The Dissertation Committee for William Eric Alexander

certifies that this is the approved version of the following dissertation:

Bridging the Gap: Unifying Transportation Planning and Operations Through Enhanced Travel Demand Modeling

Committee:

Stephen D. Boyles, *SUPERVISOR*

Chandra R. Bhat

Alexander M. Hainen

John J. Hasenbein

Randy B. Machemehl

Bridging the Gap:

Unifying Transportation Planning and Operations Through Enhanced Travel Demand Modeling

by

William Eric Alexander

Dissertation

Presented to the Faculty of the Graduate School

of the University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

May 2024

This work is dedicated in loving memory of Mary Janette McDonald Sims, who taught me to play games in a way that no one else could.

Acknowledgement

This monumental effort would not have been possible without the love and support of so many people. I am truly blessed with a strong community, and the names of those to whom I am indebted are innumerable. I wish to thank especially the faculty, staff, and student body of the University of Texas transportation engineering program for providing a place to call home these last six years. I also wish to thank all the members of my supervisory committee – the amount of knowledge, guidance, and patience they have given me is overwhelming to consider, and I am privileged to have been able to study under these titans of their fields. Additionally, I wish to thank the organizations that have supported my research financially, including the Data-Supported Transportation Operations and Planning Center (D-STOP), the Texas Department of Transportation (TxDOT), the North Central Texas Council of Governments (NCTCOG), and the United States Department of Transportation's Dwight D. Eisenhower Transportation Fellowship Program (DDEFTP). I similarly wish to thank my supervisors at Resource Systems Group for granting me the opportunity to continue this research while beginning my career in the consulting field. Finally, I would be remiss to not thank my family, both that into which I was born and that which I chose. Their unwavering encouragement has been the fuel that has kept me going throughout this process, and for that I am indescribably grateful.

Bridging the Gap:

Unifying Transportation Planning and Operations Through Enhanced Travel Demand Modeling

by

William Eric Alexander, Ph.D.

The University of Texas at Austin, 2024

SUPERVISOR: Stephen D. Boyles

Travel demand modeling is a necessary and useful tool for municipalities to better allocate resources, and much research has been developed on how best to calibrate and apply these models. This dissertation identifies and addresses a disconnect which exists between the modeling and operations fields, seeking to bridge the gap for improved performance in both the predictive capabilities of these models as well as the operational efficiency of the ensuing equilibrium that will develop between road users' route choices and managers' traffic signal timing decisions made based on these models. A major engineering contribution we detail is the development of *wrap*, a cross-platform, free-and-open-source travel demand modeling software package. Scientific contributions presented in this dissertation include a novel method for calibrating trip generation rates based on a sample of roadway volume data as well as a detailed investigation of pressure-based traffic signal timing optimization. This latter investigation introduces three novel pressure functions, offering improved route choice equilibria by balancing signal timings and drivers' anticipated efforts to minimize their cost of travel. This work also introduces preliminary contributions of practical benefit, including the use of machine learning for corridor travel time predictions and reinforcement learning for traffic signal control to better position this research for application in the field. The collaborative potential between improved travel demand models and transportation operations policies is strongly emphasized, with the contributions of this work intended to facilitate synergy in these two fields for the development of a more efficient transportation network.

Table of Contents

List of Tables

List of Figures

0. Introduction and Contributions

In the roadway transportation system, travel demand is modeled by two separate yet equally important groups: the long-range planners who predict the value of infrastructure investments and the operations managers who tune these investments to best serve the public. These are their stories.

We open with a panoramic view of the fictional city of Springfield, with a population of about 50,000 people. An economically disadvantaged city, Springfield does not invest heavily in the development resources they are required to have. As a result, the city's travel demand model is fairly weak and simplified. What's more, the traffic signals in Springfield are obsolete and only allow for fixed-time signal control, that is, without any ability to respond to changes in conditions. Thus, these signals must have their signal timings set according to the expected volume modeled in their travel demand model. Nonetheless, this model is not terribly accurate, and little is done to calibrate it due to the prohibitive costs associated with such a task.

However, Springfield does have a program by which they will periodically measure volumes on certain roadways in the city. While this approach is not an extensive measurement of travel demand across the entire Springfield network, it does provide quality data points for volumes on the selected roadways for the studied time periods. So what use is this data to the city? They do not have the resources to provide this data for all roadways, and the sensors available do not provide real-time data. In this dissertation, we detail an approach which can provide a cost-effective way to improve the efficacy of traffic management technologies without breaking the bank. Through this, we aim to bridge the gap in practice between the planning and operations functions of cities' administrations.

Modeling the State of Travel Demand

Cities build models not only of travel demand behavior, but also of their roadways to improve the state of traffic control. These models can propagate the expected traffic across a model of the roadway network and provide information on the efficiency of signal timings. But this is only as reliable as the underlying approaches' expected demand at any time, which in ordinary practice come from a travel demand model. Thus, it is notable that past research has shown the efficacy of improving certain parameters of travel demand models using sample field measurements of link volumes. To that end, we can build models which become progressively more accurate, and which therefore provide more accurate estimates of costs incurred as a result of management policies.

First, we model what we expect traffic to look like in the long run. This macro-state defines, for various population segments, the number of trips taken to/from each transportation analysis zone, the pairing of those trips, the mode choice, and the route choice. These in turn yield a predicted traffic volume on the roadways and can be discretized into various timeframes. We will summarize in Chapter 1, the literature review, the basics of long-term travel demand forecasting, with a focus on maximizing entropy in the forecasted model. In laying the foundation for the work we contribute through this dissertation, we detail in Chapter 2 our past development of a large-scale travel demand modeling software upon which we will build our models. We aim to also identify key design and implementation decisions which impact performance and provide for extensibility.

Improvements to Travel Demand Modeling

Propagation models have been developed which replicate the flow of traffic across a network accurately, but such prototypes are dependent on obtaining a reliable profile of travel demand across both the modeled network and the modeled time period. These travel demand models have also been under tremendous scrutiny in the research community, with a bevy of methods of modeling the myriad facets of trip-making behavior in a transparent and explainable manner. However, these models of long-term behavior grow stale with time and must be updated regularly, requiring additional data collection.

Past research has shown that the error in travel demand models grows as more parameters of trip-making are predicted, although this is somewhat mitigated through the solving for a user equilibrium in route choice and may necessitate additional iterations of modeling. Calibration efforts have primarily focused on tuning the matrix of modeled demand immediately prior to performing route choice, determining parameters for trip distribution and mode choice models. A common means to do so is to modify model parameters such that predicted link flows match observed traffic counts more closely. However, while trip generation is an important component in the calibration of travel demand models, we do not see evidence that trip generation parameters have been calibrated using a traffic count-based approach. Given that trip generation rates are a highly observable travel behavior component that can be estimated from simple household travel surveys, we present a means to perform such a calibration herein to address this gap in the literature. Previous work we have accomplished and will detail in Chapter 3 shows that this is an effective means of improving trip generation model parameter estimates.

Improvements to Transportation Systems Management

In addition to improvements to the modeling capabilities of cities, we also aim to improve how cities respond to those forecasts in day-to-day operations. By way of analogy, consider we seek to develop agents, that is, players of a game in which they take on the role of a traffic manager, with the aim to score as highly in the game as possible in the face of responses from the traveling public, each member of which is playing their own game and seeking to maximize their own reward. The agents may perform in a variety of manners across different scenarios, and the scores may vary depending on what rules are in place for scorekeeping. Thus, it is timely to loosely define a) the "players" we seek to investigate and b) the rules of the different games which will be played. Specifically, we present findings that improve the abilities of one class of player, a city's transportation systems management and operations professionals, to not only predict how other players (namely, the traveling public) behave but also to respond to these opponents' strategies, nudging the state of the game towards a more efficient outcome.

In Chapter 4, we investigate a series of models and policies that may improve the ability of traffic signal controllers to anticipate changes in traffic volumes attributable to travelers "playing the game" by seeking out shortcuts. We presume in the work herein that traffic signal controllers follow basic behavioral patterns, specifically the standard ring-barrier behavior, with the ability to alter the fraction of time each turning movement is allowed to proceed through an intersection. By anticipating the reactions of travelers to changes in signal timings, controllers can preemptively adapt their timing policies to mitigate the delays incurred due to signal controllers.

As shown in prior research, the iterative search for an equilibrium between signal timings and route choices is attainable given limited assumptions on the behavior of transportation network dynamics. We therefore show a mechanism that improves upon the current state of the art by more accurately reflecting the negative impacts of signal delay, especially as would impact travelers' route choices. The chosen representation of this "pressure" to accommodate traffic is of significant impact on the efficiency of the resulting traffic equilibrium, so we present a representation that both more accurately reflects the true impacts attributable to signal timings and provides for competitive performance in common performance metrics. This approach improves existing models by reflecting the long-run average signal delay in both over- and under-saturated networks, depending on generalizations common in queueing theory and static traffic assignment principles to address the weaknesses of the current state-of-the-art pressure functions.

Advancements in Modeling Vehicle Propagation and Network Operations

From the outputs of travel demand models, we can build not only a time-varying travel demand profile, but, through dynamic loading, we can also build link-level time-varying demand profiles that can further be discretized in space using traffic propagation models such as the cell transmission model, link transmission model, or car-following model-based approaches.

What's more, in addition to the volume forecast data available from these long-term models, we are also able to obtain even finer-resolution data directly from sensors in the field. These allow us to not only build short- to medium-term models of traffic demand on a link, but to predict corridor-level travel times and delays. We will detail in Chapter 5 pilot approaches we have undertaken based on recurrent neural networks which provide reliable means to predict corridor travel times from this data, thus providing a heuristic for calibrating these propagation models.

