INTEGRATION OF MICROSIMULATION AND OPTIMIZED AUTONOMOUS INTERSECTION MANAGEMENT

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL AT
THE UNIVERSITY OF MINNESOTA
BY

JACK OLSSON

IN PARTAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

DECEMBER 2018

ACKNOWLEDGEMENT

I would like to thank several people for supporting me in my endeavor of not only completing this master’s thesis, but also my master’s degree. First of all, thank you to my advisor, Dr. Michael Levin, for his guidance and sharing of knowledge and for introducing me to this fascinating research topic. Secondly, to my friends who have listened to and sympathized with my struggles and celebrated my triumphs throughout my research progress. More specifically to Pramesh Kumar, for his invaluable knowledge and teaching of the use of Gurobi, and to Rongsheng Chen, for his expertise in Aimsun’s API. I sincerely appreciate these two specific individuals’ time and talents that they have offered to me. Lastly, to my family for always encouraging and supporting my education and academic endeavors, and to my fiancé, Kelsie Valish, for being there through all of it.

DEDICATION

I dedicate this thesis work to my grandfather, John E. Olsson, who instilled in me the importance of education, supported me through it, and also served as the perfect role model as to what a hard-working person should be.

ABSTRACT

As the acceptance and implementation of autonomous vehicle technology continues to grow, further research into autonomous intersection management has become imperative. Autonomous intersection management is a type of intersection control for autonomous vehicles which eliminates the need for a traffic signal by using vehicle-to-infrastructure communication. Vehicles communicate information to an intersection manager which includes their turning intentions, and the intersection manager determines vehicle ordering and spacing such that vehicles can pass safely and efficiently through the intersection. Specifically, reservation-based autonomous intersection management, which give vehicles space-time path reservations through an intersection, has the potential to greatly increase the capacity of intersections by allowing an intersection controller to optimize the path that each vehicle takes. In this study, a mixed-integer linear program is proposed which expands on previous studies' models by giving the intersection manager more flexibility through optimizing vehicle acceleration and velocity through the intersection.

In addition, this new model was integrated with the microsimulation software Aimsun to simulate intersections under varying conditions. By utilizing Aimsun's Application Programming Interface (API), the reservation-based intersection control model developed in this study was implemented into the software. Then, various scenarios were simulated which included fluctuating the vehicle demands, altering the permitted vehicle accelerations and speeds, and modifying the safety buffer between vehicles.

The results indicate that the model proposed in this study has the capability to reduce delay and increase average speed experienced by vehicles at an intersection for different scenarios. Overall, the results show that the model offers several improvements to the existing reservation-based intersection control formulations and conventional signal controls. It has promising practical applications too, with its integration with the popular microsimulation tool Aimsun. Moving forward, this study's model and formulation will be valuable tools in implementing autonomous intersection management, especially when used in conjunction with a microsimulation program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1 INTRODUCTION

1.1 Topic Introduction

One of the hottest topics today in the transportation industry is autonomous vehicles. As the testing and use of autonomous vehicles on the country's roads has already begun, the further growth of autonomous vehicle technology is imminent. More specifically, many car manufacturers have implemented autonomous driving technologies like lane-keep assist, adaptive cruise control, and park assist. Furthermore, companies like Uber and Google have begun testing driverless vehicles in large metropolitan cities across the nation. The potential improvements they can bring to society make further research and development of this technology imperative. Some of these are improvements in safety, accessibility, and efficiency. Since connected autonomous vehicles have the capability of communicating with each other and with infrastructure, this allows for unprecedented flexibility in how vehicles operate in traffic. To accomplish this, an intersection manager can be utilized which can receive messages from vehicles that can include information like turning intentions, vehicle speed and location, and vehicle limitations such as maximum speed or acceleration. Then, the intersection manager can use this information to determine vehicle ordering and spacing and assign "reservations" to vehicles to pass through the intersection at specific times to ensure a safe traversal.

One innovative research topic in the crossover field of traffic engineering and autonomous vehicle technology is autonomous intersection management (AIM). AIM is an intriguing concept because vehicles can communicate with each other and the intersection manager to seamlessly pass through the intersection safely without using a traffic signal. Reservation-based AIM is one type of this traffic control which focuses on assigning vehicles non-conflicting space-time path reservations through the intersection. This technology has the potential to greatly increase the efficiency of intersections by allowing an intersection controller to optimize the path that each vehicle takes.

Additionally, AIM brings significant safety improvements to the public by removing potential human error when navigating through intersections. Intersections are hotspots for collisions due to the high number of conflict points, but by allowing a flawless intersection controller to dictate where vehicles should maneuver through an intersection, the risk for collisions is significantly reduced. In the big picture of AIM, time spent waiting at intersections may be mostly eliminated which can lead to enormous time savings. Further improvements in efficiency are brought by first and foremost researching more effective methods of this intersection control.

1.2 Problem Statement and Objectives of Study

Since AIM does not exist in real-world locations today, microsimulation provides the best means to practically compare and contrast its effects with variable traffic conditions compared to both existing models of reservation-based AIM and conventional intersection control. To obtain the greatest benefits from reservation-based AIM, it is necessary to optimize it for actual traffic behavior. Traffic microsimulation software provides an opportunity to accurately model this intersection control and to examine the effects of the technology on various factors. However, current traffic simulation software does not have built-in capabilities to simply construct intersections with these controls. Most of these simulation programs do have capabilities to use application programming interfaces (APIs) for detailed customization of the software. Therefore, the goal of this research project is to integrate AIM with traffic microsimulation software and to analyze the behavior of vehicles at these intersections under varying conditions compared to other forms of intersection control.

In order to do that, this study expands upon the AIM strategies developed in previous studies first from Dresner and Stone [1] and more recently from Levin and Rey [2]. The early studies of AIM utilized policies that were not optimization-based. For example, many studies used a “first-come first-served” policy which gave priority to pass through the intersection to whichever vehicle sent a request to the intersection manager first. However, this approach can lead to suboptimal results since a more efficient of vehicle ordering may exist that is not necessarily based on “first-come first-served”. Optimization of AIM, then, can lead to major improvements in delay for an entire section and for individual vehicles. In this study, the model gives the intersection manager more flexibility through optimizing vehicle acceleration through the intersection. This will allow the intersection controller to optimize vehicle trajectories based on their current position and desired turning movement while ensuring that vehicles can maneuver safely through the intersection. In current literature, there is no research on reservation-based AIM which allows the intersection controller so much freedom in controlling vehicles' speeds and accelerations as they pass through the intersection. In addition, by allowing the intersection manager to optimize acceleration, it will lead to better modeling in microsimulation.

In this study, an AIM formulation was developed as a mixed-integer linear program. This formulation was implemented in the popular microsimulation software Aimsun using the application programming interface. Then, a sensitivity analysis of the new AIM formulation was conducted and compared to two other forms of intersection control: a conventional signal and another form of AIM. Multiple scenarios were considered in the sensitivity analysis, and several parameters were varied during the evaluation. The results show how delay and average speed changed for the different forms of intersection control.

The technology presented in this research is not yet practice-ready as there are several aspects that need to be included to make it more implementable. However, this work serves as a stepping point to further develop autonomous intersection management – specifically for automobiles. More specifically, the research proposed in this plan has a two-fold contribution: providing a more flexible and improved AIM optimization model and allowing for more accurate testing of this intersection control which can bring the technology one step closer towards being a reality.

1.3 Thesis Organization

The remainder of the paper will be organized into several main sections.

Chapter 2 will provide the background and literature review of AIM on a general scale, the various policies and optimization models that can be used in AIM, and possible accommodations of human vehicles in AIM.

Chapter 3 will discuss the methodology which includes the mixed-integer linear program (MILP) that was developed for this study. More specifically, an explanation will be given of how the MILP was developed, what new benefits it provides compared to previous studies' models, and assumptions that were made in the formulation.

Chapter 4 will go into detail about the use of the application programming interface (API) that was utilized in Aimsun microsimulation to add the functionality of AIM to the program. A general overview of the Aimsun network will be provided, followed by an explanation of the coordination of Aimsun's API and the Gurobi optimization software. More specific details about the use of Gurobi in this study will be given, and the modeling choices and overall model development will be outlined.

In **Chapter 5**, the different test scenarios that were considered for the study will be summarized. This included sensitivity analyses which consider several measures of effectiveness (MOEs) and several parameters. Then, the results of the study will be presented based on these scenarios.

Lastly, **Chapter 6** will provide the conclusions that were drawn from the study as well as discuss the potential future work that could be extended from this work.

CHAPTER 2

2 BACKGROUND

Although AIM is a relatively new research subject, there is a surprisingly large scope of subtopics that have spawned from it. The topics relevant to this study can be broken down into four main categories: the background of AIM, the various policies of AIM, the optimization-based models of AIM, and the accommodation of human vehicles within AIM.

2.1 Background of Autonomous Intersection Management

While autonomous vehicle technology has not started gaining major support and advances until relatively recently, some research on AIM dates back at least twenty years [3]. However, this research mainly considered very general methods of autonomous vehicles crossing intersections according to predetermined strategies. The classic AIM was introduced by Dresner and Stone which proposed a “first-come first-served” (FCFS) reservation-based AIM in which cars acted as agents that could send and receive intersection reservations to and from an intersection manager [1]. However, this study did not consider the optimization of vehicle requests and only considered whether or not each request conflicted with an already assigned reservation.

After the introduction of AIM, Dresner and Stone also conducted a study which summarized the benefits and outcomes of reservation-based AIM, namely the reductions in delay that vehicles experience at an intersection which can be significantly reduced and even eliminated as the number of autonomous vehicles increases [4]. Furthermore, they detailed in another study how potential mechanical failures of an AIM system could be addressed, especially since AIM systems that are unequipped with failure protocol could be dangerous due to the automated nature of the vehicles [5].

Several studies also took AIM one step further to test its feasibility by taking a more practical approach. Quinlan et al. was able to test AIM in a realistic setting by creating a mixed-reality setting where an actual autonomous vehicle interacted with other autonomous vehicles that only existed in simulation [6]. Similarly, De La Fortelle and Qian discussed issues of AIM when it was used in real-world tests and suggested criteria for how to expand autonomous driving technology including safety, efficiency scalability, etc. [7].

In general, though, there are numerous studies which demonstrate the extreme improvements that AIM can provide when compared to either standard signalized intersections or stop-controlled intersections [1, 5, 8-18]. Fajardo et al. conducted an extensive study utilizing a custom simulation program to compare the performance of AIM versus signals considering many different factors like the number of left-turning vehicles and the distance of buffer between vehicles [8]. This study showed that the delay experienced by vehicles in AIM was significantly less than the delay at a standard traffic signal. Even in a real-life test, traditional traffic signals were outperformed by AIM [6].

Since the introduction of AIM, many different strategies and methods were developed in order to find a more optimal approach to AIM. Several different policies were developed in response to the FCFS, and beyond that, several optimization models were created to build upon the original strategy proposed for AIM. Over time, these concepts have become more practical due to the applicability of simulation software that can accurately model and simulate AIM.

2.2 Methods and Policies of Autonomous Intersection Management

After the introduction of the FCFS reservation-based model of AIM, several alternative AIM policies and methods have been studied. Many new policies built-upon the FCFS reservation-based policy originally created by Dresner and Stone by implementing several improvements to the original FCFS model to make it more realistic and more efficient [9-12, 19-23]. Various other methods have appeared as well that have taken a different approach to policies of AIM [13-14, 24-28].

Shortly after the concept of AIM was introduced, Dresner and Stone improved their FCFS model, which was originally very limited. Vehicles at autonomous intersections now had the capability to turn and accelerate, and it was not necessary that vehicles have any knowledge of the policy being used by the intersection manager [9]. Later, they completed a study which detailed several aspects that could be improved in AIM for both the intersection manager and for driver agents [19]. For example, intersection managers could provide a more delayed response when granting reservations to vehicles in order to provide a more optimal reservation for all vehicles. They could also give priority to certain vehicles or even treat the intersection as a market. For driver agents, vehicles could utilize the “intersection market” to bid for reservations and more flexibility could be given to vehicles for lane changing. Related to the suggestions from the study of Dresner and Stone, Au et al. actually considered how an intersection manager could receive batch reservations, or multiple reservation requests at one time, and how it could prioritize reservations based on vehicle type or wait times [22-23]. Additionally, Dresner and Stone extended the applicability of AIM to intersection control for low-volume roadways which requires no infrastructure (mechanism only relies on vehicle-to-vehicle communication) [12].

Other improvements to FCFS AIM have included extending its applicability to a multi-intersection network and introducing a planning-based model to reduce the chance that vehicles have to stop prior to entering the intersection [20-21]. Li et al. created an AIM model termed ‘ACUTA’ which expanded upon the FCFS model by allowing advanced stop location so vehicles could enter the intersection at higher speeds, non-deceleration zones – so vehicles did not have to stop if their reservation was rejected – and priority reservation [10]. This model of AIM was incorporated into the popular microsimulation software VISSIM to provide more reliable results than a custom simulation software when evaluating AIM against other conventional traffic control methods [11].

One study considered several paradoxes of reservation-based AIM [29]. Specifically, it noted and proved that signals could actually perform better than reservation-based AIM in certain situations – like when vehicles on local roads were prioritized over arterials. It found AIM performed better at symmetric intersections (i.e. downtown) versus asymmetric intersections (local/arterial). Additionally, reservation-based AIM could encourage selfish routing choices for drivers. This helped lead to the conclusion that more optimal policies of AIM than the FCFS policy could exist. In fact, Vasirani and Ossowski compared FCFS AIM to four other policies that were based on the adversarial queuing theory (AQT), which included longest-in-system, shortest-in-system, farthest-to-go, and nearest-to-source [30]. It found that, while FCFS was simpler and required less information about the vehicle, it was not as efficient as the other policies of AQT.

One popular AIM policy that has appeared more in literature is auction-based AIM [24-26]. The concept of implementing auctions into AIM was first proposed in a study by Carlino et al., and it was also tested in microsimulation [24]. Another study proposed a market-based mechanism at the intersection level in which the intersection manager assigns space-time slots to vehicles in order to pass through the intersection [25]. In two different studies, Schepperle and Böhm and Levin and Boyles implemented intersection auctions for reservation-based control with dynamic traffic assignment [26-27]. They modified tile-based AIM so that it could be applied to large networks. Furthermore, Levin compared intersection auctions to the traditional FCFS reservation-based control and found several benefits that were provided by auction control.

Other AIM methods include decentralized autonomous intersection management, vehicle back-off protocol, and implementation of a multiclass cell-transmission (CTM) model with consideration of autonomous intersections [13-14, 28]. The decentralized control system for AIM utilized the popular microsimulation software Aimsun paired with Matlab to look at benefits of AIM compared to standard signals, with a main focus in energy-use optimization [13]. Vehicle back-off protocol used vehicle messages and collision avoidance to control traffic within intersections and showed several benefits when modeled in simulation [14]. In the multiclass CTM model, tile-based reservation was combined with the legacy early method to model AIM for one intersection [28]. This study proved useful in determining the autonomous vehicle (AV) penetration rate at which AIM became more efficient than standard signal control.

While these methods and policies considered many different approaches to AIM, many opportunities still exist for improvements in efficiency. They do not consider the optimization of intersection control, which is a promising avenue in bringing more improvements to AIM. As demonstrated by Levin et al., several cases exist in which signals can outperform FCFS [29]. While the aforementioned studies showed that AIM has many benefits over standard intersection control, using optimization in AIM can still bring even more benefits and solve the situations in which AIM could be outperformed by signals.

2.3 Optimization Models of Autonomous Intersection Management

Studies concerning the optimization of intersection control started to appear shortly after the introduction of FCFS reservation-based AIM. Optimization of reservation-based AIM, optimization of time-space AIM models, and several other optimization methods are all present in literature. Even models that were derived from various intersection control policies utilize different methods of optimization.

A number of studies utilized the reservation-based approach proposed by Dresner and Stone combined with optimization algorithms in order to develop optimizable models of AIM [31-33]. De La Fortelle developed a reservation algorithm based on the FCFS reservation-based model in order to optimize the arrival time at the intersection for all reservation requests [31]. This is similar to the batch reservation policies [22-23] except that an optimization algorithm is also included in the model. Levin et al. developed an integer program optimization for tile-based intersections in order to optimize the order in which vehicles cross the intersection [32]. Additionally, this model was tested with two different objective functions (minimize travel time and minimize fuel consumption) using a transportation network model. Lastly, another study presented a centralized scheduling algorithm based on vehicle reservations, but also considered vehicle priority in the intersection control [33]. This model was tested in a self-developed simulation to demonstrate the efficiency of the optimized AIM model.

Tangent to the development of AIM and the development of optimization methods of AIM, an important relevant subtopic of intersection control considers the optimization of air traffic conflicts [34-36]. In an airspace, it is also necessary to optimize the crossing of aircraft in order to ensure safety and maximum efficiency. Chiang et al. utilized computational geometry and algorithms to route aircraft through an airspace full of conflict zones [34]. Another strategy used in studies on airspace conflict optimization is to utilize speed regulation and control combined with optimization [35-36]. Rey et al. used optimization to reduce the duration of and total number of conflict zones by adjusting and regulating aircraft speed. It was found that air conflicts could be resolved without adding much delay to individual aircrafts. The strategies used in air traffic optimization are relevant since they provide approaches that can also be used for AIM and are similar to the approach utilized in the optimized-AIM model in this study.

This leads to a similar strategy in AIM used by several papers which is concerned with the time-space optimization of vehicles at an intersection [2, 15-16, 37-43]. Two studies used an approach in which the intersection manager received vehicle information like speed, location, or acceleration and optimized the time that a vehicle should enter the intersection (or the path that it should take through the intersection) [37-38]. Another method called the "CVIC" model utilized an algorithm that found a safe maneuver for every vehicle through the intersection [15]. This model also included an algorithm that would take over in case of system failure. This study was further expanded to consider multiple intersection within the model [39]. Another approach was to apply model predictive control to optimizing AIM [16, 40]. This framework determined optimal trajectories or vehicle ordering with collision avoidance using an intersection manager considering speed limits and vehicle constraints. One study tested this framework with simulation in comparison to standard traffic signals and showed several improvements [16].

Wuthishuwong and Traechtler created a model that used a node reservation algorithm which assigned each vehicle a certain time to enter the intersection [41]. Additionally, a dynamic program in this study found the trajectory of each vehicle through the intersection with continuous movement and no collisions. Grégoire et al. proposed a mathematical framework for a continuous scheduling optimization in conjunction with vehicle priority for AIM [42]. This was done by assigning each vehicle path-velocity assignments through the intersection. Similarly, Altché and De La Fortelle utilized a mixed-integer linear program for AIM which determined velocity assignments as well as vehicle ordering through an intersection [43]. In this study, results of the model were also compared to the results from the FCFS policy. In a more recent study, a conflict-point formulation of AIM was proposed in which an intersection manager assigned time-space reservations to vehicles after mixed-integer linear program optimization [2]. This formulation was used as the foundation for the formulation that was derived in this paper's study. This type of optimization is similar to aircraft conflict resolution in that it ensures no conflict regions overlap within an intersection.

Other methods of AIM optimization have also been considered [17-18, 44]. Zohdy et al. took an approach of intersection management using cooperative adaptive cruise control, which showed major improvements in delay when compared to traffic lights [17]. In another study, polling-systems were used, in which vehicles are subject to differential constraints in conjunction with a coordination control algorithm [44]. Zhu and Ukkusuri developed a linear program based on traffic dynamics for AIM which included a bi-level optimization combining system optimal dynamic traffic assignment and AIM [18].

The methods used in the optimization of intersection control are wide-ranging. While different approaches can be used in solving AIM, advantages and disadvantages are present in each. The conflict-point formulation presented by Levin and Rey, which was expanded upon in this study, was flexible and extremely efficient in optimizing AIM. Additionally, while simulation was used frequently in analyzing the different policies and methods of intersection control, combining the use of microsimulation software with optimized AIM has not been done before.

2.4 AIM with Human Vehicles

While most studies concerning AIM are based on the assumption that all vehicles passing through a given intersection are fully autonomous, there is existing literature that actually considers methods of AIM that are accommodating of human vehicles. It will either take a significantly long period of time for all vehicles on the road to be autonomous or it may never fully happen since some people will choose to drive themselves due to monetary reasons or for the enjoyment of driving. Accordingly, it is important to consider and understand methods that can be utilized or integrated with methods of AIM so that both autonomous and human vehicles can be accommodated. Some of these methods are unrelated to AIM, but others are extensions of the classic AIM model.

One of these other approaches to AIM is contextualized traffic controlling [45]. This method of intersection control sends commands to vehicles when they have the right of way in an intersection. This is determined based on vehicle information that is sent to the controller which then tries to find the most optimal vehicle ordering. Alonso et al. considered a method in which autonomous vehicles received information about the intentions about human vehicles like position, speed, and turning intentions and then were able to make their own decisions about crossing the intersection [46]. Similarly, another study developed an approach in which in-vehicle virtual traffic lights notified the driver when they were able to pass through the intersection [47]. While these methods are not related to AIM, they propose concepts that bring more flexibility and would allow human vehicles to traverse through AIM intersections.

Other studies have tried to expand AIM specifically to be more accommodating of human vehicles [48-51]. In two separate studies, Dresner and Stone improved their AIM model by discussing how to incorporate human vehicles (with an emphasis on emergency vehicles) [48]. A method was presented in which AIM capabilities could be added to an intersection gradually as the penetration rate of AVs increased [49]. Another study was an extension of AIM that developed an algorithm with a space-time reservation scheme that worked with low percentages of non-autonomous vehicles [50]. Additionally, Qian et al. developed a priority-based coordination AIM system which supported legacy vehicles [51]. In this approach, legacy vehicles maintained a safe following distance from vehicles – allowing for human vehicles to be incorporated in AIM.

These studies that address considerations of human vehicles with AIM are relevant to this study because they demonstrate that methods exist that incorporate human vehicles into AIM. In this study, then, it is assumed that all vehicles are fully autonomous since one of the previous methods could be incorporated with the model in this study to allow for human vehicles. As the number of autonomous vehicles rises, though, the efficiency of AIM will also increase.

CHAPTER 3

3 METHODOLOGY

In this study, a mixed-integer linear program (MILP) was developed which was based on the formulation created by Levin and Rey [2]. The formulation proposed in this study, which will be referred to as AIM+, was based on a reservation scheme of AIM in which an intersection manager utilized a conflict-point formulation to assign vehicles a space-time reservation through the intersection. Each vehicle met the assigned reservation by modifying its speed to arrive at conflict points within the intersection at specific times. The assigned trajectories of all of the vehicles in the network were optimized in such a way that every vehicle could pass through the intersection collision-free and as efficiently as possible.

3.1 MILP Development

To improve the formulation presented by Levin and Rey, AIM+ gives the intersection manager more flexibility in assigning space-time reservations to vehicles by allowing the vehicles to pass through the intersection at accelerating speeds. In the previous model, vehicles passing through an intersection were only able to do so with a constant speed. By allowing vehicles to accelerate through the intersection, the intersection will operate more efficiently. Control of acceleration is also necessary to integrate AIM+ into a microsimulation environment.

