flow λ_ℓ and VOT μ_ν^ℓ for each lane ℓ, and send them through each of the mechanisms listed above. Each simulation collects payments made at the intersection as well as vehicle VOT and entry and exit times, from which we can calculate the time each vehicle spends in the intersection and how much that time cost the vehicle. Together, these costs (as well as delay experienced by vehicles without factoring in VOT) can be averaged across all vehicles in each simulation instance, giving a single number measuring the value consumed that can be compared apple-to-apples across all simulations.

Hypothesis: The average traversal time (including payments converted to time) per payment mechanism will decrease consistently from first price to second price to the externality payment mechanism. Along the other axis, the single lane, single vehicle auction will perform the worst, followed by sequencing setup, with multiple dispatch performing the best.

Experimental setup: For these experiments, taking a cue from Carlino 2013 [8], vehicles' VOTs ν are drawn from a uniform distribution between 0 and 1 value units per second and the sample size is upped to 100 compared to the standard experimental setup described in subsection 2.2.5.

The following payment mechanisms are compared

- First-price
- Second-price
- Externality

using the following eligible winning set configurations

- single lane dispatch, single vehicle per lane
- multiple lane dispatch, single vehicle per lane
- single lane dispatch, multiple consecutive vehicles with the same movement per lane

Results: The relative performance of each model is as expected, with a few slightly surprising results reported in Table 4.1. The auction setups that only allow one lane to win at a time perform substantially worse than multiple dispatch, although sequencing does slightly better than the fully single setup, as expected of a setup intended to allow slightly more than one vehicle through instead of being comparable to a stop sign.

The surprise might be in just how much better multiple dispatch performs compared to the other two, especially since unlike FCFS or traffic signals it doesn't allow more than one vehicle per lane through at a time. Granted, it's still two orders of magnitude worse than FCFS even after accounting for the fact that FCFS doesn't prioritize VOTs (0.089 average incurred cost).

Table 4.1: Incurred cost sample mean (and standard deviation) under various auction configurations ($n = 100$)

Mechanism	Single	Sequence	Multiple
1st price	74.889	69.233	10.087
	(5.946)	(6.333)	(2.838)
2nd price	73.767	68.191	9.303
	(6.006)	(6.381)	(2.842)
Externality	82.087	77.765	5.701
	(12.595)	(12.188)	(2.272)

As for payment mechanisms, as expected, the second price auction outperforms the first price auction slightly. Counterintuitively, in the single lane winner setups, the externality payment mechanism performs substantially worse than the alternatives, but it shows its true potential in the multiple dispatch setup, markedly outperforming both first and second price mechanisms for the lowest average time consumed across all configurations. This may be attributable to multiple dispatch being the only setup that can express a significant distinction between the winning, first losing, and everyone else lane sets that form the crux of the externality payment mechanism.

4.5.2 One bad actor experiments

As a supplement to the baseline comparison experiments, I conducted a set of experiments where one vehicle lies about its valuation to see if the features intended to encourage truthful reporting are working as intended.

Hypothesis: Vehicles in first price payment mechanism will show the most benefit from under-reporting their VOTs, followed by the second price payment, and finally the externality payment mechanism.

Experimental setup: The same settings from the experiments in the last subsection apply, with two additional adjustments.

First, because of the substantially better performance in the multiple dispatch auction, I focus this experiment set on that setup only.

Second, in this experiment, one vehicle in a traffic profile is chosen and its true VOT scaled down to some proportion of its original value to create its reported VOT. The NAAIMS instance is then run twice: once with the original VOT, and once with the downscaled VOT, with all else held identical (including the arrival times and characteristics of other vehicles in