A wide variety of efforts have been put forward which have advanced practitioners' abilities to model, analyze, and control traffic. In terms of modeling, techniques for representing the state of traffic on a network have advanced from the simplistic volume-delay functions of traditional route choice analysis to partial differential equations which model individual vehicles' trajectories to intensively detailed microscopic simulation techniques. While these can provide incredibly detailed information regarding the state of traffic across entire metro areas, control improvement techniques tend to aim to minimize the amount of information which must be provided, in order to reduce the complexity of the optimization problem. To do so, practitioners frequently spatially limit the scope of networks they analyze, omitting benefits that may be obtained on a more macroscopic level, such as improving throughput of a lengthy corridor. We detail in Chapter 6 preliminary work we have undertaken towards developing a standards-compliant reinforcement learning scenario which interacts with a market-competitive microsimulation software, PTV VISSIM, to simulate a real-world traffic signal with a high degree of realism as well as an initial investigation in said environment of equity in the distribution of traffic signal timings developed by a reinforcement learning agent.

How to Read this Document

As the goal of this dissertation is to bring closer together the modeling and operations functions of transportation engineering, we aim to present two approaches which improve one of these bodies of work by making better use of data from the other body. Additionally, to prepare this work for future implementation in the field, we document a large software engineering project that is foundational to this

research, as well as preliminary investigations that aim to better make use of intelligent transportation systems, the data they generate, and modern computational advancements to improve modeling and operations decision-making.

The literature review we provide in the ensuing chapter details the basic concepts relevant to this dissertation, specifically documenting common practices in industry today for predicting travel behavior, managing roadway travel through traffic signal control, and utilizing machine learning to characterize behavior which is difficult to model directly. This review of the foundations is then followed in Chapter 2 by a discussion of the largest engineering contribution of this work: the implementation of basic travel demand models including a highly efficient and scalable algorithm to perform static traffic assignment. As the experiments we describe in Chapters 3 and 4 depend on the practical benefits of this implementation, the software we have developed, *wrap*, is of central importance to the scientific contributions of this dissertation.

The first of these scientific contributions, discussed in Chapter 3, details an effective means to calibrate trip generation rates in a traditional trip-based travel demand model utilizing a sample of roadway volume count data. We specifically focus on the selection of a gradient descent method that approximates the gradient by performing an iteration of the full model for each candidate parameter to be tuned, thus providing a search direction by perturbing the parameters and evaluating the model's forecast response. This simplification addresses the highly complex nature of large travel demand models by utilizing the warm-start functionality provided in *wrap* to provide an initial, transferrable approximation that requires only light iteration to reach equilibrium, thus improving computation time over methods that directly calculate the gradient and Hessian matrix. This, in turn, gives a better approximation of traveler behavior which can be functionally implemented using common detection technology present in the field today.

Our second main contribution to the current scientific literature, presented in Chapter 4, is a collection of novel pressure functions which reflect the potential impacts of traffic signal timings on roadway network performance. These pressure functions, when iteratively balanced against modeled route choice responses in the traffic assignment problem (as implemented in *wrap*), yield improvements to the ensuing long-run equilibrium. We compare the novel pressure functions we present against two pressure functions which have stood out in this area of research and detail the flaws of these two which are addressed by the novel functions. These pressure functions, therefore, provide an opportunity for practitioners to better anticipate traveler responsive routing and to develop signal timings which can nudge the route choice equilibrium towards a better performance standard.

Finally, Chapters 5 and 6 detail initial work towards improving the operational modeling capabilities of current practice. The experiments we detail in Chapters 3 and 4 depend on a reliable ability to predict travel times based on roadway demand and to model the dynamics of traffic at a reasonable level. Therefore, we establish preliminary areas of opportunity to improve these modeling resources through common machine learning techniques, showing significant progress in corridor travel time prediction and providing a standards-compliant reinforcement learning framework that pave a way towards implementing the contributions of this dissertation in everyday practice.

1. Literature Review

Travel Demand Modeling and Forecasting

In the United States, a policy board called a **metropolitan planning organization (MPO)** is required by law for all urbanized areas with a population of at least 50,000 people. These MPOs are intended to "promote the safe and efficient management, operation, and development of resilient surface transportation systems" and to "encourage the continued improvement and evolution of […] transportation planning processes" (Office of the Law Revision Council, 2021). To do so, MPOs are required to undergo a transportation planning procedure that includes developing forecasts of travel demand for a minimum of 20 years. A similar process is required of each state, in order to "provid[e] for the development and implementation of the intermodal transportation system of the State" (Office of the Law Revision Counsel, 2021). These planning processes in essence require the responsible organizations to develop statistical **travel demand models (TDMs)** to predict not only future population changes, but also how those changes as well as other factors influence how the population interacts with state and local transportation systems.

As could be expected, MPOs and states are rather diverse in their resources, and, therefore, the complexity of their respective TDMs. These can range from simple models, which assume that change in travel demand is tied directly to the change in population for an area, to the more complex activity-based models. However, of the MPOs whose modeling techniques are publicly known, the overwhelming majority utilize the traditional **Urban Transportation Modeling System (UTMS)**, commonly referred to as the **four-step model**, consisting of sequential modeling of trip generation, trip endpoint distribution, mode choice, and traffic assignment (Walthall & Walton, 2019). Adding further to this wide array of model complexity, some MPOs may omit the mode choice phase of their models, as they assume that the overwhelming majority of travel demand is satisfied by automobile trips. On the other hand, models developed with more resources may not only model the traditional four steps but also add others such as trip time distribution. Additionally, policy implications such as those of tolling mechanisms and future vehicle technology improvements may be considered in the most complex models (Dias, Nair, Ruíz-Juri, Bhat, & Mirzaei, 2020).

In the following subsections, we will provide a review of the various steps of the UTMS process. We will then discuss two techniques that, while increasing the scope of these models dramatically, can provide more flexibility in modeling trip behavior – namely, the combined modeling of multiple trip purposes and

multiple traveler classes. As will be shown, there may be myriad parameters incorporated into these models, and each of these brings with it a certain amount of estimation error. We will then briefly discuss prior efforts to quantify and mitigate this error through model calibration procedures.

Trip Generation Modeling

To begin the process of modeling a population's travels, practitioners must estimate the number of trips that will be produced from, or attracted to, a given **Travel Analysis Zone (TAZ)**. The number of trips generated at each TAZ depends on a variety of conditions, including housing availability, employment, land use, and many others. To quantify the "generators" of these trips, modelers begin with data collection, traditionally in the form of a travel survey. These surveys began as in-person interviews or travel diaries, with travelers detailing their household trip-making history in order to build a sample distribution of trip-making frequency. These methods, however, are not only subject to sampling biases, but can also face difficulties in response rate and cost (Mukherjee & Kadali, 2022). To supplement these samples, modelers have long leaned on position-tracking technologies in vehicles, smartphones, and more to calibrate these distributions and account for skewed representativeness of survey data (Bricka & Bhat, 2006).

The most indispensable resource in performing trip generation estimation is the Institute of Transportation Engineers' *Trip Generation Manual* (Institute of Transportation Engineers, 2021). This collection of traffic study data provides trip generation rates for a wide range of land uses, ranging in specificity from the generic, such as Hotel (Land Use Code 310), to the quite precise, e.g., Coffee/Donut Shop with Drive-Through Window and No Indoor Seating (Code 938). These rates are an estimate of how many trips are generated per a certain unit relevant to the individual land use. For example, while Nursing Home (Code 620) measures trips per nursing home beds, the unit of generation for Marijuana Cultivation and Processing Facility (Code 109) is in terms of square feet of gross floor area. What's more, trips may be measured across multiple strata for the same use case, such as Church (Code 560) which can be measured by attendees, seats, or square feet of gross floor area. These rates are also delineated by weekday and indicate the time period and, optionally, the directional distribution.

The data points provide an estimation parameter that can be used in models that not only reflect existing conditions, but also for forecast environments as well. Thus, if a city, for example, is presented with a permit application for construction of a new mid-rise multi-family apartment building, they can estimate the impact that construction may have once the land use has been realized. In turn, this allows for

forecasting changes in traffic, transit usage, and more through modeling the remainder of the UTMS. Furthermore, MPOs may optionally develop more than one model scenario to account for, among other considerations, shifts in population growth, employment, and housing (Office of the Law Revision Counsel, 2021).

There are significant weaknesses in the *Trip Generation Manual*-based approach, as studies often feature sparsely populated data points or little context for the data points (such as vehicle occupancy rates). Additionally, the manual focuses heavily on vehicle-based trips, with little multimodal data available (Mukherjee & Kadali, 2022). Because of these and other weaknesses, rates are often a point of individual study for modelers, with the manual primarily acting as a reference work. Approaches for determining the appropriate rates to use in a model are varied, ranging from basic regressions to **multinomial logit (MNL)** models to models that rely on **machine learning (ML)** approaches such as **artificial neural networks (ANNs)**. While ML methods are gaining ground in efficacy and usage in generation modeling, the models they produce are often overly complicated or difficult to interpret. Thus, at times, practitioners often rely on more intuitive cross-classification, linear regression-based, or logit models, despite questions related to their ability to represent complex traveler behavior (Mukherjee & Kadali, 2022).

At completion of the trip generation phase of the UTMS, modelers have generated an estimate, for each TAZ in the model, of the number of trips produced from and attracted to the TAZ under the modeled scenario, represented as $\mathcal{P}_z$ and $\mathcal{A}_z$, where z is a TAZ in the set of modeled zones $\mathfrak{Z}$. These values, which take the form of a $2 \times n$ matrix or two vectors of length $n = |\mathfrak{Z}|$ and which we dub collectively a **production-attraction (PA) map**, are then typically fed into a **trip endpoint distribution (TED)** model, which is the topic of the next subsection. Alternately, in cities where mode choice modeling is less relevant, potentially due to substandard public transit options or a high dependence on personal vehicles, modelers may perform mode choice first to reduce model computational complexity.

