The Report Committee for Prashanth Venkatraman
Certifies that this is the approved version of the following report:

Planning for Autonomous Vehicles: Ridesharing and Traffic Control

APPROVED BY
SUPERVISING COMMITTEE:

Stephen D. Boyles, Supervisor

Christian G. Claudel

Planning for Autonomous Vehicles: Ridesharing and Traffic Control

by

Prashanth Venkatraman

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2018

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Dr. Stephen Boyles for affording me the opportunity to work with him over the last year, for his patience with my work, his enthusiasm and immense knowledge on the subject. His guidance helped me through the entirety of my research and in the writing of this report. I would also like to thank him for leading me diverse and exciting projects during my time here.

I would also like to thank Dr. Christian Claudel for agreeing to read my Master's report, for his insightful comments and guidance through the writing of this report.

My sincere gratitude to Dr. Michael Levin for introducing me to the area of autonomous vehicles planning problems and for letting me pick his brain on the many topics I have encountered through the course of my work here.

I would like to thank my fellow lab mates at The University of Texas at Austin: Rahul Patel, William Alexander, Priyadarshan Patil, Cesar Yahia and Dongxu He for the sleepless nights we have spent brainstorming for ideas to move projects forward and all the fun we have had in the process. Without them, this report would not have been possible. I have learnt immensely from each one of them and I will cherish every minute we have spent together.

Lastly, I would like to thank my parents, Venkatraman Natarajan and Vasantha Venkatraman and my sister, Priya Kumar for supporting me throughout my pursuit to becoming a better engineer and a better human being.

Abstract

Planning for Autonomous Vehicles: Ridesharing and Traffic Control

Prashanth Venkatraman, M.S.E.

The University of Texas at Austin, 2018

Supervisor: Stephen D. Boyles

This report address two problems that could drive the adoption of autonomous vehicles (AV) – shared autonomous vehicle routing (SAV) problem and the autonomous intersection management system (AIM) location problem. The SAV routing problem is finding the optimal SAV to passenger matching as well as the SAV route choice. Since widespread use of SAVs would have significant effects on traffic congestion, we develop a new tabu search heuristic for the SAV routing problem under the influence of traffic congestion. The algorithm aims to minimize traveler's travel time. It considers several adjacent solutions by repeatedly swapping travelers between SAV routes. A nearest traveler neighborhood is defined to choose travelers to consider for the swap procedure. The Sioux Falls network is used to test the performance of the heuristic with varying demand and fleet sizes. The heuristic is found to produce encouraging results in reducing the total passenger travel time. A series of experiments are performed to understand the sensitivity of the heuristic to its parameters and the effects of congestion.

The AIM location problem is the problem of optimally locating AIMs in a network so as to improve the experienced travel times in the network. Traditional traffic signals are inefficient in taking advantage of the benefits of AVs. Previous studies show that full adoption of AIMs in a network is not necessarily an improvement. This report aims to develop a framework which can be used to identify the intersections where an

implementation of AIMs is beneficial. To do so, two models are proposed. First, a regression model is developed to classify intersections based on their performance. Second, the AIM location problem is formulated as an optimization problem and a genetic algorithm is developed to identify the optimal distribution of AIMs in a network. Both approaches are tested on the downtown Austin network and are compared for their performance.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

The development of autonomous vehicles (AVs) has the potential to transform transportation as we know it. Studies in the past have shown that AVs have significant benefits when compared to human vehicles. AVs could reduce CO_2 consumption by 300 million tons per year leading to significant savings in emissions *(1)*. Currently, 30,000 people die due to road accidents and nearly 90% of all road accidents are due to human error. Fagnant and Kockelman *(2)* found that widespread adoption of AVs could reduce accidents due to human error almost entirely. AVs eliminate human error in judgement and thereby, result in safer roads. The most important benefit is the fact that AVs result in reduced reaction times and thus improved efficiency in travel. Advances in technology in the last decade and specifically, of general Artificial Intelligence systems, it is increasingly evident that AVs will transform mobility.

Yes, autonomous vehicles provide significant benefits to mobility. But how close are we to seeing AVs on the roads? Ever since the success of the first DARPA challenge, companies have invested heavily in the development of AVs. Post the success of Sebastian Thrun's Stanford team at the DARPA challenge in 2005, Google has invested heavily in bringing AVs to the street. In the last couple years, the fight to get the first AV on the roads has intensified with Alphabet's Waymo, Uber and GM gaining approval from several states to test their AVs on roads. In 2016, GM bought Cruise, a company building AV technology. Last year, GM's autonomous vehicle company, Cruise Automation showed that they had achieved level 4 autonomy on the streets of San Francisco. In the spring of 2018, Alphabet's Waymo started a pilot program in Phoenix where passengers could use an AV fleet to complete their trips. The company plans to expand the service to other parts of the country by the end of the year. Similarly, big automobile manufacturers have committed billions of dollars to produce commercial AVs in the next decade. Thus, AV industry is on the brink of rolling out fully autonomous vehicles for public use. Although the benefits of AVs are encouraging, the cities of the world are unprepared for such technology. Thus cities will have to transform themselves before AVs can hit the roads.

Smart cities will pave the way for this change by creating an ecosystem conducive to the sustainable adoption of AVs. The concept of a smart city has attracted worldwide interest in the last few years from companies, governments and universities alike. Although there seem to be many definitions to a smart city, a common theme among all of these is using information and communication technologies to make a city more intelligent and efficient across its many functions - administration, education, mobility. An intelligent mobility system could take many forms. In fact, a city a with a bus service which can be tracked using a transit mobile app could be categorized as a smart city. Thus, smart cities offer the potential of creating a connected environment which provides connected vehicles with vital information to make them more efficient such as information about road conditions and travel times.

In this report, we focus on the applications of an AV system and their relative benefits. Primarily, we focus on two novel problems that have gained traction in literature in recent times. First, we will discuss the problem of routing a fleet of shared autonomous vehicles (SAV) while considering the effects of traffic congestion. Previous studies have used agent based simulations to study the effects of AVs. However, most studies have not considered the effects of congestion caused by a large fleet of AVs and their effect on travel times. In the first section of the paper, we will present a mathematical formulation for this model which is based on the cell transmission model of traffic flow and present a tabu search heuristic designed to solve it.

In the second section of the report, we discuss the problem of optimally locating autonomous intersection management systems (AIMs), a technology designed to improve the efficiency of the road network at traffic intersections. Although traffic signals perform well under the condition of human driven vehicles, they are not designed to take advantage of the increased sensitivity and precision of autonomous vehicles. Previous studies have shown that traffic signals are not an efficient means of controlling the flow of traffic in a network. AIMs were introduced by Dresner & Stone *(33,34)* utilize the relative advantages provided by AVs, especially at intersections. Installation of such systems at intersections will go a long way to improve the performance of AVs and the network as well. In this report, we will address the problem of optimally placing AIMs in a city network which produces the best system wide benefit in total system travel time (TSTT). We develop a framework to characterize intersections which are best suited to being modeled as reservations and where to place them in a network.

Chapter 2
Shared Vehicle Routing Problem

The SAV routing problem is the problem of optimally routing a fleet of SAVs to meet the demands of passengers in a network. Initially, a set of travelers and a fleet of vehicles are distributed throughout the traffic assignment zones. Each traveler d has an origin r, a destination s, and a desired departure time t. Our goal is to assign a SAV to serve traveler d such that the traveler arrives as early as possible, but departs at or after the traveler's desired departure time. Now, a SAV v may be located at any zone in the network, including r. Depending on the location of v, it may have to make an empty repositioning trip to serve traveler d. The section develops a mixed integer linear programming formulation of the SAV routing problem, for which we will develop a tabu search algorithm.

For this paper, we focus only on SAV demand, and therefore do not include the route choices of non-SAV trips. This formulation differs from the one developed in Levin *(6)* in that it has been tailored to represent an agent-based SAV simulation using the cell transmission model *(28)*. Vehicle flows are disaggregated by whether they are carrying travelers or empty, and variables are restricted to integer values.

2.1 Literature Review

2.1.1 Shared autonomous vehicles

Currently, personal vehicles spend most of their time unused while parked at their owner's destination. Previous studies found that AVs could travel empty to avoid parking fees *(12)*. de Almeida Correia and van Arem *(13)* proposed that privately owned AVs could be used to serve multiple household members since AVs could undertake empty trips and developed a user equilibrium formulation. However, AVs could be used in ride-sharing arrangements. Fagnant and Kockelman *(3)* proposed an SAV system where multiple travelers could make use of a fleet of SAVs by scheduling trips at different times. The lack of drivers makes service costs of SAVs comparable with personal vehicles. Thus, SAVs could be used in conjunction with personal vehicles or replace them altogether.

Several studies explored the replacement of personal vehicles with SAVs *(3, 4, 14)*. Fagnant and Kockelman *(3)* observed that one SAV could replace 11 personal vehicles in a grid network. Chen et al. *(14)* found a replacement rate of one SAV for 3.7 personal vehicles which included electric SAVs in their study.

Despite the potential benefits of an SAV system, there has been little focus on the SAV routing problem. Previous studies used heuristics with agent-based simulations to route SAVs. However, these studies did not consider the effects of SAV route choice on traffic congestion, which would be considerable for large SAV fleets. Levin et al. *(5)* found that SAVs could make congestion worse than personal vehicles do due to empty repositioning. Most previous studies did not take these effects into consideration *(e.g. 3, 4, 14)*.

Optimal solutions to the SAV routing problem could reduce travel times, operating costs and waiting costs. Burns et al. *(15)* found that smaller fleet sizes reduced congestion and thus improved wait times and vehicle utilization in Manhattan. Alonso-Mora et al. *(16)* developed a mathematical model and assignment heuristic for real-time ridesharing and tested it on SAV cab rides in New York City. Although the SAV routing problem is similar to the dial-a-ride problem (DARP), it differs significantly from previous work on DARPs because the volume of vehicles involved.

2.1.2 Dial-a-ride problems

VRPs are the class of problems that involve the routing a fleet of vehicles to provide some type of service to customers distributed over a network *(7)*. The network contains a demand of travelers, each to be picked up at one location and dropped at another. The fleet of vehicles seek to provide mobility to the travelers and meet demand. The problem has many variants depending on the type of service provided. The variant relevant to the SAV routing problem is the DARP *(17)*. In this section, we seek to explore variants of

DARPs to provide context for the SAV routing problem and the heuristic we have devised.

DARPs can be classified into static and dynamic DARPs based on the nature of demand. Static DARPs *(18)* assume that all demand is known in advance and routes can be fixed in advance. Dynamic DARPs *(19)* assume a time-varying demand which appears after vehicle operations have begun. Similarly, the SAV routing problem could have static or dynamic demand. Dynamic demand increases the complexity of routing algorithms as they have to account for the uncertainty of future demand.

DARPs also have traveler constraints. Traveler constraints have been studied extensively in literature *(e.g. 20, 21)*. They often appear as desired time windows for traveler departure and arrival. Such constraints greatly expand the feasible region and increase the complexity of the formulation. Additionally, they can make the SAV routing problem infeasible due to SAV's effects on congestion and travel times. Therefore, in this paper we consider the SAV routing problem with static demand, and travelers with a desired departure time, but not a time window for arrival.

DARPs are typically formulated as integer-linear programs and solution algorithms include exact methods *(22, 23)*. Since DARPs need to be solved quickly (or in real-time), previous work has explored a variety of heuristics and metaheuristics to solve them *(e.g. 24, 25)*. The algorithm we propose differs from previous work due to the nature of the SAV routing problem. We study a problem with a large fleet of vehicles. DARPs generally assume fixed travel times between nodes in a network. Consequently, there is little literature on DARPs that are affected by and cause congestion. In this paper, we propose a heuristic to route SAVs under the effects of congestion.

2.1.3 Tabu Search Optimization and the VRP

Several methods have been developed to solve VRPs. The literature on solution algorithms can be classified as classical heuristics and metaheuristics. In this paper we focus on a family of metaheuristics called tabu search. Tabu search is an algorithm where neighbors of a solution are examined to identify the best improvement direction at each iteration.

Tabu search is one of the most widely used metaheuristics in solving optimization problems today. It has certain distinct features like the use of an adaptive memory to store information related to the search process. Tabu search looks through the solution space in a deterministic manner. When a better neighbor is found, it replaces the current solution. To avoid stagnating at local optima, tabu search accepts non-improving neighboring solutions. This policy may result in cycling.However, it avoids cycling by storing in memory neighbors previously visited in its recent search trajectory. This short term memory structure is called a tabu list. At each iteration of tabu search, we update the short-term memory. However, storing all visited solutions is time and space consuming. The tabu list usually stores distinct attributes of previous solutions and contains a constant number of tabu moves.