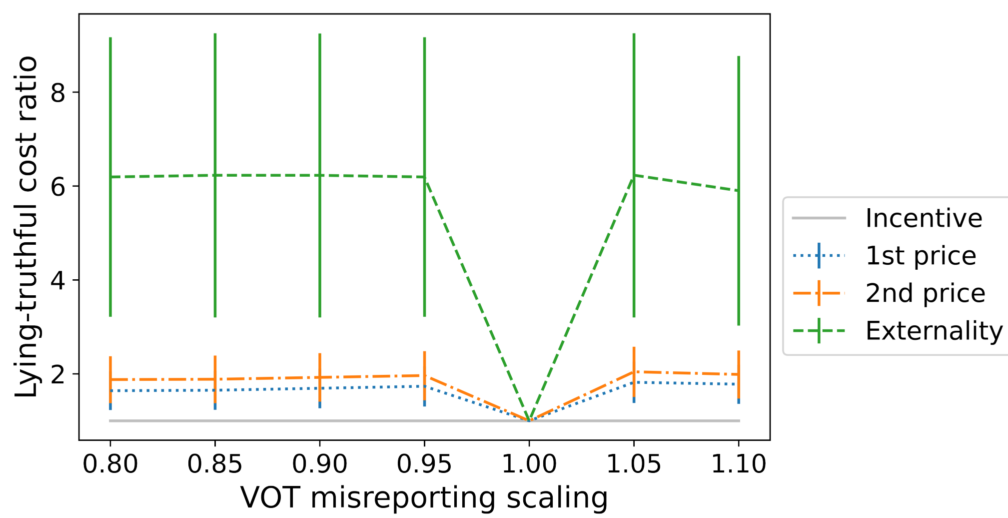

Figure 4.4: Reported VOT as a proportion of true VOT versus the ratio between the cost incurred in the misreporting case and the true case.

the simulation instance). The cost incurred by the chosen vehicle is compared across the two instances as a ratio to see to what extent misreporting its true VOT benefited it, if at all. Smaller values are better for the misreporting vehicle and worse for the payment mechanism, especially if less than 1 as that implies that the vehicle experiences less cost when misreporting their VOT than when reporting their true VOT.

Results: As you can see in Figure 4.4, we observe that vehicles don't benefit from misreporting their VOT under any payment mechanism regardless of whether they under-report or over-report. That said, they experience stronger penalties for misreporting with second price payments than first price payments, as we expect, and much more with externality payments than second price. Although these results should be taken with a large grain of salt due to the wide margins of error and the fact that these results don't consider a strategizing equilibrium, this along with the results from the direct comparison experiments are a good indicator that the externality payment mechanism is an improvement on the first and second price payment mechanisms.

4.6 Desiderata achieved and remaining challenges

This mechanisms described in this chapter addresses most, but not all of the desiderata listed in subsection 4.1.1, and does so more comprehensively than the previously defined first- and second-price auctions. Specifically, it incorporates some additional features designed to make it more optimal, strategy-proof, and fair, correcting some issues with the other auctions, but it does not do so in a way meant to be mathematically provable. In short, it compromises on desiderata 4 and 5 and possibly 6, but less so than its predecessors, in order to achieve throughput closer to intersection policies not considering valuations.

4.6.1 Aside: the exposure problem

Consider again the example setup in Figure 4.1, where all movement times are identical, there are no new arrivals (realized or expected) to consider, and $\nu_C > \nu_A > \nu_B > 0$ but $\nu_A + \nu_B > \nu_C$.

If B chose to lie and report a VOT of 0, C would win the first auction, followed by the movements of A and then B, making B move third.

If B truthfully reported it VOT for bidding, A and B both contribute to a winning bid that allows A to move before C. Under the sequenced auction setup, in the next step B would be lose to C and be forced to move third, just the same as before. The sequenced auction paradigm is strictly worse for B in this scenario, as they've subsidized the bid of A for no improvement in its delay.

Resolving this issue seems likely to require a fundamental redesign of our mechanisms and is thus considered out of scope for this chapter, but a good suggestion for future work.

4.6.2 Future work

Fully addressing the exposure problem will likely require a completely new auction design that directly bundles several auctions together instead of treating them mostly separately, with a floor at exit. This may go hand-in-hand with the creation of a provably optimal, incentive-compatible, and strategy proof mechanism for intersection auctions instead of the heuristic patchwork the first-, second-, and externality-price mechanisms described in this chapter.

On the front of the tools created for this dissertation, future work includes exploring the impacts of the auction mechanism on different intersection configurations. If extending the mechanism to further close the gap in throughput between FCFS and auctions, it'd be necessary to combine both multiple dispatch and externality into a single mechanism, as well as incorporating some additional streamlining of the auction simulation codebase to reduce runtime.