3.1.1 MILP Assumptions

The development of this formulation was made on the basis of several assumptions. First of all, it was assumed that the vehicle would utilize a constant acceleration rate from its first conflict point to its last point. Conflict points are the points at which different vehicle paths intersect, merge, or diverge. By using the assumption of constant acceleration, the kinematic equations were able to be utilized in the formulation to determine vehicle acceleration. Equation (1) was used to relate acceleration to arrival time and distance traveled.

$$d = v_i t + \frac{1}{2} a t^2 \qquad (1)$$

Since this kinematic equation contains a quadratic term, and in order to keep the formulation as a linear program so that optimization software was easily able to solve the problem, the assumption was made that the initial velocity, v_i, is zero. This assumption only existed within the MILP formulation in order to ensure that the optimization formulation was solvable. Since it was not optimal to require vehicles to come to a stop immediately before entering the intersection when considering overall efficiency of the intersection, provisions were made by establishing the first conflict point at a spot several hundred feet upstream of the intersection. This conflict point was considered as the initial conflict point for each vehicle's path, but it served more so as an abstract point which signified where the vehicle should be at rest as opposed to being an actual point of crossing of vehicle paths. The vehicle would have sufficient distance and time before entering the intersection to accelerate to a higher speed. Further provisions were made within microsimulation so that each vehicle would not physically have to come to a stop in the simulation. This will be discussed more in-depth in **Chapter 4**. **Figure 1** illustrates the trajectories and conflict points for a four-leg intersection. The trajectories for the eastbound movements are shown in red, and the conflict points for the eastbound through movement in the left lane are numbered.

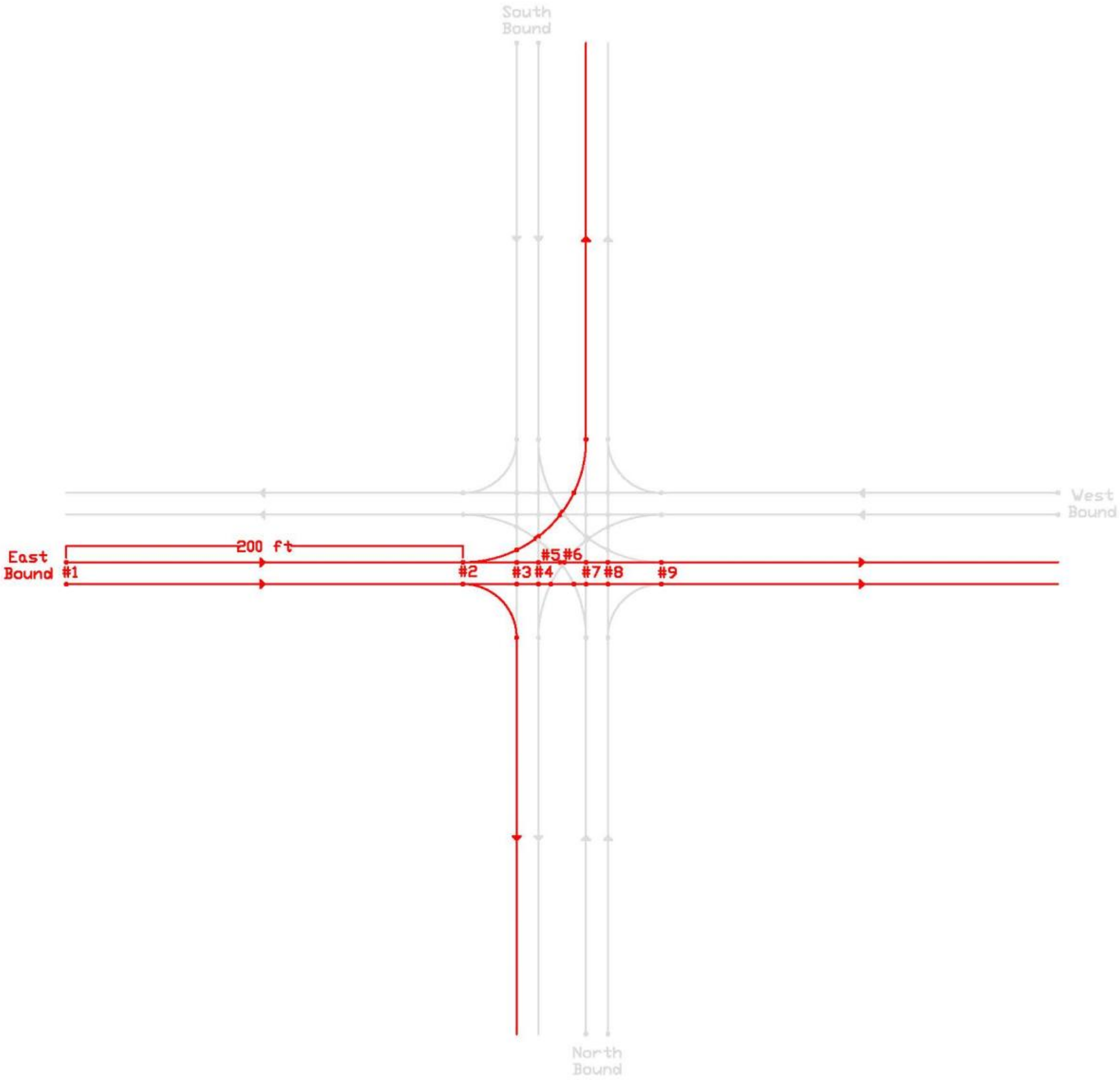


Figure 1, Intersection Conflict Points

It was also assumed that the acceleration and speed between the first and last conflict points of an intersection would be non-negative. This ensured that vehicles in the system would only accelerate on their trajectories.

Lastly, the assumption was made that autonomous vehicles utilizing the intersection controlled by AIM+ would not cause a collision. Since every vehicle has a list of specific conflict points, in the case of a malfunction, it's possible to determine if a vehicle has deviated from its path. The intersection manager can then take action to alert other vehicles. When considering the technology of connected autonomous vehicles, it is important to note that they are (or will be) capable of transmitting basic safety messages (BSMs) which include information like vehicle location and speed. These messages that are broadcasted by vehicles can be used to help eliminate collisions at intersections controlled by AIM. The National Highway Traffic Safety Administration has worked on implementing a policy requiring new vehicles to have the technology to broadcast these messages which can be emitted nearly every 0.1 seconds [52]. If vehicles are able to broadcast BSMs, an intersection manager can notice if a vehicle stops broadcasting messages for a certain time period (i.e. a malfunctioning vehicle does not send BSMs for 0.5 seconds). In the case of a vehicle malfunction in an intersection, the intersection manager can then send out an alert to tell all vehicles approaching the intersection to slow down and come to a stop to prevent any potential collisions. Any vehicles that are already in the intersection at the time of a malfunction can appropriately and safely accelerate and avoid the trajectory of the malfunctioning vehicle and finish exiting in the intersection. In fact, several studies have been conducted that consider how BSMs can be used to prevent collisions [53-54]. This technology would pair well with AIM to ensure maximum efficiency and maximum safety at intersections.

To go more in-depth, a half second can be utilized, as a generalization, as the point at which an intersection manager sends out an alert because it has not received a message from a vehicle that will be entering the intersection. In reality, this threshold may be more or less than half a second, but regardless of the number, kinematics can be used to describe how to ensure that all vehicles with potential conflicts with the malfunctioning vehicle can come to a stop safely. After 0.5 seconds of not receiving BSMs from a certain vehicle, the intersection manager will assume this vehicle has malfunctioned. Accordingly, it can then send a signal to all vehicles to slow down. Assuming that the malfunctioning vehicle has an initial velocity of 40 mph and will decelerate at 11.2 ft/s^2 (the value commonly used for vehicle deceleration by the American Association of State Highway and Transportation Officials), the time it takes that vehicle to come to rest can be calculated according to equation (2) below.

$$v_f = v_i + at \tag{2}$$

Therefore, it will take a malfunctioning vehicle approximately 5.2 seconds to come to rest. For all other vehicles, they will continue to travel at their initial speed of 40 mph for 0.5 seconds until alerted by the intersection manager to come to a stop. At that point, the vehicles will have approximately 4.7 seconds (5.2 seconds minus 0.5 seconds) to come to a stop. The required deceleration rate to accomplish this is 12.4 ft/s^2 which can also be calculated from equation (2). According to the National Cooperative Highway Research Program Report 600, mean deceleration rates of 17.7 ft/s^2 were attainable under good traction conditions [55]. While this rate is higher than the normal comfortable deceleration rate, it is still reasonable to assume that vehicles could utilize this deceleration rate in case of a malfunction at an intersection.

While this example is more general, it illustrates the capability of utilizing BSMs as a way to alert an intersection manager that a vehicle is malfunctioning. By knowing approximately where a vehicle has started malfunctioning, the intersection manager can know the trajectory of how the vehicle will come to rest. Accordingly, the intersection manager can calculate required deceleration rates for each vehicle so that they can also stop before crossing the path of the malfunctioned vehicle. While no collisions occur within the simulated scenarios in Aimsun, this example illustrates how BSMs and an intersection manager can work together to ensure that the intersection can still operate efficiently. However, it’s also important to consider situations where BSMs aren’t received by the intersection due to other reasons (i.e. temporary communication blockage). In these situations, using the current algorithm, the intersection manager would still assume that a vehicle has malfunctioned and a signal would be sent to vehicles. This issue would create cases where vehicles are falsely alarmed of a malfunction They would still begin to come to a stop and incur delay and vehicle speed would be affected. However, once the intersection manager has started receiving messages from the vehicle again, an alert can be sent back to vehicles to assign them new paths based on their current location and speed.

3.1.2 Notation

Several variables will be defined prior to introducing the constraints of the AIM+ MILP formulation. These variables can be found in **Table 1**.

Table 1, MILP Formulation Variables

$\mathcal{V}$	set of all vehicles in network	e_i	earliest time that vehicle i can reach initial conflict point
ρ_i	path vehicle i takes through intersection	$d_i(c_i^1, c_i^n)$	distance vehicle i travels between intersection entry point and conflict point n
c_i^0	Initial conflict point for vehicle i located some specified distance upstream of the intersection	D_i	width of vehicle i
c_i^1	intersection entry point for vehicle i	L_i	length of vehicle i
c_i^N	intersection exit point for vehicle i (last conflict point for vehicle i)	w	congested wave speed
c_i^n	nth conflict point on path of vehicle i	$\underline{a_i}$	minimum acceleration that vehicle i can travel through the intersection
$t_i(c_i^N)$	time at which vehicle i reaches intersection exit point	$\overline{a_i}$	maximum acceleration that vehicle i can travel through the intersection
$\tau_i(c_i^N)$	time it takes vehicle i to completely pass through intersection exit point	λ_i	equivalent to $\sqrt{\frac{1}{a_i}}$. Written as such to ensure linear constraints
$t_i(c_i^n)$	time at which vehicle i reaches conflict point c_i^n	$\underline{U_i}$	minimum velocity that vehicle i can travel through the intersection
$\tau_i(c_i^n)$	estimated time it takes vehicle i to pass through conflict point c_i^n	$\overline{U_i}$	maximum velocity that vehicle i can travel through the intersection
$t_i(c_i^1)$	time at which vehicle i reaches intersection entry point	$\delta_{ij}(c_i^n)$	binary variable denoting if vehicle i and vehicle j overlap at point c_i^n
$\tau_i(c_i^1)$	time it takes vehicle i to completely pass through intersection entry point	M_{ij}	large positive constant for separation constraint
f_i	priority or weight of each vehicle, where $f_i > 0$		

The key purpose of an intersection manager in AIM is to determine every vehicle's arrival time at each conflict point, because these must be assigned such that a smooth and collision-free traversal can be made through the intersection. Each vehicle's respective list of conflict points depended on its turning movement and approach. For example, a left-turning vehicle would have more conflict points than a right-turning vehicle, because a left-turning vehicle crosses the paths of through-moving vehicles. As previously mentioned, this list of conflict points includes the initial conflict point which was some specified distance upstream of the intersection. It should be noted that when $n = 0$, this refers to the initial conflict point, $n = 1$ refers to the intersection entry point, and $n = N$ refers to the intersection exit point (the last conflict point).

3.1.3 Decision Variables and Known Inputs

As discussed, the intersection manager determines the time of arrival for each vehicle i at each of its conflict points c_i^n – which is denoted by $t_i(c_i^n)$. This is one of the decision variables of the AIM+ formulation in addition to the acceleration term, λ_i, and the binary separation variable, δ_{ij}. The term $\tau_i(c_i^n)$, which defines the time required for vehicle i to pass through a conflict point c_i^n, is based on the decision variables and other known values of the formulation. These known variables include the distance between two conflict points, $d_i(c_i^{n-1}, c_i^n)$, the minimum vehicle acceleration, $\underline{a_i}$, and the minimum vehicle velocity, $\underline{U_i}$, and are constant exogenous values. The other variables of the formulation are directly output by Aimsun and will be discussed further in **Chapter 4**.

3.1.4 Vehicle Ordering

For every vehicle that was approaching an intersection, it was possible to calculate its earliest arrival time (e_i) to the intersection entry point based on parameters like location, speed, and acceleration. Accordingly, every vehicle $i \epsilon \mathcal{V}$ was not able to arrive at the intersection at any time earlier than its earliest arrival time. Therefore:

$$t_i(c_i^0) \geq e_i \quad \forall\, i \in \mathcal{V} \tag{3}$$

Furthermore, it was necessary to ensure correct vehicle ordering for vehicles that had the same initial conflict point (c_i^0). By doing this, it ensured that vehicles were not assigned arrival times that were earlier than the arrival times of vehicles in front of them in the same lane. This applied to all pairs of vehicles i and j in the set of vehicles $\mathcal{V}$ for every shared conflict point along each vehicle's path. If $e_i < e_j$, then vehicle j could not enter conflict point c_i^n until vehicle i had arrived at that point – denoted by $t_i(c_i^n)$ – and finished passing through the point – denoted by $\tau_i(c_i^n)$. A constraint was introduced which guaranteed this vehicle following order which essentially ensured that the first vehicle to enter a lane was the first vehicle to exit that lane and enter the intersection (like first-in first-out).

$$t_i(c_i^n) + \tau_i(c_i^n) \leq t_j\big(c_j^n\big) \quad \forall i,j \in \mathcal{V}: c_i^0 = c_j^0, e_i < e_j, \forall c_i^n \in \rho_i \cap \rho_j: n \in [1, N_i] \qquad (4)$$

3.1.5 Travel Time between Conflict Points

As a vehicle traversed its path between conflict points, its speed and acceleration were constrained so that it did not exceed a value that was reasonable for each unique vehicle. Each vehicle had a maximum velocity that was assigned based on the speed limit of the link and the intended turning movement of the vehicle. For example, vehicles making through movements at the intersection had higher maximum velocities than vehicles that were turning left or right. This was done to more accurately model turning movements at an intersection. In addition, each vehicle was assigned a minimum velocity so that its velocity at least reached a reasonable and appropriate velocity upon entering the intersection. The constraint on this velocity for every vehicle was derived from the definition of velocity:

$$\frac{2d_{i(c_i^0,c_i^N)}}{\overline{U_i}} \leq t_i(c_i^N) - t_i(c_i^0) \leq \frac{2d_{i(c_i^0,c_i^N)}}{\underline{U_i}} \quad \forall i \in \mathcal{V} \qquad (5)$$

The variables within the constraint were rearranged such that the decision variables $(t_i(c_i^n))$ were isolated from the remaining terms of the equation. Average velocity is $U_{ave} = \frac{d_{i(c_i^0,c_i^N)}}{t_i(c_i^N) - t_i(c_i^0)}$, the distance between the first conflict point and the last conflict point for vehicle i over the time it takes to travel between those two points. While this constraint, by definition, concerned the average velocity of the vehicle, this equation could be easily modified to be constrained by the maximum velocity. $\frac{\overline{U}}{2} = U_{ave}$ if acceleration is constant and the vehicle is starting from rest (at the initial conflict point). Combining the two equations $U_{ave} = \frac{\overline{U}}{2}$ and $U_{ave} = \frac{d_{i(}c_i^0, c_i^N)}{t_i(c_i^N) - t_i(c_i^0)}$ led to equation (5). The $\underline{U}$ value was constrained identically.

The vehicle acceleration was also constrained in a similar manner so that vehicles utilized a reasonable acceleration as they traversed through the intersection and so that the acceleration was within the capabilities of each individual vehicle. With the MILP assumption of zero initial velocity, constant acceleration can be related to distance and time by $a_i = \frac{d_{i(}c_i^0, c_i^n)}{(t_i(c_i^n) - t_i(c_i^0))^2}$. This created an issue since the acceleration term is quadratic in the decision variable of arrival time. To linearize the constraints, a substitution was made for the acceleration term so $\lambda_i = \sqrt{\frac{1}{a_i}}$ or $a_i = \frac{1}{\lambda_i^2}$. This allowed for constraining of acceleration for each vehicle by:

$$\sqrt{\frac{1}{\overline{a_i}}} \leq \lambda_i \ \leq \sqrt{\frac{1}{\underline{a_i}}} \quad \forall i \in \mathcal{V} \tag{6}$$

This solved the non-linearity issue since λ_i is a decision variable, and it is not a quadratic term in the constraint. Next, the λ_i term was constrained according to the kinematic equation (1) shown previously. The initial velocity term was assumed to be zero, and the acceleration term was substituted for λ_i to obtain the constraint:

$$t_i(c_i^n) - t_i(c_i^0) = \lambda_i \sqrt{2d_i\left(c_i^0,\ c_i^n\right)} \quad \forall\ i \in \mathcal{V}, \forall c_i^n \in \rho_i : n \in [1, N_i] \tag{7}$$

This constraint allowed the assigned acceleration for each vehicle to constrain the arrival time at conflict points based on the kinematic equation in equation (1). This constraint is still linear because the term in the square root (the distance) is a constant.

3.1.6 Time Passing through Conflict Points

Since vehicles do not pass through conflict points instantaneously, it was necessary to know how long it took each vehicle to fully pass through a conflict point. Then, a constraint was added to the formulation based on this to calculate the time that a vehicle occupies a conflict point. Two separate constraints were introduced to the formulation: one for the first conflict point where the vehicle was starting from rest and one for the remainder of the conflict points in a vehicle's path ρ_i.

The two constraints for the estimated passing time through a conflict point, $\tau_i(c_i^n)$, share two terms. This included a term based on the vehicle headway and a term based on kinematics which considered the actual time that a vehicle spent driving through a given point. The constraint for the estimated passing time through the initial conflict point is:

$$\tau_i(c_i^0) = \frac{L_i}{w} + \frac{L_i \lambda_i}{\sqrt{2 d_i(c_i^0, c_i^1)}} \quad \forall\, i \in \mathcal{V} \tag{8}$$

The first term of the equation considered the headway with the vehicle length, L_i, divided by the backwards wave speed, w. This term accounts for the reaction time headway between vehicles. Although autonomous vehicles do not require as large of a reaction time as human-driven vehicles need, a smaller reaction time headway between vehicles could still be needed. The second term was derived from the kinematic equation $v_f^2 = v_i^2 + 2ad$, where the acceleration variable was written in terms of λ_i. Since the initial velocity was assumed to be zero, the v_f term simplified to $\frac{\sqrt{2 d_i(c_i^0, c_i^1)}}{\lambda_i}$ where the distance describes the length between the initial conflict point and the intersection entry point. This was an estimate of the velocity of the vehicle during the passage through the initial conflict point since it was actually describing the velocity of the vehicle at the intersection entry point (c_i^1). The time it took to pass through the point is estimated by taking the length of the vehicle divided by this velocity. Thus, this was a slight underestimation of the vehicle's actual passing time through the initial conflict point but was sufficient for this application since this underestimation would only become an issue if the spacing between vehicles is significantly small.

A similar constraint was introduced to constrain the passing time of each vehicle through its remaining conflict points:

$$\tau_i(c_i^n) = \frac{L_i}{w} + \frac{L_i(t_i(c_i^n) - t_i(c_i^{n-1}))}{d_i(c_i^{n-1}, c_i^n)} + \frac{D_i}{2\underline{U_i}} \quad \forall\, i \in \mathcal{V}, \forall c_i^n \in \rho_i : n \in [1, N_i] \qquad (9)$$

The first term of this constraint is identical to that shown in equation (8). The estimated passing time through each conflict point in this constraint was determined in a different manner – based on the vehicle's speed when it arrived at that conflict point. In other words, the distance between the conflict point of interest and the previous conflict point divided by the time used by the vehicle to travel between those two points was used as the vehicle's velocity through the conflict point. This was a slight overestimation of the vehicle's passing time through the conflict point since the vehicle was actually accelerating at all times, but a constant velocity was used in the constraint. An overestimation, though, will never lead to a collision since vehicles are actually passing through a conflict point faster than what is expected. Then, the actual passing time was found by taking the vehicle length divided by this velocity (represented in terms of distance and time). In addition, the last term of the constraint added an additional buffer to the passing time. Since conflict points are only one-dimensional, when vehicles were occupying a conflict point, they were taking up some additional space based on their width. Accordingly, this last term took the vehicle's width divided by two (because a vehicles width is split halfway on each "side" of a conflict point). Then, this value was divided by the vehicle's minimum velocity to guarantee a sufficient time buffer at conflict points shared by multiple vehicles. Adding this term to the previous constraint was not necessary since it concerned only the initial conflict point on each vehicle's path. Since the initial conflict point is not actually a crossing point of trajectories, it was not necessary to add this buffer to account for the vehicle's width.

3.1.7 Conflict Point Separation

The binary decision variable $\delta_{ij}(c_i^n)$, which was already briefly mentioned, denoted whether vehicles i and j overlap at conflict point c_i^n. Each pair of vehicles that shared a conflict point was designated two decision variables δ_{ij} and δ_{ji}. If vehicle i arrived before vehicle j, then $\delta_{ij} = 1$ and $\delta_{ji} = 0$ (or vice versa). Additionally, if vehicle i arrived before vehicle j, $t_j(c_j^n) \geq t_i(c_i^n) + \tau_i(c_i^n)$ was necessary to prevent a collision. To handle the two cases, a large positive constant M_{ij} was introduced into the conflict point separation constraints:

$$t_i(c_i^n) + \tau_i(c_i^n) - t_j(c_j^n) \leq \left(1 - \delta_{ij}(c_i^n)\right) M_{ij} \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, \forall c_i^n \in \rho_i \cap \rho_j: n \in [1, \mathrm{N}_i] \qquad (10)$$

This constraint held for every pair of vehicles that did not have the same initial conflict point and for every conflict point that was shared in their paths. Furthermore, to guarantee this separation using the previous method, it was necessary to introduce another constraint concerning δ_{ij}:

$$\delta_{ij}(c_i^n) + \delta_{ji}(c_i^n) = 1 \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, i < j, \forall c_i^n \in \rho_i \cap \rho_j: n \in [1, \mathrm{N}_i] \qquad (11)$$

To ensure no vehicle overlapping, this constraint guaranteed that for every shared conflict point by a pair of vehicles (not including the initial conflict point), that vehicle i will arrive at the conflict point before vehicle j or vice versa.

Lastly, the decision variable δ_{ij} was established as a binary variable in the MILP formulation for every pair of vehicles that did not have the same initial conflict point and for every conflict point that was shared in their paths. Thus:

$$\delta_{ij}(c_i^n) \in \{0,1\} \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, \forall c_i^n \in \rho_i \cap \rho_j: n \in [1, \mathrm{N}_i] \qquad (12)$$

3.1.8 Objective Function

The objective function of this formulation decided how the intersection manager should prioritize the flow of traffic. Several objective functions could be implemented in this formulation such as maximizing throughput, ensuring FCFS, or minimizing delay. This was given by:

$$\min \sum_{i \in \mathcal{V}} f_i \left(t_i(c_i^N) + \tau_i(c_i^N)\right) \qquad (13)$$

In this equation, f_i denoted the priority or weight of each vehicle. For the purposes of this research, the objective function aimed to minimize the total time that vehicles spent driving in the intersection (or maximizing the throughput), so $f_i = 1$ for all vehicles. The value of f_i could change to other non-negative values like the earliest arrival time e_i to obtain a fairer optimization approach.