Several studies have employed tabu search to solve VRPs *(8, 9, 10, 11)*. One of the first attempts to apply tabu search to the VRP is due to Willard *(8)*, where the solution is transformed into a giant tour by replication of the depot. Neighborhoods are defined as all feasible solutions that can be reached from the current solution by 2-opt or 3-opt exchanges. Gendreau, Hertz, and Laporte *(26)* developed Taburoute with a similar structure but with several innovative features. For instance, the neighborhood is defined as all the solutions that can be reached from the current solution by removing a vertex from its current route and inserting it into another route containing one of its p neighbors using a Generalized Insertion method (GENI) *(27)*. Taillard *(11)* proposed an algorithm similar to Taburoute, where the neighborhood is defined by λ-interchange mechanism and uses standard insertions. A novel feature of Taillard's algorithm is the decomposition of the main problem into sub-problems by partitioning into concentric sectors centered at the depot.

2.2 Formulation

2.2.1 Traffic network

Consider a traffic network $G = (N, A)$ with set of nodes N and set of links A. Nodes are separated into junctions and zones. Let $Z \subseteq N$ be the set of zones - Locations that satisfy one or both of the following properties: 1) passenger origins or destinations; or 2) depots at which SAVs may park indefinitely. SAVs only travel to a zone to pick-up/drop-off a traveler or to park, and therefore zones are the origins and destinations of SAV trips. Each SAV is initially parked at a zone r. To carry travelers, or travel to a traveler's origin, the SAV will depart from r for another zone s, and travel through the network to reach s.

To combine the formulation with simulation, we track SAV movements and demand at discrete time steps of Δt. Our simulation uses a time step of $\Delta t = 6$ seconds, which is a typical value for our traffic flow model. Let T be the end of the time horizon under consideration. We choose T large enough that it gives the SAVs enough time to serve all travelers. Due to the difficulty in solving this formulation, we created an agent-based simulation integrated with dynamic network loading to evaluate solutions. SAVs are modeled as autonomous agents that move through the network carrying travelers and following an exogenously specified route assignment. For more details on this agent-based simulation, we refer the reader to Levin et al. *(5)*.

2.2.2 Traffic flow

We use the cell transmission model *(28)*, a Godunov approximation of the kinematic wave model (*29)* for traffic flow that has been used in both analytical *(30)* and simulation-based models *(31)*. Each link $a \in A$ is divided into an ordered set of *cells*, discrete spatial intervals that vehicles travel through. The cell length, Δx_a, is chosen so that $\frac{\Delta x_a}{\Delta t} = u_a^{\mathrm{f}}$, where u_a^{f} is the free flow speed on link a. Therefore, each vehicle can travel through at most one cell per time step. Let C be the set of all cells in the network. The primary decision variables are the transition flows between cells (and between cells and zones). These determine when SAVs depart, where they are destined to, which route they take, and when they pick up travelers.

Let $n_i^s(t)$ be the number of SAVs in cell i destined for $s \in Z$ carrying travelers, and let $\tilde{n}_i^s(t)$ be the number of SAVs in cell i destined for s that are empty. Let $y_{ij}^s(t)$ and $\tilde{y}_{ij}^s(t)$ be the transition flows of traveler-carrying and empty SAVs, destined for s, from cell i to cell j at time t, respectively. Cell occupancies evolve via conservation of flow:

$$n_j^s(t+1) = n_j^s(t) + \sum_{i \in \Gamma^-(j)} y_{ij}^s(t) - \sum_{k \in \Gamma^+(j)} y_{jk}^s(t) \quad (2a)$$

$$\tilde{n}_j^s(t+1) = \tilde{n}_j^s(t) + \sum_{i \in \Gamma^-(j)} \tilde{y}_{ij}^s(t) - \sum_{k \in \Gamma^+(j)} \tilde{y}_{jk}^s(t) \quad (2b)$$

where $\Gamma^+(i)$ and $\Gamma^-(i)$ are the forward and backward star (cells) of i, respectively. Notice that the decision variables $y_{ij}^s(t)$ and $\tilde{y}_{ij}^s(t)$ determine the turning movements at i and SAV route choice. We distinguish between the route choice of SAVs carrying travelers ($y_{ij}^s(t)$) and empty SAVs ($\tilde{y}_{ij}^s(t)$) because empty trips do not affect the total person travel time.

Transition flows are constrained by the kinematic wave theory. Let $S_i(t)$ be the *sending flow* of cell i at time t – the maximum number of vehicles that could leave i at time t. Let $R_i(t)$ be the *receiving flow* of cell i at time t – the maximum number of vehicles that could enter i at t. Then

$$\sum_{s \in Z} \sum_{j \in \Gamma^+(i)} \left(y_{ij}^s(t) + \tilde{y}_{ij}^s(t) \right) \leq S_i(t) \quad (3)$$

and

$$\sum_{s \in Z} \sum_{i \in \Gamma^-(j)} \left(y_{ij}^s(t) + \tilde{y}_{ij}^s(t) \right) \leq R_j(t) \quad (4)$$

The sending and receiving flows are determined by the fundamental diagram:

$$\sum_{j\in\Gamma^+(i)} y_{ij}^s(t) \le n_i^s(t) \tag{5a}$$

$$\sum_{j\in\Gamma^+(i)} \tilde{y}_{ij}^s(t) \le \tilde{n}_i^s(t) \tag{5b}$$

$$S_i(t) \le q_i^{\max} \tag{5c}$$

$$R_i(t) \le q_i^{\max} \tag{6a}$$

$$R_i(t) \le \frac{w_i}{u_i^{\mathrm{f}}}\left(n_i^{\max} - n_i(t)\right) \tag{6b}$$

where $q_i^{\max}$ and $n_i^{\max}$ are the capacity and maximum occupancy (jam density) of cell i, respectively, and $\frac{w_i}{v_i^{\mathrm{f}}}$ is the ratio of the congested wave speed to the free flow speed.

2.2.3 SAV-traveler behavior

SAVs pick-up and drop-off travelers at zones, and in addition, may park at some of the zones while not in use. Let $p_r(t)$ be the number of SAVs parked in zone r at time t. The initial locations of SAVs, $p_r(0)$, is an input parameter, and additionally defines the SAV fleet size. All SAVs must be parked at the end of the time horizon:

$$\sum_{r\in Z} p_r(0) = \sum_{r\in Z} p_r(T) \tag{7}$$

In a slight overload of the notation, let $y_{iz}^s(t)$ and $\tilde{y}_{iz}^s(t)$ be the number of traveler-carrying and empty SAVs moving from cell i to zone z destined for zone s at time t, respectively. SAVs destined for s cannot enter other zones before arriving at s. Therefore if $z \ne s$, then $y_{iz}^s(t) = \tilde{y}_{iz}^s(t) = 0$. Similarly, let $y_{zi}^s(t)$ and $\tilde{y}_{zi}^s(t)$ be the number of traveler-carrying and empty SAVs entering cell i from zone z destined for zone s at time t. Also let $\Gamma^{-}(z)$ and $\Gamma^{+}(z)$ be the predecessor and successor cells of zone z, respectively, and include z in the forward and backward star of cells where appropriate.

When SAVs arrive at a zone, they become parked. Only SAVs parked at zone r can depart from r for other zones. Therefore, the number of parked SAVs evolves through conservation of flow:

$$p_r(t+1) = p_r(t) - \sum_{s \in Z} \sum_{i \in \Gamma^{+}(r)} \left(y_{ri}^{s}(t) + \tilde{y}_{ri}^{s}(t) \right) + \sum_{i \in \Gamma^{-}(r)} \left(y_{ir}^{r}(t) + \tilde{y}_{ir}^{r}(t) \right) \tag{8}$$

The number of departing SAVs cannot exceed the number of parked SAVs:

$$\sum_{s \in Z} \sum_{i \in \Gamma^{+}(r)} \left(y_{ri}^{s}(t) + \tilde{y}_{ri}^{s}(t) \right) \leq p_r(t) \tag{9}$$

Zones without available parking can be modeled by requiring equation (9) to hold with equality.

Let $d_r^s(t)$ be the exogenous number of travelers wishing to depart r for s at time t. Also let $w_r^s(t)$ be the number of travelers waiting at r for travel to s at time t. Travelers can only depart when picked up by an SAV. Therefore, the numbers of waiting travelers evolve as follows:

$$w_r^s(t+1) = w_r^s(t) + d_r^s(t) - \sum_{i \in \Gamma^{+}(r)} y_{ri}^{s}(t) \tag{10}$$

with

$$\sum_{i \in \Gamma^{+}(r)} y_{ri}^{s}(t) \leq w_r^s(t) \tag{11}$$

$w_r^s(t)$ maintains a waiting list of travelers, and the demand $d_r^s(t)$ is added to this waiting list at each time step. Departing passengers are removed from the waiting list. To ensure that SAVs service all travelers, we add the following constraint:

$$w_r^s(T) = 0 \tag{12}$$

2.2.4 Mixed integer program

We now combine the above constraints into a mixed integer linear program:

$$\min \quad Z = \sum_{t=0}^{T} \left(\sum_{s \in Z} \left(\sum_{i \in C} n_i^s(t) + \eta \sum_{r \in Z} w_r^s(t) \right) \right)$$

subject to:

$$n_j^s(t+1) = n_j^s(t) + \sum_{i \in \Gamma^-(j)} y_{ij}^s(t) - \sum_{k \in \Gamma^+(j)} y_{jk}^s(t) \qquad \forall j \in C, \forall s \in Z, \forall t \in [0,T]$$

$$\tilde{n}_i^s(t+1) = \tilde{n}_j^s(t) + \sum_{i \in \Gamma^-(j)} \tilde{y}_{ij}^s(t) - \sum_{k \in \Gamma^+(j)} \tilde{y}_{jk}^s(t) \qquad \forall j \in C, \forall s \in Z, \forall t \in [0,T]$$

$$\sum_{j \in \Gamma^+(i)} \left(y_{ij}^s(t) + \tilde{y}_{ij}^s(t) \right) \le q_i^{\max} \qquad \forall i \in C, \forall t \in [0,T]$$

$$\sum_{j \in \Gamma^+(i)} y_{ij}^s(t) \le n_i^s(t) \qquad \forall i \in C, \forall s \in Z, \forall t \in [0,T]$$

$$\sum_{j \in \Gamma^+(i)} \tilde{y}_{ij}^s(t) \le \tilde{n}_i^s(t) \qquad \forall i \in C, \forall s \in Z, \forall t \in [0,T]$$

$$\sum_{s \in Z} \sum_{i \in \Gamma^-(j)} \left(y_{ij}^s(t) + \tilde{y}_{ij}^s(t) \right) \le q_j^{\max} \qquad \forall j \in C, \forall t \in [0,T]$$

$$\sum_{s \in Z} \sum_{i \in \Gamma^-(j)} \left(y_{ij}^s(t) + \tilde{y}_{ij}^s(t) \right) \le \frac{w_i}{v_i^{\mathrm{f}}} \left(n_i^{\max} - \sum_{s \in Z} \left(n_i^s(t) + \tilde{n}_i^s(t) \right) \right)$$

$$\forall i \in C, \forall t \in [0,T]$$

$$\sum_{r \in Z} p_r(0) = \sum_{r \in Z} p_r(T)$$

$$p_r(t+1) = p_r(t) - \sum_{s \in Z} \sum_{i \in \Gamma^+(r)} \left(y_{ri}^s(t) + \tilde{y}_{ri}^s(t) \right) + \sum_{i \in \Gamma^-(r)} \left(y_{ir}^r(t) + \tilde{y}_{ir}^r(t) \right)$$

$$\forall r \in Z, \forall t \in [0,T]$$

$$\sum_{s \in Z} \sum_{i \in \Gamma^+(r)} \left(y_{ri}^s(t) + \tilde{y}_{ri}^s(t) \right) \le p_r(t) \qquad \forall r \in Z, \forall t \in [0,T]$$

$$w_r^s(t+1) = w_r^s(t) + d_r^s(t) - \sum_{i \in \Gamma^+(r)} y_{ri}^s(t) \qquad \forall r \in Z, \forall s \in Z, \forall t \in [0,T]$$

$$\sum_{i \in \Gamma^+(r)} y_{ri}^s(t) \le w_r^s(t) \qquad \forall r \in Z, \forall s \in Z, \forall t \in [0,T]$$

$$w_r^s(T) = 0 \qquad \forall r \in Z, \forall s \in Z$$

$$y_{ij}^s(t) \in \mathbb{Z}_+ \qquad \forall s \in Z, \forall i \in C, \forall j \in C, \forall t \in [0,T]$$

$$\tilde{y}_{ij}^s(t) \in \mathbb{Z}_+ \qquad \forall s \in Z, \forall i \in C, \forall j \in C, \forall t \in [0,T]$$

The objective function is to minimize the total passenger travel time, where η is the weight placed on waiting time. We used $\eta = 1$ in our experiments, but η could be calibrated based on passenger interests. We require that $p_r(0) \in \mathbb{Z}_+$ for all $r \in Z$ and that $d_r^s(t) \in \mathbb{Z}_+$ for all $(r,s) \in Z^2$ and for all times t. Combined with the constraints that $y_{ij}^s(t)$ and $\tilde{y}_{ij}^s(t)$ are integers, these ensure that all traveler and vehicle variables (parked vehicles, waiting travelers, and cell occupancies) remain integers as well through their conservation equations.