Endpoint Distribution Modeling

Trip endpoint distribution models at a fundamental level exist to "express the percentage (probability) […] of trips made by users […] going to destination d, given the origin zone o […]" (Cascetta, 2009, p. 185). The earliest aggregated models included the simplistic Fratar growth model (Fratar, 1954), which can be used for trips transiting a metro area or for small cities that expect little change; as well as intervening opportunities modeling (Stouffer, 1940), which deals with individual behavior and lends itself to statistical

calibration. However, the far-and-away most popular method to model trip endpoint distribution is the gravity model (Easa, 1993, pp. 797-798).

Gravity models concentrate their efforts on basing trip destination shares based on both the number of trips attracted to a TAZ relative to those attracted to others, as well as the cost of travel between the production and attraction TAZs, with the former being proportional to the share of trips and the latter being inversely (Cascetta, Pagliara, & Papola, 2007). Gravity models are founded on the information theory concept of entropy maximization (Wilson, 1970), also known as information minimization. This method seeks to develop unbiased distributions (in this case, of trip endpoints) in the face of limited information (Jaynes, 1957).

To do so accurately, gravity models depend on understanding the likelihood of an endpoint selection. For travelers, the attractiveness of an OD pairing is dependent not only on the cost perceptions of the pairing, but also how much a traveler perceives the cost as an impediment to travel. This creates a sort of nested attractiveness function, wherein the true cost is estimated by the traveler, then a decision is made regarding whether or not this perceived cost forms an undue impediment to travel.

At the completion of the trip endpoint distribution step of the UTMS, a matrix $\mathcal{T}$ has been developed of dimension $n \times n$, for a model with n TAZs. This matrix, which we refer to as a **production-attraction (PA) matrix**, defines, for each pair of TAZs $(\mathfrak{o}, \mathfrak{d})$, the number of trips $t_{\mathfrak{o}\mathfrak{d}}$ that are produced at origin TAZ $\mathfrak{o}$ *and* are attracted to destination TAZ $\mathfrak{d}$. This matrix is then traditionally used as the input to a travel mode choice model, which we detail in the next subsection.

Mode Choice Modeling

The purpose of mode choice modeling is to determine, for a set of t trips between origin TAZ $\mathfrak{o}$ and destination TAZ $\mathfrak{d}$, the likelihood of a traveler choosing a particular mode of travel, such as driving a personal vehicle, walking, or riding public transit. In doing so, statistical models are developed which can represent travelers' discrete choices based upon cost (or, alternately, utility) perceptions associated with each mode option. A wide variety of models have been developed for performing this task.

Let the set $\mathfrak{M}$ represent the collection of travel modes available in a model. Upon completion of the mode choice step of the UTMS, the model yields a set of matrices $\boldsymbol{T} \coloneqq \{\mathcal{T}_{\mathfrak{m}} : \mathfrak{m} \in \mathfrak{M}\}$ in which, for a given matrix $\mathcal{T}_{\mathfrak{m}} \in \boldsymbol{T}$, each matrix element $t_{\mathfrak{o}\mathfrak{d}\mathfrak{m}}$ indicates, for each pair of TAZs $(\mathfrak{o}, \mathfrak{d})$, the number of trips travelers make which use mode $\mathfrak{m}$. The individual matrices $\mathcal{T}_{\mathfrak{m}}$ are thus called ***modal* production-attraction (MPA) matrices**.

Up until now, we have considered trips to be made solely by *travelers*, but we must now shift our focus to view trips as being made by *vehicles*. In this view, we must consolidate person-trips into vehicle-trips according to the occupancy of the vehicle of the mode in question. For now, we consider this to be consistent across each TAZ pair, i.e., there is a scalar quantity $o_{\mathfrak{m}}$ that represents the average occupancy of all of mode $\mathfrak{m}$'s vehicles. Thus, for each matrix $\mathcal{T}_{\mathfrak{m}}$ representing the number of person-trips, there exists a corresponding matrix $\tilde{\mathcal{T}}_{\mathfrak{m}}$ in which each element $\tilde{t}_{\mathfrak{o}\mathfrak{d}\mathfrak{m}} \in \tilde{\mathcal{T}}_{\mathfrak{m}}$ is defined as $\tilde{t}_{\mathfrak{o}\mathfrak{d}\mathfrak{m}} = \frac{t_{\mathfrak{o}\mathfrak{d}\mathfrak{m}}}{o_{\mathfrak{m}}}$. We term this corresponding matrix $\tilde{\mathcal{T}}_{\mathfrak{m}}$ an **origin-destination (OD) matrix**. Note, however, that the aggregation of occupancy rates per mode is unnecessary – it is equally feasible to believe that, for certain TAZ pairs, occupancy may be higher or lower than average. Thus, the occupancy factor *o* may be dependent not only on the mode, but also on origin and destination.

Once an OD matrix has been created, we have effectively quantified the demand for trips of certain modes for a modeled period. However, travelers may, over the course of their trips, utilize multiple facilities in their trip, be they of the same mode or otherwise. To put this into example, suppose a traveler chooses to ride by bus for a trip, but must make a transfer midway through the trip to another bus. In a more complex example, a driver will usually make trips that involve traversing multiple distinct roads while en route to their destination.

Given that there may be multiple routing options for any particular OD pair, and that the costs associated with these options may change depending on the volume of travelers using it, the final step of the UTMS iteratively calculates an equilibrium in which travelers seek to minimize their individual travel costs. In the next subsection, we detail this process, known commonly as either traffic assignment or route choice modeling.

Route Choice Modeling

Once an OD pair and mode of travel have been selected for a given trip, it is next necessary for a traveler to determine what route they will follow. While this may be very simple in cases such as transit trips, in which only one route is viable due to alternate routes' higher perceived costs, in many cases, especially those of automobile trips, the costs associated with multiple routes may vary greatly depending on the conditions on the roadways. The general purpose of this step of the UTMS is to determine the volume of travelers using an individual facility (e.g., on a given roadway or train). However, these facilities are usually treated as congestible goods, meaning that the volume associated with them alters the cost of these facilities as they are perceived by travelers. This creates an economic game – how should multiple

independent agents choose their routes such that they will minimize their travel costs, while others are doing the same and are, in the process, competing with all other travelers for a consumable good (the capacity of the individual facilities)?

The oft-cited original description of this game in the context of route choice is that of Wardrop (Wardrop, 1952). This work defined the concept of a **user equilibrium (UE)** assignment, as opposed to a **system optimal (SO)** assignment. The assignment (that is, the complete description of routing for all trips) for a SO solution seeks to minimize the total cost incurred, summed across all travelers; by contrast, a UE solution, as described by Wardrop, has two criteria it must meet: a) the costs incurred "on all the routes *actually used* are equal, and less than those which would be experienced by a single vehicle on any unused route" (emphasis added) and b) the average cost is minimized. While Wardrop initially focused solely on travel times as costs, the principle generalizes well to any form of cost function.

This principle was later codified by Beckmann et al. in 1956 (Beckmann, McGuire, & Winsten, 1956) as a mathematical optimization problem with the following objective function, with $c_{\mathfrak{ij}}(\cdot)$ representing a generalized cost of travel between nodes $\mathfrak{i}$ and $\mathfrak{j}$ in the set of modeled nodes $\mathfrak{N}$:

$$\min_{x,h} \sum_{\mathfrak{ij} \in \mathfrak{L}} \int_0^{x_{\mathfrak{ij}}} c_{\mathfrak{ij}}(x)dx \qquad \{1\}$$

The constraints on the objective function are three-fold:

$$x_{\mathfrak{ij}} = \sum_{\pi \in \Pi} \begin{cases} h_\pi & if\ \mathfrak{ij} \in \pi \\ 0 & otherwise \end{cases} \quad \forall\, \mathfrak{ij} \in \mathfrak{L} \qquad \{2\}$$

$$\sum_{\pi \in \Pi_{\mathfrak{od}}} h_\pi = \tilde{t}_{\mathfrak{od}} \quad \forall\, \mathfrak{od} \in \tilde{\mathcal{T}} \qquad \{3\}$$

$$h_\pi \geq 0 \quad \forall\, \pi \in \Pi \qquad \{4\}$$

Let us consider these constraints individually. The first, Equation {2}, requires that, for every link* $\mathfrak{ij}$ in the set of all modeled links $\mathfrak{L}$, the total number of link users is equal to the sum of all users of all paths π that includes link $\mathfrak{ij}$. Equation {3} in turn requires that, for all OD pairs $\mathfrak{od}$ in the assigned OD matrix $\tilde{\mathcal{T}}$, all demand $\tilde{t}_{\mathfrak{od}}$ between the two TAZs is served by some path π in the set of all possible paths between the

* Generally, links refer to any facility used to move between two nodes (i.e., locations) $\mathfrak{i}$ and $\mathfrak{j}$ in a network. This may be a transit line, but for our purposes here we treat links as roads.

two zones, denoted as Π_{bb}. Lastly, Equation {4} is a sanity-check requirement: no path can carry less than zero flow, as negative flow is not a concept in the realm of physical transportation.

Beckmann et al.'s above non-linear optimization problem is one of significant complexity, and as such, it has fostered a massive swath of research into algorithms which seek to solve it for various scenarios. In addition to a wide variety of solution mechanisms, several extensions to the original problem have been proposed, including those that incorporate stochasticity in traveler behavior and elastic travel demand. To unify the multitude of approaches, Bliemer et al. developed a general framework (Bliemer, Raadsen, Brederode, Bell, & Wismans, 2014) which aims to express all algorithms in the form of an iterative process of:

- network loading, i.e., the simulation of an assignment solution flowing across the network,
- route choice, i.e., an estimate of how travelers interpret travel costs associated with the amount of flow on each facility,
- performance measurement, in which a determination is made as to whether the chosen objective function continues to improve or, instead, has converged to a local optimum[†], and, if it has not converged,
- a shifting of path flows towards lower-cost routes.