3.2 Mixed-Integer Linear Program

Considering the constraints and objective function that were previously outlined [equations 3-13], these were combined into one formulation to establish the mixed-integer linear program (below) for AIM+. This formulation was utilized for this study and eventually integrated into microsimulation software (which will be discussed in **Chapter 4)**.

$$\min \sum_{i \in \mathcal{V}} f_i \left(t_i\left(c_i^N\right) + \tau_i\left(c_i^N\right) \right) \tag{14}$$

$$s.t. \quad t_i\left(c_i^0\right) \geq e_i \quad \forall\, i \in \mathcal{V} \tag{15}$$

$$\tau_i\left(c_i^0\right) = \frac{L_i}{w} + \frac{L_i(\lambda_i)}{\sqrt{2d_i(c_i^0, c_i^1)}} \quad \forall\, i \in \mathcal{V} \tag{16}$$

$$\tau_i(c_i^n) = \frac{L_i}{w} + \frac{L_i(t_i\left(c_i^n\right) - t_i\left(c_i^{n-1}\right))}{d_i(c_i^{n-1}, c_i^n\,)} + \frac{D_i}{2\underline{U_i}} \quad \forall\, i \in \mathcal{V}, \forall c_i^n \in \rho_i : n \in [1, N_i] \tag{17}$$

$$t_i(c_i^n) - t_i\left(c_i^0\right) = \lambda_i \sqrt{2d_i\left(c_i^0,\, c_i^n\right)} \quad \forall\, i \in \mathcal{V}, \forall c_i^n \in \rho_i : n \in [1, N_i] \tag{18}$$

$$\sqrt{\frac{1}{\overline{a_i}}} \leq \lambda_i \ \leq \sqrt{\frac{1}{\underline{a_i}}} \quad \forall i \in \mathcal{V} \tag{19}$$

$$\frac{2d_{i(}c_i^0, c_i^N)}{\overline{U_i}} \leq t_i\left(c_i^N\right) - t_i\left(c_i^0\right) \leq \frac{2d_{i(}c_i^0, c_i^N)}{\underline{U_i}} \quad \forall i \in \mathcal{V} \tag{20}$$

$$t_i(c_i^n) + \tau_i(c_i^n) \leq t_j\left(c_j^n\right) \quad \forall i, j \in \mathcal{V}: c_i^0 = c_j^0, e_i < e_j, \forall c_i^n \in \rho_i \cap \rho_j : n \in [1, N_i] \tag{21}$$

$$t_i(c_i^n) + \tau_i(c_i^n) - t_j\left(c_j^n\right) \leq \left(1 - \delta_{ij}(c_i^n)\right) M_{ij} \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, \forall c_i^n \in \rho_i \cap \rho_j : n \in [1, N_i] \tag{22}$$

$$\delta_{ij}(c_i^n) + \delta_{ji}(c_i^n) = 1 \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, i < j, \forall c_i^n \in \rho_i \cap \rho_j : n \in [1, N_i] \tag{23}$$

$$\delta_{ij}(c_i^n) \in \{0,1\} \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, \forall c_i^n \in \rho_i \cap \rho_j : n \in [1, N_i] \tag{24}$$

The decision variables of the formulation were the arrival times for each vehicle at each of its conflict points $t_i(c_i^n)$, the binary vehicle separation variable $\delta_{ij}(c_i^n)$, and the acceleration term, λ_i. As previously mentioned, $\tau_i(c_i^n)$ was determined based on predefined inputs which included distances between conflict points, vehicle lengths, and the backwards wave speed. In order to implement this optimization formulation, coordination between Aimsun microsimulation and Gurobi was established.

CHAPTER 4

4 INTEGRATION WITH MICROSIMULATION

The second part of this study involved implementing the MILP into the microsimulation software Aimsun. Current microsimulation software does not come with built-in capabilities to simulate AIM, so the Aimsun Application Programming Interface (API) was used to integrate this new capability into the software. The Aimsun Next version of Aimsun was utilized for this study. Microsimulation was chosen as the analysis tool in this study due to its flexibility and robustness in providing accurate and in-depth results. Since AIM does not exist at intersections today, simulation software must be used to test its effectiveness.

4.1 Aimsun Intersection Details

The intersection that was utilized for this microsimulation case study was a symmetrical four-leg intersection. By considering a symmetrical intersection, the left-turn and right-turn trajectories could be modeled as quarter circles. This allowed for straight-forward calculation of conflict-point locations. However, AIM+ could be implemented for intersections with any geometry. Vehicle turning radii can be determined in conjunction with the lane spacing and intersection width in order to determine the location of the coordinates for all conflict points in any given intersection. Each approach of the intersection (henceforth notated as north, east, south, and west approaches) contained two lanes. The inner lane for each approach was used for left-turns and through movements (aka left-through). The outer lane for each approach was used for through movements (aka right-through) and right-turns. Additionally, the width of the lanes, the length of the incoming and outgoing links, and the width of the intersection were all chosen during the creation of the network.

Prior to calculating each vehicle’s arrival time at each of its conflict points, it was necessary to extract the location of each conflict point so that overlapping vehicle trajectories could be determined. A vehicle’s conflict points were determined based on two parameters:

vehicle approach (north, east, south, or west) and vehicle movement (left-turn, through-left, through-right, and right-turn). Knowing the vehicle turning radii, lane spacing, and intersection width allowed for the determination of all possible vehicle trajectories. The points where trajectories cross signify all of the conflict points for the intersection. Since the intersection was symmetrical, the coordinates of each conflict point could be determined using geometry. Knowing the approach and movement of a vehicle gave its trajectory, which was made up of a list of specific conflict-points (with known coordinates).

The distance between various conflict points was a known constant which was based on the intersection geometry. This was especially important since many of the constraints of the MILP required knowing the distance between either two consecutive conflict points or between the initial conflict point and any of the remaining conflict points. Once the intersection had been created, it was possible to establish the coordination between Aimsun and Gurobi.

4.2 Aimsun and Gurobi Coordination

To solve the proposed MILP, the optimization software Gurobi was utilized in this study. Gurobi Optimizer is a popular mathematical programming tool which has the capability to solve a wide range of optimization problems including linear programming, MILP, and mixed-integer quadratic programming (MIQP) [56]. The Python API of Gurobi was used in this study since the microsimulation software that was used in this study, Aimsun, also has a Python API.

As evidenced by the discussion of the methodology, several input parameters were necessary prior to running Gurobi to obtain an optimal solution. The majority of these inputs were output from Aimsun during a simulation, but some other parameters were chosen as a user-defined default value. The necessary input parameters for AIM+ into Gurobi for each vehicle i, as well as the source and the value of each parameter, are shown in **Table 2**.

Table 2, Gurobi Parameters

Parameter	Source	Value
Vehicle Identification	Aimsun API	Ascending integer number
Vehicle Length, L_i	Aimsun API	11.48 ft – 14.76 ft
Vehicle Width, D_i	Aimsun API	6.56 ft
Vehicle Approach	Aimsun API	0 (west), 1 (north), 2 (east), 3 (south)
Vehicle Movement	Aimsun API	0 (left), 1 (through-left), 2 (through-right), 3 (right)
Earliest Arrival Time, e_i	Aimsun API (based on a calculation using location and speed)	Decimal value dependent on simulation time, speed, and location
Minimum Velocity, $\underline{U_i}$	User-Defined	10 ft/s
Maximum Velocity, $\overline{U}_i$	Aimsun API	25 mph – 55 mph (lower during turn movements)
Minimum Acceleration, $\underline{a}_i$	User-Defined	1.5 ft/s^2
Maximum Acceleration, $\overline{a}_i$	Aimsun API	8.53 ft/s^2 – 11.15 ft/s^2
Backwards Wave Speed, w	User-Defined	∞ (since headway between AVs is negligible)

The Aimsun API for this study was developed in such a way that every time a vehicle entered the network, the values of the above parameters were exported and appended to a CSV file (referred to as “Aimsun Output”). This information, in turn, was used as an input by Gurobi. The Gurobi formulation was run once over a specified time interval. This calculated the optimal arrival time at each conflict point for each vehicle, considering all of the vehicles that had been output by Aimsun to the CSV file up to that point in time.

Gurobi then created three separate CSV files. One included the output information on each vehicle’s arrival time and passing time at each conflict point (referred to as the “Gurobi Output”). Another utilized this information to calculate and output the velocity of each vehicle at every time step (0.1 seconds) so that it arrived at each conflict point at the time specified by the Gurobi Output (this was referred to as “Time Step Velocities”). These time step velocities changed at a constant rate due to the constant acceleration, and the velocities took effect at some point upstream of the intersection. Lastly, the vehicle information originally output by Aimsun plus the Gurobi Output for a specific set of the vehicles considered in the Gurobi run were stored to a CSV file (referred to as "Previous Aimsun Output"). This set was made up of vehicles that were added to the Aimsun Output file shortly before Gurobi was run. They potentially would not enter the intersection before the next run of Gurobi which was determined based on the arrival times at each vehicle's initial conflict point specified in the Gurobi Output. The arrival times of this set of vehicles were set as a constraint for the following Gurobi run to ensure that vehicles from the next iteration were not assigned arrival times that conflicted with vehicles that already had path assignments.

Next, the Aimsun API was used to read in this information to modify the speed of the appropriate vehicle at the specified time step to ensure that each vehicle would pass through the intersection seamlessly and collision-free. **Figure 2** depicts a flowchart showing the progression described above between Aimsun and Gurobi.

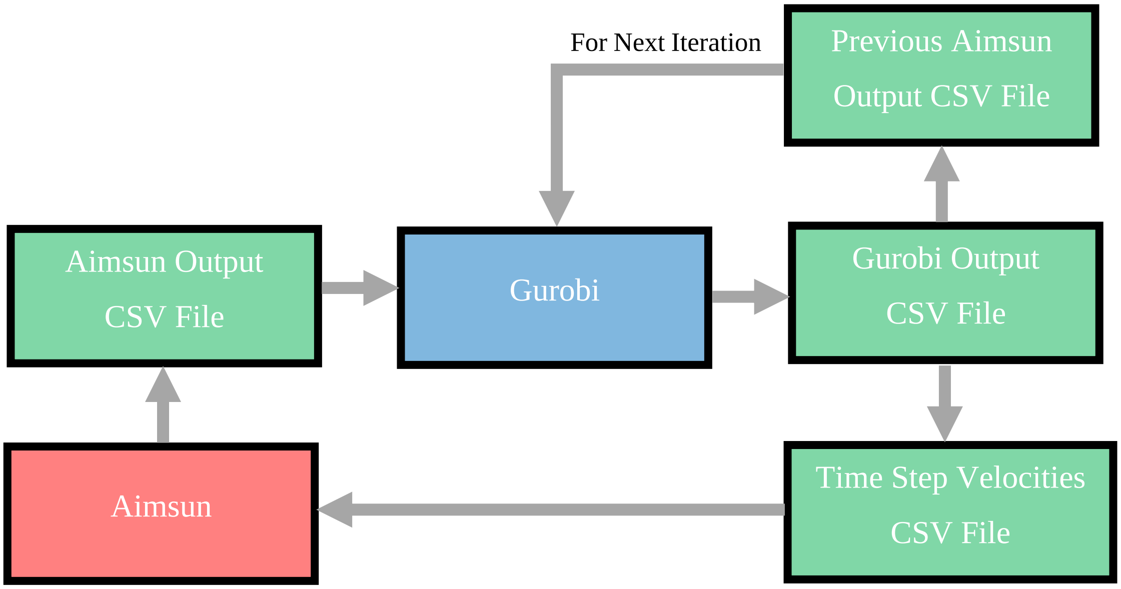


Figure 2, Aimsun-Gurobi Coordination Flowchart

4.3 Gurobi Specifications within Aimsun

The coordination between the Aimsun simulation software and the Gurobi optimizer required the establishment of several methods before any simulation trials could be run. In an ideal situation, Gurobi would be run once for a one-hour simulation in order to find the most optimal solution. However, a vehicle’s planned path may not be known to the intersection manager until it is within some specific distance. Accordingly, a rolling horizon approach was adopted for this study, where only a specific set of vehicles were considered within each run of Gurobi.

The rolling horizon approach used in this study was useful because it only considered vehicles that were within a certain arrival time of the intersection. It also maintained a reasonable computation time when looking for the optimal solution within Gurobi. While simulation is paused while looking for an optimal solution, it is important to have a reasonable computation time in order to make AIM+ more realistic since traffic cannot be paused in real life to look for a solution. In addition, the intersection manager used an “as-soon-as-possible” (ASAP) policy in which vehicles were assigned their time-space trajectory through the intersection as soon as they entered the link leading up to an intersection and Gurobi was run for the next iteration. This method was chosen over an “as-late-as-possible” (ALAP) policy, because Levin and Rey found that the benefits of using ALAP over ASAP were small and also required much higher computation times [2]. As mentioned in the previous section, a subset of the vehicles which already had their time-space trajectory assigned to them from the prior run of Gurobi had their path set as a constraint for the next run of Gurobi. That way, vehicles that entered links later did not overlap with already assigned paths.

In order to establish a balance between optimality and reasonable computation time, 20 seconds was chosen as the frequency with which Gurobi would be run during a simulation. A solution for each vehicle’s optimal trajectory, then, had to be determined prior to the vehicle needing to alter (raise or lower) its speed. Accordingly, the length of an incoming link to an intersection needed to be long enough so that there was adequate time for the intersection manager to obtain information about each incoming vehicle and run Gurobi at some multiple of 20 seconds. Every vehicle continued to drive towards the intersection throughout this time. The vehicles from the previous Gurobi run that were considered in the following run were selected based on their assigned arrival time at their initial conflict point in reference to the simulation time that Gurobi was last run. For example, for a Gurobi run that occurred at 10 minutes into the simulation, if a vehicle was assigned an arrival time at the intersection entry point of 10 minutes and 30 seconds, this gives a time difference of 30 seconds. For most of the scenarios considered in this study (which will be discussed further in **Chapter 5**), if this time difference was 35 seconds or more, then the vehicles arrival times would be considered as fixed values in the next run of Gurobi. This was chosen because vehicles that had their information obtained by the intersection manager more than five seconds before the next run of Gurobi would not have any conflicting trajectory with vehicles in the next run since they were sufficiently separated (both in distance and time). In general, this time threshold can be chosen by considering how much time it would take a vehicle to fully cross an intersection based on the intersection geometry (i.e. number of lanes that must be crossed) and the vehicle speed. For example, an intersection with only one lane per approach and a speed limit of 55 mph does not take much time for vehicles to cross. As such, a lower time threshold could be used. On the other hand, an intersection with four lanes per approach and a speed limit of only 35 mph would need a higher time threshold since vehicles will spend more time in the intersection.

4.4 Aimsun Modeling Choices and Assumptions

Several modeling choices and assumptions were made while constructing the intersection in Aimsun that was utilized in the analysis for this study. This included both the physical geometrical construction of the intersection and also the behavior of the vehicles in AIM+.

4.4.1 Intersection Construction Details

In this study, a symmetrical intersection was created to allow for simple extraction of conflict point coordinates. However, AIM+ can be extended to any intersection if the location of conflict points can be determined. **Figure 3** below shows the intersection created in Aimsun that was used for the analysis in this study.

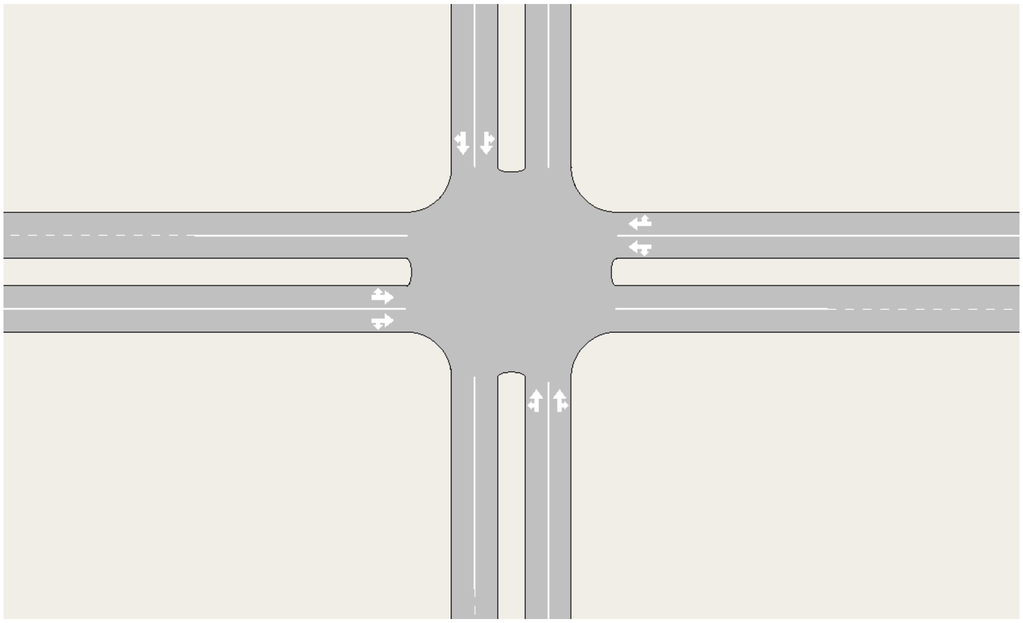


Figure 3, Aimsun Intersection

The width of the intersection was 100 feet with a left-turn radius of 62 feet, a right-turn radius of 27 feet, and a lane width of 11 feet. No turn-lanes were considered in this intersection. The standard incoming and outgoing link length used for the intersection was 2500 feet (or approximately a half-mile). This is similar to link/road lengths that would appear between intersections for a real road network. The coordinates of the conflict points for the intersection were found utilizing the geometry equations given in Levin and Rey [2]. One key difference was the introduction of the aforementioned initial "conflict-point" which was located 200 feet upstream of the intersection entry point for each lane of the intersection. As discussed in the previous chapter, this point served as the location where the vehicle was at-rest so that the MILP formulation could be maintained as a convex optimization problem. This location was chosen to be far enough back so that vehicles were able to accelerate to a reasonable speed before entering the intersection.

4.4.2 Incoming Link Vehicle Velocity

A vehicle's arrival time at each of its conflict points was based on its earliest arrival time (as discussed in **Chapter 3**). To calculate this parameter for each vehicle, it was necessary to know a vehicle's location and speed. Then, the time it took to arrive to the initial conflict point was determined by calculating the distance between its actual location and its initial conflict point and then dividing by its speed. Adding this value to the simulation time gave the earliest arrival time. Since AIM+ assumes 100% autonomous vehicles, it is plausible to assume that all vehicles approaching an intersection are approaching it at a constant velocity. Therefore, the intersection modeled in Aimsun assumed that vehicles were approaching the intersection at a constant velocity, which allowed for straightforward calculation of earliest arrival time. This assumption was valid for this model only until the vehicle changed its speed so that it could meet its intersection reservation at the correct time and speed. This constant velocity that the vehicle actually used was dependent on the speed limit for the roadway. This assumption could be easily relaxed in future work, but was a trivial matter concerning the scope of the research in this study.

4.4.3 Lane-Changing Behavior

In this Aimsun model of AIM+, the lane-changing behavior was simplified on the incoming links. All of the necessary vehicle lane-changing was assumed to occur prior to vehicles entering the links of the model. Since the conflict point arrival times that were assigned to vehicles were based on the vehicle’s location and speed, changing lanes could interfere with a vehicle meeting its assigned arrival on-time and at the correct speed. On the outgoing links from the intersection, vehicles were free to lane-change.

4.4.4 Vehicle Acceleration Provisions

As previously mentioned, the MILP formulation operated under the assumption that vehicles came to rest at the initial conflict point. However, this assumption is both unrealistic and undesirable for autonomous vehicles. Thus, provisions were made in the Aimsun model so that vehicles were not actually required to come to rest at the initial conflict point. All vehicles still had to meet their reservation within the intersection that was assigned by the intersection manager. However, as long as each vehicle arrived at its intersection entry point at the assigned time and velocity, the vehicle had some flexibility in modifying its speed so that it did not have to come to a stop upstream of the intersection at the initial conflict point.

To allow for this accommodation in the model, a piecewise acceleration function was determined for each vehicle. This function was based on each vehicle's initial constant velocity, its required intersection entry speed, and its assigned arrival times at the initial conflict point and the intersection entry point. The piecewise acceleration function prompted each vehicle to start decelerating at a constant rate at some specific distance upstream from the initial conflict point. After passing the initial conflict point, the vehicle began to accelerate at the constant rate specified by Gurobi. These changes from deceleration to acceleration near the initial conflict point were done so gradually, with no sudden jump in velocity. Utilizing these already known values mentioned above, the velocity value at which the decreasing vehicle velocity started to increase (represented by V_x) could be determined according to equation (25). V_i denotes the vehicle's initial constant velocity which was utilized until the vehicle began decelerating at the point d feet upstream of the intersection entry point, V_f denotes the vehicles final velocity upon accelerating and reaching the intersection entry point, and t^* represents the time of arrival at the deceleration starting point.

$$V_x = \frac{d - V_i\left(t_i\left(c_i^0\right) - t^*\right) - V_f\left(t_i\left(c_i^1\right) - t_i\left(c_i^0\right)\right)}{t_i\left(c_i^1\right) - t^*} \tag{25}$$

This acceleration function was used to determine the velocity for each vehicle at every time step starting at the distance d feet upstream from the intersection entry point until the vehicle entered the intersection, upon which its velocity was specified within the Time Step Velocities CSV file. As briefly mentioned, the time step within Aimsun was set to the minimum of 0.1 seconds in order to ensure that a vehicle's velocity was increased or decreased as gradually as possible. The upstream point d at which vehicles began to decelerate was generally chosen to be 1000 feet prior to the initial conflict point (which was 200 feet prior to the intersection entry point). This distance ensured that vehicles would not have to decelerate rapidly, and it also ensured that vehicles were not prompted to change their velocity before their path and time step velocities had been assigned by Gurobi. This meant that vehicles began to have their speed controlled by their assigned time-path 1200 feet before the intersection entry point. The incoming links to the intersection were long enough to ensure this was possible, and the speed of the vehicles was sufficiently low to guarantee that vehicles would not arrive to the 1200-foot upstream point before receiving an intersection reservation. **Figure 4** shows an example of how this piecewise acceleration could look, where $V_i = 40$ mph, $V_f = 30$ mph, and V_x was determined to be 25 mph.

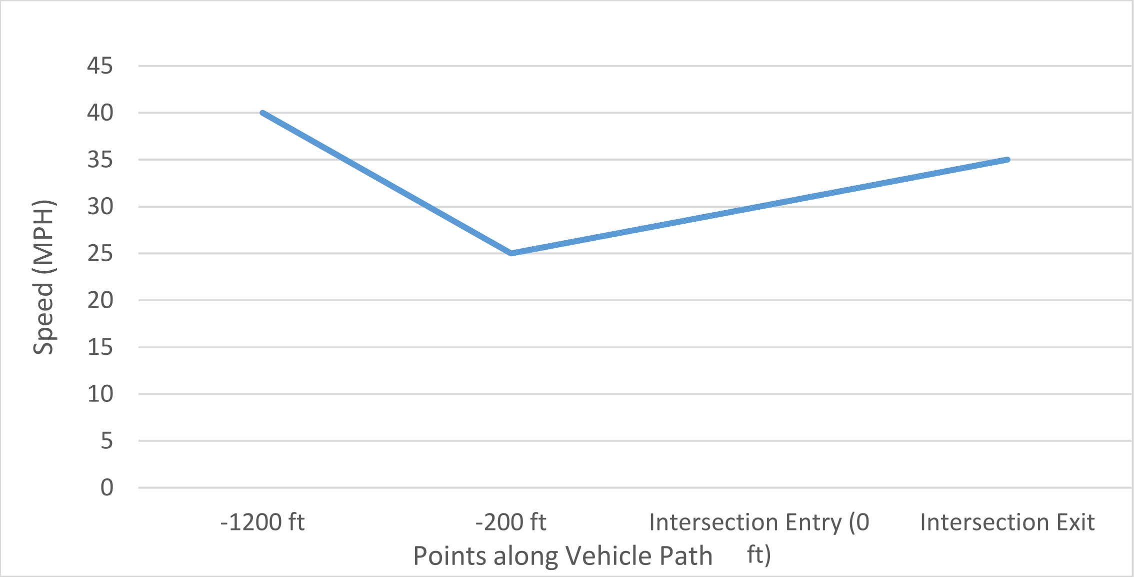


Figure 4, Piecewise Acceleration Function

The summarized purpose of this piecewise acceleration function was that each vehicle's velocity was altered every 0.1 second time step in a unique way so that it arrived at the intersection entry point at the correct time and speed according to the optimal solution found by Gurobi. This precise arrival at the intersection entry point was imperative to ensure a collision-free traversal through the intersection. The implementation of this acceleration function removed the need for having each vehicle come to a rest at the initial conflict point – which is much more realistic and desirable.