This formulation is substantially more difficult to solve than most previous work on VRPs because the traffic flow model allows congestion to form. Because this problem, a variant of the VRP problem, is NP-hard, we develop a tabu search algorithm to find a good solution on larger networks.

2.3 Network Loading

When passengers wish to depart, they enter a waiting list at their origin, and remain there until picked up by an SAV. SAVs initially are parked at some zone, and enter the network via source cells. They travel along their assigned route until reaching their destination sink cell, at which point they exit the network and park until needed again. Network loading is a forward simulation procedure; we start at $t = 0$. At each time t we calculate the state of the network at time $t + 1$, based on the state at time t and the SAV routing assignment.

The origin-destination pairs and passenger demands are exogenous variables specified with the network. This is typical of the traffic assignment literature; the origin-destination matrix is known, but the vehicle route choices are not.

2.4 Tabu search algorithm

In this section, we develop a tabu search algorithm to solve the SAV routing problem. We will give a brief description of the problem and follow with a discussion of each step of the algorithm. For the SAV routing problem, consider an initial feasible passenger-

SAV assignment S in the feasible region of the MILP. For any assignment S, we associate the objective function

$$Z(S) = \sum_{t=0}^{T}\left(\sum_{s\in Z}\left(\sum_{i\in C} n_i^s(t) + \sum_{r\in Z} w_r^s(t)\right)\right)$$

The objective function consists of two components summed over the time horizon, T- the time spent by travelers waiting for an SAV and the time spent by the traveler in traveling from origin to destination. The goal of tabu search in the VRP literature is to improve on an initial solution by swapping travelers from their paths to reach neighboring solutions. In this regard, we have developed a tabu search procedure which attempts to minimize total waiting time by swapping passengers from their current SAV with another passenger in a different SAV. At each iteration, a traveler is swapped with neighboring traveler, with a similar departure time, from a candidate list of potential neighbors, the nearest traveler neighborhood, resulting in a new traveler-SAV assignment, $S^{'}$. Let S^* denote the assignment which minimizes the objective.

We define the steps involved in tabu search below followed by an elaborate discussion about each of the steps.

1. Choose an initial traveler-SAV assignment, S. Set $S^* = S$ and $Z(S^*) = Z(S)$. Define an empty tabu list.
2. Iterate through the traveler list and perform the following steps for each traveler:
   - For each traveler d, define the traveler's nearest traveler neighborhood, N_d. Check whether the swapping d into SAV m_n is tabu(is forbidden).
   - If not, swap d with each traveler in N_d, and compute the objective $Z(S^{'})$.
   - If $Z(S^{'}) < Z(S)$, set $S^* = S$ and $Z(S^*) = Z(S^{'})$. Add the traveler and the traveler's previous SAV to the tabu list. Update the tabu list.
3. Return to step 2 for the next traveler.

2.4.1 Initial Solution

As mentioned earlier, each traveler has a desired time of departure from the traveler's origin. We assume that the traveler does not have a constrained time of arrival at the traveler's destination as this could prevent feasibility. We first define an initial solution to the SAV routing problem, and improve it through the algorithm. The procedure to find an initial solution can be summarized as follows:

a. Sort the travelers based on their desired departure times.
b. Iterate through the sorted list of travelers and assign each traveler randomly to the next available SAV.
c. Use time-dependent Dijkstra's shortest path algorithm to route each SAV from its current location to the traveler's origin and from the traveler origin to the traveler's destination for each traveler. Here, the shortest paths are based on average link travel times from the previous five minutes of simulation.
d. Compute the value of the objective for the traveler assignment, $Z(S)$ and set $S^* = S, Z(S^*) = Z(S)$. This calculation involves running an agent-based simulation.

2.4.2 Algorithm

This section describes the algorithm to find a new traveler-SAV assignment which minimizes the total person travel time at each iteration. Let us define $T(i)$ as the total travel time at iteration $i.$ Each traveler's total travel time has two components – the time spent waiting for an SAV and the time spent in traveling from the traveler's origin to destination. Now, we seek to minimize the time a traveler spends waiting for an SAV by placing each traveler in the SAV route that produces the largest decrease in traveler waiting time. To achieve this, we swap each traveler from the traveler's current SAV with another traveler in a different SAV from the other traveler's nearest traveler neighborhood. We use Dijkstra's shortest path algorithm, using average link travel times over the previous 5 minutes of simulation to route SAVs. We perform this swap procedure for every traveler in our network iteratively. We define a tabu list which is a map of traveler-taxi swaps that were performed in the recent search trajectory.

2.4.2.1 Nearest Traveler Neighborhood

The nearest traveler neighborhood is a candidate list of potential neighbors with whom a traveler can be swapped so that the total waiting time decreases. A traveler n (in SAV m_n) is said to be in traveler d's (in SAV m_d) nearest traveler neighborhood (N_d) iff:

1. The traveler has a desired departure time that is close to that of traveler d. In our algorithm, we used a departure time range of 15 minutes centered at traveler d's departure time; and
2. Assigning SAV m_n to traveler d and SAV m_t to traveler n reduces the waiting time for both travelers.

The swaps in N_d are sorted in decreasing order of the reduction in the waiting time of travelers □ and n. The size of the neighborhood can vary depending on the size of the problem, i.e. the number of travelers and SAVs in the network. In the experiments conducted, we found that a neighborhood size of 5 was sufficient. A neighborhood size larger than 5 did not necessarily improve the solutions.

2.4.2.2 Tabu List

The tabu list is a mapping of traveler and the SAVs they were swapped from in the recent search trajectory of the algorithm. Each element in the list is a tabu move, i.e. a swap of a traveler into the SAV is prohibited. The duration of time that an element in the list remains tabu depends on the size of the network and fleet used. In our experiments, we used a tabu duration of 5 iterations to perform sufficiently well. The rationale is that within five moves the algorithm diversifies enough to prevent it from visiting a previously visited solution. We have further discussed the sensitivity of the algorithm to this parameter in the results.

2.4.2.3 The Swap

For each traveler d(in SAV m_d) in the network, we perform the following swap procedure:

- We define a traveler's nearest traveler neighborhood, N_d.
- For each traveler n in N_d, we swap traveler d with traveler n to obtain S' and compute $Z(S')$. For example, consider SAV m_d with a traveler assignment of (1 5 7 9 10 16) and SAV m_n with a traveler assignment of (3 6 12 15 18). We swap traveler 9 with traveler 12 in their respective SAV routes, i.e m_d'= (1 5 7 12 10 16) and m_n' = (3 6 9 15 18). We denote this assignment as S'.
- If the move is tabu, the swap is disregarded unless $Z(S') < Z(S^*)$.
- If $Z(S') < Z(S^*)$, set $S^* = S'$and if $Z(S^*) = Z(S')$. Update the tabu list.

2.4.2.4 Stopping Criterion

We want our algorithm to exhaust the search space of all possible swaps. However, we do not know the optimal solution beforehand and hence, we wish to bound the running time of the algorithm also. Thus, it is essential to define a stopping criterion to reduce the number of wasteful iterations. We have defined a stopping criterion wherein the search stops once the algorithm reaches 1000 consecutive swaps without an improvement in the objective value. We found the algorithm to converge quickly given this criterion.

2.5 Numerical Results

This section presents the numerical experimental results used to test the heuristic. The experiments were performed on the Sioux Falls network *(32)*, which is extensively used for demonstration. We implemented the MILP and the tabu search heuristic in Java on an Intel i5-4590 CPU clocked at 3.3GHz with 12GB of memory.

We demonstrate the algorithm on the well-known Sioux Falls network which consists of 24 nodes and 76 links. The demand was based on a trip table uniformly distributed over the time horizon, and scaled proportionally.

2.5.1 Improvement from Tabu Search

To demonstrate the improvement from tabu search compared to the initial solution, we test the heuristic on a moderately congested network with a demand of 9639 travelers with a time horizon of 10 hours and varying size of the SAV fleet. We define two test measures to help with our analysis − average vehicle travel time(AVTT) and average passenger travel time(APTT). This was done to understand the effects of varying fleet size on congestion in the network and the corresponding increase in travel times. The experiment was repeated with a traveler demand of 2408, and 4823 travelers. For each demand, the experiment was repeated with SAV:traveler ratios of 1:5, 1:3 and 1:2. Table 1 shows the results of the experiments.

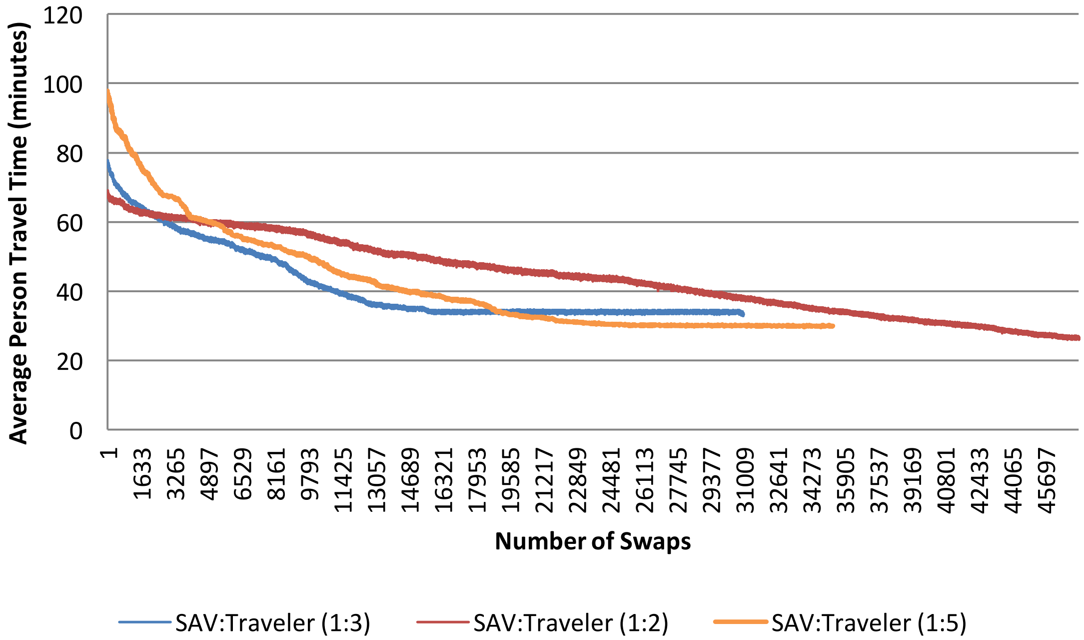
2.5.2 Effect of Demand

We can clearly see that congestion affects the travel times of the travelers. The increase in the demand increases congestion in the network as evidenced from the increase in APTT. From the results, we can see that the AVTT and AVTT increase significantly with increase in the demand for the same SAV:traveler ratio (for SAV:traveler ratio 1:3, APTT increases from 52.84 minutes to 77.87 minutes and AVTT increases from 20.53 minutes to 32.06 minutes) which are reduced significantly by the tabu search procedure.

2.5.3 Effect of Fleet Size

Figure 1 shows the effects of varying fleet sizes on congestion in the network. The increase in the fleet size reduces the initial APTT due to the larger availability of SAVs in the network. This increases the congestion increases the AVTT (from 23.65 minutes to 40.89 minutes for 9639 travelers). The results from Table 1 clearly show that the tabu search procedure decreases the AVTT and APTT significantly in such congested networks.

Figure 1 Variation of average person travel time with number of swaps with congestion



2.6 Sensitivity Analysis

In this section, we discuss the sensitivity of the algorithm to the parameters that define the algorithm. For each parameter, we ran the algorithm on a network at 33% demand – 9639 travelers, with a fleet size of 3333 SAVs with a time horizon of 10 hours.

2.6.1 Effect of Tabu List Duration

The experiments run showed us that the larger the tabu duration, the less time the algorithm took to converge. We ran experiments with tabu sizes of 5, 10, 100 and 1000. The algorithm did not show a significant improvement in running time for small tabu sizes – 1034 minutes for a tabu size of 5. However, at larger tabu sizes, it took significantly lesser time to converge – 896 minutes for a tabu size of 100.

2.6.2 Effect of Nearest Traveler Neighborhood Size

We found that varying the neighborhood size did not improve the performance of the algorithm. In fact, the algorithm met the stopping criteria and stopped abruptly as the neighborhood size was increased. This is due to the fact that the neighborhood is arranged in the decreasing order of improvement to the objective function. As a result, swaps beyond the first five entries of the neighborhood do not result in an improvement in the objective.