Chapter 5

Conclusion

Since the concept of AIM first came out in the early 2000s, autonomous intersection management has achieved some fame, maturing into a deep area of study at the crossroads of computer science and transportation engineering that I'm proud to contribute to with my dissertation.

In this thesis, I've developed a next-generation AIM simulation framework and implementation, NAAIMS, with compatibility, ease of development, and modularity as its foci. Using this simulation framework, I've developed a new AIM variant, stochastic automated intersection management, and further formalized an AIM subfield, intersection priority auctions, as both key contributions to the research field in their own rights and to demonstrate the utility and capability of NAAIMS.

Under the umbrella of stochastic AIM, this dissertation has developed the base framework for stochastic reservations and tilings, extended the concept of maximum tolerable crash incidence rate to both stochastic tiles and deterministic tile reservations, and sketched the viability of stochastic AIM as a topic for future research using simulated experiments. These empirical studies in NAAIMS use purpose-built model of stochastic vehicle movement informed by results from real human driving simulator experiments to show that guided, human-driven vehicles have the potential to bridge the gap between the status quo and the fully autonomous vehicle paradigm necessary for full realization of AIM's potential for maximizing throughput.

In the auction realm, I've formalized intersection priority auctions for individual vehicles into a consistent framework and refined first price and second price auction mechanisms to accommodate value of time bids and my new developments. Specifically, these refer to priority auctions that support dispatching from multiple lanes and sequences of vehicles in the same auction without the need for pre-planned signal phases, and modular payment mechanisms that allow for a new mechanism, externality payments, to improve upon the demonstrably flawed first- and second-price auctions and payment mechanisms. These auction configurations are tested for average value-weighted delay and incentive for misreporting, again using NAAIMS, further demonstrating the advantages of the new, modular strategy for simulating a wide scope of automated intersections.

5.1 Future Work

Given that the main focus of this dissertation was to implement a modular automated intersection management simulator to facilitate future study, there's a wide breadth of extendable research topics, much of which was already detailed in earlier chapters of this thesis.

Key avenues of future work in priority auctions come in the form of developing a more refined theoretical auction mechanism that is closer to provably optimal, strategy-proof, and incentive compatible that can avoid the exposure problem described in section 4.6.1. This new auction mechanism will likely be a stark departure from the framework detailed in Chapter 4 as it will need to bundle together sequences much more tightly than the externality payment mechanism does.

For auction research more directly tied to the configurations described in this dissertation, one marginal improvement would be to combine the multiple dispatch and sequenced auctions in order to reap the benefits in delay reduction from both approaches.

In the stochastic arena, the movement model described in Chapter 3, with its assumption of independence between, is intended only to be first sketch to demonstrate that stochastic reservations are potentially viable. They'll need a more realized movement model that incorporates information like correlated latitudinal and longitudinal deviation in order to fully characterize their benefits compared to traffic signals.

Our collaborators at the University of Washington are already preparing for a second round of driving simulator experiments informed by the results we've seen doing analysis for our NAAIMS experiments. Our feedback includes testing human drivers on a larger variety of acceleration profiles, and providing concrete guidance for lateral movement in addition to a pace car.

Finally, there are several research topics to explore within the NAAIMS framework itself, many of which already have their own API hooks in the NAAIMS codebase. A few key topics are bringing forward features from Dresner and Stone's AIM4 simulator into NAAIMS, such as support for multiple connected intersections, semi-autonomous vehicles, and traffic signals.

These goals are supplemented by extensions to the NAAIMS feature set that target more realistic kinematics, especially on turns and curves, exploring different intersection configurations and tiling shapes, applying AIM concepts to lane-changing between intersections.

The name of the game is to create the most fully featured AIM simulator for comprehensive study of a future for automated intersections' that's simply not the same.

Appendices

Appendix A

Soonest exit

The soonest exit problem involves finding the lowest time t and corresponding exit velocity v of a vehicle from a road lane. The base soonest exit scenario is quite simple, as it only involves accelerating to the speed limit and then staying there as it approaches the intersection, so the only question is whether the vehicle reaches the speed limit v_{max} before it reaches the end of the intersection.