4.4.5 Aimsun Vehicle Static Parameters

Several static parameters for every vehicle class have default values when creating a network in Aimsun. These parameters ensure that vehicles obey the car-following model – which is quintessential in traffic microsimulation. However, when modeling autonomous vehicles in microsimulation, some of these parameters needed to be altered since autonomous vehicles have more flexibility when driving due to the reduced/negligible margin of driving error.

The two main parameters that were changed using the Aimsun API were the sensitivity factor and the minimum distance between vehicles. The sensitivity factor is a parameter that specifies how a following vehicle estimates the deceleration of a leader vehicle [57]. If a leading vehicle begins to decelerate, the following vehicle will also begin decelerating. Because autonomous vehicles have the capability of automatically knowing if a leading vehicle will begin decelerating, it is not necessary for a vehicle to start decelerating in advance on its own. Another related default vehicle parameter is the minimum distance between vehicles. This parameter prompts vehicles to automatically start decelerating if they are within a certain distance of another vehicle. Since autonomous vehicles do not require this strict spacing between each other, the minimum distance between vehicles can be lowered significantly.

CHAPTER 5

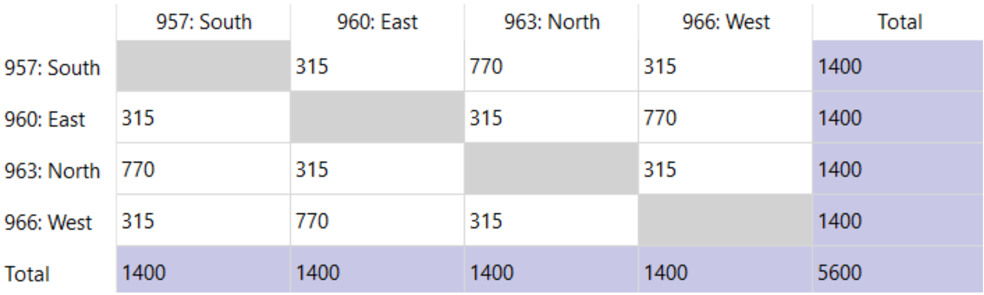
5 SENSITIVITY ANALYSES AND RESULTS

In this study, three forms of intersection control were evaluated and compared utilizing Aimsun: AIM+, AIM with FCFS, and signal-control. Several different measures of effectiveness (MOEs) were considered, and multiple parameters were varied during the scenarios in the analysis.

5.1 Aimsun Simulation Details

Each different scenario was made up of multiple runs (with a minimum of two), where the actual number depended on the resulting confidence interval. Aimsun automatically calculated the average values of different MOEs over all of the replications of each scenario. Each run was made up of 60 minutes of simulation, with one minute of warm-up time. This warm-up time was sufficient enough so that vehicles on each link were able to arrive to the intersection before the recording of the MOEs began.

The demand and the routing choices of each scenario was determined by using an origin-destination (OD) matrix. Since the intersection was made up of four incoming links and four outgoing links, four nodes were created which supplied the demand to all of the incoming links. Within the OD matrix, the number of vehicles traveling from each incoming link to each outgoing link was specified (which varied depending on the scenario). **Figure 5** shows an example of what this OD matrix looks like within Aimsun where the names specify the approach.



	957: South	960: East	963: North	966: West	Total
957: South		315	770	315	1400
960: East	315		315	770	1400
963: North	770	315		315	1400
966: West	315	770	315		1400
Total	1400	1400	1400	1400	5600

Figure 5, Aimsun OD Matrix

5.2 Analysis Scenario Measures of Effectiveness

Two main MOEs were used in the sensitivity analysis to determine how AIM+ affected these factors under varying situations. The MOEs used in the sensitivity analysis were:

- Average delay – measured in seconds or seconds/mile
- Average speed – measured in miles/hour (mph)

These MOEs were directly output by Aimsun after each simulation. Delay time per vehicle was calculated by taking the difference between the expected travel time – which is the travel time under ideal conditions (i.e. no congestion and vehicles are able to travel at the posted speed limit without slowing or stopping) – minus the actual travel time. The average speed was the average speed experienced by all vehicles throughout their entire journey through the network.

5.3 Sensitivity Analyses

Two other forms of intersection control were also compared to AIM+ in the analysis: conventional traffic signal control and AIM with FCFS policy. This allowed for comparison of AIM+ both to the intersection control that exists today and to another form of AIM. Both of these other forms of intersection control were also modeled similarly in Aimsun with some minor differences that will be discussed later in this chapter.

It is also important to note that, as mentioned previously, cases of malfunctioning vehicles were not modeled in the simulations. Using the malfunction-handling algorithm mentioned in **Chapter 3**, vehicles approaching the intersection would have to come to a stop if a vehicle stopped transmitting BSMs. This holds true even if a vehicle stops transmitting BSMs for cases that don’t involve a malfunctioning vehicle. If these occurrences happen frequently (including when they occur unnecessarily), the intersection will not be able to operate as efficiently. As such, it should be noted that the results presented from these analyses consider scenarios without malfunctions, and the operational capabilities of AIM+ could become worse depending on the frequency of vehicle malfunctions.

In the sensitivity analyses for the three forms of intersection control, both MOEs were utilized to evaluate the operational capabilities. In addition, up to five different categories of parameter-varying scenarios were considered in the sensitivity analyses. These five categories were safety buffer, maximum speed, intersection demand, turn ratio, and approach ratio. These parameters will be discussed more in-depth later in the chapter.

5.3.1 Scenario Standard Parameter Values

During the analysis of each parameter category, the remaining parameter values were held constant for most cases. The standard values for the parameters were:

- Intersection demand: 5600 vehicles per hour for the entire intersection
- Approach ratio: 1:1:1:1 (even split of the demand among all four approaches)
- Turn ratio: 22.5%/55%/22.5% split of the total demand for left-turns, through movements, right-turns respectively
- Safety buffer: 0.0 seconds of extra time headway between vehicles (except for some cases in maximum speed category)
- Maximum speed: 40 miles/hour

In a large majority of the cases, only one parameter was varied for each scenario so that the impact of solely that parameter on the operational capabilities of each intersection control could be assessed.

5.3.2 Traffic Signal Control Intersection

A signalized intersection was considered in the analysis in order to determine what improvements in efficiency AIM+ could bring to today's methods of intersection control. The construction of the traffic-signal controlled intersection in Aimsun was slightly different than the model used for AIM+, but the same skeleton layout of the intersection was used (i.e. link lengths, lane widths, etc.). For the analysis of the signal-controlled intersection, the goal was to create a signalized intersection that was able to reasonably handle the demand that utilized the intersection. By doing this, a fair comparison could be made between AIM+ and the signalized intersection.

For the signalized intersection in Aimsun, dual left-turn lanes and one right-turn lane with storage lengths of 500 feet were added to the model in order to serve the higher demand of turning vehicles. Without the addition of these lanes, the results of the MOEs can be skewed if, for example, left-turning vehicles spend an unrealistic and unreasonable amount of time waiting in a queue to turn left when very few gaps are present. This modification ensured that the MOEs were affected by the intersection control and not the intersection geometry. **Figure 6** shows an image of the signalized intersection in Aimsun.

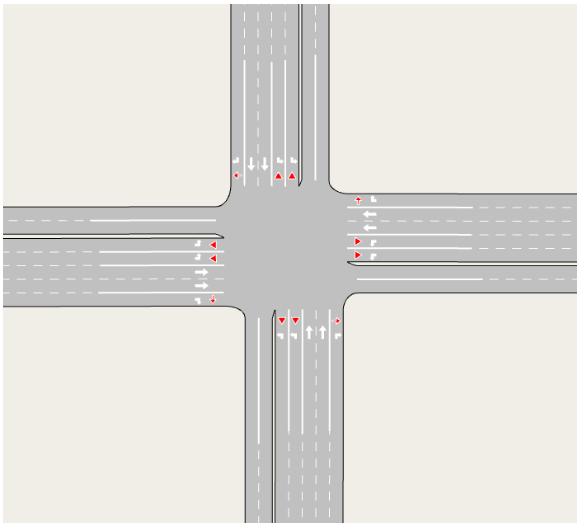


Figure 6, Signalized Intersection in Aimsun

Other provisions made for the signalized intersection included permitting lane changing so that vehicles were not stuck behind a non-moving queue and allowing for a slight variability in the speed used by vehicles (in reference to the speed limit). This was done by changing the speed acceptance within the vehicle class in Aimsun. To ensure that the most optimal signal timings were used in the simulations of the signalized intersection, Synchro by Trafficware was used for each scenario considered in the analysis. This signal timing optimization used protected phases for the left-turns and varying cycle and phase lengths. The unique signal timing obtained from Synchro for each scenario was then inputted into the Aimsun model.

5.3.3 AIM with FCFS Intersection

The analysis of AIM with FCFS policy was important, because it showed the benefits that AIM+ could provide in comparison to another method of AIM. The FCFS AIM policy was chosen for this analysis over other AIM policies because it was the first policy that was developed for AIM. Additionally, it represents a realistic policy due to its fairness of letting the first vehicle that enters the vicinity of the intersection be the first vehicle to pass through the intersection. The intersection used for AIM with FCFS in Aimsun was identical to that of AIM+. The major difference between the two came with the addition of two constraints to the MILP formulation. Equations (26) and (27) show these two additional constraints.

$$t_i(c_i^0) \leq t_j\left(c_j^0\right) \quad \forall i, j \in \mathcal{V}, e_i < e_j \tag{26}$$

$$t_i(c_i^N) \leq t_j\left(c_j^N\right) \quad \forall i, j \in \mathcal{V}, e_i < e_j \tag{27}$$

The introduction of these two constraints ensured that the vehicle that had the smaller earliest arrival time (i.e. it entered the network first) arrived at its initial conflict point and exited the intersection before the vehicle that entered the network at a later time.

5.4 Sensitivity Analyses Scenarios and Results

After running all of the scenarios for AIM+, the signalized intersection, and AIM with FCFS, the results were compiled for the scenarios and the MOEs previously specified. The results for each of the five varying parameters: safety buffer, maximum speed, intersection demand, turn ratio, and approach ratio will all be presented individually. For the scenarios that are applicable, AIM+ will first be compared to a signalized intersection, then to AIM with FCFS. Lastly, the trends of AIM+ individually will be discussed for each of the five parameters. All the results show the standard error for a 95% confidence interval.

5.4.1 Intersection Demand Scenarios and Results

Intersection demand for the analysis scenarios was considered as the total number of vehicles that entered the network from all four approaches of the intersection. The range for this parameter was 3200 vehicles per hour up to 8800 vehicles per hour. Given two lanes per approach, this range came to 400 vehicles per hour per lane up to 1100 vehicles per hour per lane. By knowing the speed limits that were set for each link and the backwards wave speed, it is possible to calculate the lane capacity for the network in Aimsun. Since the backwards wave speed (w) and free flow speed (u_f) (the speed limit) were known values, the capacity (Q) can be calculated using the jam density (k_j) for a link based on the triangular fundamental diagram according to equation (28).

$$Q = \frac{k_j u_f w}{w - u_f} \tag{28}$$

This information can be useful when deciding what ranges of demand should be tested on the intersection. However, this parameter had the largest impact on the computation time of the simulation, so 8,800 vehicles per hour was chosen as the maximum to keep the computation time within a reasonable range. **Figure 7** shows the change in computation time for one hour of simulation as the total intersection demand rose from 3200 vehicles per hour up to 8800 vehicles per hour for one replication of each scenario. The computer used for simulation had 16.0 GB of RAM with an Intel Xeon processor at 3.80 GHz.

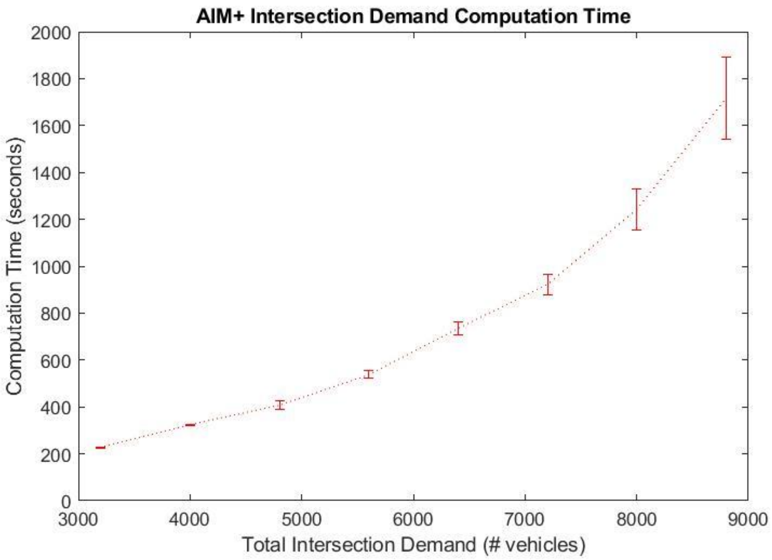


Figure 7, AIM+ Intersection Demand Computation Time

Several potential modifications could be made that could lead to an improvement in the computation time. First, lowering the horizon time will also lower the computation time. Since Gurobi would be run more frequently, less vehicles would be considered in each run.

This modification could also work for cases where the link free flow time is lower than the computation time. Since there would be less decision variables, the computation time would be faster. Secondly, a time-cap could be set for each run of Gurobi. By doing this, there is the chance that the most optimal solution to AIM+ would not be found for each run. However, the solution that is found within the time period would still be sufficient to ensure safe traversals for all vehicles through the intersection. Lastly, the computation could be reduced by using faster processors or dedicated hardware for solving this MILP.

5.4.1.1 Intersection Demand Results: AIM+ Versus Signal Control

The most pertinent comparison that was made between two forms of intersection control is the overall intersection demand. Intersection demand is a parameter that commonly fluctuates in real-world cases at intersections (i.e. if a signal is in a rural or urban location, if it is peak hour or non-peak hour, etc.). This made it important to evaluate how the forms of intersection control operated under fluctuating demand. **Figure 8a and Figure 8b** show the average delay and the average speed experienced by vehicles as intersection demand increased for AIM+ and traffic signal control.

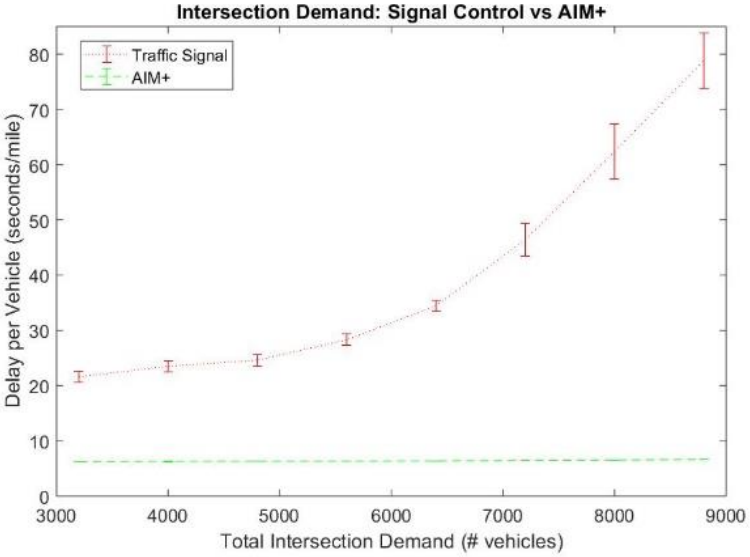


Figure 8a, AIM+/Signal: Intersection Demand and Delay

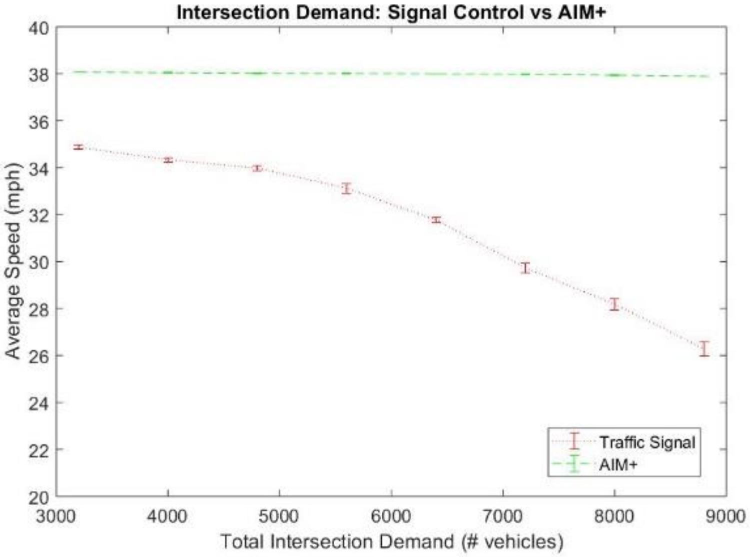


Figure 8b, AIM+/Signal: Intersection Demand and Speed

The intersection with a traffic signal as the form of traffic control was much more impacted by increasing demand than AIM+. In fact, the delay and average speed with AIM+ appear to be unaffected by fluctuating demand when compared to the magnitude at which the signalized intersection was affected by the demand. The signalized intersection saw an increase in delay of approximately 60 seconds between 3200 vehicles per hour and 8800 vehicles per hour. This increase largely occurred between 6000 vehicles per hour and higher. Additionally, the average speed per vehicle decreased from approximately 35 mph at 3200 vehicles per hour to 26 mph at 8800 vehicles. The larger drop also began to occur at approximately 6000 vehicles per hour.

5.4.1.2 Intersection Demand Results: AIM+ Versus AIM with FCFS

Considering the effect of increasing intersection demand on AIM with FCFS versus AIM+, similar trends to the signal control scenarios for intersection demand were noted for delay and average speed – as shown in **Figure 9a and Figure 9b**.

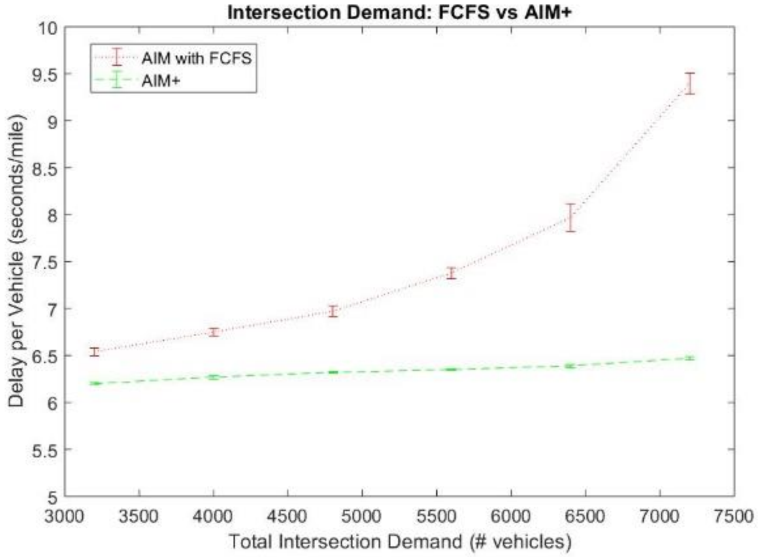


Figure 9a, AIM+/AIM FCFS: Intersection Demand and Delay

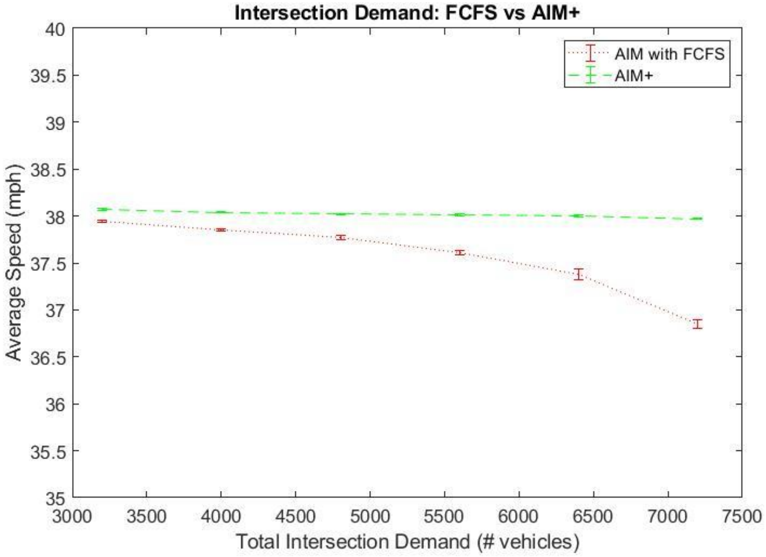


Figure 9b, AIM+/AIM FCFS: Intersection Demand and Speed

The significant jump in both delay and average speed for AIM with FCFS occurred right around the 6,400 vehicles per hour mark. However, it was interesting to note that the difference between AIM+ and AIM with FCFS at lower demands was not very pronounced. The difference in delay for AIM with FCFS and AIM+ ranged from approximately half a second to approximately three seconds (with AIM+ performing better). Similarly, AIM with FCFS saw a difference in average speed ranging from approximately 0.2 mph up to more than 1 mph in favor of AIM+.

5.4.1.3 Intersection Demand Results: AIM+

Probably most interesting within the individual sensitivity analysis of AIM+ was the varying of the intersection demand parameter. **Figure 10a and Figure 10b** show how delay and average speed were affected by increasing intersection demand – split evenly among all approaches.

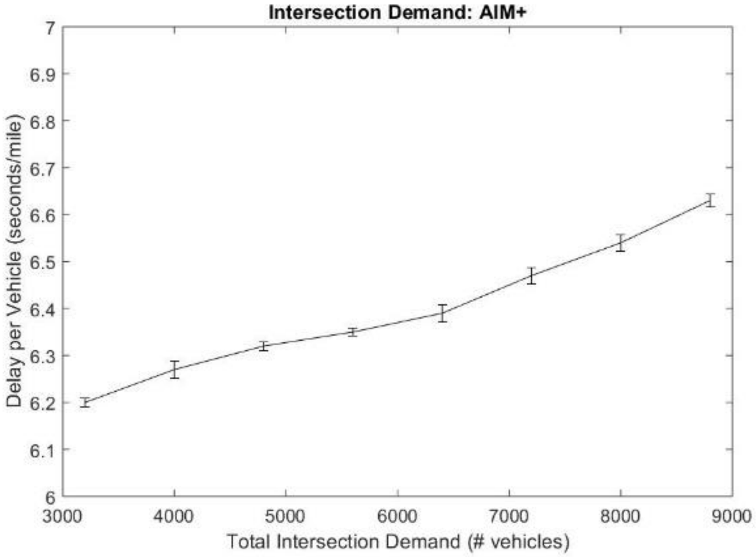


Figure 10a, AIM+: Intersection Demand and Delay

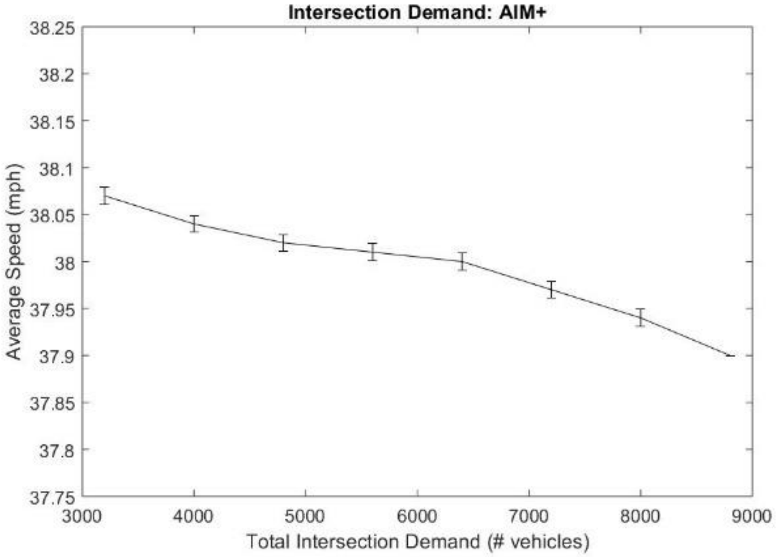


Figure 10b, AIM+: Intersection Demand and Speed

Similar yet opposing trends for average delay and average speed can be noted. The delay per vehicle increased slightly as the demand increased from 3200 vehicles per hour to 8800 vehicles per hour. The total delay increase, though, was less than half a second per mile. For the average speed, a downward trend was seen, with only a total change in average speed of approximately 0.2 mph across the whole spectrum of demand.