We next delineate two major subclasses of this assignment problem – those assignment procedures which treat all departure times as incurring the same network conditions, known as **static traffic assignment (STA)**; and those which incorporate departure time from a given location in the network as an extra dimension of the problem, known as **dynamic traffic assignment (DTA)**. Generally speaking, STA and DTA models have a balance of pros and cons between them. While STA models are less computationally complex and offer a guaranteed unique solution, they suffer from a lack of realism due to their failure to incorporate the effects of time, and as a result, on congested links it is not uncommon for the UE solution to feature links with volumes higher than their stated capacities. By contrast, DTA models provide far more detail on network performance, including estimates of queue length, but require far more computation time and space, and there is no guarantee that a solution exists *at all*, much less that a unique one exists (Boyles, Melson, Rambha, & Duthie, 2013).

[†] As will be discussed shortly, in variants of the traffic assignment problem which constrain the problem as a convex optimization problem, this is also the global optimum.

In keeping with Bliemer et al.'s framework, we first consider the network loading models utilized in STA and DTA models. The differences in this component represent the largest divergence of the two, as static models assume that traffic flow is invariant with time, whereas the opposite is true in dynamic models (Patriksson, 2015). This is to say that, while in STA models, the travel costs associated with a link c_{ij} are merely convex functions of the long-term equilibrium flow x_{ij} on those links, in the more complex DTA models, links' flow and storage capacity change at each discrete timestep of the model. While the size of these timesteps is a design choice, an upper bound on their duration is the shortest amount of time required to traverse any link in the network in free-flow conditions (Boyles, Lownes, & Unnikrishnan, 2022).

In the remaining steps of the framework, the differences between STA and DTA models are less pronounced. In the case of route choice, the calculation of path costs generally follows the same procedure – that of the lowest cost path problem, with the added complication of time dependency in DTA models. Performance measurement is of high import to STA models, as the quest to find the unique solution of the convex optimization problem terminates when there is no improvement made in the objective function {1}. On the contrary, since no equilibrium solution is guaranteed to exist for DTA models, the performance measurement step is less critical in the route choice framework.

Finally, one significant difference between STA and DTA models is how, once one or more "shortcut" paths (i.e., ones with a lower cost) have been identified, flow from the previous network loading step is shifted from higher-cost paths to the newly identified shortcut paths. In STA literature, there exist three overarching categories of flow-shifting algorithms: link-based, which merely track the amount of flow on each link in the network; path-based, in which flow is tracked across all used paths for all origins; and bush-based, in which a directed acyclic graph (hereafter referred to as a **bush**) is built for each origin and which tracks the relative share of flow passing through each node i flowing across every link ij emanating from the node (or, alternately, that of every link ji flowing *into* the node). While the first of these two are compatible as well with DTA models (with some adjustments), bush-based approaches have not made significant inroads into DTA models due to the added structural complexity associated with tracking the extra dimension of time (Boyles, Lownes, & Unnikrishnan, 2022).

Time Distribution Modeling

One consequence of incorporating time as an added dimension in TDMs is that the departure times for trips are not known a priori and must therefore be modeled as an extra step in the UTMS. Since the

characteristics of transportation facilities may change throughout a day, travelers' behavioral choices may change as well. While each of the traditional four steps of the UTMS may be influenced by time of day, the complexity of TDMs (both in terms of computational time and required memory) rises significantly as more steps are modified to consider departure time.

A bevy of literature has been developed on the departure time choices of travelers, with initial efforts focusing on commuters' trips and later work expanding to trip purposes whose time constraints are more flexible. One common practice has been the discretization of a weekday into a series of travel periods, such as morning off-peak, morning peak (i.e., rush hour), mid-day off-peak, afternoon peak, and evening off-peak periods, though more recent efforts have turned attention to continuous-time models (Bhat & Steed, 2002).

These two types of approaches (discrete vs. continuous) may serve different purposes. STA models, for example, are not influenced at all by departure time, so practitioners may choose to model route choice for multiple discrete periods with independent STA models incorporating modified facility conditions for each time period (Model Development Group, NCTCOG Transportation Department, 2007). Alternately, if a modeler has developed a DTA model for an entire day, she may wish to seek out a continuous departure time distribution which can be discretized to match the model's timestep size.

Population Segmentation

Until now, we have considered trip-making behavior writ large. That is, we have disregarded certain factors which strongly influence the behavioral choices travelers make, especially due to the travelers' individual characteristics, their trip purposes, and their scheduling and trip combination patterns. In this subsection, we detail the impact of modeling diversity in these factors to create models which more accurately portray the vivid variety of a population.

One of the largest distinguishers in trip-making behavior is the purpose for which a trip is taken. The most common example of this is trips made as commuter trips, which are typically made during the peak hours, and which are typically drawn from heavily residential areas to areas of high employment, as well as vice versa. However, not all trips are alike in this regard, with non-work trips having more flexibility in their departure time (Bhat & Steed, 2002) as well as more destination alternatives to choose from.

Further complicating travel behavior is the concept of trip chaining, in which travelers combine multiple trips, either of the same or different purposes, to reduce the amount of travel. These combined trips are

called tours, and strategies for modeling the tour-making behavior of travelers have been under considerable scrutiny as part of the activity-based modeling approach.

Finally, we consider the fact that not all travelers are alike in their resources. To model different classes of traveler, practitioners often delineate a population along a variety of classifications such as income level, car ownership, and number of children (Model Development Group, NCTCOG Transportation Department, 2007). In turn, these influence the parameters of TDMs greatly. For example, the number of trips for the purpose of taking children to school will be negligible for travelers without children, but for households that have kids, this trip generator can't be ignored.

Another manner in which traveler classification is frequently influential is in route choice. Consider a network which features a large presence of toll roads – travelers with high income will be more likely to not see these tolls as a significant cost to use these facilities, compared with lower income travelers that may see a significant burden in the tolls. This ties closely with the concept of a traveler's **value of time (VOT)**, that is, the amount of money a traveler would expect their travel time to be worth, usually expressed as a unit of currency per unit of time. Not only does this factor tie closely into a person's routing behavior, but it also determines whether a traveler is more likely to use public transit as their travel mode. If a trip requires more travel time when performed via transit than via driving, but the cost of parking at the destination is significantly higher than the transit fare, travelers with low VOT are more likely to opt for the slower option.

This effect can also be seen in classifying trip-makers according to vehicle ownership. In a household that does not own a car, travelers are dependent on public transportation or non-motorized travel modes such as biking or walking, leading to the term "captive riders" for these travelers. What's more, for households with more persons than cars, trip chaining can become even more complex, as family members may drive one another to their respective destinations, with one traveler retaining possession of the vehicle, then returning to pick up the other members once all activities are completed.

These population classifications influence every step of the UTMS, as has been shown for trip generation, mode choice, route choice, and trip chaining. For trip endpoint distribution, then, it should come as no surprise that these factors are significant in destination choice. This gives rise to the concept of an impedance to mobility. Also referred to as a travel friction factor, this concept is primarily utilized in the trip distribution step to determine the relative attractiveness of various destinations. These factors "must be established between and within zones to weigh such factors as time and cost of transportation that

discourage travel between areas" (Bevis, 1959). It then follows that these impedances are functions of the characteristics of population segments as well as the OD pair's costs. One means of defining this impedance function $I(\cdot)$ is through a gamma function (Model Development Group, NCTCOG Transportation Department, 2007). As shown in the below equation, this method incorporates three parameters (i, j, and k) which can be set for each population segment.

$$I(c_{od}) = i \cdot c_{od}^{-j} \cdot e^{-kc_{od}} \quad \{5\}$$

Consider that these parameters can be set independently for each modeled population segment's behavioral patterns. Just as well, the trip generation rates act as behavioral parameters in the UTMS, as do the parameters of the various mode choice models. The models of the individual steps of the UTMS can thus be combined, for each modeled population subsegment, into a larger trip demand model. These models take as input a set of demographic data about travelers, a set of trip cost data associated with the available modes of travel, and the expected behavioral parameters, outputting an OD matrix for the modeled subsegment. Going even further, the step of route choice may also be added to the mix, which requires additional information regarding the flow of traffic across the transportation network as well as further behavioral parameters regarding the routing preferences of travelers.

Efforts have been underway to efficiently solve these combined models since at least the late 1960's (Evans, 1976), with two primary approaches – those that are based on the traditional sequence of the steps of the UTMS, and those that "state the behavioral assumptions, translate them into mathematical conditions, and seek solutions that satisfy" them (Bar-Gera & Boyce, 2003). The former approach, while maintaining the traditional structure of models, suffers from an inconsistent treatment of costs that may change as a result of other steps in the model, most notably travel times. To address this, models often feature a feedback mechanism which iteratively updates the perceived cost inputs based on a previous iteration's predicted costs (Model Development Group, NCTCOG Transportation Department, 2007; Bar-Gera & Boyce, 2003). As for the latter approach, by restricting the model components to convex optimization problems, the iterative nature of the traditional approach is obviated, and computation time can be dramatically reduced. However, this has the practical effect of restricting certain model design components such as traffic flow and mode choice models (Bar-Gera & Boyce, 2003).

There are myriad design decisions that must go into developing a TDM, with each potentially having significant consequences on the validity of model forecasts. It has been shown that errors in initial steps of the UTMS propagate and amplify through the later steps, but that the attainment of a user equilibrium

in route choice may provide stability in output link predictions (Zhao & Kockelman, 2002). This was shown not only for models without feedback, but also with models which not only iterate through multiple feedback loops, but which also integrate with a land use model (Pradhan & Kockelman, 2002). In this work, it is suggested that the only parameters which hold significant long-run impacts are the growth rates for population and employment.