Complications arise when the vehicle we're calculating the soonest exit for is not the first vehicle in its lane. Then, we need to know the exit time and velocity of the preceding vehicle before we can find the soonest exit time and velocity of our study vehicle. If the preceding vehicle is slower than the study vehicle, now the actions it might take are no longer limited to accelerating and staying at constant speed, but might also include braking as well.

Let $x(t, t_b)$ be the distance covered by the the study vehicle in time t, of which t_b is spent braking (with the rest spent either accelerating or at the speed limit), and $x_p(t)$ be the same for the preceding vehicle. t_{crit} will be the time at which the preceding vehicle reaches v_{max}.

Because of this, NAAIMS approaches the problem like this:

1. Find the fastest exit $x(t, 0)$ and check if it's the soonest exit, returning if $x(t_{\text{crit}}, 0) <= x_p(t_{\text{crit}})$.

2. Find the slowest reasonable exit $x(t, t_{\text{max}})$ (i.e., that doesn't require braking to 0 velocity and waiting) and check if it's still too fast and causes a collision by checking if $x(t_{\text{crit}})$ is greater than $x_p(t_{\text{crit}})$. If so, return invalid as the study vehicle cannot not collide with the preceding vehicle (something has gone wrong if this is the case). A.1

3. Binary search $x(t_{\text{crit}}, t_b)$ over t_b with direction indicated by $x_p(t_{\text{crit}}) - x(t_{\text{crit}}, b)$. Return the soonest exit found with error up to the length of a simulated timestep.

Most of the math required to find the soonest exit is trivial (simply accelerate and clip to the speed limit), but the exits that require any braking are complex enough to show the derivation of.

To do so, I'll summarize the relevant parameters before diving into the explanations. Note that I've slightly redefined x to be the fixed distance the study vehicle needs to cover to reach the intersection instead of a function of time, as we're finding these exits to compare with the preceding exit in the binary search.

v_0	starting velocity
v_e	exit velocity
a	acceleration
b	braking rate (defined to be negative)
t_b	time spent braking
t_a	time spent accelerating
t	total time to reach the intersection $t_a + t_b$
x_a	distance spent accelerating
x_b	distance spent braking
x	distance to cover $x_a + x_b$

A.1 Slowest exit (lower bound)

The slowest eligible exit accelerates to (and potentially stays at) the speed limit, before braking as much as it needs to just come to a complete stop at the intersection line.

Fix $v_e = 0$ and find t_a (and consequently t_b) w.r.t. t.

$$v_0 + at_a + b(t - t_a) = v_e$$

$$v_0 + at_a + bt - bt_a = 0$$

$$(a - b)t_a = -bt - v_0$$

$$t_a = bt + \frac{v_0}{b - a}$$

Next, setup the distance equation and solve for t.

$$\begin{aligned}
x_a &= v_0 t_a + \frac{1}{2} a t_a^2 \\
x_b &= (v_0 + a t_a)(t - t_a) + \frac{1}{2} b (t - t_a)^2 \\
x &= x_a + x_b \\
&= v_0 t_a + \frac{1}{2} a t_a^2 + (v_0 + a t_a)(t - t_a) + \frac{1}{2} b (t - t_a)^2 \\
&= a t t_a - \frac{1}{2} a t_a^2 + \frac{1}{2} b (t - t_a)^2 + t v_0 \\
&= a t \frac{bt + v_0}{b - a} - \frac{1}{2} a \left(\frac{bt + v_0}{b - a} \right)^2 + \frac{1}{2} b \left(t - \frac{bt + v_0}{b - a} \right)^2 + t v_0 \\
2(b - a)x &= a b t^2 + 2 b t v_0 + v_0^2 \\
0 &= (ab) t^2 + (2 b v_0) t + (v_0^2 - 2(b - a)x) \\
t &= \frac{-b v_0 - \sqrt{2b(b - a)(ax + \frac{1}{2} v_0^2)}}{ab}
\end{aligned}$$

A.2 Fixed t_b, free exit velocity

The other potential soonest exits that use braking are those on the slower side of the binary search, close to the slowest exit. They still accelerate/stay at the speed limit and then slow down, but they don't come to a complete stop before the intersection.