2.6.4 Effect of Stopping Criterion

The algorithm converged prematurely for small values of the stopping criterion but quickly improved as the stopping criterion was increased. For a stopping value of 10 iterations without improvement, the algorithm was found to stop with average passenger travel time dropping from 57 minutes to 42 minutes. However, as the stopping value was improved to 50, the algorithm was found to improve the solution.

Table 1. Results from tabu search - congestion aware experiment

Number of Travelers	SAV: Traveler Ratio	Size of SAV Fleet	Initial AVTT (minutes)	Final AVTT (minutes)	Initial APTT (minutes)	Final APTT (minutes)	Solution time (hours)
9639	1:5	2000	23.65	18.59	97.85	29.69	19.6
	1:3	3333	32.06	27.41	77.87	32.91	15.8
	1:2	5000	40.89	32.98	69.17	25.95	18.8
4823	1:5	1000	20.79	16.91	84.17	24.74	7.21
	1:3	1667	21.23	17.69	59.09	15.23	5.3
	1:2	2500	22.88	17.59	44.76	11.22	7.6
2406	1:5	500	20.15	15.72	93.38	19.68	2.35
	1:3	833	20.53	17.42	52.84	14.17	2.21
	1:2	1250	20.93	16.97	47.16	10.54	3.25
14500	1:5	3000	18.6	13.14	100.27	30.09	21.48
	1:3	5000	18.72	12.87	83.69	30.17	45.75
	1:2	7500	18.74	15.14	72.57	40.07	64.01

Chapter 3

Network Optimization: Placement of AIMs in the city of Austin

This chapter presents the formulation and methodology of two methods used to develop quantitative measures of the benefits and costs associated with replacing a traditional signal with a reservation-based smart intersection. The solution methodology taken to solve the problem is twofold. First, a regression model is used from simulated data to quantitatively identify a deployment strategy by calculating costs and benefits from a parsimonious set of independent variables. Second, a dynamic user equilibrium traffic assignment model is developed to determine a system optimal grouping of smart and signalized intersections that provides the minimum TSTT.

In an effort to generalize the effect of individual intersection characteristics on the differential impact of travel times of smart versus signalized control, a multiple regression model is estimated using a mixture of DTA simulation data and available intersection characteristics such as signal properties and turning demand. The model input data is collected from subsets of signalized intersections in different large-scale city networks. The aim of this model is to apply the regression to a subset of a network's intersections and output a ranking of the best smart intersection candidates. This ranking can then be tested in simulation to evaluate model accuracy, and the model can be used to easily develop smart intersection selection and prioritization rules.

Finding a system optimal grouping of smart and signalized intersections in a network giving the minimum TSTT is a more difficult task. Due to the bi-level nature of the optimization problem with the second layer defined by the solving of DUE on a large-scale network, the overall problem is np-hard to solve exactly. Thus, a metaheuristic is used to find a solution. In order to find a system optimal grouping of smart and signalized intersections in a network, a genetic algorithm is used. Section 3.2 details the specific algorithm used to find a feasible solution and its steps.

3.1 Literature Review

Traffic signals are not designed to make the best use of the intersection capacity during a phase. For example, many turning movements are restricted and there are significant gaps in the flow of traffic. Such precautions are necessary to prevent accidents in human driven vehicles. However, these will not take advantage of the power of AVs. Dresner & Stone *(33,34)* proposed a reservation-based intersection control to make use of the AV technologies to increase intersection utilization.

Reservation based controls are designed such that every vehicle entering an intersection communicates wirelessly with an intersection manager and places a reservation to traverse the intersection at a given time. The intersection manager then simulates the requests in a grid of space-time tiles and accepts or rejects each request depending on if conflicts arise between reservations. Previous literature has studied the possibility of using different mechanisms to decide which reservation to prioritize when conflicts occur. Most studies *(33,34,35,36)* have used a first-come-first-serve (FCFS) policy where the vehicle that placed the first request is given priority over other vehicles approaching an intersection. Other policies have been proposed to handle conflicts such as prioritizing emergency vehicles *(37)* and a system where an auction is held at each intersection where vehicles entering the intersection bid to access it *(38,39,40)*.

Fajardo et al. *(35)* and Li et al. *(36)* found that FCFS reservations provided savings in delays beyond optimized traffic signals. The computational complexity of the simulation of vehicles through the intersection space-time tiles model have limited microsimulation based studies to small networks *(41)* or modified the reservation protocol to reduce its capacity *(42)*. Levin and Boyles *(43)* proposed the conflict region model to alleviate the computational issues by replacing the simultaneous occupancy checks of the tile-based reservation protocol with capacity constraints.

The reservation model can be extended for human driven vehicles as well. For AVs, the communication of a reservation with the intersection manager takes place through an in-built wireless communication system. Humans might be able to communicate their ETA at an intersection and speed through an app on their smart phones. However, following a reservation protocol can be confusing and even hazardous for human driven vehicles. Bento et al. *(44)* and Qian et al. *(45)* studied the possibility of integrating human vehicles into the reservation model. The conflict region model was found to scale up for DTA on large city networks without compromising on the key characteristics of reservation controls. Bento et al. *(44)* proposed reserving additional safety margins for human vehicles, and Levin et al. *(46)* implemented this into the conflict region model. Dresner & Stone *(47,48)* proposed inserting a cycling green light into the reservation protocol to allow human vehicles to move.

Patel et al. *(49)* used the cell transmission model to study the effects of reservation based intersection control for AVs on congestion in large networks. In their study, they found that the FCFS policy resulted in significant congestion in some scenarios - local road-arterial intersections that are close together, and at high demand. In such scenarios, it would be beneficial to use signal controls. While FCFS reservation controls have significant advantages to signal controls, it is imperative to understand the dynamic of the network and to optimally place reservations in order to take full advantage of the reservation policy.

In this report, we wish to develop methods to identify subsets of intersections in a network where reservation based intersection control would provide system-wide benefit in terms of congestion. Two methods are proposed to tackle this problem. First, a regression based approach to generalize intersections as being better suited to reservation-based control/signalized control based on their characteristics and characteristics of the network. Second, a genetic algorithm is used in coordination with a dynamic traffic assignment (DTA) model to solve for dynamic user equilibrium (DUE) to find optimal allocation of reservations and signals in the network.

3.2 Methodology

3.2.1 A Mesoscopic Dynamic Traffic Assignment Model

This subsection serves to define the dynamic traffic assignment model (DTA) used to simulate all networks when solving for dynamic user equilibrium (DUE), including the model specifications and assumptions. The mentioned model is a custom implementation of a mesoscopic DTA model in Java which can capture human-driven (HVs) and autonomous vehicle behavior either separately or together on any network. A multiclass cell transmission model (CTM) is used to propagate flow through network links based on hydrodynamic flow theory. To model reservation-based intersection control, a conflict region model is used which divides an intersection region into larger and simplified conflict regions, each with a capacity, either accepting or rejecting vehicle requests based on some priority function. The priority function assumed in this study is a first come first serve (FCFS) function in which the first vehicle to make a request with the intersection manager is processed by the manager first. The DTA model solves for DUE using the method of successive averages (MSA) to a convergence defined by a 1% relative gap. Primarily, to simulate AV behavior, AVs are assumed a 0.5 second reaction time compared to HVs which have a 1 second reaction time. This difference in reaction time leads to increased roadway capacity caused by increased backward wave speed and reduced following headways as the proportion of AVs in a network increases, with the greatest increased capacity associated with 100% AVs. It is also assumed that only AVs can use the TBR intersections, and so for the sake of this study, all DTA simulation runs are evaluated with only AVs in the networks.

For reference, the two real city networks used in simulation include the downtown Austin, TX and downtown Dallas, TX networks with a total demand of 62,783 and 167,592 vehicle trips over a 4 hour observation period respectively. This total demand is considered 100% demand as the next two sections may refer to a proportion of this demand.

3.2.2 A Multiple Regression Model

The goal of the multiple regression formulated in this section is to quantitatively predict any intersection's relative utility under reservation control compared to traditional signal control, given some basic intersection characteristics. If this can be accomplished, then a set of intersections can effectively be ranked according to this utility and smart intersection deployment strategies can be easily developed based on quantitative metrics.

3.2.2.1 Formulation

To summarize the utility or performance of any intersection in a network, a difference in effective TSTT of an intersection under signal control and TBR control is used as the primary response variable. To obtain regression input dataset, a set of Nintersections is selected from a parent network. In order to effectively measure the effect on system-wide TSTT of a single intersection in a large network, a small subnetwork is created involving the observed intersection and its immediately adjacent links and nodes, with the nodes only acting as origins and destinations. The intersection is then assumed a control (TBR or signal) and is solved for DUE using a DTA simulator and a user specified origin-destination demand matrix. The response variable is then found by subtracting the TSTT under TBR control from the TSTT with signalized control. Multiple predictor variables described in section 3.2.2.2 are obtained from given network input data and from simulation. Specific methods and sources of data collection are presented in section 3.2.2.3. The general regression formula is as follows:

$$TSTT_{signal} - TSTT_{TBR} = \Delta TSTT = FFTT + \vec{\beta} * \vec{X}$$

where β is the vector of variable coefficients and $X$is the vector of predictor variables.

To encapsulate effects of different levels of demand on an intersection, the single intersection network is solved for DUE with the DTA simulator under each of six different demand levels: $d = \{10\%, 30\%, 50\%, 75\%, 85\%, 100\%\}$. The origin-destination (OD) demand matrices for the listed demands are obtained through simulation, detailed in section 3.2.2.3. With Nintersections selected to observe from the parent network and six demand levels considered, a total of $N * 6$ data points are found to estimate the desired multiple regression model in which a $\Delta TSTT$ is predicted given some intersection characteristics, $\vec{X}$.

The lower the $\Delta TSTT$, the better the intersection performs under TBR control compared to signal control. By obtaining this response variable, intersections can effectively be ranked according to the differential utility between the two controls.

3.2.2.2 Variables

The set of possible predictor variables to be used in the regression model to predict the difference in TSTT between an intersection under signal control and TBR control are as follows:

Table 2. Description of variables used in Multiple regression model

Predictor Variable	Description of Variable	Variable type
Number of phases	Total number of signal phases across a cycle	Number of phases
Cycle length	Signal's time of one complete phasing cycle	Time (seconds)
Number of moves	Total number of non-restrictive turning movements for the intersection. Turning movements are defined by an approach link and an exit link.	Number of turning movements
Average lane vehicle count	Average number of total vehicles using the intersections incoming and outgoing lanes	Number of vehicles/hour/lane
Number of through turns	Total cumulative through demand of the intersection across all approaches	Number of vehicles
Number of left turns	Total cumulative left turn demand of the intersection across all approaches	Number of vehicles
Number of right turns	Total cumulative right turn demand of the intersection across all approaches	Number of vehicles
Minimum length	Minimum length of a link entering or exiting the intersection	Length in feet
Maximum length	Maximum length of a link entering or exiting the intersection	Length in feet
Average length	Average length of a link entering or exiting the intersection	Length in meters
Minimum link capacity	Minimum capacity of a link entering or exiting the intersection	Number of vehicles/hour
Total link capacity	Total cumulative capacity of all links entering or exiting the intersection	Number of vehicles/hour

3.2.2.3 Data Collection

The primary response variable for this regression model is a difference in TSTT between an intersection under signal control and TBR control. To find the TSTT of an individual intersection, a new subnetwork of only the specified intersection is created. The new single-intersection network contains only one intersection which acts as a real intersection to move vehicles across links, and all directly adjacent nodes to the intersection which merely act as the network's origins and destinations in which vehicles enter and exit the network. The new network also contains centroid nodes and connectors which act as the network's loading points in which vehicles exit and enter. To define the subnetwork's origin-destination demand for a specific control at any given demand level, a DTA simulation of the larger parent network is run at the desired demand level with all eligible intersections assuming the specified control. (The demand level describes the proportion of total demand seen in a network.) The "eligible" intersection set is defined in section 3.2.2.4. A cumulative count of vehicles is kept during simulation for each possible turning movement in each intersection, and because the new subnetworks only contain one intersection each, the counts for each turning movement in the parent run can be combined to define origin-destination demand for each intersection. For each intersection, this demand is then set as the new subnetwork OD-demand matrix, a DTA simulation run of the network is completed under TBR and signal control with a TSTT output for each, and a $\Delta TSTT$ is found through the difference.

Average lane vehicle count, through turns, right turns, and left turns are all predictor variables which were found through simulation as well. While running the parent network DTA simulation to find OD-demand matrices for each intersection as described above, a cumulative count of vehicles using each possible turning movement was recorded which was simply translated into total through, left and right turns for the intersection. The average lane vehicle count was found by averaging the total experienced vehicle counts for all incoming and outgoing lanes in the intersection. All other potential variables defined in section 3.2.2.2 were obtained through city network datasets.