Model Calibration

Regardless of long-term influence on travel demand models, it is useful to develop accurate parameter sets for TDMs in order to better predict short-term travel behavior. While the true parameters of travel demand behavior are unique to each individual traveler, aggregating travelers of similar backgrounds reduces the complexity of the modeling challenge presented in modeling travel demand. Returning to our discussion of entropy maximization, modelers are tasked with finding the model which most accurately aligns with collected observation data from the field while still maintaining computational feasibility. To do so, a common approach is to begin with a model based on sample data from surveys or other means of data sampling, then to calibrate the models as more data is acquired. In doing so, a set of OD matrices can be estimated which reflect true traveler behavior more closely than those output by the sample data-based models.

Willumsen provides an overview of early techniques for **OD matrix estimation (ODME)**, dividing approaches into three techniques: estimating the parameters of a gravity distribution model (either linear or nonlinear), using Wardropian equilibria, and maximizing entropy (Willumsen, 1978). Among early efforts following the first approach are those of Robillard (Robillard, 1975) which used linear regression and a simple gravity distribution model together with proportional assignment. This type of model often fails to consider congestion as a factor in route choice, however, requiring more data or calibration to produce an estimate. These approaches can fail in an inner-city model, as gravity models are weak predictors when trips are short or travel outside a survey area (Willumsen, 1978). An example of the second approach is that of Nguyen (Nguyen, 1976) which considers the impact of congestion on route choice. However, the solution is unlikely to be unique, given how underdetermined the bilevel problem is. Nevertheless, many researchers have built off this method, with Cascetta and Russo demonstrating the non-uniqueness of the solutions in Monte Carlo simulation (Cascetta & Russo, 1997). Finally, Willumsen proposed an early instance of the third type (Willumsen, 1978).

These models can be solved quickly but have their fair share of shortcomings. They require knowledge of many link volumes (total trip count across the entire network, or productions and attractions at most major facilities), then do not consider planning data such as household surveys in their calculations, and they are sensitive to inconsistencies in observed volumes. As shown by Marzano et al., the high number of unknowns associated with direct OD matrix manipulation lead to a greatly underdetermined optimization problem when attempting to validate models in practice against limited count data (Marzano, Papola, & Simonelli, 2008). We aim to address this shortcoming in work we present in Chapter 3, in which we select a small subset of model parameters to calibrate to maintain determinacy with the number of counts available. Later efforts presented new models that focus on entropy maximization and information minimization, as well as solution techniques, that allow for inclusion of prior estimated OD matrix data in developing a new OD matrix estimate, and they appear to perform better than the gravity model alone in certain instances where the assumptions of the gravity model are unreasonable (Van Zuylen & Willumsen, 1980).

Combining the entropy maximization strategy with equilibrium assignment does not guarantee a solution under all approaches, but if the flow observations represent equilibrium, a model can generate the same results as a combined distribution and assignment model, owing to the parallel nature of the underlying statistical structures of the two (Fisk, 1989). Bernardin Jr. et al. develop an approach utilizing genetic algorithm heuristics and composite log likelihood maximization, but which depends on an initial demand model parameter set provided a priori and is “computationally intense” (Bernardin Jr., Trevino, Slater, & Gliebe, 2015). Cipriani et al. aim to adjust dynamic traffic demand matrices, i.e. OD profiles, in a computationally efficient manner that depends on a stochastic approximation search, but this loses the benefit of behavioral explainability that comes with calibrating models at earlier phases in the modeling pipeline (Cipriani, Florian, Mahut, & Nigro, 2011). Our work aims to retain this explainability by modifying solely those parameters which can be validated against traveler behavior observations.

Cascetta et al. develop a “quasi-dynamic” framework that aims to improve performance by assuming that destination choice varies significantly less than total demand over the course of a day, but this again does not address behavioral components of travel demand (Cascetta, Papola, Marzano, Simonelli, & Vitiello, 2013). Chen et al. propose a microsimulation-based approach which calibrates activity-based models’ utility-maximizing nested logit parameters, but their approach does not directly impact raw trip generation rates and is also limited in scalability due to its dependence on microsimulation, a computationally complex task (Chen, Prakash, Lima de Azevedo, & Ben-Akiva, 2020). Efforts have shown

promise in utilizing anonymized cell phone location data as a calibration target, but the efforts of Doorley et al. do not build a travel demand model but rather construct an OD matrix directly (Doorley, Alonso, Grignard, Maciá, & Larson, 2020) and that of Bwambale et al. calibrates trip productions per zone using a synthetic population, rather than adjusting an overarching behavioral parameter (Bwambale, Choudhury, Hess, & Iqbal, 2020). By contrast, the efforts we detail in Chapter 3 utilize modeling simplifications from static traffic assignment to improve efficiency and scalability while retaining the behavioral comprehensibility of traditional travel demand models.

Traffic System Management and Operations

In this section we define the basics of traffic signal control and the history of approaches thereto. We first begin with the basics of signal timing policy design, with a special focus on framing such a design problem in the context of queueing theory. We next detail technological advancements that have provided the ability for signal controllers to detect vehicles utilizing the roadway and respond in rudimentary fashion. Next, we examine novel advancements in applying adaptive modeling to the signal timing policy design problem. As these approaches are dependent on the ability to accurately model traffic propagation, we follow this with an overview of traffic flow propagation models. We conclude this section with a discussion of the metrics which are used to evaluate the performance of signal timing policies.

Signal Timing Basics

The traditional underlying mechanism for traffic signal control policies is the "ring-barrier" system, as illustrated in Figure 1. The diagram is broken up into two horizontal sections of compatible phases, one for the major street (on the left of the diagram) and another for the minor street (on the right). These combine to represent a discrete time period (of variable length), dubbed a "cycle," during which all active phases are served at least once[‡]. Between each of these sections is a demarcation called a "barrier," which acts as a safety checkpoint – no phase on one side of the barrier may begin until all other phases on the other side of the barrier are no longer in service.

Inside each horizontal section, the diagram shows two layers, called "rings." These can be considered as smaller-scale versions of the larger sections, as they encompass a set of phases, only one of which may be active at a time for a given ring. For example, in the figure, Phases 1 and 2 are mutually exclusive, but

[‡] Once a cycle is completed, a new (possibly identical) cycle begins.

either may be served in conjunction with Phases 5 or[§] 6, and vice versa. None of these four phases can be in service at the same time as any phase from across the barrier (i.e., Phases 3, 4, 7, or 8).

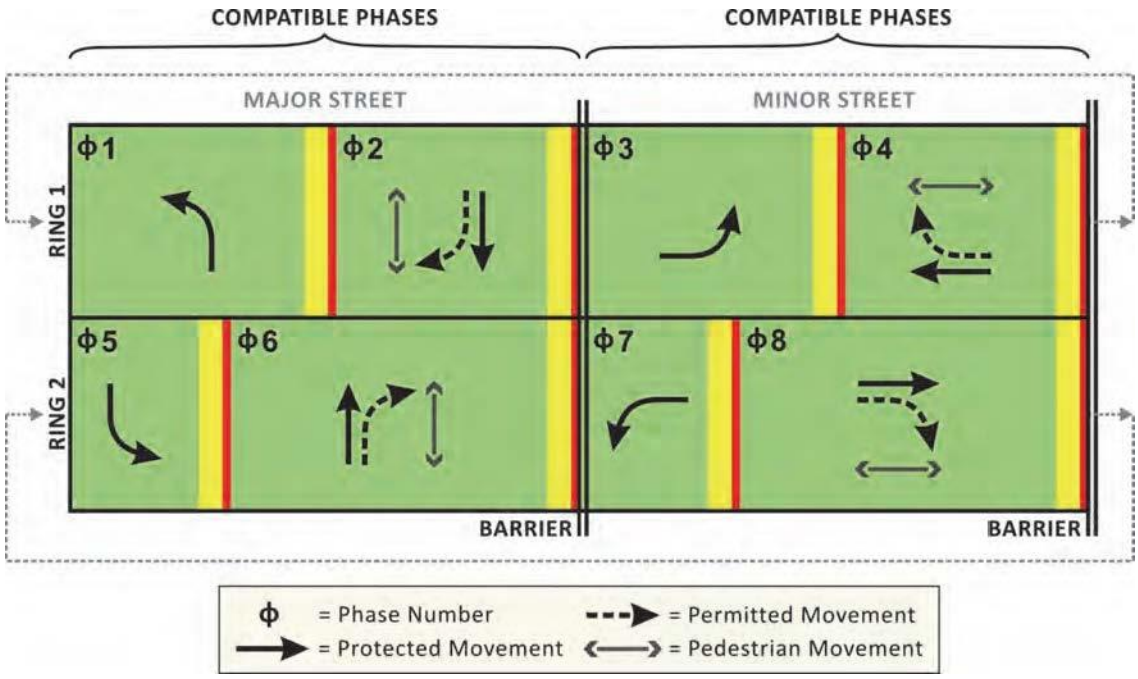

Figure 1: Basic ring-and-barrier diagram (Urbanik, et al., 2015)

A common consideration in traffic signal control is the idea of "startup lost time," in which the first few seconds of a green phase are lost due to human perception-reaction time, acceleration time, as well as distracted driving. We further incorporate an "all-red" period as a safety mechanism. These mechanisms, which act as a set-up/tear-down transition time, allow drivers to safely cross the intersection if they enter it just as the signal turns red. The amount of lost time is dependent, therefore, on the service rate, i.e., how long it takes vehicles to traverse the intersection. Since this is a safety mechanism, the value is typically set such that the combined all-red and startup lost time will encompass an overwhelming majority of services, usually around the 90th percentile of the service time distribution.