5.4.2 Approach Ratio Scenarios and Results

The approach ratio was defined as the percentage of the total intersection demand that utilized a specific approach of the intersection (i.e. north leg). Three different sets of scenarios were considered in the analysis of the approach ratio. The first set of scenarios increased the percentage of total demand that was experienced on just one leg of the intersection. **Table 3** shows the first set of scenarios with the split of total intersection demand over the four approaches (north, east, south, and west) as a percentage.

Table 3, Approach Ratio Scenarios Part 1

	Percentage of Total Intersection Demand			
Approach\Scenario	AR1	AR2	AR3	AR4
North Leg (SB)	32.5%	40.0%	47.5%	55.0%
East Leg (WB)	22.5%	20.0%	17.5%	15.0%
South Leg (NB)	22.5%	20.0%	17.5%	15.0%
West Leg (EB)	22.5%	20.0%	17.5%	15.0%

As **Table 3** shows, the demand on the north leg increased from 32.5% up to 55%, while the demand from the remaining three legs was proportionally subtracted. This could be comparable to traffic conditions near an event where a large percentage of the demand is loaded onto just one approach of an intersection.

The second set of scenarios increased the percentage of total demand experienced on two of the legs (the north and south legs). **Table 4** shows the second set of scenarios considered in the approach ratio analysis of AIM+.

Table 4, Approach Ratio Scenarios Part 2

	Percentage of Total Intersection Demand		
Approach\Scenario	AR5	AR6	AR7
North Leg (SB)	31.25%	37.50%	43.75%
East Leg (WB)	18.75%	12.50%	6.25%
South Leg (NB)	31.25%	37.50%	43.75%
West Leg (EB)	18.75%	12.50%	6.25%

In this set of scenarios, the demand was progressively added onto the north and south approaches and subtracted proportionally from the other two approaches. This is similar to an intersection where there is an obvious major road and an obvious minor road. The percentage of total demand on the north and south approaches increased from approximately 31% each up to approximately 44% each, as shown in **Table 4**.

A final scenario considered an approach ratio where there was no demand originating from, or traveling to, one of the legs of the intersection. In this case, the intersection behaved like a T-intersection. The demand was split evenly among the remaining approaches (33% of demand each). **Table 5** shows the exact distribution of the demand over the approaches.

Table 5, Approach Ratio Scenarios Part 3

	Percentage of Total Intersection Demand
Approach\Scenario	AR8
North Leg (SB)	33.33%
East Leg (WB)	33.33%
South Leg (NB)	33.33%
West Leg (EB)	0.00%

An important difference to note with this scenario concerns the standard parameter values that were utilized. In this scenario, the turn ratio was split 50% to 50% between left and right turns for the "T-approach" (the east leg). For the remaining two approaches, the through movements had 66% of the total approach demand, and the turn movement (either left- or right-turn) had 33% of the total approach demand.

5.4.2.1 Approach Ratio Results: AIM+ Versus Signal Control

Based on the first set of scenarios previously detailed in **Table 3**, the effect of gradually increasing the overall proportion of demand on just one approach for a signalized intersection is shown below in **Figure 11a and Figure 11c** and for AIM+ in **Figure 11b and Figure 11d**.

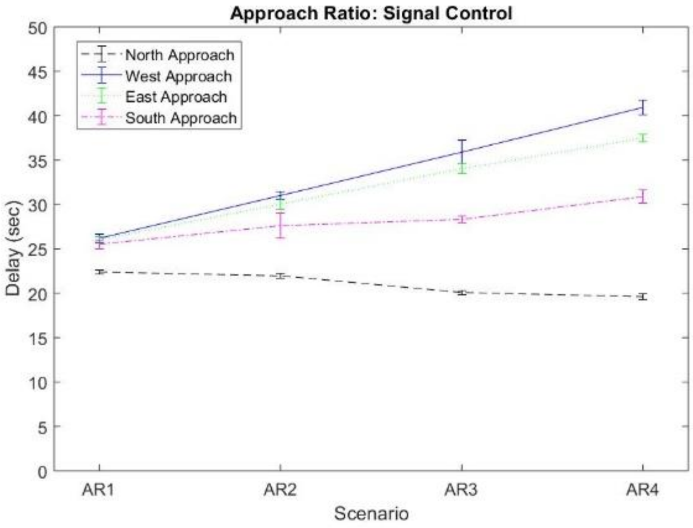


Figure 11a, Signal: Approach Ratio and Delay Part 1

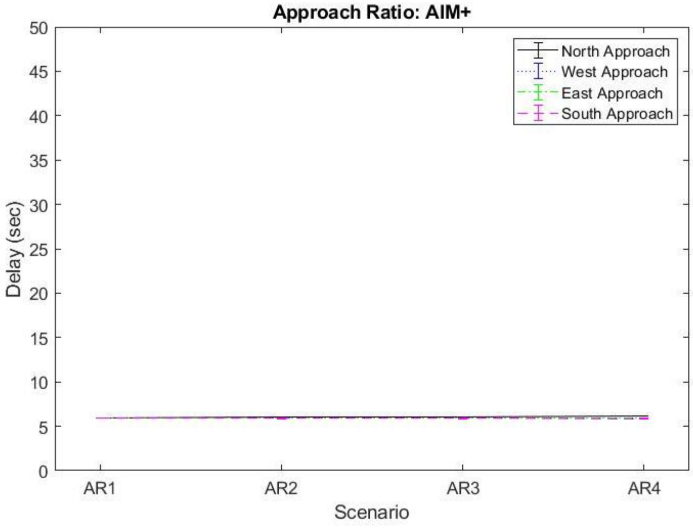


Figure 11b, AIM+: Approach Ratio and Delay Part 1

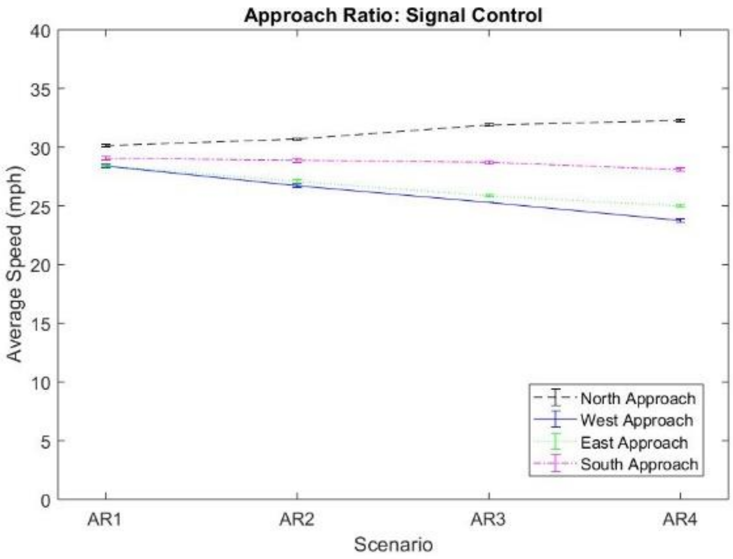


Figure 11c, Signal: Approach Ratio and Speed Part 1

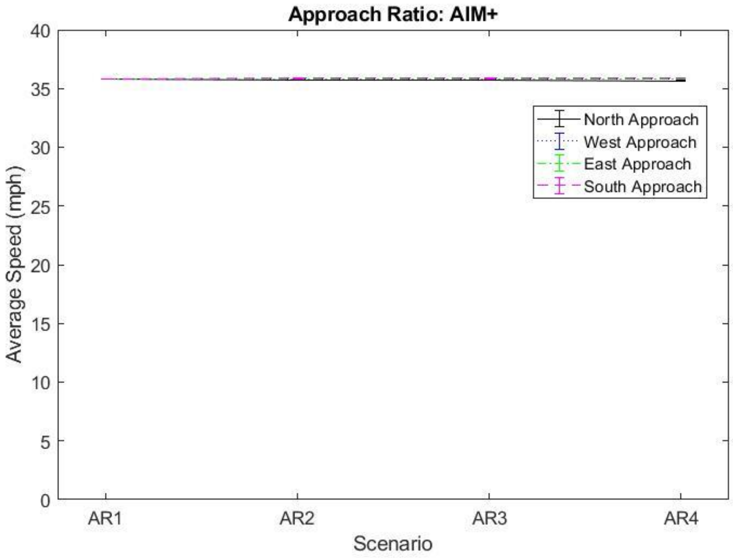


Figure 11d, AIM+: Approach Ratio and Speed Part 1

Figure 11a and Figure 11c show that all four approaches notice somewhat of a significant change in both delay and average speed for traffic signal control. The MOEs for AIM+, however, remain relatively constant (as shown in **Figure 11b and Figure 11d)**. In **Figure 11a** describing the signalized intersection, the delay for the north approach actually decreased by a few seconds (approximately 2 seconds), which can be explained by the changing phase length for the north approach as the demand increased. The other three approaches saw an increase in delay ranging between 5 seconds and 15 seconds.

Furthermore, the average speed for the signalized intersection on the north approach increased slightly by about 3 mph, while the other approaches experienced decreases in average speed between 1 mph and 4 mph. All of the approaches for the signalized intersection experienced at least 15 additional seconds of delay more than AIM+. For the average speed, AIM+ approaches experienced average speeds of at least 4 mph higher.

The second set of scenarios was described in **Table 4**. Below, **Figure 12a and Figure 12c** show the impact of loading more demand on the north and south approaches on the delay and average speed for a signalized intersection. **Figure 12b and Figure 12d** show the counterpart results for AIM+.

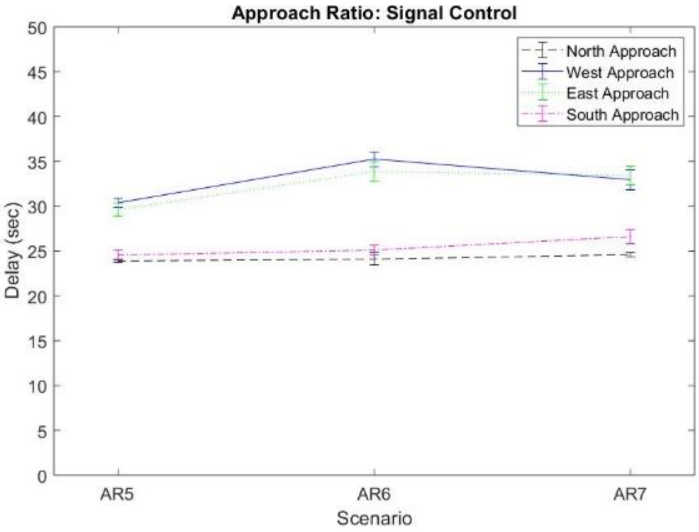


Figure 12a, Signal: Approach Ratio and Delay Part 2

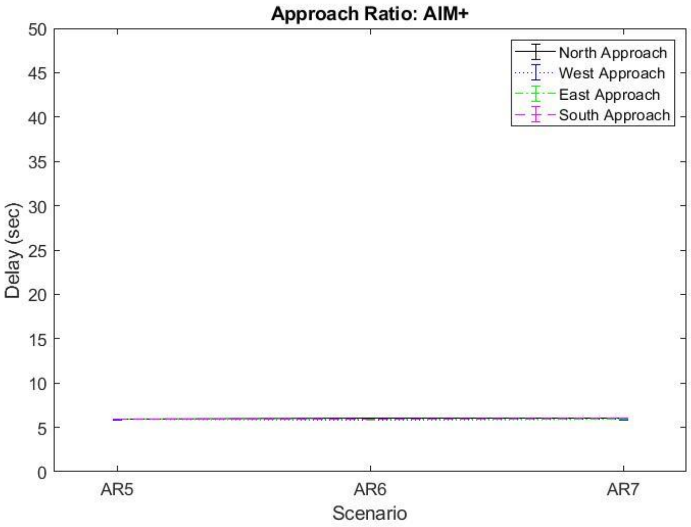


Figure 12b, AIM+: Approach Ratio and Delay Part 2

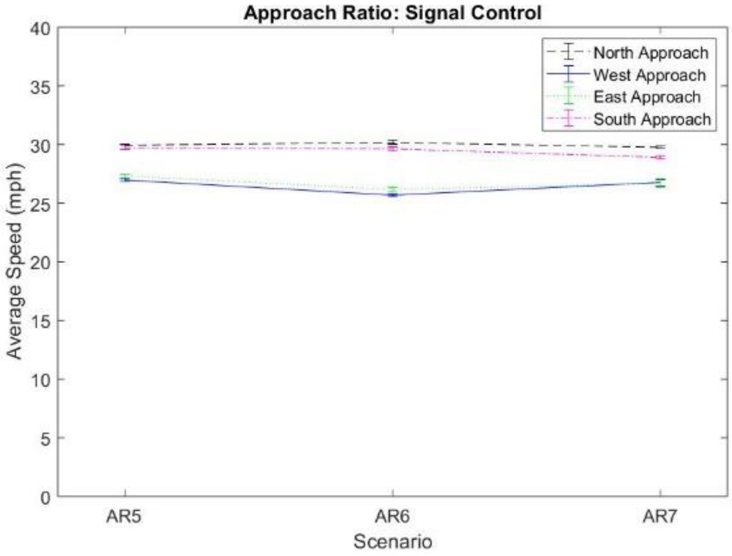


Figure 12c, Signal: Approach Ratio and Speed Part 2

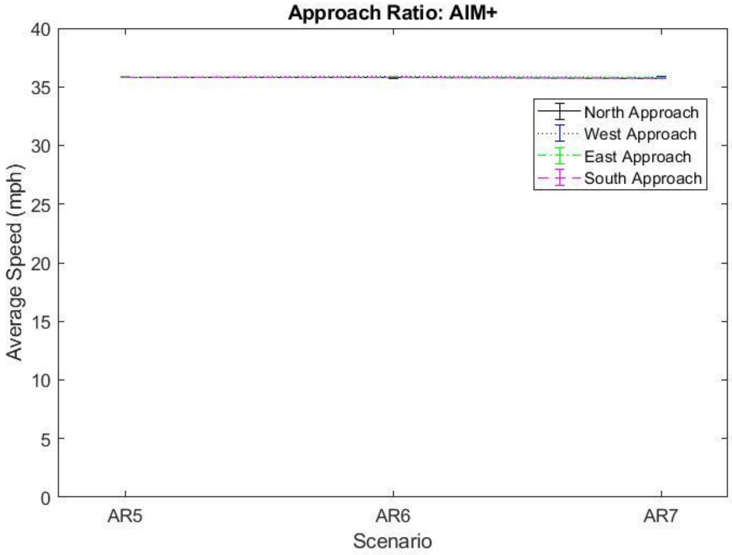


Figure 12d, AIM+: Approach Ratio and Speed Part 2

In these scenarios, there was not much overall change in either delay or average speed for the signalized intersection or AIM+. While the overall differences between AIM+ and signal control were large for both delay and average speed, the trends experienced by both for the above scenarios was relatively the same. One difference was that in AIM+, the performance (in terms of delay and average speed) was better for the east and west approaches, which had lower amounts of demand. On the other hand, for the signalized intersection, the west and east approaches had worse values for delay and average speed.

AIM+ approaches experienced, on average, 16 seconds of delay less than the signalized intersection approaches. For the average speed, AIM+ approaches had an average speed that was 5 mph more than the signalized intersection approaches. These results point to the fact that AIM+ is more flexible and capable of handling unbalanced approach ratios than a signalized intersection is – especially when one roadway has much higher volumes in both directions than the other roadway.

The last scenario, which considered a T-intersection as detailed in **Table 5**, was also analyzed for a signalized intersection. **Figure 13a and Figure 13c** illustrate the impact of traffic signal control on the delay and average speed respectively. Similarly, **Figure 13b and Figure 13d** show the corresponding results for the AIM+ scenarios.

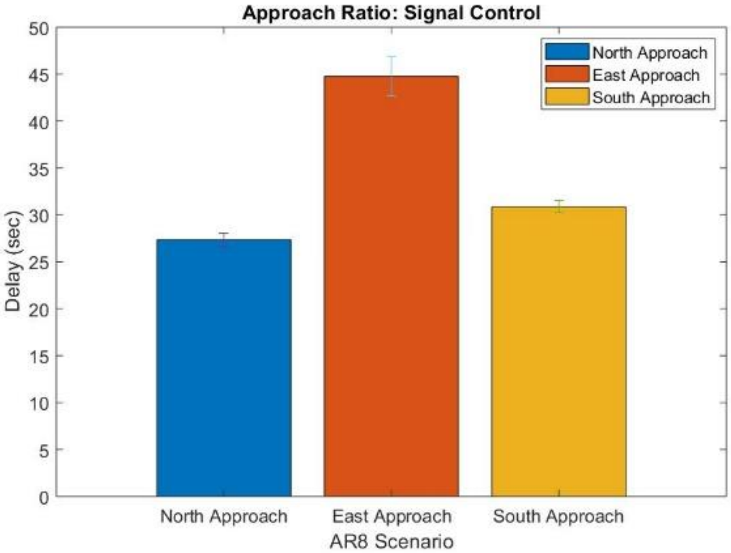


Figure 13a, Signal: Approach Ratio and Delay Part 3

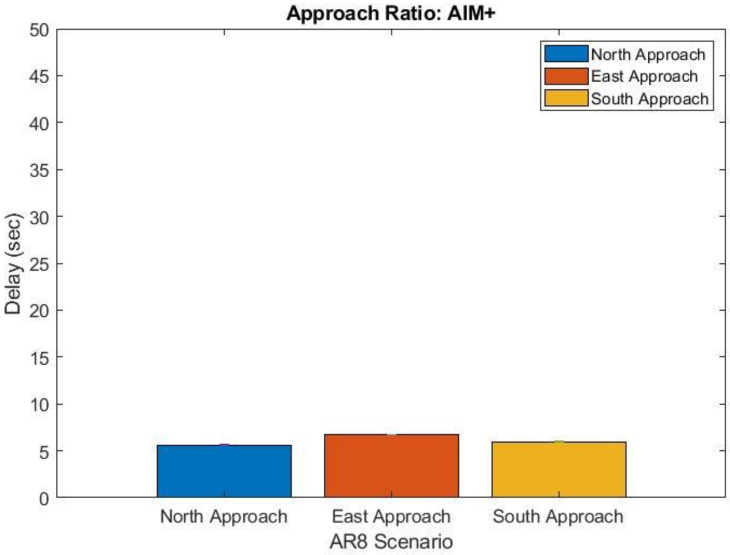


Figure 13b, AIM+: Approach Ratio and Delay Part 3

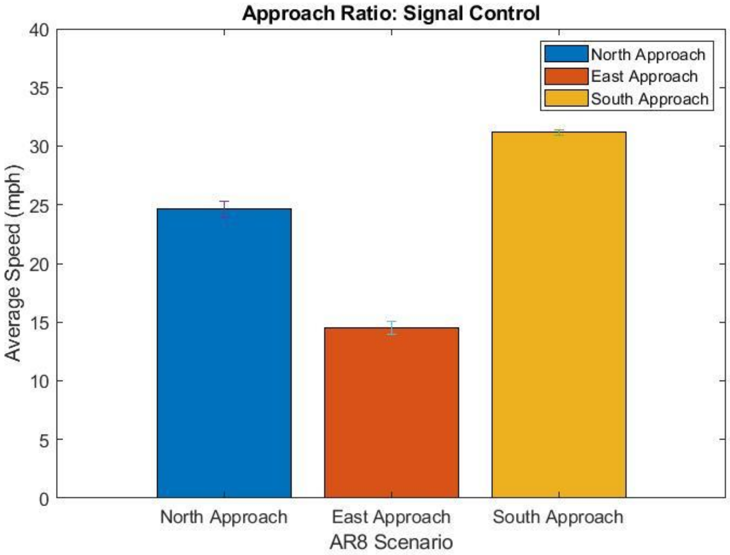


Figure 13c, Signal: Approach Ratio and Speed Part 3

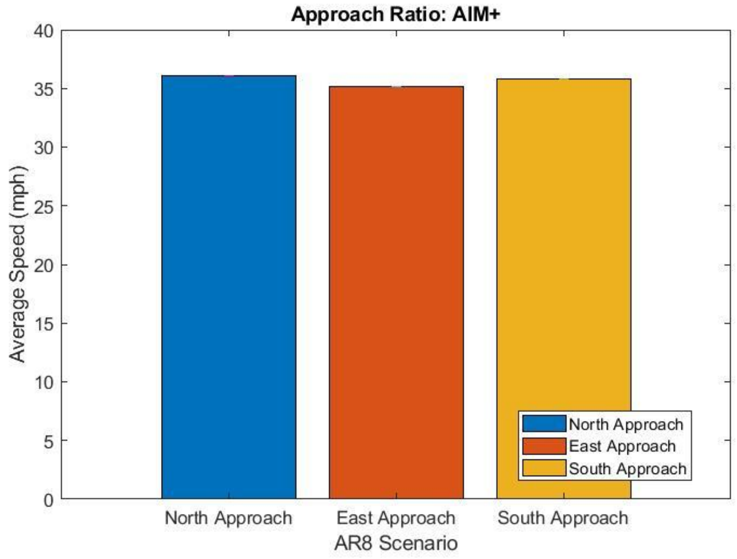


Figure 13d, AIM+: Approach Ratio and Speed Part 3

The relative delay and average speed differences at each approach for the signalized intersection matched those for AIM+. The magnitude of differences between each of the approaches was much more pronounced for the signalized intersection, though. For example, the difference in delay between the north approach and the east approach was approximately 20 seconds. For AIM+ this same difference was only about one second. Similarly, the difference between the average speed for the north and east approaches was approximately 10 mph for the signalized intersection and less than 1 mph for AIM+. A similar pattern was noted when comparing the south approach to the other approaches. The absolute difference between delay for AIM+ and the signal control intersection was at least 20 seconds for each approach, where AIM+ outperformed the signalized intersection. As for average speed, AIM+ had higher average speeds ranging from 5 mph for the south approach up to 20 mph for the east approach.

5.4.2.2 Approach Ratio Results: AIM+ Versus AIM with FCFS

The three sets of scenarios concerning approach ratio were also considered in the analysis of AIM with FCFS. **Figure 14a and Figure 14c** illustrate the impact on delay and average speed for AIM with FCFS. The AIM+ results for the first set of approach ratio scenarios (which were already shown on a different y-axis scale in **Figure 11b and Figure 11d**) are shown in **Figure 14b and Figure 14d**.