Much the setup is like before, but the target variable is redefined w.r.t. t_b and v_e isn't set to 0. Instead, v_e is solely dependent on t and the given values so we can proceed straight to the distance functions.

Start by finding t using the same formulas as last time redefined w.r.t.

the known t_b.

$$\begin{aligned}
x_a &= v_0(t - t_b) + \frac{1}{2}a(t - t_b)^2 \\
x_b &= (v_0 + a(t - t_b))t_b + \frac{1}{2}bt_b^2 \\
x &= x_a + x_b \\
&= v_0(t - t_b) + \frac{1}{2}a(t - t_b)^2 + (v_0 + a(t - t_b))t_b + \frac{1}{2}bt_b^2 \\
&= \frac{1}{2}at^2 - \frac{1}{2}at_b^2 + \frac{1}{2}bt_b^2 + tv_0 \\
t &= \frac{-v_0 + \sqrt{a^2t_b^2 - abt_b^2 + 2ax + v_0^2}}{a}
\end{aligned}$$

Now let's find v_e in terms of t using the substitution pattern from the last subsection.

$$v_0 + a(t - t_b) + bt_b = v_e$$

$$v_0 + at - at_b + bt_b = v_e$$

$$v_e = v_0 + at + (b - a)t_b$$

Appendix B

Stochastic deviations

The movement model takes the original time spent in the intersection and pumps it up or down by adjusting the time taken to accelerate to the speed limit.

B.1 Reach $v_{\max}$ before exit

Given the starting velocity, speed limit, distance to cover, and time to cover that distance in, find the time the vehicle needs to spend at the speed limit, the time the vehicle needs to spend accelerating, and at what acceleration value.

v_0	starting velocity
v_e	exit velocity
a_a	acceleration
t_a	time spent accelerating
t_e	time to exit in
x_e	distance to exit

$$
\begin{aligned}
x_e &= v_0 t_a + \frac{a_a}{2} t_a^2 + v_{\max}(t_e - t_a) \\
v_{\max} &= v_0 + a_a t_a \\
a_a &= \frac{v_{\max} - v_0}{t_a} \\
x_e &= v_0 t_a + \frac{v_{\max} - v_0}{2 t_a} t_a^2 + v_{\max}(t_e - t_a) \\
x_e &= v_0 t_a + \frac{v_{\max} - v_0}{2} t_a + v_{\max} t_e - v_{\max} t_a \\
x_e &= (v_0 + \frac{v_{\max} - v_0}{2} - v_{\max}) t_a + v_{\max} t_e \\
t_a &= \frac{x_e - v_{\max} t_e}{v0 + \frac{v_{\max} - v0}{2} - v_{\max}} \\
a_a &= \frac{v_{\max} - v_0}{t_a}
\end{aligned}
$$

Bibliography

[1] About auction strategies for intersection management when human-driven and autonomous vehicles coexist. *Multimedia Tools and Applications*, 80:15921–15936, 2021.

[2] Mojtaba Abdolmaleki, Yafeng Yin, and Neda Masoud. A unifying graph-coloring approach for intersection control in a connected and automated vehicle environment. *SSRN*, 2021.

[3] Tzu-Chiu Au, Shun Zhang, and Peter Stone. Autonomous intersection management for semi-autonomous vehicles. In *Routledge Handbook of Transportation*, pages 116–132. Routledge, 2016.

[4] Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Traffic lights with auction-based controllers: Algorithms and real-world data, 2017.

[5] Gulsevi Basar and Mecit Cetin. Auction-based tolling systems in a connected and automated vehicles environment: Public opinion and implications for toll revenue and capacity utilization. *Transportation Research. Part C, Emerging Technologies*, 81:268–285, 2017.

[6] Martin Beckmann, Charles B McGuire, and Christopher B Winsten. Studies in the economics of transportation. Technical report, 1956.