In this report, two parent networks, downtown Dallas and downtown Austin, were used to obtain regression input datasets and estimate the respective regression models. A total of 174 and 152 intersections were observed from the downtown Austin and downtown Dallas networks respectively.

3.2.2.4 Assumptions

The following assumptions were made while developing the regression model:

- The set of intersections that can be switched to autonomous intersections are the set of traffic signals in the network. The model does not consider the set of merges, diverges, or stop sign controlled intersections. This is assumed as signalized intersections are likely to be the priority intersections to change as TBR provides little system-wide benefit when applied to non-signalized intersections, as shown in previous studies (Patel & Levin, 2016).
- All DTA simulations are run using a demand composed only of AVs. Because HVs are assumed to not use TBR intersection control, a demand of only AVs is required for stable results and analysis between networks with mixtures of TBR and signal intersections.
- Because an independent subnetwork is created for each intersection containing only the intersection, it is assumed for the DTA simulation that the intersection is independent of all others in the network. Other intersections do not impact the observed intersection over time, however as described in section 3.2.2.2, OD-demand matrices are found through observed vehicle counts in the parent DTA simulation run. The parent DTA simulation does include all intersections (which are eventually isolated as subnetworks), and so demand in the subnetwork OD matrices is from a complete network with intersection interactions.

3.2.2.5 Model Metrics

The following metrics will be used to evaluate the performance and accuracy of the estimated linear regression models, including their variables and other specifications.

The R^2, also known as the coefficient of determination is a statistical measure of how close the actual data are to the fitted regression model. It is measured as:

$$R^2 = SS_{reg}/SS_{total}$$

where $SS_{reg}$is the total variance in the data explained by the model and SS_{total} is the total variance in the data.

The second statistic used to measure the relative significance of each variable in the model. A standard $t - test$ is conducted for each predictor variable with a confidence interval threshold of 95% ($t_{.95} = 1.645$), translating to a variable being significant if: $t_{variable} \geq 1.645$with 95% confidence. Although this defines a threshold for significance, some variables may be chosen to remain in the model if some other significance is seen.

3.2.3 A Genetic Algorithm for System Optimal Placement of Reservation based Intersections

The goal of the Genetic Algorithm (GA) approach is to identify the spatial orientation of smart intersections and traditional intersections in the network that provides the best benefit in terms of TSTT. However, it is impractical to assume that a smart intersection can be installed at every intersection in the network. Therefore, this section explores two sub problems. First, it identifies the system optimal allocation of reservations in the network which produce the best system wide benefits. The second experiment finds the optimal allocation of TBRs that produces the best benefit when there is a limit to the number of TBRs that can be installed. In both cases, the GA is used in coordination with a dynamic traffic assignment model to find the dynamic user equilibrium solution.

The following were assumed while developing the model:

- The set of intersections that can be switched to Autonomous intersections are the set of traffic signals in the network. The model does not consider the set of merges, diverges, left and right turns.
- In the dynamic traffic assignment model, the only demand in the network is the demand due to AVs

3.2.3.1 A Background on Genetic Algorithms

A genetic algorithm (GA) is a class of computational methods inspired by genetic evolution used to solve constrained and unconstrained optimization problems. In a genetic algorithm, an initial random population of candidate solutions are created. An evaluation metric is used to determine the fitness value of each of these parents. At each step of the algorithm, a pair of parents are selected at random from the population to reproduce to create child individuals. A genetic algorithm follows a set of steps at each iteration to create children which ensures that the best characteristics of the parent generation are preserved. In our target problem, each individual in the population is a specific orientation of the network where each intersection is modeled either as a traffic signal or as a tile-based reservation. The characteristic which evolves from one generation to the next is the orientation of each of these intersections in the network. The algorithm is designed such that the orientation of each intersection which results in the best observed TSTT is retained in future generations. These steps are outlined below. Section 3.2.3.3 describes the specific design of the GA used to solve the problem at hand. A pseudocode of the algorithm is presented in Section 3.2.3.2

3.2.3.2 Pseudocode

In this section, we define the steps involved in a GA followed by a detailed discussion of each of the steps. A pseudocode of the algorithm is presented below:

1. Create an initial random population of n individuals with random intersection controls

2. Evaluate the goodness of each individual in the population to the problem
3. Sort the population in decreasing order of the fitness measure
4. While iteration < maxIterations, repeat
    a. Select a pair of the individuals in the population at random as candidate parents to cross over to form children.
    b. Generate child individuals by crossing over candidate parents.
    c. Introduce mutation in the children. Evaluate the goodness of the child individuals created.
    d. Sort the population in decreasing order of the fitness measure.
    e. Remove the lowest performing 50 individuals from the population.

3.2.3.3 Genetic Algorithm Steps

Candidate Selection

A pair of individuals are selected as parents at random from the best performing individuals of the current generation. This depends on the proportion of the current population we choose to keep at each step of the algorithm. For example, if we wish to create 20% children at each step of the algorithm, the GA chooses parents from the best performing 80% of the current generation.

Crossover Probability

The crossover probability gives how often crossover will occur between parents. This helps decide which characteristics of each parent enters a child in the next iteration. We have developed a heuristic to determine this probability which is described below. The probability is determined by:

$$Crossover\ Probabiltiy = 0.5 + 0.5 * \frac{|TSTT(Parent1) - TSTT(Parent2)|}{TSTT(100\%\ Signals) - TSTT(100\%\ TBRs)}$$

The denominator is the upper bound on observable difference in TSTT between two different orientations of the network - one where every intersection is modeled as a traffic signal and one where every intersection is modeled as a TBR. The crossover probability is designed to ensure that the parent with the better TSTT has a higher chance of having its intersection orientation passed on to the child.

Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation to the next. It is usually applied with a low probability. There are many types of mutations used in genetic algorithms. In this model, we have chosen an inversion mutation. An inversion mutation is one in which the orientation of each intersection in the individual is inverted if the inversion probability is met. Mutation is done to introduce a degree of diversity to the genetic structure of each generation. In the experiments, the probability of mutation has been set to at 0.7%

3.2.3.4 Model Inputs

Population Size

The population size is the number of created at the beginning of the GA. The population size often determines the quality of the solutions obtained from a GA. A small population size will result in quicker convergence but may result in the algorithm getting trapped in a local optimum. However, a large population will slow the model down significantly. It is prudent to choose a population size that avoids both of these pitfalls. The experiments use a population size of a 100 parents to start our GA.

Proportion of Children

The proportion of children is a parameter which determines the proportion of children to generate at each iteration of the GA. Consequently, this affects the quality of the population in each generation.

Maximum TBRs

This parameter limits the number of tile-based reservations in each individual in the population. This has been modeled as a parameter in the model since it was observed that an all TBR network performed significantly worse in some scenarios.

Convergence (Max Iterations)

The convergence criteria is the maximum number of iterations to be completed before the algorithm terminates. The max iterations criterion suffers from extremes. A small value for maximum iterations will result in the algorithm converging prematurely, sometimes before it hits a local optimum. Similarly, a large value of maximum iterations will slow the algorithm from converging and may result in cycling after it reaches a local optimum. It is thus, prudent to choose a convergence criterion that avoids both these problems.

3.2.3.5 Model Output - DTA DUE Solution

When a new individual is generated, a goodness calculation is performed to evaluate how good the solution is. In the problem at hand, the goodness calculation is the TSTT observed from solving the specific orientation of the network to equilibrium using a Simulation based Dynamic Traffic Assignment model.

3.3 Experimental Results

This section presents experimental results of applying the linear regression and the genetic algorithm models, as described in Section 3.2, to a large-scale city network. The presented regression model is used to find a ranking of the “best” candidate intersections in the network to assume reservation control and the genetic algorithm is used to find an optimal subset of intersections in various set sizes which minimize the TSTT of the network. The network used for all experimental simulation results is the downtown network of Austin which contains 1,247 links, 546 intersections (174 signalized), 171 zones, and a total demand of 62,783 vehicle trips over a 4-hour observation period. This network includes the smaller arterial networks discussed in Patel et al. *(49)* including

Congress Avenue and Lamar & 38th St where paradoxes in the benefit of TBR were seen. The DTA model used in this section is described in Section 3.2.1.

3.3.1 Linear Regression Results

With the input data for the Dallas, TX network, presented in Section 3.2.2, a linear regression model was estimated with the goal of identifying significant intersection characteristics that affect the relative system-wide utility of a single intersection under reservation control compared to signal control. It is the primary goal that with an accurate regression model, a set of eligible (signalized) intersections in a network can be ranked in terms of their relative utilities. A subset of these intersections, starting with the intersection holding the largest utility value and moving in decreasing order, can be chosen to assume reservation control and provide the most benefit to the system's TSTT compared to any other same-size subset of the eligible set. Recall that the relative utility is $\Delta TSTT = TSTT_{signal} - TSTT_{TBR}$, so a larger $\Delta TSTT$ means a larger benefit to the system with the intersection under TBR control. This section presents the estimated regression model and results of applying this ranking system to a network in simulation.

3.3.1.1 Model Outputs

This section presents the regression model used to predict $\Delta TSTT$ and the variables found to be significant in identifying an intersection suitable for reservation control compared to signal control.

Table 3 presents the results of the regression model. In using the intersections from Dallas as training data for the regression model, it is seen that the model is able to predict 75% of the variation in the data for the City of Austin network. This is evident from the R^2 value of 0.754.

Table 3. Summary of the results of linear regression model

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.868	.754	.752	360.81758

3.3.1.2 Significant Variables

From the pool of variables presented in section 3.2.2.2, the variables presented in Table 4 were found to be significant predictors for the regression model. The relative importance of each variable can be gauged by the magnitude of the coefficient of the variable in the model. This is also evidenced from the value of t-value for each variable at 95% confidence level. Recall that a variable is considered significant if $t_{variable} \geq 1.645$. Table 4 presents the relative importance of each variable in predicting the difference in TSTT for each intersection. From the model, it is evident that cycle length, number of moves, number of through turns, number of left turns, number of right turns and minimum length have high t-values and are thus, significant predictors in the model. However, the minimum capacity of link has t-value which is lower than the set threshold value. It has been retained in the model because it is indicative of queue spillback in the network because a link with a low capacity entering an intersection could result in spill back if there is significant congestion in the network.

Table 4. Summary of the significant variables in the regression model

Coefficients				
Model		Unstandardized Coefficients		t
		B	Std. Error	
1	(Constant)	-717.296	146.626	-4.892
	Cycle length	3.286	.295	11.126
	Number of moves	9.495	4.411	2.153
	Number of Through turns	.261	.011	23.621
	Number of left turns	.430	.028	15.412
	Number of right turns	.414	.031	13.191
	Minimum length	.409	.231	1.767
	Minimum link capacity	-.025	.022	-1.140

3.3.1.3 Regression Results in Simulation

Before testing the regression model in simulation by applying it to the Austin intersections, the input data used to train the regression is tested to ensure this base data offers desirable results in simulation. To test the base data, the $\Delta TSTT$ objective function values for each eligible intersection in the Austin network are used to rank the intersections in descending order of their objectives, leaving the intersections which perform better as reservations to improve system-wide congestion, higher in the ranking list. To evaluate the ranking, an experiment with each of a different proportion (including 20%, 40%, 60%, and 80%) of the highest ranked intersections is conducted by forcing this subset of intersections to assume TBR control and the rest to assume signal control. For example, in the 40% experiment, intersections ranked 1 through 70 (40% of 174 eligible intersections) are chosen to take on TBR control, and the rest signal control. This subset of intersections is predicted to provide more system-wide benefit to TSTT than any other same sized subset of intersections. A DTA simulation was then run for each before mentioned proportion of the top TBR candidate intersections. To analyze the margin of effect seen in TSTT for all presented results, two control or base case scenarios are run to provide a bound on currently known TSTT's. The first is the case of the network with only signal control and the second is the case of the network with only TBR control for all eligible intersections, yielding a TSTT of 6443.22 hours and 4560.14 hours respectively. Keep in mind that all experiments in this study are run with a demand of only autonomous vehicles.