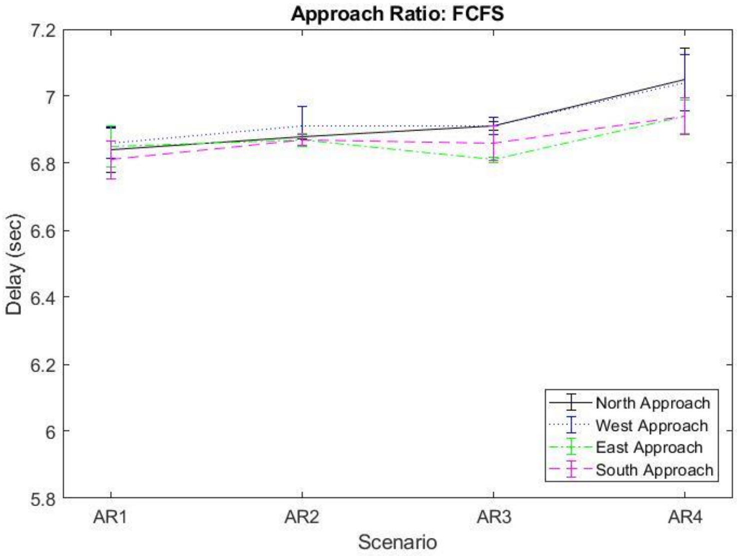


Figure 14a, AIM FCFS: Approach Ratio and Delay Part 1

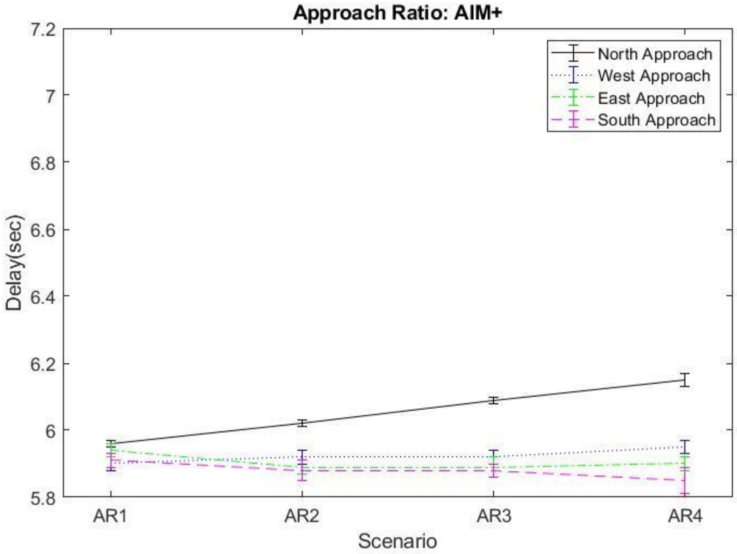


Figure 14b, AIM+: Approach Ratio and Delay Part 1

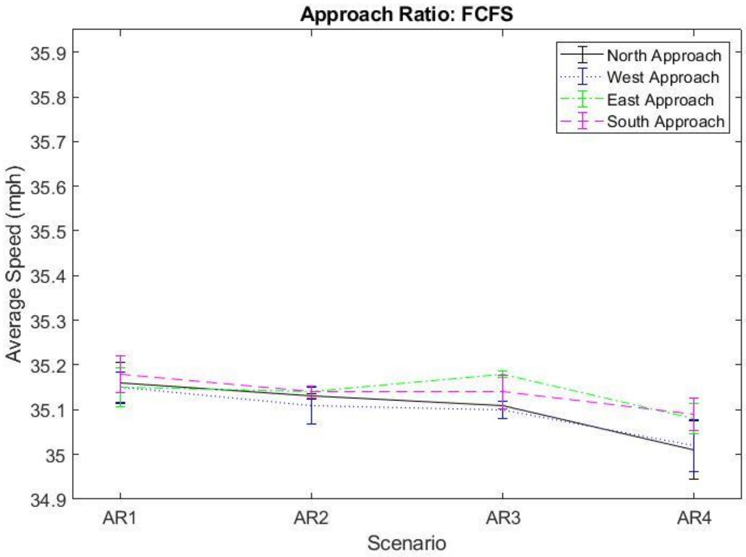


Figure 14c, AIM FCFS: Approach Ratio and Speed Part 1

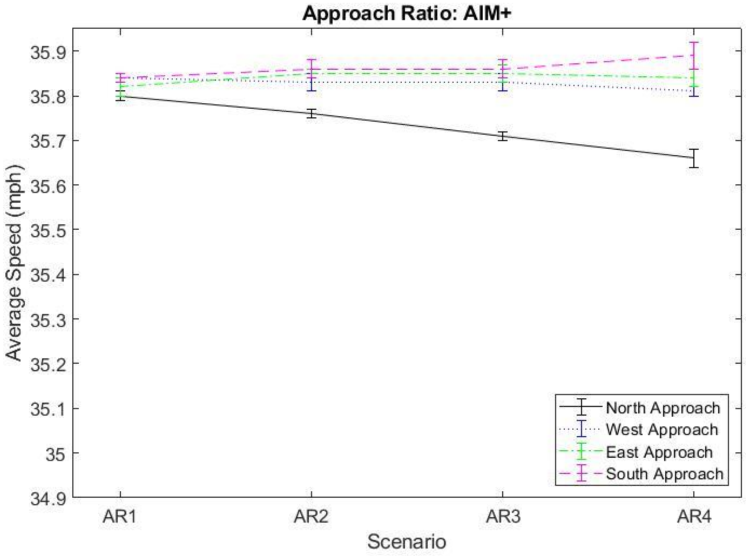


Figure 14d, AIM+: Approach Ratio and Speed Part 1

Compared with AIM+, AIM with FCFS had, on average, approximately one second more of delay across the scenarios AR1-AR4. Additionally, AIM with FCFS saw a decrease in average speed of about 0.7 mph compared to AIM+. Trends were comparable for all approaches with the exception of the north approach, which had the most delay for all four scenarios and was affected most. In AIM with FCFS, the trends for all four approaches were approximately the same for both delay and average speed. In AIM+, though, the north approach saw more delay and a lower average speed than the other approaches. The conclusion drawn from this was that individual approaches were more sensitive to increasing demand for AIM+, but considering the overall values for the MOEs, AIM+ still performed better.

Figure 15a and Figure 15c shows the results for AIM with FCFS considering the second set of approach ratio scenarios, in which the demand was progressively added to two opposing approaches (north and south). **Figure 15b and Figure 15d** show the AIM+ counterpart results (which were also shown in **Figure 12b and Figure 12d** with a different scale on the y-axis).

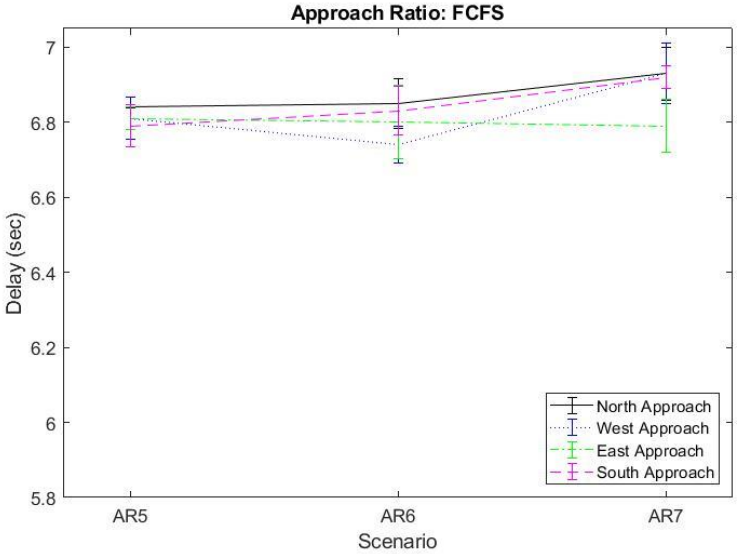


Figure 15a, AIM FCFS: Approach Ratio and Delay Part 2

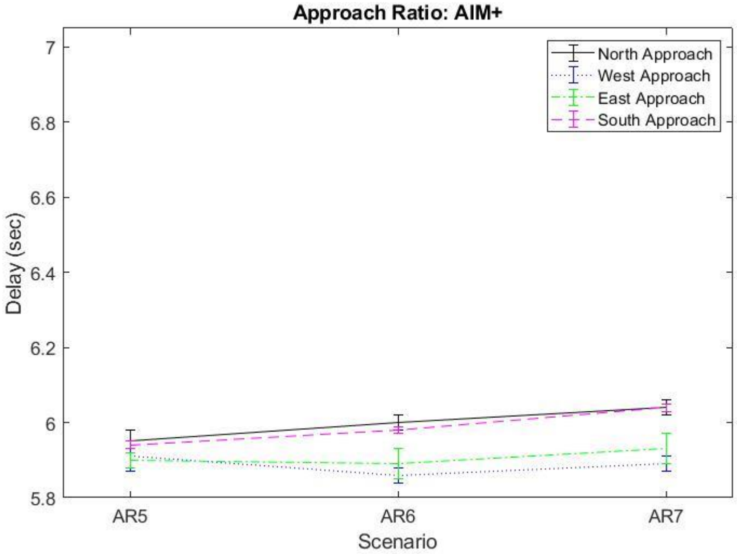


Figure 15b, AIM+: Approach Ratio and Delay Part 2

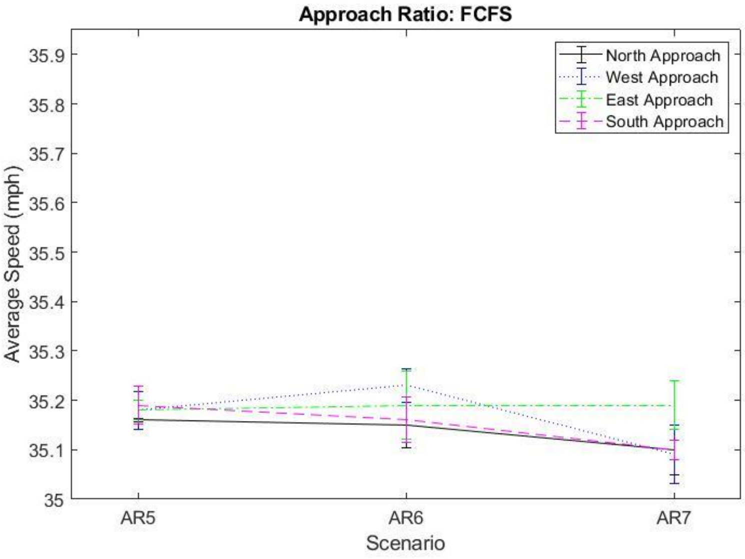


Figure 15c, AIM FCFS: Approach Ratio and Speed Part 2

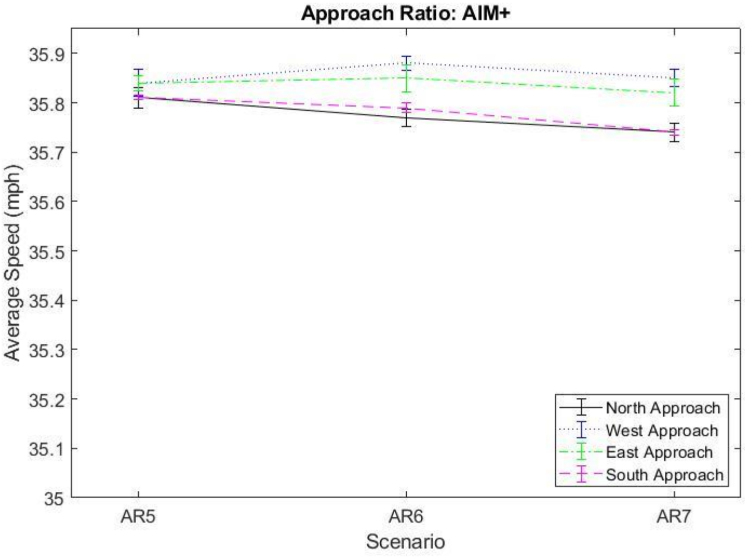


Figure 15d, AIM+: Approach Ratio and Speed Part 2

For the second set of scenarios, AIM with FCFS performed almost identically to AIM+ in terms of the trends observed for delay and average speed. It can be observed that the delay for the north and south approaches (which had the larger amount of delay) was very slightly higher, and the average speed was just slightly lower than AIM+. Similar to the first set of scenarios for approach ratio of AIM with FCFS, the delay decrease and average speed increase with AIM+ is important to note. The difference in delay was approximately 0.9 seconds in these scenarios, and the average speed difference was approximately 0.6 miles per hour.

The results for the T-intersection scenario for AIM with FCFS are shown in **Figure 16a and Figure 16c** and in **Figure 16b and Figure 16d** for AIM+ (also shown in **Figure 13b and Figure 13d**).

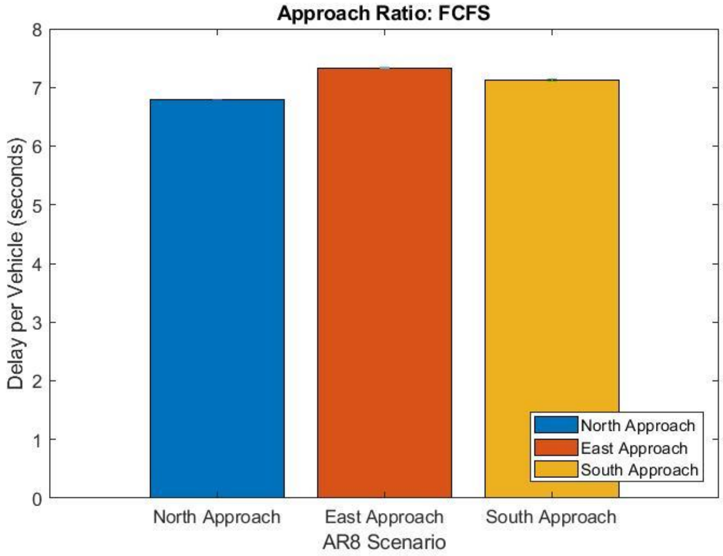


Figure 16a, AIM FCFS: Approach Ratio and Delay Part 3

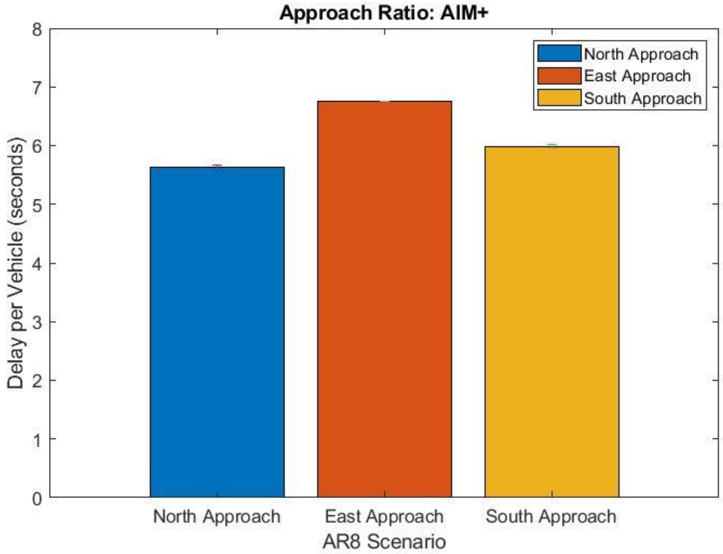


Figure 16b, AIM+: Approach Ratio and Delay Part 3

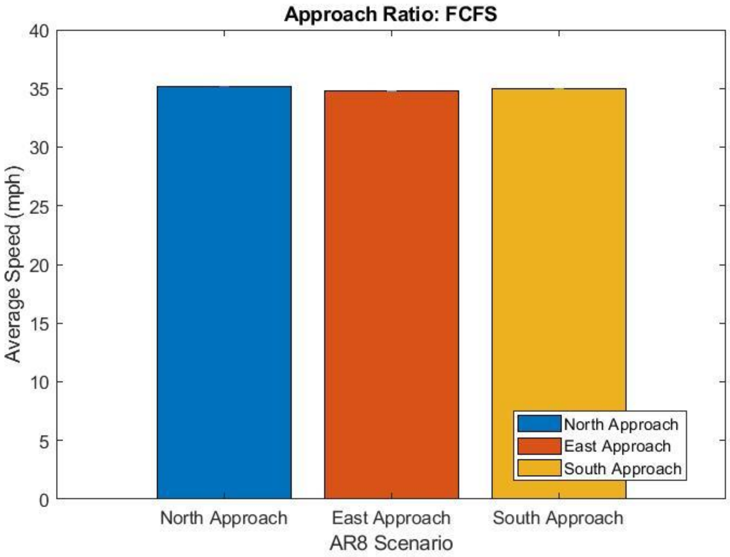


Figure 16c, AIM FCFS: Approach Ratio and Speed Part 3

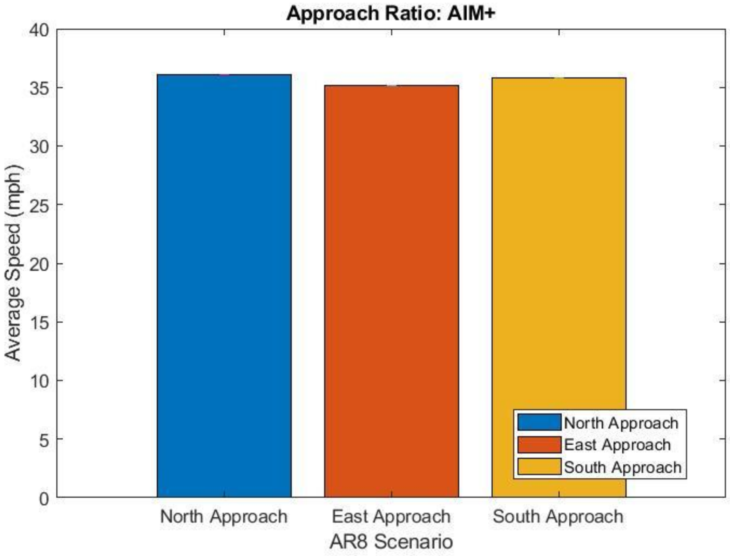


Figure 16d, AIM+: Approach Ratio and Speed Part 3

Similar to the signalized intersection scenario, the delay and average speed for AIM with FCFS matched the same trends for AIM+. The difference in delay and average speed was seen for this scenario as well. Delay increased by about one second for the north and south approaches in AIM with FCFS, but the delays were approximately the same for the east approach. For the average speed, there was a decrease of approximately 0.8 miles per hour for AIM with FCFS for the north and south approaches. The decrease in average speed for the east approach was not as significant at 0.4 miles per hour, although more noticeable than the change in delay for the approach.

5.4.2.3 Approach Ratio Results: AIM+

The trends of results for the approach ratio scenarios in AIM+ specifically were also evaluated to assess the sensitivity of AIM+. The figures shown in this section were also shown in sections 5.4.2.1 and 5.4.2.2 but were recreated here on a smaller y-axis so that the trends of the different approaches could be analyzed more specifically. **Figure 17a and Figure 17b** show the impact of loading the demand on one leg on the delay per vehicle and the average speed for each approach using AIM+ as the form of intersection control.

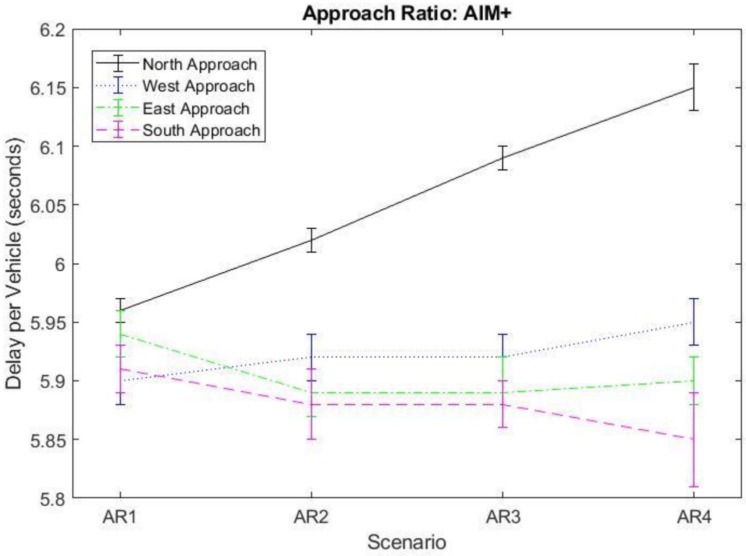


Figure 17a, AIM+: Approach Ratio and Delay Part 1

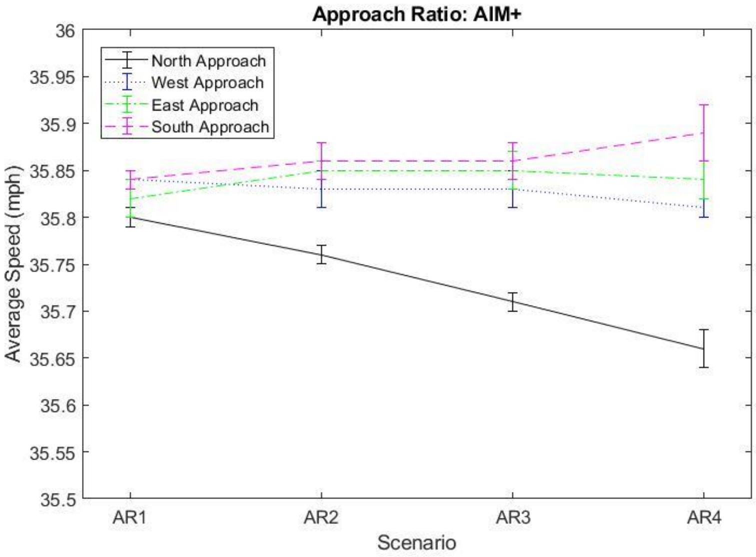


Figure 17b, AIM+: Approach Ratio and Speed Part 1

The delay and average speed per vehicle remained relatively constant for all three approaches across the four scenarios. A very slight decrease in delay and increase in average velocity for the south approach were seen. For the north approach, there was a much more obvious change in both delay and average speed. The delay increased by approximately 0.2 seconds and the average speed decreased by approximately 0.1 mph. While this showed that adding demand to one approach can affect its performance, again, it is important to note the very small overall change that occurred from both extremes of the scenarios.

Figure 18a and Figure 18b show how increasing the demand on two of the approaches affected the delay and average speed per vehicle.

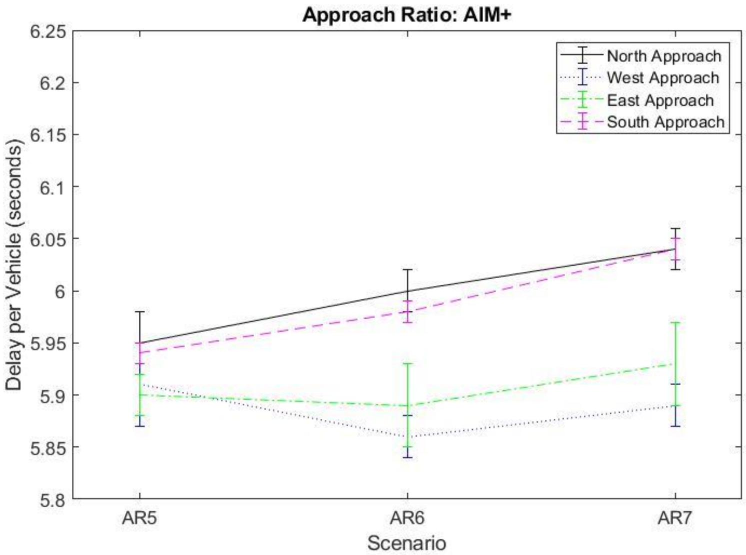


Figure 18a, AIM+: Approach Ratio and Delay Part 2

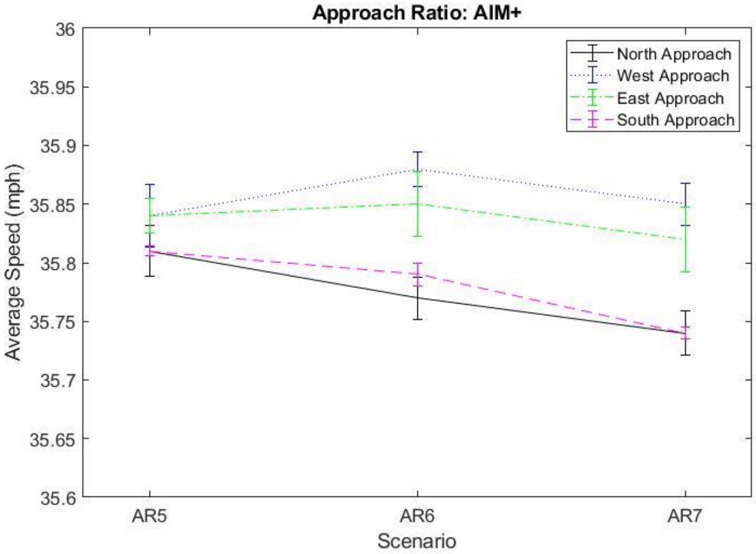


Figure 18b, AIM+: Approach Ratio and Speed Part 2

Again, there was not much change in either the delay or the average speed for the west and east approaches, which had progressively less demand for each scenario. In this second set of scenarios, a similar trend was noted to that of the first set of scenarios for approach ratio. In these scenarios, though, the south and north approach were both impacted by the added demand, which was reasonable since vehicles further upstream on links had to wait for vehicles in front of them to arrive to the intersection for the reservation. The delay increased by only approximately 0.05 seconds per vehicle for the north and south approaches, and the average speed decreased by only approximately 0.05 mph.