[7] Giacomo Cabri, Luca Gherardini, and Manuela Montangero. Auction-based crossings management. In *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, GoodTechs '19, page 183–188, New York, NY, USA, 2019. Association for Computing Machinery.

[8] Dustin Carlino, Stephen D Boyles, and Peter Stone. Auction-based autonomous intersection management. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 529–534. IEEE, 2013.

[9] Lei Chen and Cristofer Englund. Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, 2016.

[10] Xiangdong Chen, Meng Li, Xi Lin, Yafeng Yin, and Fang Hed. Rhythmic control of automated traffic—part i: Concept and properties at isolated intersections. *Transportation Science*, 55(5):969–987, 2021.

[11] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[12] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery.

[13] Richard W Denney, Eddie Curtis, and Paul Olson. The national traffic signal report card. *ITE Journal*, 82(6):22–26, 2012.

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

[15] Kurt Dresner and Peter Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 2 of *AAMAS '04*, pages 530–537. IEEE Computer Society, 2004.

[16] Yiheng Feng, K. Larry Head, Shayan Khoshmagham, and Mehdi Zamanipour. A real-time adaptive signal control in a connected vehicle environment. *Transportation Research. Part C, Emerging Technologies*, 55:460–473, 2015.

[17] Yiheng Feng, Chunhui Yu, and Henry X Liu. Spatiotemporal intersection control in a connected and automated vehicle environment. *Transportation Research. Part C, Emerging Technologies*, 89(C):364–383, 2018.

[18] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[19] Qing He, K. Larry Head, and Jun Ding. Pamscod: Platoon-based arterial multi-modal signal control with online data. *Transportation Research.*

Part C, Emerging Technologies, 20(1):164–184, 2012.

[20] B. G Heydecker. Objectives, stimulus and feedback in signal control of road traffic. *Journal of Intelligent Transportation Systems*, 8(2):63–76, 2004.

[21] L.L. Hoberock. A survey of longitudinal acceleration comfort studies in ground transportation vehicles, 1976-07.

[22] Steven Hwang, Ashis G Banerjee, and Linda Ng Boyle. Predicting driver's transition time to a secondary task given an in-vehicle alert. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–7, 2020.

[23] S Ilgin Guler, Monica Menendez, and Linus Meier. Using connected vehicle technology to improve the efficiency of intersections. *Transportation Research. Part C, Emerging Technologies*, 46:121–131, 2014.

[24] Christian P. Janssen, Linda Ng Boyle, Wendy Ju, Andreas Riener, and Ignacio Alvarez. Agents, environments, scenarios: A framework for examining models and simulations of human-vehicle interaction. *Transportation Research Interdisciplinary Perspectives*, 8:100214, 2020.

[25] Michael W Levin and Stephen D Boyles. Intersection auctions and reservation-based control in dynamic traffic assignment. *Transportation Research Record*, 2497(1):35–44, 2015.

[26] Michael W Levin, Stephen D Boyles, and Rahul Patel. Paradoxes of reservation-based intersection controls in traffic networks. *Transportation Research. Part A, Policy and Practice*, 90:14–25, 2016.

[27] Michael W Levin, Hagen Fritz, and Stephen D Boyles. On optimizing reservation-based intersection controls. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):505–515, 2017.

[28] Michael W Levin and David Rey. Conflict-point formulation of intersection control for autonomous vehicles. *Transportation Research. Part C, Emerging Technologies*, 85:528–547, 2017.

[29] Ning Li and Linda Ng Boyle. Allocation of driver attention for varying in-vehicle system modalities. *Human Factors*, 62(8):1349–1364, 2020. PMID: 31887066.

[30] Zhuofei Li, Lily Elefteriadou, and Sanjay Ranka. Signal control optimization for automated vehicles at isolated signalized intersections. *Transportation Research. Part C, Emerging Technologies*, 49:1–18, 2014.

[31] Xi Lin, Meng Li, Zuo-Jun Max Shen, Yafeng Yin, and Fang He. Rhythmic control of automated traffic—part ii: Grid network rhythm and online routing. *Transportation Science*, 55(5):988–1009, 2021.