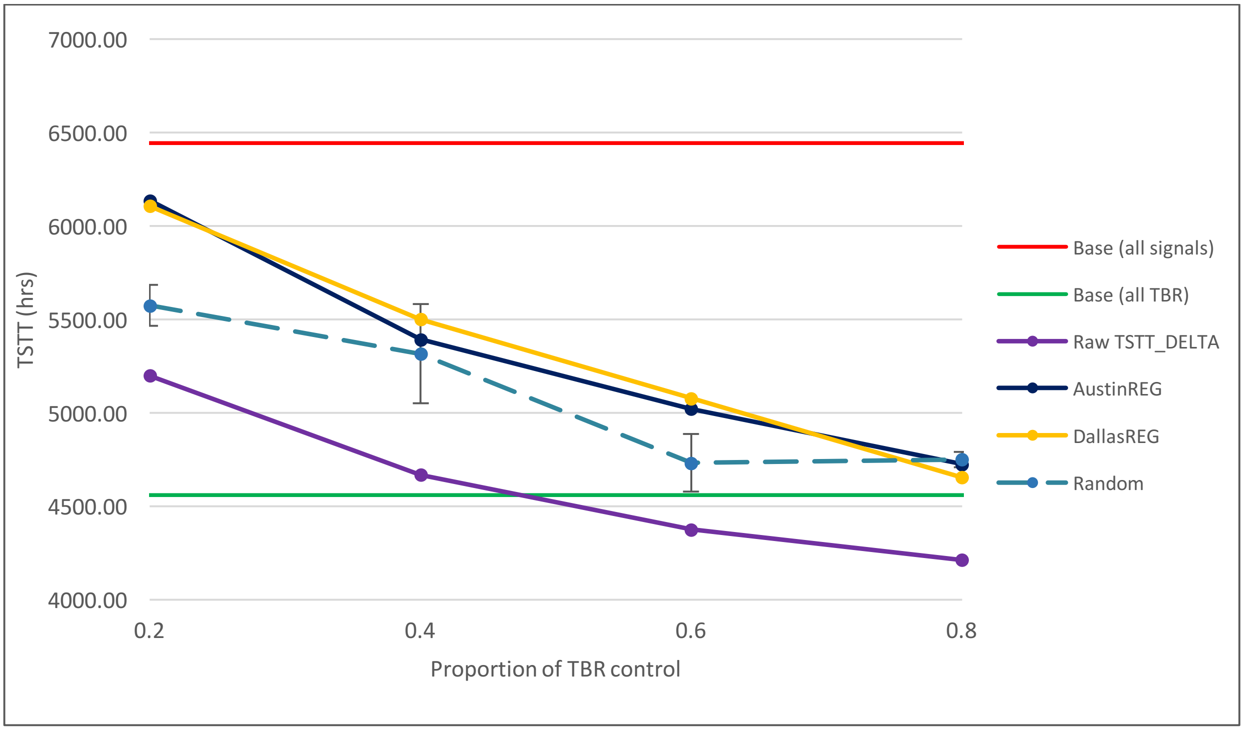
As shown in the graph in Figure 2, results for the base input data at 20% TBR controlled intersections show a 20% drop in TSTT from the network with only signals which is to be somewhat expected. In general, TSTT decreased at a slightly decreasing rate as the proportion of TBR control in the network increased which is also to be expected. However, the base data results actually show a 4% and 8% lower TSTT at 60% and 80% TBR control, respectively, compared to the base case with 100% TBR control. The graph in Figure 2 shows a red and green horizontal line showing the 100% signal and 100% TBR base cases respectively, and the Raw TSTT_DELTA data points show the decreased TSTT of the 60% and 80% experiments, and even show that at 40% TBR control, the difference in TSTT with the 100% TBR control base case is quite small. These decreased TSTT's, although not large in size, show that more system-wide benefit can be seen using only a fraction of the reservation control. This also backs the notion that some intersections, typically lower in the ranked list of TBR candidates, tend to help the traffic system as signals. In the 100% TBR base case, great improvement is seen over the 100% signals base case and there is no real sign of TBR causing any increased congestion compared to a better performing and smaller subset of possible TBR intersections, partially due to dynamic route choice. As vehicles can dynamically alter paths based on their observation of the network's state, they can avoid such TBR intersections which become very congested, however this can lead to congestion in other portions of the network. Such rerouting due to paradoxical queue spillback seen near TBR intersections can be avoided with the right intersections.

Next, the Dallas-based regression is applied to the 174 eligible Austin intersections to predict their relative utilities, and the intersections are ranked in descending order as was done with the base input data. The same experiments for the predicted ranked list of intersections were then conducted and shown on the graph in Figure 2. Results show the same general trend of decreasing TSTT as the proportion of TBR control increased held, however all simulation outputs of TSTT were approximately 17% higher than those found in the pre-regression input data experiments. This large difference is not expected as the predicted regression model showed a somewhat close fit to the data with an R^2 of 0.754. Although the regression results outperformed the 100% signals base case, this was to be expected of almost any network with a subset of intersections using reservation control. At a TBR proportion of 80%, there does seem to be a TSTT falling very close to the base case of 100% TBR control showing that a similar result can be seen with 35 less reservation intersections, however it does not outperform the 100% case like the data the regression was modeled after.

To confirm the validity of a set of downtown Dallas intersections being able to accurately predict a set of Austin intersections, the Austin network base characteristic data was used to estimate a new regression model so that the Austin network could predict the relative utilities of its own intersections. DTA simulation results show a set of points nearly identical to that seen in the Dallas-based regression, shown in Figure 2. This confirmed that intersections from the two different networks seem to have no significant difference in their respective characteristics' relation to their differential impact on system-wide congestion under the two intersection controls. Finally, it is not clear the magnitude of the regression ranked list's inaccuracy compared to the original input data only through simulation. As a comparison and means of testing the accuracy of the regression's predicted results, for each analyzed proportion of TBR control in a network, 20 networks are randomly generated subject to the number of TBRs remaining equal to the respective proportion and their TSTT values are averaged for a new data point. As shown in Figure 2 with the dashed plotted line, on average, the randomly generated networks actually outperformed both regressions for each TBR proportion, suggesting that the ranked list predicted by the regression was inaccurate.

The shortcomings of the regression model are most likely due to uncaptured interdependencies between intersections. Results show that the base data obtained through the simulation of single-intersection networks yielded a high performing ranking of intersections in terms of their differential utility under reservation and signal control. This suggests that some intersections are better for system-wide congestion under TBR control compared to others, and that this trend cannot be captured linearly. Most likely, because demand for the single-intersection networks for each intersection were directly obtained through simulation of the parent network, some intersection dependency was captured even though intersections were tested as independent. The regression also only identifies single intersections as TBR controlled and does not identify the clustering of intersections where reservations could provide the most benefit. This interdependency between intersections may heavily impact the system-wide effect that a single intersection carries, most likely through demand allocation. A group of intersections may not necessarily improve the TSTT unless they are placed optimally in the network. Regression results showed the most significant predictor variables to be different turning demands which may be true, however these most likely cannot captured linearly, as previously stated.

Figure 2: A summary of regression and raw data results in simulation across TBR control proportions



3.3.2 Genetic Algorithm Results

The regression identifies the intersections with characteristics desirable for a reservation. However, the regression model is incapable of identifying the optimal subset of intersections in a network which would produce the best benefit in terms of TSTT. As discussed in section 1.x, previous studies found that a system with 100% reservations did not necessarily improve TSTT. The methodology explained in section 2.3 was used to identify the subset of intersections in the city of Austin that produce the best benefit when modeled as a reservation. Two sets of problems were solved using the GA – to identify the system optimal allocation of reservations in the network which produce the best system wide benefits and to the find the optimal allocation of TBRs that produce the best benefit when the number of TBRs is limited to 35 and 70 intersections. In all these cases, the GA was used in coordination with a dynamic traffic assignment model to find the dynamic user equilibrium solution. In the model used, an initial population of 200 individuals was used with a proportion of 0.75 of the population kept at each iteration. The probability of mutation was set at 0.7% per intersection and the convergence criterion was set at a maximum number of iterations of 100.

3.3.2.1 System Optimal GA

For the system optimal experiment, the initial population had 200 individuals with 50-50 split between traffic signal and reservations. The initial solution for the system optimal genetic algorithm (SOGA) is much lower than that produced by the solution from the regression. In fact, SOGA starts with a TSTT that is comparable and quickly reaches the levels produced when 100% of intersections are modeled as reservations. The algorithm terminates abruptly just as it begins to converge as evidenced from Figure 3. This indicates that for a larger proportion of reservations, a larger convergence criterion must be used. Figure 4 shows that the proportion of reservations in the solution increases gradually with the number of iterations. The proportion of reservations in the solution when the GA terminated is at 0.9. This validates the paradox observed in earlier studies.

Figure 3. Variation of TSTT with number of iterations

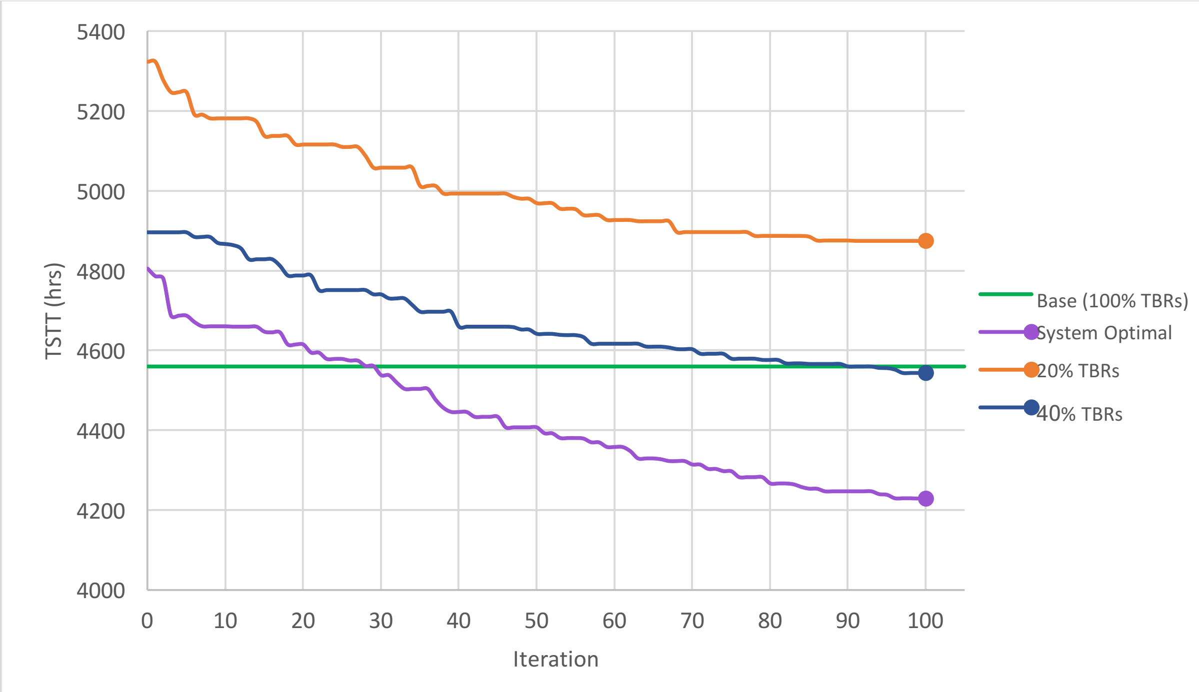
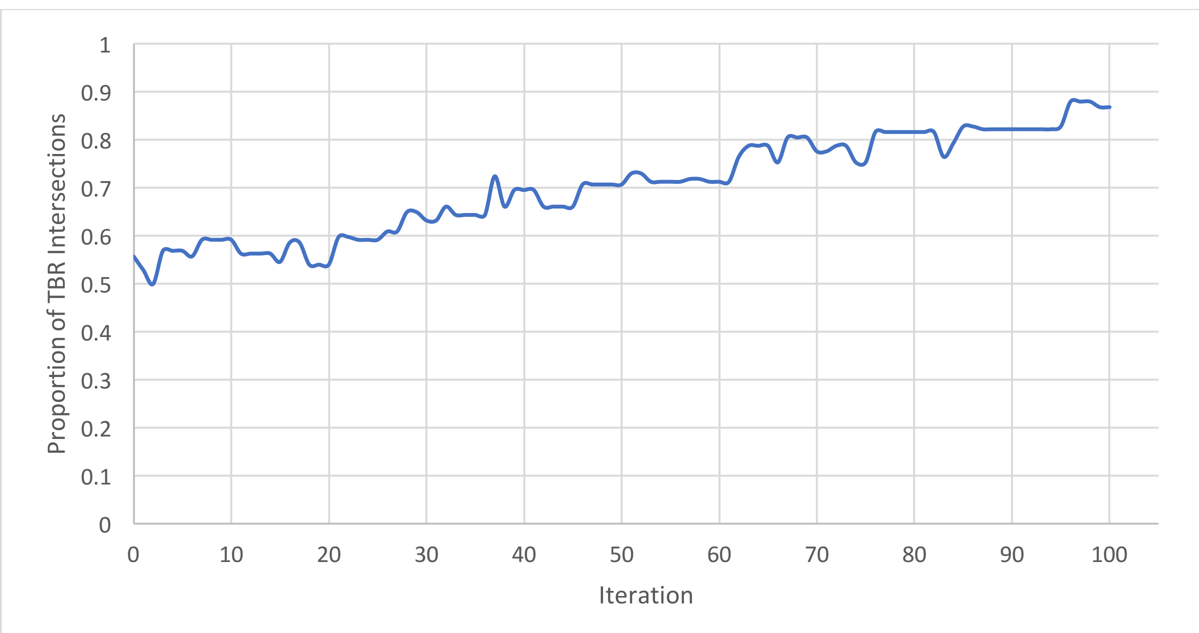


Figure 4. Variation of proportion of TBR with number of iterations



3.3.2.2 Limited TBRs GA

The limited TBR experiment is similar to the SOGA with the exception that there is now a limit on the number of reservations in the network. An initial population of 200 individuals were produced containing 20% and 40% TBRs in each individual. The population was then evolved by crossing the parent generation to produce children. At each iteration, 0.75 of the parent generation were kept in the population. The probability of mutation was kept at 0.7% and the termination criterion was kept at 100 iterations again. It is evident from Figure 3 that the TSTT drops with increase in the proportion of TBRs in the network. For the 40% TBR experiment, the GA evolves to produce results comparable with the default 100% TBR case. This indicates that a 40% TBR network can provide the benefit of a 100% TBR network if the location of TBRs is chosen wisely. This is also indicative of the paradoxes observed in earlier studies. Although the system-wide benefit of implementing TBRs increases with the proportion of TBRs in the system, the improvement is not uniform. From Figure 3, the system optimal GA reaches the 100% TBR solution at iteration 30. At this point, the proportion of TBRs in the model is 65% (approx. from Figure 4). However, the limited 40% TBR solution reaches achieves the 100% TBR solution by iteration 90. This implies that an optimal placement of a limited number of TBRs can achieve the system-wide benefit of a network with a higher number of TBRs.