Figure 19a and Figure 19b show the average delay and average speed respectively for the T-intersection scenario.

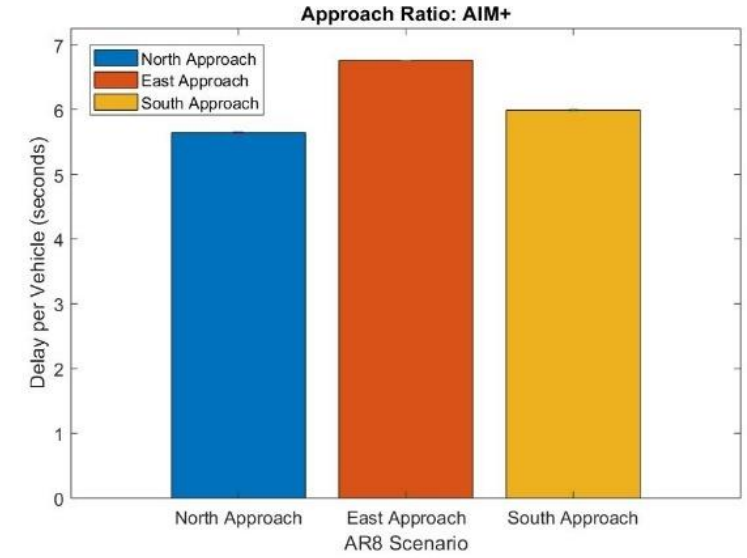


Figure 19a, AIM+: Approach Ratio and Delay Part 3

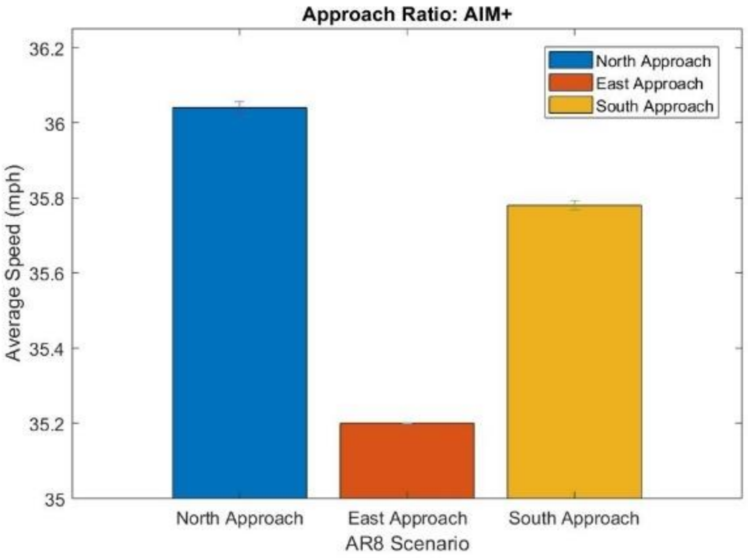


Figure 19b, AIM+: Approach Ratio and Speed Part 3

In this case, the east approach was the approach which was required to turn either left or right at the intersection. Compared to the south and north approaches, the east approach had a lower average delay by 0.5 seconds and 1 second and a lower average speed by approximately 0.6 and 0.8 mph respectively. The south approach performed slightly worse than the north approach which could be expected since the north approach only had through movements and right-turns and not any left-turn movements, which have more impact on the delay and speed.

5.4.3 Safety Buffer Scenarios and Results

Safety buffer is the time spacing between vehicles at conflict points within the intersection. This spacing can be increased incrementally to provide additional time-space between vehicles which may make the traversal through the intersection more realistic and comfortable for the rider. In the analysis, values for the safety buffer ranged from 0.0 additional seconds up to 0.35 additional seconds of spacing between vehicles. **Figure 20** shows illustrations of the different increasing spacing of vehicles.

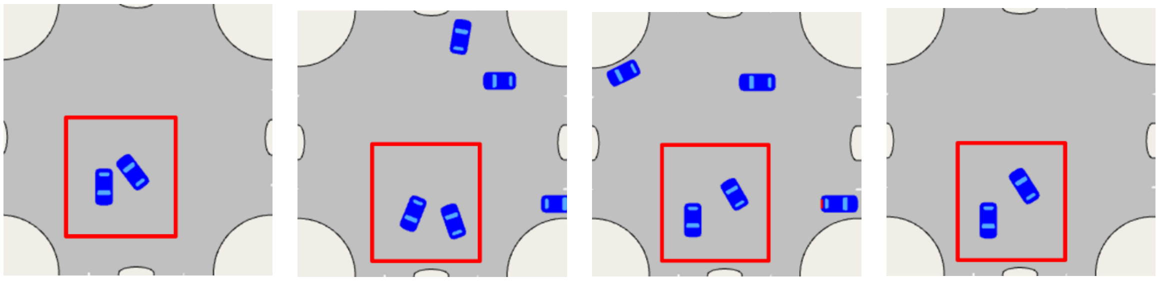


Figure 20, Safety Buffers (0.0 seconds, 0.1 seconds, 0.2 seconds, 0.3 seconds respectively)

This entire range was not used for all comparisons in the analysis due to the large increase in computation time that occurs as this safety buffer increases (since vehicles require more spacing when passing through the intersection less gaps are available). Furthermore, the safety buffer scenarios were not considered in the analysis of the signalized intersection. These were excluded since predefined safety buffers are non-existent with human vehicles, so there is no pertinence in considering them for the signalized intersection evaluation.

5.4.3.1 Safety Buffer Results: AIM+ Versus AIM with FCFS

Since AIM with FCFS does concern autonomous vehicles, it was possible to compare how the two forms of AIM operate as the safety buffer between vehicles increases. **Figure 21a and Figure 21b** compare the functionality of both AIM with FCFS and AIM+ considering delay per vehicle and average speed with respect to increasing safety buffer.

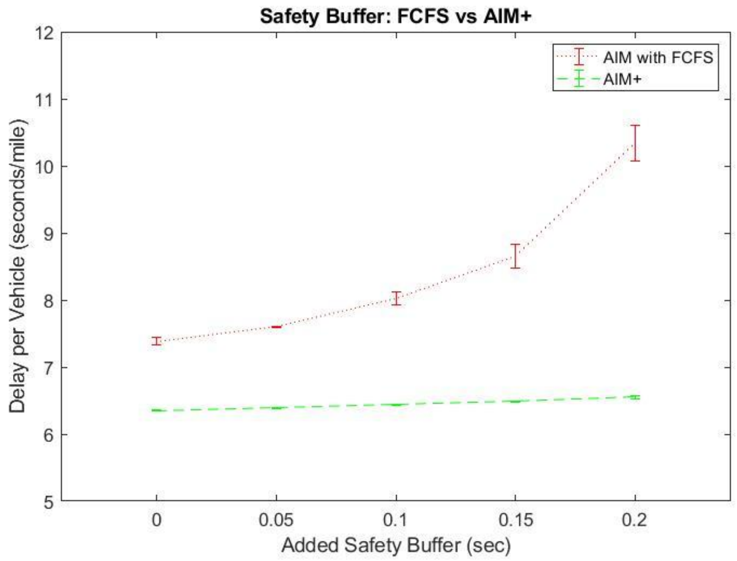


Figure 21a, AIM+/AIM FCFS: Safety Buffer and Delay

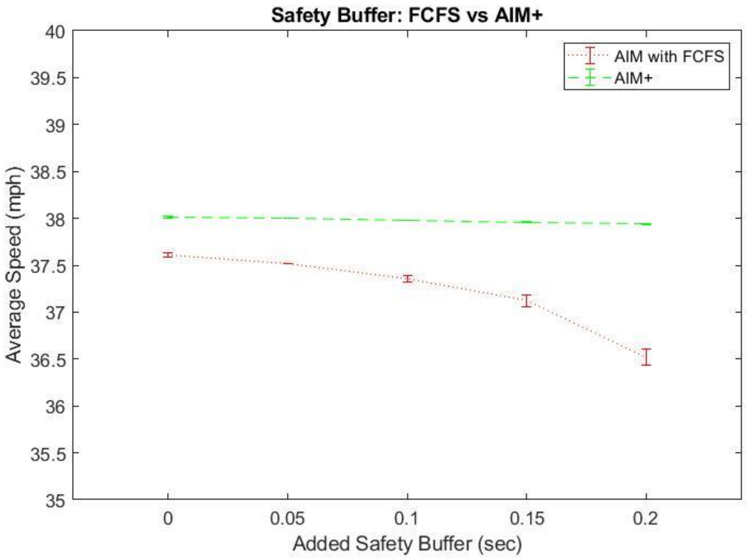


Figure 21b, AIM+/AIM FCFS: Safety Buffer and Speed

For the delay, there was a much larger upward trend for AIM with FCFS than for AIM+. Even at the point where no safety buffer was added, the delay for AIM with FCFS was approximately one second larger than AIM+. As the safety buffer increased, this difference became much more pronounced, and at 0.15 seconds of added safety buffer, there was a much more dramatic increase in delay for AIM with FCFS. While the delay for AIM with FCFS was not significantly different than AIM+, AIM+ still performed better. Average speed saw a similar (yet opposite) trend, in which the average speed for AIM with FCFS was slightly worse than AIM+ when no safety buffer was added, but around 0.15 seconds of added safety buffer, the average speed started to decline more rapidly for AIM with FCFS.

5.4.3.2 Safety Buffer Results: AIM+

Figure 22a and Figure 22b show the effect of increasing the safety buffer on the average delay experienced per vehicle (in seconds per mile) and the average speed for every vehicle (in miles per hour) considering AIM+ as the form of intersection control.

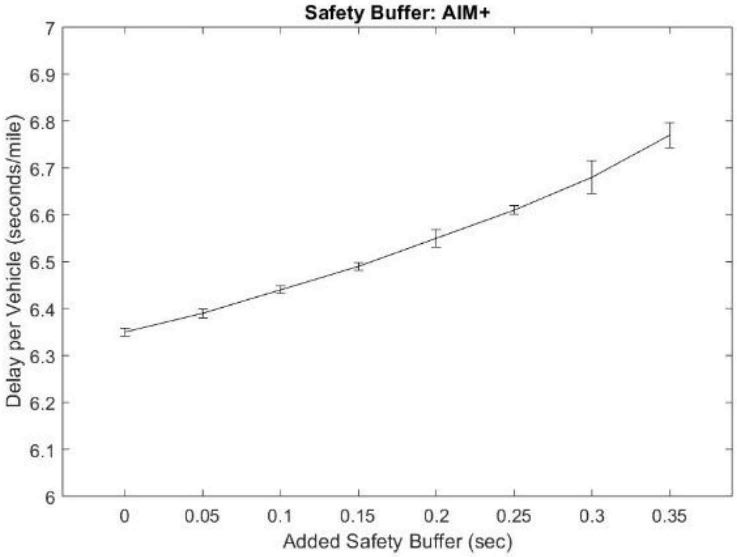


Figure 22a, AIM+: Safety Buffer and Delay

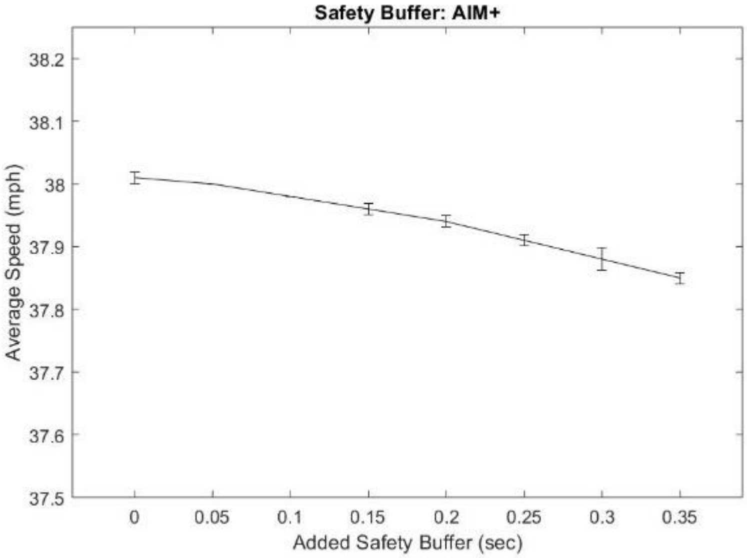


Figure 22b, AIM+: Safety Buffer and Speed

In **Figure 22a**, a gradual upward trend in delay experienced per vehicle is visible, which is reasonable since a vehicle should expect to experience greater delay if more spacing between vehicles is required. However, it is important to note the overall change in delay between no additional buffer added (0.0 seconds) and 0.35 seconds (the maximum) added. Providing this additional safety buffer (as was illustrated in **Figure 20**) only contributed to an additional delay of approximately half a second per mile per vehicle. For the average speed in **Figure 22b**, a gradual downward trend in average speed can be observed. Again, though, the change from 0.0 seconds to 0.35 seconds of added buffer only led to an overall decrease in average speed of approximately 0.15 miles per hour.

5.4.4 Turn Ratio Scenarios and Results

The turn ratio parameter is defined by the percentage of the total intersection demand that turns left, right, or passes through the intersection. While the percentage of turning vehicles was held constant for the intersection demand scenarios, in reality, there are intersections that experience much higher (or lower) left-turning volumes. For these scenarios, the percentage of the total demand making right-turns was held constant at 20%. This was done since right-turning vehicles have little impact on the delay or average speed of the intersection since the number of conflict points along their trajectories is minimal. For left-turns and through movements, the remaining 80% of demand was split at increasing increments of 10% (i.e. 10% left-turning and 70% through, 20% left-turning and 60% through, up to 70% left-turning and 10% through). Turn ratio scenarios were excluded for the signalized intersection scenarios since the provided dual left-turn lanes led to a dissimilar comparison to AIM+ turn ratio scenarios.

5.4.4.1 Turn Ratio Results: AIM+ Versus AIM with FCFS

Since the Aimsun intersection used for the analysis of AIM with FCFS was the same as that used for the analysis of AIM+, a fair comparison was made between the two considering varying amounts of turn ratios. **Figure 23a and Figure 23b** details the results of this comparison for AIM+ and AIM with FCFS for delay and average speed. Note that the x-axis denotes the percentage of the total demand that was making left-turn movements.

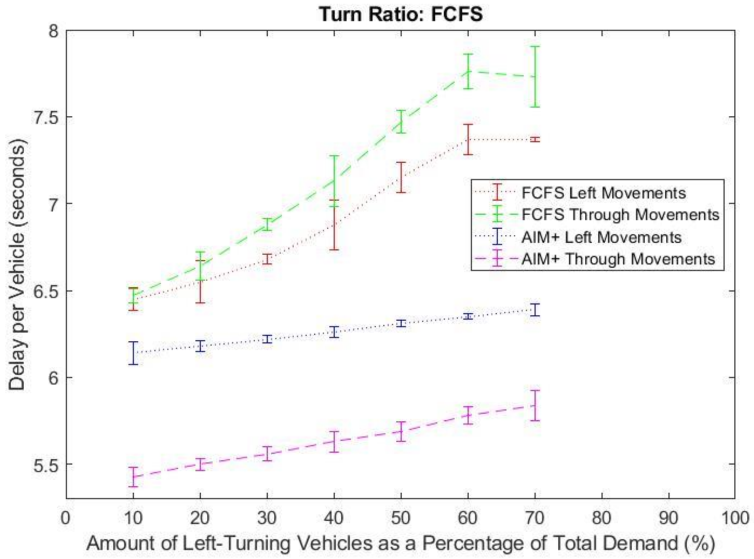


Figure 23a, AIM+/AIM FCFS: Turn Ratio and Delay

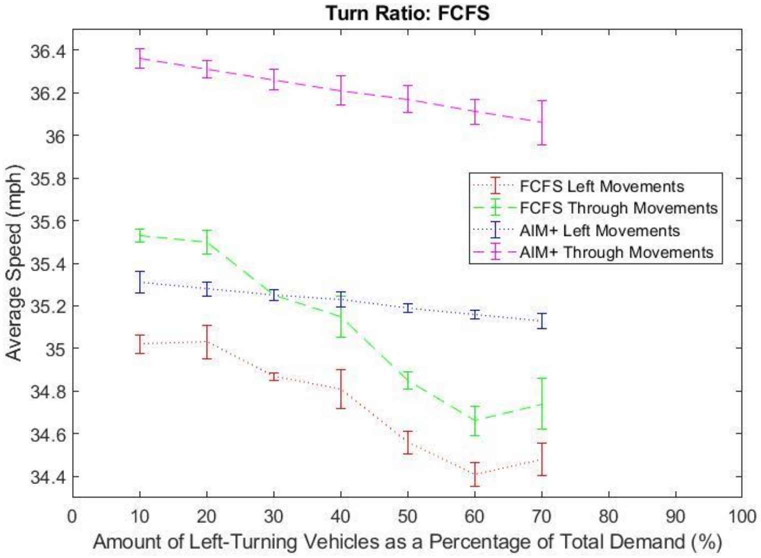


Figure 23b, AIM+/AIM FCFS: Turn Ratio and Speed

The results for these scenarios were interesting because an almost identical trend can be noted between the left movements and the through movements for AIM+ and the left movements and through movements for AIM with FCFS. Looking at the average delay at low percentages of left-turning vehicles, the delays were approximately the same for AIM with FCFS. As that percentage increased, though, a more defined increase in delay was noted. For average speed using AIM with FCFS, left movements experienced, on average, half a mile per hour difference at 10% of the total demand serving left movements. Contrasting to the pattern of delay, this difference actually became slightly less pronounced as the number of left-turning vehicles increased. Looking at the results for the turn ratio scenarios for AIM+, similar trends, yet more gradual, were observed for delay and average speed for both movements. Left-turn movements experienced between 0.4 seconds and 1 second less delay in AIM+, and through movements experienced around 2 seconds less delay for AIM+. For the average speed in AIM with FCFS, left-turning vehicles experienced an increase in average speed ranging from 0.3 mph to 0.8 mph and through moving vehicles saw an increase in average speed between 0.8 mph and 1.4 mph.

5.4.4.2 Turn Ratio Results: AIM+

Considering only AIM+, **Figure 24a and Figure 24b** depict how the increasing percentage of left-turning vehicles affects the average delay per vehicle and the average speed. These figures were also shown in the section above but were represented again here in order to observe the AIM+-specific trends for the turn ratio scenarios.

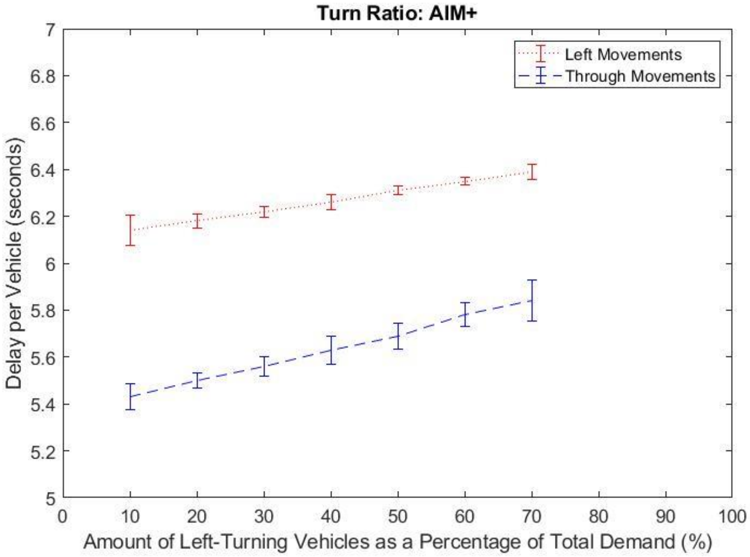


Figure 24a, AIM+: Turn Ratio and Delay

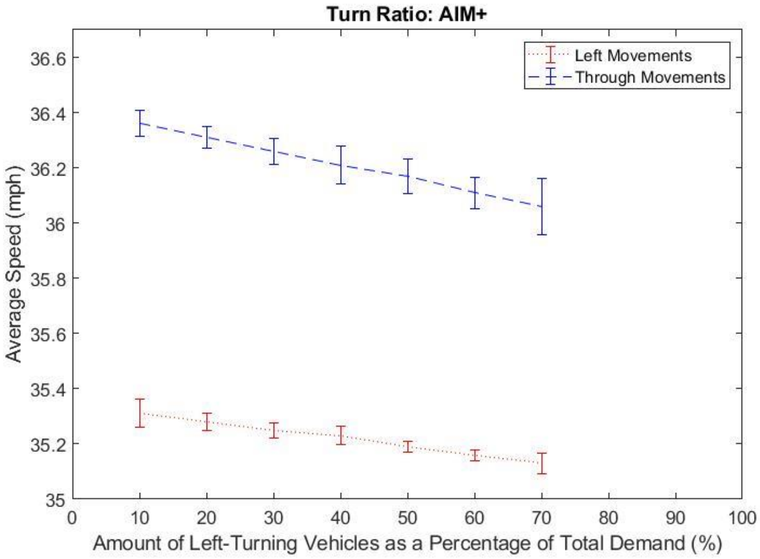


Figure 24b, AIM+: Turn Ratio and Speed

As the percentage of left-turning vehicles increased while the demand remained constant, the delay experienced by both left and through movements increased – but only by approximately 0.2-0.3 seconds. Since the introduction of more left-turning trajectories created more conflict points with other paths, the delay increased for both left and through movements since more maneuvering through conflict points was required as the number of left-turning vehicles increased. The delay for left-turning movements was, on average, larger than that of through movements. Correspondingly, the average speed for both of these movements was observed to gradually decrease by approximately 0.2 mph as the percentage of left-turning vehicles increased. The average speed of left-turning vehicles was approximately 1 mph less than that of through-moving vehicles. Similar trends for left and through movements were noted from these results, but the through movements were impacted slightly more by the increasing percentage of left-turning vehicles versus the left movements.

5.4.5 Maximum Speed Scenarios and Results

The last parameter considered in the analyses was the maximum speed. The maximum speed parameter is comparable to changing the speed-limit for the network. In AIM scenarios, this speed was the constant initial velocity that vehicles utilized upon entering the network until they were instructed to change their velocity by the intersection manager.