In practice, these considerations provide a lower bound on the amount of time that a phase can be in service, but a policy choice must be made to determine the time spent servicing that phase. At a minimum, traffic signal timing plans must define a duration for each inter-barrier interval as well as the "split" values which define, within the bounds of the safety constraints, how time is allocated to each phase. In situations where a series of signalized intersections are located in close proximity, progression – the ability

[§] Exclusive or.

of vehicles to traverse the sequence of intersections without stopping or slowing – can be established with common cycle lengths across the intersections, with their cycles "offset" by an interval of time corresponding to the travel time between each intersection (Day, et al., 2014).

Deciding on a signal timing policy which operates efficiently for all phases to be served is a complex task, however, especially in the face of limited information. Possibly the most well-known approach for determining a basic traffic signal timing plan is that of Webster (Webster, 1958). This method for creating timing plans according to the ratio of an approach's estimated flow to its saturation flow, and this may be well-suited for isolated intersections (Papageorgiou, Diakaki, Dinopoulou, Kotsialos, & Wang, 2003).

Later developments in the signal timing field have yielded methods which perform better when timing multiple intersections within close proximity of one another. The best-known of these is TRANSYT, which establishes a fixed cycle length for all intersections of the corridor, then allocates the phase timings according to the estimated volumes for each movement (Papageorgiou, Diakaki, Dinopoulou, Kotsialos, & Wang, 2003). However, these approaches depend on the accuracy of historical information, and therefore fail to adapt to real-time conditions.

Actuated Signal Control

To address the drawbacks of depending on historical traffic data, sensors have been developed that provide a means to measure the state of traffic in real time. The most basic form of these detectors is the loop magnetometer detector, a simple loop of wire embedded in the pavement at the stop bar. When a vehicle passes over this loop, the metal in the chassis induces an electric current which indicates the presence of a vehicle. Other common approaches include sensors which measure the pressure in a pneumatic tube which is compressed by vehicle tires as well as video- and radio frequency-based approaches (Jain, Saini, & Mittal, 2019).

These technologies provide information regarding a vehicle's presence at any given moment of time and can measure the count of vehicles that have passed over it if configured for such. This in turn allows for the signal timing pattern to, for example, maintain service on a busy main road rather than serving the side street if the detectors do not indicate that a vehicle is waiting for service on the side street (Day, et al., 2014). What's more, advances in technology have created a means to measure other traffic statistics such as speed, weight (Jain, Saini, & Mittal, 2019), and queue lengths (Liu, Wu, Ma, & Hu, 2009).

As a result of these advances, a wide variety of performance metrics are available to practitioners, most notably of delay, but also of progression. Combined with communications technologies, these metrics

provide better data analysis in near-real time to aid in improving timing policies as well as identify areas for improvement of the physical roadway network (Day, et al., 2014).

Adaptive Signal Control

The greatest innovation to arise from the availability of new traffic data is the ability to provide predictive models of traffic patterns. These "adaptive" systems modify the parameters of signal plans in response to the measurements obtained from the field, although algorithms which implement this strategy vary widely (Day, et al., 2014). One of the first such instances of this was SCOOT, which adapted TRANSYT to respond to traffic, calculating performance metrics in real time and updating signal timing plans if a superior choice is found (Papageorgiou, Diakaki, Dinopoulou, Kotsialos, & Wang, 2003).

Other means such as CRONOS and RHODES seek to solve model-based optimization problems, either periodically or over a rolling horizon. While these approaches do not explicitly consider the phase splits, cycle lengths, or offsets (Papageorgiou, Diakaki, Dinopoulou, Kotsialos, & Wang, 2003), the models must translate their decisions into actual means of signal control. Much like a locomotive traveling down tracks which are being laid immediately before it, this approach effectively builds a ring-barrier plan in real-time (Day, et al., 2014). As with earlier techniques, these approaches tend to rely on delay as the performance metric to optimize (Papageorgiou, Diakaki, Dinopoulou, Kotsialos, & Wang, 2003).

Propagation Modeling

One category of adaptive techniques is that of approaches which create a model of traffic flow using variables to model traffic propagation. These are not only dependent on the measurements obtained from the field but also on the accuracy of the underlying traffic propagation model (Papageorgiou, Diakaki, Dinopoulou, Kotsialos, & Wang, 2003).

The key requirement of such a model is that it be able to describe the travel time from one intersection to the next, such that optimization techniques may evaluate properly the impact of candidate signal timing policies. The simplest form of these models may merely provide the travel time or delay to free-flow travel time as an algebraic function of the volume, such as the volume-delay functions used in STA. The most common form of static volume-delay functions is the BPR function (Bureau of Public Roads, 1964), though other options such as conical functions (Spiess, 1990) are also found in practice (Model Development Group, NCTCOG Transportation Department, 2007).

More technically complex than static volume-delay functions are models which have been developed for DTA, as these not only measure travel time but also can represent queue lengths as well as basic dynamics of traffic flow. These approaches, based on the Lighthill-Whitham-Richards hydrodynamic traffic flow model, provide more accurate descriptions of the traffic state while remaining relatively computationally tractable even for larger networks. Two principal versions of this form exist – the cell transmission model and the link transmission model. While these are similar in their approaches to solving the partial differential equations associated with traffic flow theory, they each provide advantages over one another, with the cell-based model providing discretized approximations of the state of traffic across a link and link-based models providing per-vehicle location information (Boyles, Lownes, & Unnikrishnan, 2022).

In simulation-based approaches, individual vehicles propagate according to the behavior of a driver, with car-following models used to describe vehicles' expected interactions with other vehicles traversing the same link, as well as gap-acceptance models detailing the risk aversion of drivers when a dilemma arises, either from a traffic signal, vehicles from another roadway, or some other cause. The development of these models is of keen interest to the field (Aycin & Benekohal, 1999), as is their calibration from real-world data (Papathanasopoulou & Antoniou, 2015).

One recent development of note is the implementation of digital twin concepts in transportation modeling, as a means to provide additional data when real-world data is scarce. This has been utilized not only to provide an adaptive signal control method directly (Dasgupta, Rahman, Lidbe, Lu, & Jones, 2021), but also to provide additional performance metrics for reinforcement learning-based adaptive signal control approaches (Aslani, Mesgari, & Wiering, 2017). We detail in Chapter 6 an effort to incorporate the realism offered by microsimulation software within a reinforcement learning environment that meets the standard organizational structures and interfaces of current RL practice.

Performance Measurement

Of the approaches we have discussed thus far, each focuses efforts on reducing delay as the metric which should be minimized on a system-wide basis. However, this can create side-effects that may not be readily visible when this is taken to the extreme. For example, models may, in the course of pursuing a minimal delay, create large disruptions in progression (Papageorgiou, Diakaki, Dinopoulou, Kotsialos, & Wang, 2003). In doing so, traditional vehicles will consume more fuel and emit more greenhouse gas, so mitigating emissions may serve as a more preferable objective to mitigate the impacts of climate change (Stevanovic, Stevanovic, Zhang, & Batterman, 2009).

Another consideration which has received significant attention of late is that of equity in transportation. Consider that the delay and increased emissions caused by traffic signals create an economic externality absorbed by travelers. Thus, it is reasonable to aim to equitably distribute these externalities in selecting traffic signal timing plans (Li & Ge, 2014). However, it is not always clear how equity in transportation should be defined.

One means to measure equity, borrowed directly from the field of economics, is the Gini coefficient, a zero-to-one measure of statistical dispersion of income (Gini, 1936). Provided that delay and emissions impacts can be priced, it would seem a logical extension that the measure of dispersion of these costs can also be represented well by the Gini coefficient. However, we know of no applications in the literature of this metric to the impacts of traffic signal controllers. In Chapter 6 we detail an initial investigation which incorporates this equity metric as a component in RL agent performance evaluation, showing that equitable distribution of traffic signal delay is by no means a given and should be considered when developing "optimized" traffic signal timing plans.

Machine Learning in Traffic Management

This section provides an overview of several applications of machine learning techniques in traffic management, particularly those that relate to real-time detection and control mechanisms. We first discuss the application of unsupervised learning towards pattern detection or feature extraction in the realm of travel demand modeling and traffic state detection. We next provide a background into supervised learning in transportation, especially as a means of computer vision used for detection. Finally, we provide a discourse on the application of reinforcement learning as a means to improve policymaking in automated transportation control systems.

Unsupervised Learning

Perhaps the most abstract form of model development we will discuss herein, the strategy of unsupervised learning creates a model based solely on the inputs, with no concept of what models *should* look like (Bhavsar, Safro, Bouaynaya, Polikar, & Dera, 2017). That is, unsupervised learning aims to identify patterns in the data that can be investigated after the fact. For instance, one approach depends on vehicular data from a vehicle's OBD-II port in conjunction with smartphone sensor data to categorize driving styles (Gace, Pevec, Vdovic, Babic, & Podobnik, 2021). Another seeks to identify traveler mode choice using GPS trajectory records (Christos & James, 2020).

Given the somewhat unpredictable nature of the unsupervised methods, they are of little relevance to the tasks we propose herein. However, we acknowledge that unsupervised learning has the ability to identify patterns which may prove useful as descriptors of the state of traffic, either on a macro- or microscopic basis.

Supervised Learning

To give more guardrails to the model development process, supervised learning provides output data in addition to inputs, providing examples of how the agent is expected to learn. Two major categories of supervised learning approaches are those which seek to classify inputs into a discrete set of categories, for example the traditional levels of service (A through F); and those which seek to approximate a function through regression to the data (Bhavsar, Safro, Bouaynaya, Polikar, & Dera, 2017).