3.3.3 Network Trends

In order to observe trends and find commonalities between methods of finding optimal TBR placement, experimental results are plotted onto maps using the ArcGIS software. In Figure 5, both the pre-regression base data and the Dallas-based regression solutions are mapped in the Austin network. Interestingly, both maps show most of the TBR intersections in series along corridors. Although both maps show similar clusters of TBR control, the regression map shows little to no reservations in the center of downtown, whereas the base data map shows short corridors of reservations along highly congested streets. Both show TBR corridors along Lamar Blvd, 1st St and 15th St, all of which are known to be major arterial roads and/or become very congested in peak hour traffic.

As shown in Figure 6, the genetic algorithm mapping at a 20% proportion of TBRs shows a similar trend in TBR corridors along major arterials and roadways such as Lamar Blvd and 1st St, however these corridors are not as continuous as those seen in the regression and base data results, with the exception of 15th St. On 15th St, the GA placed 7 consecutive TBR intersections of which there are almost no directly adjacent signalized/TBR intersections. This aligns with the prediction that those intersections which are not within very close proximity to other minor intersections would do better as reservations as there is no significant interruption of major flow through the corridor. The 20% TBR GA solution had the lowest TSTT of all 20% TBR proportion experiments. Similar trends are seen in the map of the 40% TBRs GA solution in Figure 6. The GA places TBR intersections in longer chains of multiple intersecting corridors, especially in the central downtown areas. Aside from some of the intersections in the middle of central downtown, most TBR intersections tend to have few signals within close proximity and are clustered mainly in straight corridors.

As seen previously and in Table 5, the pre-regression raw data at 80% TBRs, when simulated, actually slightly outperformed the GA's system optimal solution which came to an 86% proportion of TBR control in the network. This could be due to the fact that the genetic algorithm was allowed too much freedom when adding to the proportion of TBRs, and skipped over a more optimal solution as was found in the pre-regression base results. Although TSTT was approximately the same between the two, as seen in Figure 7, the orientation of signals across the network was different. The base data ranking of intersections left more clusters of signals crowded together compared to the GA which had a wider spread of signal locations. Still, similarities were seen in signal placement at certain nodes such as around the UT campus and in parts of central downtown. The same trend of TBR corridors is seen in both map visualizations of the two results which gave the two lowest TSTT values seen across all experiments in the study.

Figure 5: Intersection placements found by pre-regression raw input data (left) and the Dallas-based regression (right)

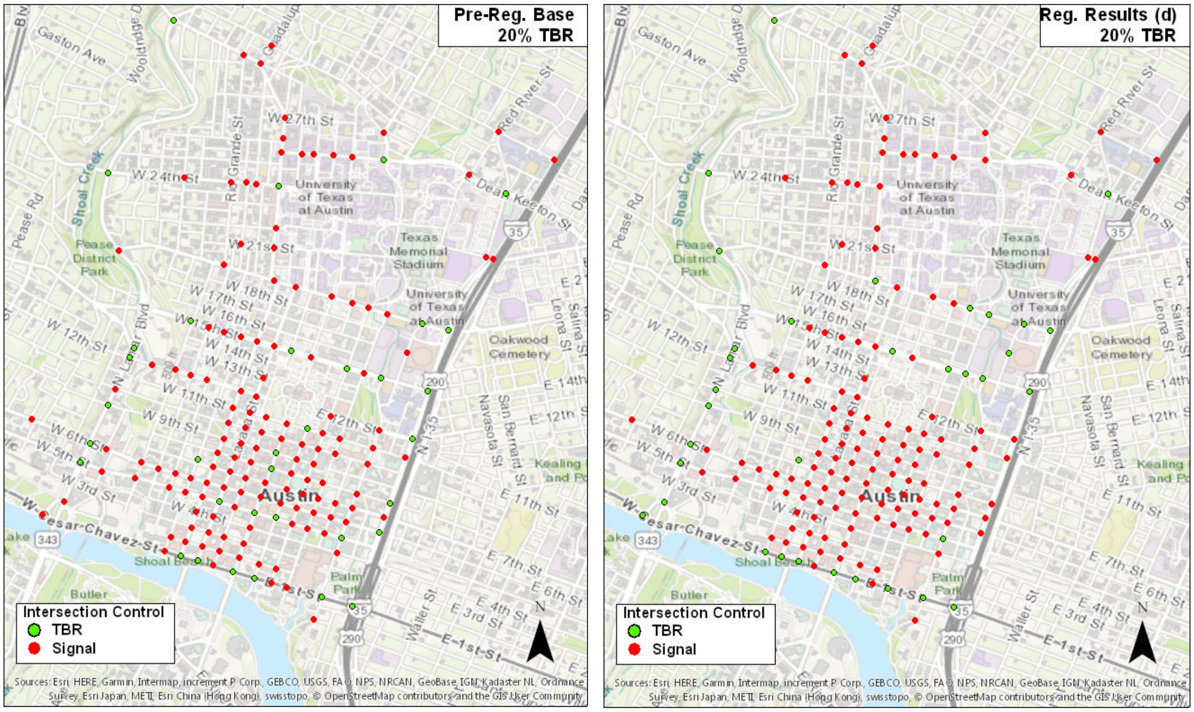


Figure 6: Intersection placements found by the limited reservation GA with 20% TBRs (left) and 40% TBRs (right)

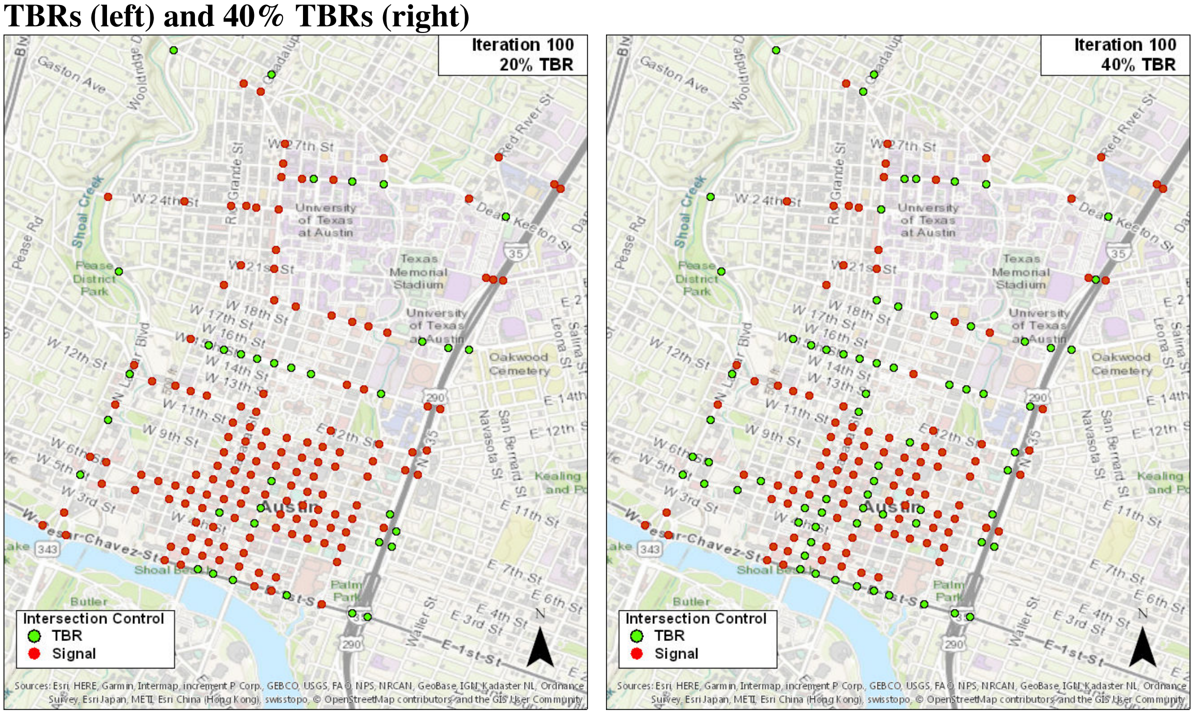
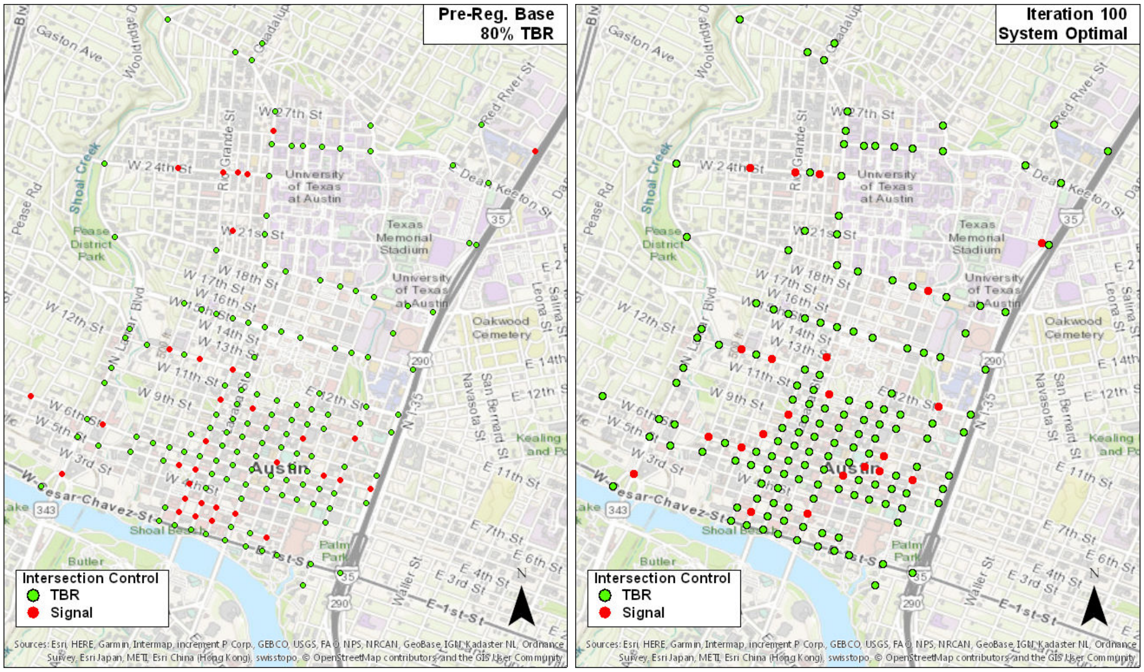


Table 5: A summary of used methods and TSTT

Proportions	TSTT						
	Base Cases	Dallas-based Reg.	Austin-based Reg.	Raw TSTT_DELTA	20% TBR GA	40% TBR GA	SO GA
0% TBR	6443.2						
20% TBR		6107.3	6136.8	5199.0	4873.9		
40% TBR		5501.0	5393.9	4669.4		4543.1	
60% TBR		5077.8	5020.7	4375.6			
80% TBR		4655.2	4725.1	4214.0			
100% TBR	4560.1						
86% TBR							4229.2

Figure 7: Intersection placements by pre-regression raw input data at 20% TBRs (left) and the system optimal GA solution



Chapter 4
Conclusion

This report looked at two potential technologies and their impact to the road network, with specific focus on their impact on AV adoption. The first problem discussed the shared autonomous vehicle routing problem. The first part of the report discussed a tabu search heuristic to solve the MILP formulation for the SAV routing problem. Most previous work on the SAV routing problem has not considered the effects of congestion on SAV route choice. Tabu search is a metaheuristic that must be tailored to the shape of the particular problem at hand. The success of the solution can be attributed to the inherent nature of tabu search in diversifying the search using an adaptive memory structure, tabu list, and the nature of swaps performed. The results obtained on a series of experiments on the Sioux-Falls network indicate that the heuristic is successful in routing SAVs under congestion as seen from the decrease in the average person travel times.