This parameter was varied from a minimum of 25 mph to a maximum of 55 mph. Speeds outside of this range are unlikely to be used at intersections where signals are present. The varying of this parameter required several minor modifications to the Aimsun model. First, the length of the links had to be slightly extended to ensure that there was sufficient time for Gurobi to run before a vehicle reached the point at which it should start decelerating. Additionally, at some of the cases of higher maximum speed (around 50 mph and 55 mph), it was necessary to modify the vehicle class so that each vehicle's maximum acceleration was sufficient enough for a vehicle to attain the maximum speed when starting from rest. For scenarios where the maximum velocity was set to lower values, a safety buffer was added to provide sufficient spacing between vehicles. Lastly, the threshold time at which vehicles were considered in the next Gurobi run varied for these scenarios since vehicles would arrive at the intersection more quickly or more slowly than with the standard parameter value of 40 mph.

The maximum speed category was not considered for the signal control intersection since the changing speed limit has little impact on the signal timing and, thus, the MOEs as well. Additionally, it was not considered in the analysis of AIM with FCFS.

5.4.5.1 Maximum Speed Results: AIM+

Since intersections can be located at roadways of varying, the effect on AIM+ of increasing the speed limit was tested. **Figure 25a and Figure 25b** illustrate the effects on delay and average speed from increasing the maximum speed for the vehicles in the Aimsun.

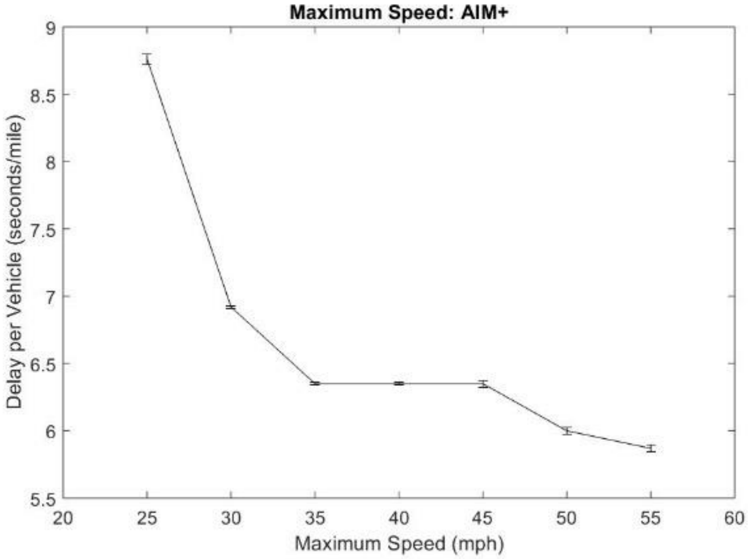


Figure 25a, AIM+: Maximum Speed and Delay

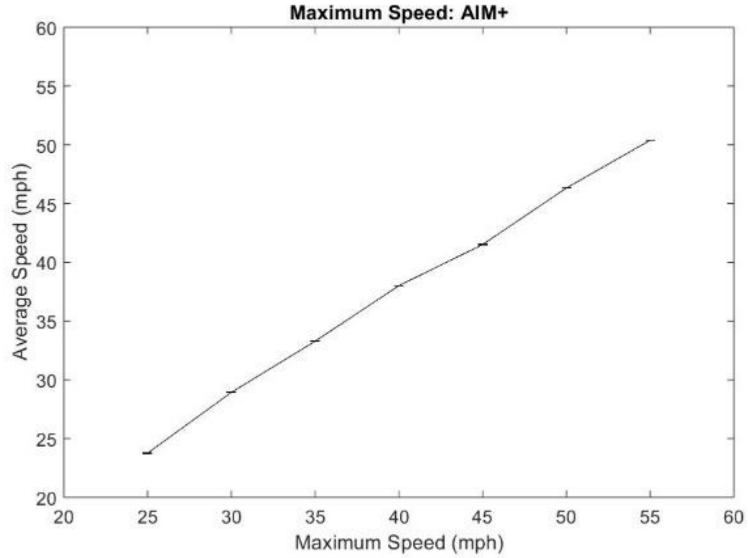


Figure 25b, AIM+: Maximum Speed and Speed

From **Figure 25a**, a sporadic downward trend in delay can be observed. Between 25 mph and 35 mph, there was a more sudden drop in the average delay experienced per vehicle (a drop of about 2.5 seconds). After 35 mph, the delay per vehicle remained relatively unchanged until a slight decrease was experienced between 45 mph and 55 mph (a drop of about half a second per mile). **Figure 25b** provides a good illustration of the capabilities of AIM+ to handle fluctuating speed limits. The average speed approximately matched the maximum speed that was set for the network. As the maximum speed reached 35 mph and more, the average speed started to decrease slightly. This could be because vehicles needed to slow down more to make turns than would be necessary at lower speeds like 25 mph.

5.4.6 Results Summary

In summary, the effects of the changing parameters on the three forms of intersection control were evaluated and presented. AIM+ was shown to outperform the signalized intersection and AIM with FCFS for all scenarios. This demonstrates the tremendous improvements that AIM+ can provide in terms of average delay and average speed. A summary of the different comparisons and sensitivity analyses is presented in this section.

5.4.6.1 AIM+ Versus Traffic Signal Control Results Summary

Comparing the results from the simulation of the signalized intersection to the results for AIM+ showed major differences. Both the delay and the average speed at signalized intersections were much worse with traffic signal control. Fluctuating the amount of demand experienced on each approach yielded mostly similar results between AIM+ and signalized intersection control in terms of the trends noted. However, the overall difference in the values of the MOEs was quite significant. **Table 6** details some of the approximate percentage reduction ranges for delay and percentage increase ranges in average speed when comparing AIM+ to standard traffic signals. The reduction and increase percentages for the approach ratio scenarios considered the most conservative approach (i.e. the approach with the lowest delay or the highest average speed for signals and the approach with the highest delay or the lowest average speed for AIM+).

Table 6, AIM+ vs Traffic Signal Control Results Summary

Scenarios	Delay Reduction with AIM+	Average Speed Increase with AIM+
Intersection Demand	71% - 91%	9% - 46%
Approach Ratio Part 1	69% - 73%	11% - 19%
Approach Ratio Part 2	75%	19%
Approach Ratio Part 3	78%	16%

5.4.6.2 AIM+ Versus AIM with FCFS Results Summary

While the overall differences in delay and average speed were not as drastic as the differences when AIM was compared to signals, it was still obvious that AIM performed better than AIM with FCFS in all situations. It was reasonable that AIM with FCFS generally followed the same trends as AIM+ since they are based on the same formulation. **Table 7** summarizes the approximate percentage reductions in delay and percentage increases in average speed for the various scenarios when comparing AIM+ to AIM with FCFS. For the turn ratio scenarios and the approach ratio scenarios, the percentages considered the most conservative approach (i.e. the approach/movement with the lowest delay or the highest average speed for AIM with FCFS and the approach/movement with the highest delay or the lowest average speed for AIM+).

Table 7, AIM+ vs AIM with FCFS Results Summary

Scenarios	Delay Reduction with AIM+	Average Speed Increase with AIM+
Safety Buffer	14% - 36%	1% - 3%
Intersection Demand	4% - 31%	0% - 3%
Turn Ratio	5% - 12%	0% - 1%
Approach Ratio Part 1	11% - 12%	2%
Approach Ratio Part 2	11% - 13%	2%
Approach Ratio Part 3	1%	0%

5.4.6.3 AIM+ Sensitivity Analysis Results Summary

Overall, the results for all of the parameter scenarios showed the robustness of AIM+ under varying conditions. While fluctuating the parameters did affect the delay and average speed, the overall impact was generally very minimal. Most scenarios saw less than one second of added delay or 1 mph decrease from the lower extreme to the upper extreme. The sensitivity analysis proved that AIM+ was flexible in dealing with changing parameters, while ensuring that the overall performance of the intersection was maintained.

CHAPTER 6

6 CONCLUSIONS

The research presented in this study accomplished several items. First, an introduction of the topic of AIM was provided, and a literature review was presented encompassing several key milestones and aspects of research concerning AIM. Then, the methodology was presented which was later integrated with microsimulation. After discussing the scenarios and parameters considered in the analysis, the results were presented.

6.1 Main Research Contributions

The research contributions of this research involved three main components. First, an improved optimization model of AIM was developed, which was based on a conflict-point reservation-based model of AIM. This was accomplished by developing a mixed-integer linear program formulation (MILP) which assigned arrival times at all of the conflict points on a vehicle’s path using a specific acceleration. Second, this AIM+ formulation was integrated with Gurobi and Aimsun using Aimsun’s Application Programming Interface (API). Lastly, this AIM capability that was integrated with microsimulation was utilized to model AIM+ for different scenarios. Different parameters were considered like safety buffer, maximum speed, intersection demand, turn ratio, and approach ratio. Furthermore, AIM+ was compared to other forms of intersection control such as traffic signal control and another type of AIM based on a “first-come first-served” principle. The research conducted in this study was unique because optimization-based AIM had not yet been integrated into off-the-shelf popular simulation software.

The results that have been gathered from this study are very promising for the efficiency of AIM+. The robustness of AIM+ as a form of AIM was clearly demonstrated with the results that were presented in the sensitivity analyses. Despite the fluctuation of the various parameters in the analysis, the measures of effectiveness used (delay and average speed) were barely affected. Additionally, significant reductions in delay and significant increases in average speed were observed when AIM+ was compared to a standard traffic signal. Even when comparing AIM+ to AIM with FCFS, AIM+ performed better in all metrics.

6.2 Future Work

Although the improvements that AIM+ can bring to road users are major, several things must still be studied and implemented into AIM+ before it can be realistically used at real intersections. The results that were presented in the sensitivity analyses were based on scenarios that were more idealistic. Upon expansion of AIM+, further simulation scenarios should be tested to better understand its overall capabilities and efficiency. While several of the proposed aspects of the potential future work can bring improvements to safety and practicability, they could also have an impact on the efficiency of AIM+. Specifically, if large safety buffers are used to prevent issues with malfunctions or if BSM-triggered halts are required even in false alarm cases, efficiency could be significantly impacted.

First, the model of AIM+ proposed in this study could be extended to a multi-intersection network. When considering a corridor or transportation network of several consecutive AIM-equipped intersections, there may be even more savings in delay or improvements in vehicle speed. These consecutive intersections could be designed to be coordinated even with AIM+. This could be accomplished by having an intersection manager automatically send vehicle information to the next intersection manager for the vehicles that would exit the intersection before the next run of Gurobi. This would allow the next intersection to know about incoming vehicle locations and speeds as soon as possible. However, since networks with consecutive intersections may have shorter links (and hence shorter travel times), vehicles need to have sufficient time to output their information, receive an intersection reservation, and modify their speed appropriately.

Secondly, the formulation could be modified to allow for fluctuating vehicle accelerations through the intersection. This would give the intersection manager yet even more flexibility in solving for the most optimal vehicle paths through the intersection. Additionally, the computational efficiency of the proposed AIM+ model could be improved within the Aimsun API and Gurobi solver. This would allow for the testing of even larger demands or even larger safety buffers within a reasonable amount of time. This is especially important if AIM+ is utilized at multiple intersections since multiple yet different optimization problems would need to be solved simultaneously.

Lastly, the AIM+ formulation should be developed further in order to accommodate other modes of traffic. This should include at least pedestrians, cyclists, transit vehicles, and shared vehicles. Since this study only concerned automobiles, it is important to add accommodations for these other forms of traveling as well in order to make it more realistic and precise for real-world applications. More specifically, pedestrians and cyclists can be accommodated at crosswalks by allowing them some specified crossing time. During this crossing time, vehicles should not be assigned arrival times that intersect the crosswalk that is being utilized until some time after the pedestrian or cyclist has finished crossing. This accommodation could have significant impacts on the operational capability of AIM+ since pedestrians would require much more time to cross an intersection than an automobile. Future simulations could show that the intersection efficiency would not be as impacted by pedestrians if platoons of vehicles moved through the intersection instead of random arrivals of vehicles. Additionally, in order to ensure that AIM+ is a socially equitable technology, modifications should include prioritizing these other modes of traffic (i.e. having transit vehicles receive assigned arrival times that are earlier than what is assigned for automobiles). While this can be easily done within the objective function of the formulation, a prioritization scheme should be developed so that vehicles with higher occupancies receive more priority than other vehicles.

6.3 Future of Autonomous Vehicles and AIM

As this research clearly indicated, the improvements to efficiency that both autonomous vehicles and AIM can bring to our society is revolutionary. Furthermore, the improvements in safety are opportunities that cannot be ignored. As policy for autonomous vehicles continues to take a more definite shape, these technologies become closer to being a reality in the transportation industry. With further research that will be conducted on these topics, it is obvious that more groundbreaking discoveries and concepts will be realized that will further advance the transportation world.

BIBLIOGRAPHY

[1] Dresner, Kurt, 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, pp. 530-537. IEEE Computer Society, 2004.

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

[3] Naumann, Rolf, Rainer Rasche, and Jürgen Tacken. "Managing autonomous vehicles at intersections." IEEE Intelligent Systems and their Applications 13, no. 3 (1998): 82-86.

[4] Dresner, Kurt, and Peter Stone. "Traffic intersections of the future." In Proceedings of the national conference on artificial intelligence, vol. 21, no. 2, p. 1593. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[5] Dresner, Kurt, and Peter Stone. "Mitigating catastrophic failure at intersections of autonomous vehicles." In Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3, pp. 1393-1396. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[6] Quinlan, Michael, Tsz-Chiu Au, Jesse Zhu, Nicolae Stiurca, and Peter Stone. "Bringing simulation to life: A mixed reality autonomous intersection." In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 6083-6088. IEEE, 2010.

[7] De La Fortelle, Arnaud, and Xiangjun Qian. "Autonomous driving at intersections: combining theoretical analysis with practical considerations." In ITS World Congress 2015.

[8] Fajardo, David, Tsz-Chiu Au, S. Waller, Peter Stone, and David Yang. "Automated intersection control: Performance of future innovation versus current traffic signal control." Transportation Research Record: Journal of the Transportation Research Board 2259 (2011): 223-232.

[9] Dresner, Kurt, and Peter Stone. "Multiagent traffic management: An improved intersection control mechanism." In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 471-477. ACM, 2005.

[10] Li, Zhixia, Madhav V. Chitturi, Dongxi Zheng, Andrea R. Bill, and David A. Noyce. Next-generation intersection control algorithm for autonomous vehicles. TRB 92nd Annual Meeting Compendium of Papers: No. 13-2185. 2013.

[11] Li, Zhixia, Madhav Chitturi, Dongxi Zheng, Andrea Bill, and David Noyce. "Modeling reservation-based autonomous intersection control in VISSIM." Transportation Research Record: Journal of the Transportation Research Board 2381 (2013): 81-90.

[12] Dresner, Kurt, and Peter Stone. "Replacing the Stop Sign: Unmanaged Intersection Control." In the Fifth Workshop on Agents in Traffic and Transportation Multiagent Systems, pp. 94-101. 2008.

[13] Makarem, Laleh, Minh-Hai Pham, André-Gilles Dumont, and Denis Gillet. "Microsimulation modeling of coordination of automated guided vehicles at intersections." Transportation Research Record: Journal of the Transportation Research Board 2324 (2012): 119-124.

[14] Ball, Rudi, and Naranker Dulay. "Enhancing traffic intersection control with intelligent objects." Urban Internet of Things Towards Programmable Realtime Cities (2010).

[15] Lee, Joyoung, and Byungkyu Park. "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment." IEEE Transactions on Intelligent Transportation Systems 13, no. 1 (2012): 81-90.

[16] Kamal, Md Abdus Samad, Jun-ichi Imura, Akira Ohata, Tomohisa Hayakawa, and Kazuyuki Aihara. "Coordination of automated vehicles at a traffic-lightless intersection." In Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on, pp. 922-927. IEEE, 2013.

[17] Zohdy, Ismail H., Raj Kishore Kamalanathsharma, and Hesham Rakha. "Intersection management for autonomous vehicles using iCACC." In Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, pp. 1109-1114. IEEE, 2012.

[18] Zhu, Feng, and Satish V. Ukkusuri. "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment." Transportation Research Part C: Emerging Technologies 55 (2015): 363-378.

[19] Dresner, Kurt, and Peter Stone. "Multiagent traffic management: Opportunities for multiagent learning." In Learning and Adaption in Multi-Agent Systems, pp. 129-138. Springer, Berlin, Heidelberg, 2006.

[20] Hausknecht, Matthew, Tsz-Chiu Au, and Peter Stone. "Autonomous intersection management: Multi-intersection optimization." In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 4581-4586. IEEE, 2011.

[21] Au, Tsz-Chiu, and Peter Stone. "Motion Planning Algorithms for Autonomous Intersection Management." In Bridging the Gap between Task and Motion Planning. 2010.

[22] Au, Tsz-Chiu, Neda Shahidi, and Peter Stone. "Enforcing Liveness in Autonomous Traffic Management." In AAAI. 2011.

[23] Shahidi, Neda, Tsz-Chiu Au, and Peter Stone. "Batch reservations in autonomous intersection management." In The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3, pp. 1225-1226. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

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

[25] Vasirani, Matteo, and Sascha Ossowski. "A market-inspired approach for intersection management in urban road traffic networks." Journal of Artificial Intelligence Research 43 (2012): 621-659.

[26] Levin, Michael W., and Stephen D. Boyles. "Intersection auctions and reservation-based control in dynamic traffic assignment." Transportation Research Record: Journal of the Transportation Research Board 2497 (2015): 35-44.

[27] Schepperle, Heiko, and Klemens Böhm. "Auction-based traffic management: Towards effective concurrent utilization of road intersections." In E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on, pp. 105-112. IEEE, 2008.

[28] Levin, Michael W., and Stephen D. Boyles. "A multiclass cell transmission model for shared human and autonomous vehicle roads." Transportation Research Part C: Emerging Technologies 62 (2016): 103-116.
[29] Levin, Michael W., Stephen D. Boyles, and Rahul Patel. "Paradoxes of reservation-based intersection controls in traffic networks." Transportation Research Part A: Policy and Practice 90 (2016): 14-25.
[30] Matteo, V., and O. Sascha. "Evaluating policies for reservation-based intersection control." In Proceedings of the 14th Portuguese Conference on Artificial Intelligence, EPIA, vol. 9, pp. 21-25. 2009.
[31] de La Fortelle, Arnaud. "Analysis of reservation algorithms for cooperative planning at intersections." In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pp. 445-449. IEEE, 2010.
[32] Levin, Michael W., Hagen Fritz, and Stephen D. Boyles. "On optimizing reservation-based intersection controls." IEEE Transactions on Intelligent Transportation Systems 18, no. 3 (2017): 505-515.
[33] Zhang, Kailong, Arnaud De La Fortelle, Dafang Zhang, and Xiao Wu. "Analysis and modeled design of one state-driven autonomous passing-through algorithm for driverless vehicles at intersections." In Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on, pp. 751-757. IEEE, 2013.
[34] Chiang, Yi-Jen, James T. Klosowski, Changkil Lee, and Joseph SB Mitchell. "Geometric algorithms for conflict detection/resolution in air traffic management." In Decision and Control, 1997., Proceedings of the 36th IEEE Conference on, vol. 2, pp. 1835-1840. IEEE, 1997.
[35] Rey, David, Christophe Rapine, Rémy Fondacci, and Nour-Eddin El Faouzi. "Minimization of potential air conflicts through speed regulation." Transportation Research Record 2300, no. 1 (2012): 59-67.
[36] Rey, David, Christophe Rapine, Rémy Fondacci, and Nour-Eddin El Faouzi. "Subliminal speed control in air traffic management: Optimization and simulation." Transportation Science 50, no. 1 (2015): 240-262.
[37] Yan, Fei, Mahjoub Dridi, and Abdellah El Moudni. "Autonomous vehicle sequencing algorithm at isolated intersections." In Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on, pp. 1-6. IEEE, 2009.
[38] Zohdy, Ismail, and Hesham Rakha. "Optimizing driverless vehicles at intersections." In 10th ITS World Congress Vienna, Austria. 2012.
[39] Lee, Joyoung, Byungkyu Brian Park, Kristin Malakorn, and Jaehyun Jason So. "Sustainability assessments of cooperative vehicle intersection control at an urban corridor." Transportation Research Part C: Emerging Technologies 32 (2013): 193-206.
[40] Kim, Kyoung-Dae, and Panganamala Ramana Kumar. "An MPC-Based Approach to Provable System-Wide Safety and Liveness of Autonomous Ground Traffic." IEEE Trans. Automat. Contr. 59, no. 12 (2014): 3341-3356.
[41] Wuthishuwong, Chairit, and Ansgar Traechtler. "Vehicle to infrastructure based safe trajectory planning for Autonomous Intersection Management." In ITS telecommunications (ITST), 2013 13th international conference on, pp. 175-180. IEEE, 2013.
[42] Gregoire, Jean, Silvère Bonnabel, and Arnaud de La Fortelle. "Optimal cooperative motion planning for vehicles at intersections." arXiv preprint arXiv:1310.7729 (2013).

[43] Altché, Florent, and Arnaud De La Fortelle. "Analysis of optimal solutions to robot coordination problems to improve autonomous intersection management policies." In Intelligent Vehicles Symposium (IV), 2016 IEEE, pp. 86-91. IEEE, 2016.

[44] Miculescu, David, and Sertac Karaman. "Polling-systems-based control of high-performance provably-safe autonomous intersections." In Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, pp. 1417-1423. IEEE, 2014.

[45] Wu, Jia, Abdeljalil Abbas-Turki, and Abdellah El Moudni. "Contextualized traffic controlling at isolated urban intersection." In The 14th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2010, vol. 29, pp. 1-6. 2010.

[46] Alonso, Javier, Vicente Milanés, Joshué Pérez, Enrique Onieva, Carlos González, and Teresa De Pedro. "Autonomous vehicle control systems for safe crossroads." Transportation Research Part C: Emerging Technologies 19, no. 6 (2011): 1095-1110.

[47] Olaverri-Monreal, Cristina, Pedro Gomes, Michelle Krüger Silvéria, and Michel Ferreira. "In-vehicle virtual traffic lights: A graphical user interface." In Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on, pp. 1-6. IEEE, 2012.

[48] Dresner, Kurt, and Peter Stone. "Human-usable and emergency vehicle-aware control policies for autonomous intersection management." In Fourth International Workshop on Agents in Traffic and Transportation (ATT), Hakodate, Japan. 2006.

[49] Dresner, Kurt M., and Peter Stone. "Sharing the Road: Autonomous Vehicles Meet Human Drivers." In IJCAI, vol. 7, pp. 1263-1268. 2007.

[50] Bento, Luís Conde, Ricardo Parafita, Sérgio Santos, and Urbano Nunes. "Intelligent traffic management at intersections: Legacy mode for vehicles not equipped with v2v and v2i communications." In Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on, pp. 726-731. IEEE, 2013.

[51] Qian, Xiangjun, Jean Gregoire, Fabien Moutarde, and Arnaud De La Fortelle. "Priority-based coordination of autonomous and legacy vehicles at intersection." In Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, pp. 1166-1171. IEEE, 2014.

[52] NHTSA. "Vehicle-to-Vehicle Communication." NHTSA. October 30, 2018. Accessed December 09, 2018. https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication.

[53] Liu, Jun, and Asad J. Khattak. "Delivering improved alerts, warnings, and control assistance using basic safety messages transmitted between connected vehicles." Transportation research part C: emerging technologies 68 (2016): 83-100.

[54] Azimi, Seyed Reza, Gaurav Bhatia, Ragunathan Raj Rajkumar, and Priyantha Mudalige. "Vehicular networks for collision avoidance at intersections." SAE International Journal of Passenger Cars-Mechanical Systems 4, no. 2011-01-0573 (2011): 406-416.

[55] Campbell, John L. Human factors guidelines for road systems. Vol. 600. Transportation Research Board, 2012.

[56] Gurobi. "Gurobi Optimizer." Mathematical Programming Solver | Gurobi. Accessed October 15, 2018. http://www.gurobi.com/products/gurobi-optimizer.

[57] Aimsun. Aimsun Next User's Manual. Aimsun Next 8.2. 2014-2017.