Most applicable to this work is the application of supervised learning to perform the task of computer vision of traffic video data. This provides a means to cheaply and effectively detect the presence of vehicles or pedestrians and measure the characteristics of traffic flow (Loce, Bernal, Wu, & Bala, 2013). One of the most valuable applications of this is in the real-time estimation of queue lengths at congested intersections (Liu, Wu, Ma, & Hu, 2009). This task remains a difficult one, so we aim to obviate the need for queue length estimation in our experiments presented in Chapter 4.

Reinforcement Learning

Reinforcement learning formalizes the task of learning in a way not seen in the prior two approaches. The method depends on an agent iteratively training a model based on a state representation vector, a set of actions which the agent can select for a given state, and a scalar reward function which critiques the performance of an outside process in response to the previous state and action choice of the agent. The agent must develop a policy model which maps a state description to an action selection, aiming to maximize the reward it obtains over the long run.

As we have discussed, the availability of detector data and probe data from individual vehicles has provided a wide variety of state representations available to practitioners, but these tend to divide into either aggregate (network- or link-wide) or disaggregate (per-vehicle) representations of vehicle positions and trajectories. While more high-resolution representations may provide a more precise description of the state of traffic, they not only significantly increase the computational complexity of model training, but there has been significant question as to whether the additional information they provide yields a better agent performance (Genders & Razavi, 2018). Regardless of representation, any real-world

application of this technique would require the chosen actions to be translated into signal state changes (Day, et al., 2014), usually in the form of the standards of the National Transportation Communications for Intelligent Transportation System Protocol (NTCIP) (National Transportation Communications for Intelligent Transportation System Protocol, n.d.).

Generally speaking, actions available to agents are typically represented as the discrete choice of the turning movements to serve at any given time. This may also be discretized as the decision to keep the signal phasing the same or transition to serving a different set of movements (Haydari & Yilmaz, 2020). This approach has been applied in the case of a digital twin-backed state representation, to significant success (Dasgupta, Rahman, Lidbe, Lu, & Jones, 2021). There are, alternately, approaches that consider a continuous action space which determines the amount of time to be applied to the next phase (Haydari & Yilmaz, 2020).

Perhaps the most critical component in the reinforcement learning paradigm is, well, the critic, also known as the reward function. Two of the most common metrics used in the literature for optimizing signal timings are delay and queue length (Haydari & Yilmaz, 2020), while other metrics of consideration include backpressure (Gregoire, Qian, Frazzoli, de La Fortelle, & Wongpiromsarn, 2015), as well as network emissions (Levin & Boyles, 2017; Kővári, Szőke, Bécsi, Aradi, & Gáspár, 2021). Recent developments have also proposed reward metrics which incorporate metrics of equity (Yan, Zhang, Büscher, & Burgard, 2020). However, the metrics involve the calculation of an equity factor that is neither intuitive nor guaranteeing a unique value, nor does it appear to be backed by prior literature. It remains to be seen what impacts would result from training using a reward function that involves more common equity measurements such as the Gini-style dispersion of delay we described previously.

Reinforcement learning is a common topic in signal optimization, with a bevy of research in recent literature. A good overview of current research from a more transportation-oriented viewpoint is provided by Noaeen et al. (Noaeen, et al., 2022). Wei et al. and Yau et al. provide a more computing-focused perspective (Yau, Qadir, Khoo, Ling, & Komisarczuk, 2017; Wei, Zheng, Gayah, & Li, 2021). A noticeable gap is present in most modern work – that most models either do not consider responsive traveler route choice at all (Harth, Langer, & Bogenberger, 2021), do so indirectly by using function approximation (La & Bhatnagar, 2010; Alegre, Ziemke, & Bazzan, 2021) or present a route choice model that is, in and of itself, an independent learning agent (Tavares & Bazzan, 2012; Chu, Wang, Codecà, & Li, 2019; Shi, Gu, Yang, Li, & Chu, 2023). While the first two approaches fail to consider directly the behavioral response of travelers, the latter one attempts to build a model to approximate route choice behavior in

a manner that depends on a computationally expensive model that must calculate each individual traveler's route choice. We detail in Chapter 4 an approach based on the principles of static traffic assignment and pressure-based signal optimization to attain the response-modeling benefits of these learning agent-based approaches while utilizing a far simpler and more efficient model.

Optimizing Route Choice and Traffic Signal Timings Cooperatively

The approaches above, however, ignore the efforts made in route choice modeling as discussed before. The fundamental assumption that, in the long run, travelers' route choices adjust as they learn, is validated by Smith and van Vuren (Smith & van Vuren, 1993) and underpins a tremendous amount of work in this area, including an equilibrium approach by Wei et al. that models learning agents seeking to reach the Nash equilibrium (Wei, Wu, Wu, Shen, & Li, 2021).

We also wish to discuss the bevy of work which models travelers' route choices directly. Several efforts have been undertaken to develop and solve multi-level optimization problems which balance both travelers' route choices and signal timings (Gartner & Al-Malik, 1996; Sun, Benekohal, & Waller, 2006; Hajbabaie & Benekohal, 2015). Smith has long been a leader in this area, developing the P0 control policy (Smith, 1979) and showing its good performance, stability, and usefulness for maximizing capacity under most circumstances (Smith, Liu, & Mounce, 2015; Smith & Liu, 2015; Smith, Viti, Huang, & Mounce, 2023).

While the iterative equilibrium-based framework Smith et al. propose is quite attractive, the choice of pressure function used to represent the state of travel demand on a roadway is of note. Wang et al. critique P0 as being difficult to observe, substituting their own pressure function which incorporates a "hybrid" dynamical system and extrapolating the control policy to general networks with multiple OD pairs, showing improvements over P0 (Wang, Li, Yue, & Mao, 2019). These two pressure functions will be discussed in more detail in Chapter 4, in which they act as a baseline against which we can compare performance of the novel pressure functions we contribute.

2. Developing a Free-and-Open-Source Travel Demand Modeling Implementation

In this chapter, we discuss our software package *wrap* (SPARTA Lab, 2017). Built originally as a component of a course project, *wrap* was devised as an implementation of Robert Dial's static traffic assignment method Algorithm B (Dial, 2006). Initial design choices were to make the project free-and-open-source software[**] to better serve the model development community, and Java was chosen as the programming language due to its cross-compatibility capabilities as well as strong typing enforcement to encourage good object-oriented programming practices.

In pursuing these programming best practices, we developed a detailed framework and a well-documented implementation for static route choice that solves the traffic assignment problem efficiently, both in computation time and storage space. The target audience of *wrap,* though, is the community of travel demand modelers writ large, and as such, what started as a simple route choice course project began to experience scope creep. Given that route choice is but one component of the UTMS, the initially scoped project was of little use without other modeling components.

To complement the route choice implementation, the *wrap* framework was expanded to include interfaces for performing trip generation, spatial distribution, and mode choice, with data structures clearly defining and efficiently storing the demands modeled by implementations of the framework, as well as structured parameters and input data on which they typically depend, such as travel impedance functions and inter-zonal travel cost matrices. In pursuing the research detailed in Chapter 3, we further extended this to include additional components not universally present in all travel demand models – those of trip balancing, multiple population segments, multiple values of time, multiple trip purposes, and time-of-day distribution. Additionally, to incorporate traffic signal timing impacts on roadway networks in Chapter 4, *wrap* was outfitted with basic ring-barrier signal control support and a pressure-based signal timing optimization technique.

The *wrap* software consists of multiple packages, each containing modules dedicated to a specific purpose, some more general than others. The root package, "`edu.utexas.wrap`" contains the basic scripts which perform individual model calculations at a high level. This package also contains some basic

[**] *wrap* is licensed under the GNU Public License, version 3.0.

enumerations on which models may operate, specifically the trip purposes and time periods which may distinguish different models.

The sub-packages underneath the root each group relevant modules together in a cohesive package, and most are covered in greater detail later in this chapter. There are two exceptions to this: the "`test`" package and the "`util`" package. The former contains unit and integration tests which can be run to confirm proper implementation of the modules underlying complex model computations.

The latter is a collection of modules whose organizational relevance may bridge multiple areas, or none at all. Many of these modules generate components used in later model calculations but do not perform calculations themselves, instead primarily performing input-output operations. Other modules are used for proper error handling, and still others are useful for performance measurement of a model's output route choice. The organization of the "`util`" package is rather loose and subject to further refinement, as it currently represents a "junk drawer" of software modules. An additional package, "`gui`" remains under development and is not considered stable at this time.

Network Representation

Underlying each UTMS model is a network containing the links and nodes in the network, i.e., the roads and intersections. In addition, travel survey zones (TSZs, also known as centroids) are used as special nodes which represent the origins and destinations of travelers' trips. Each of these modules are stored in the "`net`" package. The most important class in this package is the `Graph` module, which acts as the top-level organizational structure for the model of the physical network. The `Graph` class maintains both a forward- and reverse-star representation of the network, as well as a collection of the centroids in the network.

The links in a network may be of several different types, but all are bound by certain requirements as described in the `Link` interface. These require that any link will be able to provide a free-flow traversal time, a capacity, a method to retrieve its travel time under the current flow, and methods to get and modify the amount of flow on the link, as well as others. Different link classes may define different methods of calculating travel time, may restrict flow to only certain vehicle classes, may include tolls, or provide other functionality.

One particularly notable example is centroid connectors, which are artificial links (they do not represent a real-world roadway) that are generally considered as always uncongested and may not be used for

through traffic. Several other variations are possible, and the "net" package includes two tolled links which calculate link travel times using the BPR and conic delay functions.

The TravelSurveyZone class is used for storing demographic and area type data related to each zone which can be used in trip generation to describe the number of trips made to and from each zone. All zones are associated with an entry in the AreaClass enumeration, describing the general type of business in the area. Generally, all trip demand is associated with a zone or pair of zones, as detailed in a later section. These zones may be associated with an enclosing RegionalAnalysisArea object that may provide more demographic information.