[32] Gongyuan Lu, Yu(Marco) Nie, Xiaobo Liu, and Denghui Li. Trajectory-based traffic management inside an autonomous vehicle zone. *Transportation Research. Part B: Methodological*, 120:76–98, 2019.

[33] National Highway Traffic Safety Administration. Preliminary statement of policy concerning automated vehicles. Technical report, NHTSA, Washington, DC, 2013.

[34] New York State Department of Transportation. AVERAGE ACCIDENT RATES FOR STATE HIGHWAYS BY FACILITY TYPE (BASED ON ACCIDENT DATA January 1, 2015 TO December 31, 2016). 2016.

[35] M Quinlan, Tsz-Chiu Au, J Zhu, N Stiurca, and P Stone. Bringing simulation to life: A mixed reality autonomous intersection. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6083–6088. IEEE, 2010.

[36] H Schepperle and K Bohm. Auction-based traffic management: Towards effective concurrent utilization of road intersections. In *2008 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, pages 105–112. IEEE, 2008.

[37] Heiko Schepperle and Klemens Böhm. Agent-based traffic control using auctions. In *Cooperative Information Agents XI*, Lecture Notes in Computer Science, pages 119–133. Springer Berlin Heidelberg, Berlin, Heidelberg.

[38] Guni Sharon and Peter Stone. A protocol for mixed autonomous and human-operated vehicles at intersections. In *Autonomous Agents and*

Multiagent Systems, Lecture Notes in Computer Science, pages 151–167, Cham, 2017. Springer International Publishing.

[39] Richard H Thaler. Anomalies: The winner's curse. *The Journal of Economic Perspectives*, 2(1):191–202, 1988.

[40] The National Advanced Driving Simulator. minisim. 2020.

[41] Diana Toader. Aim4 patch for collision avoidance. University of Texas at Austin Computer Science, March 2017.

[42] Mark VanMiddlesworth, Kurt Dresner, and Peter Stone. Replacing the stop sign: Unmanaged intersection control for autonomous vehicles. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1413–1416, 2008.

[43] Samir Varma and Daniel Safarti. System and method for traffic decongestion, Jul 2020.

[44] Matteo Vasirani and Sascha Ossowski. Accommodating driver preferences in reservation-based urban traffic management. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, volume 1 of *AAMAS '10*, page 1379–1380, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[45] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1), 1961.

[46] Fo Vo Webster. Traffic signal settings. Technical report, 1958.

[47] C.K Wong and S.C Wong. Lane-based optimization of signal timings for isolated junctions. *Transportation Research. Part B: Methodological*, 37(1):63–84, 2003.

[48] Yafeng Yin. Robust optimal traffic signal timing. *Transportation Research. Part B: Methodological*, 42(10):911–924, 2008.

[49] Chunhui Yu, Yiheng Feng, Henry X Liu, Wanjing Ma, and Xiaoguang Yang. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation Research. Part B: Methodological*, 112(C):89–112, 2018.

[50] Chunhui Yu, Weili Sun, Henry X Liu, and Xiaoguang Yang. Managing connected and automated vehicles at isolated intersections: From reservation- to optimization-based methods. *Transportation Research. Part B: Methodological*, 122(C):416–435, 2019.

[51] Roman Zakharenko. Traffic priority mechanisms. *Transportation Science*, 54(5):1211 – 1224, 2020.

Vita

After completing a B.S. and M.S. in civil and environmental engineering at UC Berkeley, Carlin Liao joined Dr. Boyles's group in Fall 2018. His past work includes development of city-scale modeling, traffic assignment, and visualization software as well as the study of food, energy, and water systems. Since joining The University of Texas at Austin, he has earned a M.S. in statistics and refined his research interests to focus on black-box optimization of traffic models and the development of novel approaches for automated intersection management.

Between degrees, Carlin has worked as a data scientist and machine learning engineer for Apple, co-founded the Cal Hacks Foundation as the chief financial officer, and developed a full stack tool for analyzing traffic impacts as an R&D contractor for Arup.

Permanent address: Ernest Cockrell, Jr. Hall (ECJ) 6.202
301 E. Dean Keeton St. Stop C1761
Austin, TX 78712-1172

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.