The heuristic was designed with the objective of minimizing the total person travel times. The heuristic could be adapted to solve for other objectives such as total vehicle travel time, total waiting time. One of the shortfalls of the heuristic is that it focuses on reducing the waiting time of the travelers. This could increase the number of empty repositioning trips of SAVs and hence, total system travel time (TSTT) does not decrease significantly. A better heuristic could be designed to reduce the number of empty repositioning trips and thus, improve TSTT as well.

In this report, we assumed that SAVs serve only one traveler at a time. Future work can easily accommodate ride-sharing constraints in the formulation and study the effects of ride-sharing on congestion. This heuristic could also be used in a combined mode and route choice model to study how travelers choose between SAVs and personal vehicles. Since we seek to model congestion due to vehicle flow owing to a large number of passengers, our formulation does not explicitly track individual travelers. Future work can modify the formulation to track individual travelers which helps track the longest waiting time for individual travelers.

The second part of the report explored the problem of optimally locating autonomous intersections in a network such that the experienced TSTT is minimized. The study discussed three methods to identify the intersections where reservations would produce the best system-wide benefits in terms of TSTT in Austin. In this section, the three methods are discussed in detail with regard to their relative utility in addressing the problem at hand. The raw data approach and the regression model approach look to identify the best performing intersections when modeled as reservations while the GA targets the best spatial orientation of a limited number of TBRs which produce the best system-wide benefits.

The raw data method ranks intersections in the decreasing order of $\Delta TSTT$ when the intersection is modeled as a reservation and as a traffic signal. Although this approach may be naïve in identifying a solution, it produces encouraging results in terms of TSTT. In fact, a 50% TBR raw data solution produces the TSTT observed by the base 100% TBR solution. This indicates that each intersection has an effect on TSTT and optimizing for each intersection can produce significant benefits in terms of TSTT. However, the raw data approach cannot identify the interdependencies within the network and hence, cannot identify congestion due to queue spillback at intersections with low capacity links.

The regression approach alleviates some of the drawbacks of the raw data approach. The regression model helps quantify the characteristics which contribute to the performance of an intersection modeled as a traffic signal and as a reservation. It is a method which can be used to characterize intersections as TBR or signal simply based on the properties of the intersection. However, the regression model did not produce satisfactory results in doing so. This is partly due to the fact that there may be several interdependencies in the data which the regression is incapable of identifying. It is entirely possible that data from the Dallas network was insufficient for the model to learn enough about the subtle differences between different intersections and hence, the model did not generalize well enough when used to predict on the Austin network. This was evident from the different $\Delta TSTT$ observed between the raw data approach and the regression. Although the regression was able to generalize sufficiently well in most cases, there were some intersections where the regression produced a significantly different $\Delta TSTT$ than the raw approach. The regression and the raw data approach look to address a fundamental aspect of the problem – they try to classify an intersection as a TBR or a signal based on its characteristics. However, neither model does well enough to capture subtle interactions in the network.

The GA approach is different from the previous two approaches in that it captures the interactions in the network like congestion due to spillback from low capacity intersections. However, the GA cannot identify classify intersections as TBR or traffic signal from the characteristics of each intersection and neither is it designed to do so. The GA approach is an optimization problem which identifies the spatial orientation of a given number of TBRs which produce the best system-wide benefits. This is evident from the discussion in section 3.2. For example, among the 40% TBR solutions for the raw approach, regression approach and the GA, the GA produces the best TSTT. Thus, it is imperative to find a middle ground between the regression approach and the GA approach in solving the problem of identifying intersections to model as TBRs.

Trends in the networks for the optimal placement of smart intersections were primarily seen in the form of corridors of consecutive smart intersections. These corridors are highly congested roadways within the Austin network including streets such as Lamar Blvd and 1st street and seemed to be primarily targeted by all methods of intersection placement. Many networks such as the regression resulting network and the raw data resulting network appeared to have many similar features including similar corridors in areas of the network in which few adjacent intersections were present, however some slight differences in smart intersection placement caused a rather large difference in TSTT between the two networks. The placement of smart intersections in corridors with few adjacent intersections seems to be a trend seen enough to assume a general rule of thumb out of, however further experimentation would need to be done to confirm the relation between smart intersection placement and clustering.

It would be wise to combine the relative benefits of the GA and regression approaches to identify a solution that produces the best system-wide benefits for a given budget of TBRs. One of the drawbacks of the GA model is that the initial population consists of individuals where TBRs are randomly distributed in the network. Due to this randomization, it is necessary to produce a large number of individuals in the initial population so as to ensure every possible orientation of the network is captured. It would be desirable to reduce the number of individuals in the initial population because such a change would drastically reduce the running time of the GA and would also ensure that the GA converges to the optimal solution quickly or at least converges to a local minimum quickly. It is worth exploring the possibility of using the results of the regression to identify the initial population of the GA. This would enable the GA to solve the problem quickly and also combine the benefits of both approaches.

Future work includes obtaining more data for a possibly different type of nonlinear regression as well as developing different optimal solution heuristics and algorithms tailored to the DTA model used. Including clustering into an algorithm or conducting isolated intersection experiments would be useful in deciding the relation between smart intersection proximity and system-wide congestion and would help to uncover possible network interdependencies between intersections and generalize them.

Bibliography

1. McKinsey & Company (2016), Automotive Revolution – Perspective Towards 2030: How the Convergence of Disruptive Technology-driven Trends Could Transform the Auto Industry; at www.mckinsey.de/files/automotive_revolution_perspective_towards_2030.pdf.

2. Fagnant, D. J., & Kockelman, K. (2015). Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, *77*, 167-181.

3. D. J. Fagnant and K. M. Kockelman. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transportation Research Part C: Emerging Technologies, 40:1–13, 2014.

4. D. J. Fagnant, K. M. Kockelman, and P. Bansal. Operations of a shared autonomous vehicle fleet for the austin, texas market. Transportation Research, 28, 2015.

5. M. W. Levin, T. Li, S. D. Boyles, and K. M. Kockelman. A general framework for modeling shared autonomous vehicles. Computers, Environment, and Urban Systems, 2017b.

6. Levin, M. W. (2017). Congestion-aware system optimal route choice for shared autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, *82*, 229-247.

7. P. Toth and D. Vigo. The vehicle routing problem. Society for Industrial and Applied Mathematics, 2001.

8. Willard JA. Vehicle routing using r-optimal tabu search. Master's thesis, The Management School, Imperial College, London. 1989.

9. Osman, Ibrahim Hassan. "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem." *Annals of operations research*41.4 (1993): 421-451.

10. I.H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41:421–451, 1993.
11. E.D. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23:661–673, 1993.
12. M. W. Levin and S. D. Boyles. Effects of autonomous vehicle ownership on trip, mode, and route choice. Transportation Research Record: Journal of the Transportation Research Board, (2493):29–38, 2015a.
13. G. H. de Almeida Correia and B. van Arem. Solving the user optimum privately owned automated vehicles assignment problem (UO-POAVAP): A model to explore the impacts of self-driving vehicles on urban mobility. Transportation Research Part B: Methodological, 87:64–88, 2016.
14. T. D. Chen, K. M. Kockelman, and J. P. Hanna. Operations of a shared, autonomous, electric vehicle fleet: implications of vehicle & charging infrastructure decisions. Transportation Research Part A: Policy and Practice, 2016.
15. L. D. Burns, W. C. Jordan, and B. A. Scarborough. Transforming personal mobility. The Earth Institute, 2013.
16. J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus. On-demand high- capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences, page 201611675, 2017.
17. J.-F. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. Annals of Operations Research, 153(1):29–46, 2007.
18. G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: a classification scheme and survey. Top, 15(1):1–31, 2007.
19. G. Berbeglia, J.-F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. Eu- ropean journal of operational research, 202(1):8–15, 2010.
20. J.-J. Jaw, A. R. Odoni, H. N. Psaraftis, and N. H. Wilson. A heuristic algorithm for the multi- vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B: Methodological, 20(3):243–257, 1986.

21. T. Gschwind and S. Irnich. Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. Transportation Science, 49(2):335–354, 2014. Gschwind and Irnich, 2014
22. J.-F. Cordeau. A branch-and-cut algorithm for the dial-a-ride problem. Operations Research, 54(3):573–586, 2006.
23. S. Ropke and J.-F. Cordeau. Branch and cut and price for the pickup and delivery problem with time windows. Transportation Science, 43(3):267–286, 2009.
24. G. Berbeglia, J.-F. Cordeau, and G. Laporte. A hybrid tabu search and constraint program- ming algorithm for the dynamic dial-a-ride problem. INFORMS Journal on Computing, 24(3):343–355, 2012.
25. O. B. Madsen, H. F. Ravn, and J. M. Rygaard. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Annals of Operations Research, 60(1):193–208, 1995.
26. M.Gendreau, A. Hertz,andG. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40:1276–1290, 1994.
27. M. Gendreau, A. Hertz, and G. Laporte. New insertion and post optimization procedures for the traveling salesman problem. *Operations Research*, 40:1086–1094, 1992.
28. C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B: Methodological, 28(4):269–287, 1994.
29. Lighthill, M. J., & Whitham, G. B. (1955, May). On kinematic waves. II. A theory of traffic flow on long crowded roads. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences (Vol. 229, No. 1178, pp. 317-345). The Royal Society.
30. Ziliaskopoulos, A. K. (2000). A linear programming model for the single destination system optimum dynamic traffic assignment problem. Transportation science, 34(1), 37-49.
31. Ziliaskopoulos, A. K., & Waller, S. T. (2000). An Internet-based geographic

information system that integrates data, models and users for transportation applications. Transportation Research Part C: Emerging Technologies, 8(1), 427-444.

32. Transportation Networks for Research Core Team. *Transportation Networks for Research*. https://github.com/bstabler/TransportationNetworks. Accessed June, 2017.
33. Dresner, K., & Stone, P. (2004, July). 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.
34. Dresner, K., & Stone, P. (2006, July). 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.
35. Fajardo, D., Au, T. C., Waller, S., Stone, P., & Yang, D. (2011). Automated intersection control: Performance of future innovation versus current traffic signal control. *Transportation Research Record: Journal of the Transportation Research Board*, (2259), 223-232.
36. Li, Z., Chitturi, M., Zheng, D., Bill, A., & Noyce, D. (2013). Modeling Reservation-Based Autonomous Intersection Control in VISSIM. *Transportation Research Record: Journal of the Transportation Research Board*, (2381), 81-90.
37. Dresner, K., & Stone, P. (2006, May). 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*.
38. Schepperle, H., & Böhm, K. (2008, July). 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.

39. Vasirani, M., & Ossowski, S. (2010). *A market-based approach to accommodate user preferences in reservation-based traffic management*. Technical Report ATT.
40. Carlino, D., Boyles, S. D., & Stone, P. (2013, October). Auction-based autonomous intersection management. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on* (pp. 529-534). IEEE.
41. Hausknecht, M., Au, T. C., & Stone, P. (2011, September). Autonomous intersection management: Multi-intersection optimization. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (pp. 4581-4586). IEEE.
42. Carlino, D., Depinet, M., Khandelwal, P., & Stone, P. (2012, September). Approximately orchestrated routing and transportation analyzer: Large-scale traffic simulation for autonomous vehicles. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on* (pp. 334-339). IEEE.
43. Levin, M. W., & Boyles, S. D. (2015). Intersection Auctions and Reservation-Based Control in Dynamic Traffic Assignment. In press in *Transportation Research Record*.
44. Bento, L., Parafita, R., Santos, S., & Nunes, U. (2013, October). 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.
45. Qian, X., Gregoire, J., Moutarde, F., & De La Fortelle, A. (2014, October). 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.
46. Levin, M.W., & Boyles, S.D. (2015c). A multiclass cell transmission model for human and autonomous vehicle roads. Accepted for publication in *Transportation Research Part C: Emerging Technologies*.
47. Dresner, K., & Stone, P. (2006, May). 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*.

48. Dresner, K. M., & Stone, P. (2007, January). Sharing the Road: Autonomous Vehicles Meet Human Drivers. In *IJCAI* (Vol. 7, pp. 1263-1268).
49. Patel, R., Levin, M.W. and Boyles, S.D., 2016. Effects of autonomous vehicle behavior on arterial and freeway networks. *Transportation Research Record: Journal of the Transportation Research Board*, (2561).