Market Segmentation

As trip behavior varies widely depending on demographics, it is natural to aim to segment populations based on relevant characteristics when performing model calculations. To do so, we provide the ability to segment markets based on various attributes. The "marketsegmentation" package defines a general MarketSegment interface which provides only one requirement - that each market segment provide a method which is used to get the segment's attribute data.

The purpose of these segments is to provide a mechanism to describe to trip generators how much value should be put on specific zones by providing some sort of attribute data, without having to define different behaviors for each type of market segmentation. By performing this high-level abstraction, we can use the same methods to generate trips for market segments that differentiate based on households in an income group as we do for market segments that differentiate based on employment in retail industries.

Several types of market segment exist in the "marketsegmentation" package, including (but not limited to) segments for household size, income group, and number of vehicles. In addition, combination market segments exist which can combine many of these, for example the IncomeGroupWorkerVehicleSegment class, which segments markets based on the number of households in a specific income group with a specific number of vehicles and workers.

Demand

Each step of the four-step UTMS communicates with the adjacent steps through some representation of travel demand. For example, trip generation (detailed in the next section) generates production-attraction (PA) maps which are relayed to the trip distribution step. The different representations of demand,

namely PA maps, aggregate PA matrices, mode-specific PA matrices, and OD matrices, are all defined broadly as interfaces in the "demand" package.

This package also includes several useful utilities for manipulating the data encapsulated in these representations, including passthrough modules which multiply each entry by a fixed value, and a module for transposing a matrix's entries. Each of these depends on a concrete implementation of the underlying data structures, which are contained in the "demand.containers" package.

The demand container implementations each implement the necessities of their interfaces while maintaining unique behaviors that make them more useful in certain circumstances. For example, DemandMaps, the basic level of representing demand at each zone in the network, may be of the EmptyDemandMap class, in which case every zone's demand is zero; or they may be backed by a HashMap, which is slow and requires more storage space but is flexible to changes in network size; or they may be FixedSizeDemandMaps which are fast and simple but are rigidly defined and less useful in certain network design problems.

More details can be found in the documentation for each class. Additionally, experimental modules are under construction which are backed by a database rather than in-memory storage. These modules are found in the "demand.containers.database" package.

Matrix Manipulation

At multiple points in the development of a model, it may be desirable to split a given map or matrix into multiple different instances or to otherwise multiply a map or matrix's entries by some factor. In the "demand" package, classes can be found which act as fixed multiplier passthrough maps and matrices. These maps and matrices take as input a predecessor map or matrix and a function which should be applied thereto.

These passthrough matrices and maps evaluate this function lazily, i.e., they do not perform any calculations until the end result is necessary. In this case, the process of performing these calculations can be sped up dramatically if the results are not to be needed at the same time, or not for a while. Further development of these classes can increase the speed with which the function evaluation procedures can be performed, but they at present act as reliable and quick solutions to a common task in model construction.

In some phases of building a UTMS model, it may be desirable to combine multiple maps or matrices together along commonalities such as income group. To help with this, collector classes are provided in the "`util`" package which act as a terminal operation for streams of maps or matrices. These essentially create a new child matrix or map associated with multiple parents, and the given demand in the child matrix or map is simply the sum of the parent matrices' or maps' demands. These allow for quick combination of multiple demand containers, as no summation is done at collection time, but is rather delayed until needed. This works well with the stream-based operation of multiple modules, supporting parallel computation for many steps in UTMS models.

Trip Generation

Trip generation exists at a base level to develop demand maps, i.e., one-to-one mappings from a zone to a value representing the productions and another representing the attractions from that zone. Each trip purpose and market segment generates trips differently, so organizing the trip generation step is complex. Four primary types of trip generation exist in our framework - basic, area-specific, fixed-proportion, and rate-proportion, which we will now detail.

The basic and area-specific trip generator modules are so-called "primary generators", in that their trip generation does not depend on trips generated before and instead depends only on the characteristics of the zones in question. The basic trip generator depends on a market segment-specific rate and the market segment's underlying attributes alone. In addition, a set of area type-specific rates can be provided to scale up or down a specific area type's zones' trip vector uniformly across all market segments. The `AreaSpecificTripGenerator` behaves similarly, but with the added distinction that the area type rates vary across each market segment. This simplifies the organization of trip generation by not requiring multiple trip generators for the same trip purpose.

The fixed- and rate-proportion generators are so-called "secondary generators" in that they do depend on trips generated from a different trip generator. The fixed-proportion generator duplicates a demand map and multiplies it by a fixed proportion (hence the name) to describe secondary trips generated by a primary map. The rate-proportion generator combines trips across multiple segments, determines the shares of each zone's trips per market segment, and uses the ratio of the primary and secondary generators' rates to determine a multiplier for each zone. For more details, see the documentation for these modules.

Trip Balancing

Following trip generation, which describes independently where trips begin and end, the trips must be distributed in a way that links the two together. Trip distribution, to be discussed in a later section, links these two, but depends on the total number of attractions in a PA map equaling the total productions. The "`balancing`" package contains an interface for how a balancer module should interact with PA maps, and three module implementations that balance trips differently.

One method of balancing trips across demand maps is to simply copy the productions from each zone to the attractions from each zone, thus creating balance by ensuring every zone's productions and attractions are equal. This simple mechanism is handled in the `Prod2AttrCopyBalancer` module and could easily be reversed through development of an `Attr2ProdCopyBalancer`, if this becomes desirable.

A more complex method of balancing is found in the `Prod2AttrProportionalBalancer` and the analogous `Attr2ProdProportionalBalancer`. These modules calculate the total productions and attractions in a PA map and adjust each zone's productions (in the case of the `Prod2Attr`) or attractions (the `Attr2Prod`) by the ratio of total productions to attractions in order to guarantee the two totals are identical. These modules operate across all zones in the network but can easily be modified to only balance trips across the zones' respective regional analysis areas.

Trip Distribution

Trip distribution, the linking of trip productions to trip attractions, takes place in the aptly-named "`distribution`" package. This package includes the abstract class `TripDistributor`, which defines that any trip distribution module must accept a PA map and distribute it to an aggregate PA matrix[††]. Additionally, the package includes the `FrictionFactorMap` interface, which defines that any friction factor map implementation must be capable of returning a factor for any two zones.

The most common method of trip distribution utilizes a gravity model. This model iteratively balances factors associated with the productions and attractions for each TSZ, then uses the product of a zone

[††] Note that this behavior is consistent with trip interchange models, i.e., models that perform distribution prior to mode choice. Trip end models, which perform mode choice before distribution, are not dissimilar in functionality, but will require further development in the *wrap* repository to accommodate.

pair's productions, attractions, factors, and the friction factor between the two to determine the number of trips between the two zones. Further details on such a model are not in the scope of this chapter, but the *wrap* repository's "`distribution`" package contains a working implementation of such a model which calculates these values in parallel.

The friction factor map implementation used in the Dallas-Fort Worth Regional Travel Model leverages both a table of travel costs between all zones in the network and a predefined mapping from travel costs to a factor (Model Development Group, NCTCOG Transportation Department, 2007). The `CostBasedFrictionFactorMap` class implements this strategy by interpolating the factors associated with each zone pair's costs on demand, and thus lazily evaluates these computations, saving time.

Mode Choice

Mode choice takes two distinct forms in the UTMS: trip end and trip interchange splitting, with the difference being in whether mode choice is performed before or after trip distribution[‡‡]. As a result, two different interfaces are defined in the "`modechoice`" package in *wrap*: `TripEndSplitter` and `TripInterchangeSplitter`, respectively. Little development has been undertaken in trip end splitting, so we focus here on trip interchange splitting, the more common method. The trip interchange interface specifies that an aggregate PA matrix and market segment be given, which in turn create an output stream of modal PA matrices. This stream allows for parallel computation in later phases.

Multiple methods of performing mode choice exist, and the interface for these allows for most every conceivable method to cooperate with the remainder of the *wrap* framework. For now, only one method of mode choice has been completed: the `FixedProportionSplitter`. This method takes in a map from each mode to its share of trips for a given market segment, and simply multiplies all zones' trips by this fraction. In doing so, the `FixedProportionSplitter` can leverage the speed of the multiplier classes (discussed earlier), while still demonstrating the functionality of the interface specification.

[‡‡] The motivation behind this distinction is that urban areas with lackluster transit systems may have captive riders whose mode choice is more important than their route. This is not yet implemented in *wrap* but is provided as an interface for future extensibility.

Traffic Assignment

Traffic assignment (variously route choice, route assignment, etc.) is far and away the most computationally complex step in any UTMS model. The scope of the *wrap* package was originally solely to handle this complicated problem in a fast and extensible manner, which led to extensions to include other steps in the UTMS. As a result, a large portion of the source code in *wrap* is dedicated to the implementation of Algorithm B, a bush-based solver of the traffic assignment problem. The implementation in the *wrap* framework is found in the "`assignment`" package and is capable of performing calculations even on the largest of networks known today.

To improve scalability of the traffic assignment implementation in *wrap*, several design choices were made that contribute to the efficiency of *wrap*'s Algorithm B implementation. First, a hefty effort was made to parallelize as much as possible of the optimization problem. The interdependency of flow shifting on individual bushes restricts the ability to shift flow in parallel. However, performance evaluation on individual bushes, as well as identification of links which can be considered candidates for addition or removal from the bush can be parallelized relatively easily. Additionally, to mitigate the impact of loading and unloading a bush into memory, the *wrap* implementation of Algorithm B departs from the original design by equilibrating each bush multiple times per overarching iteration. An illustration of the modified Algorithm B is shown in Figure 2.

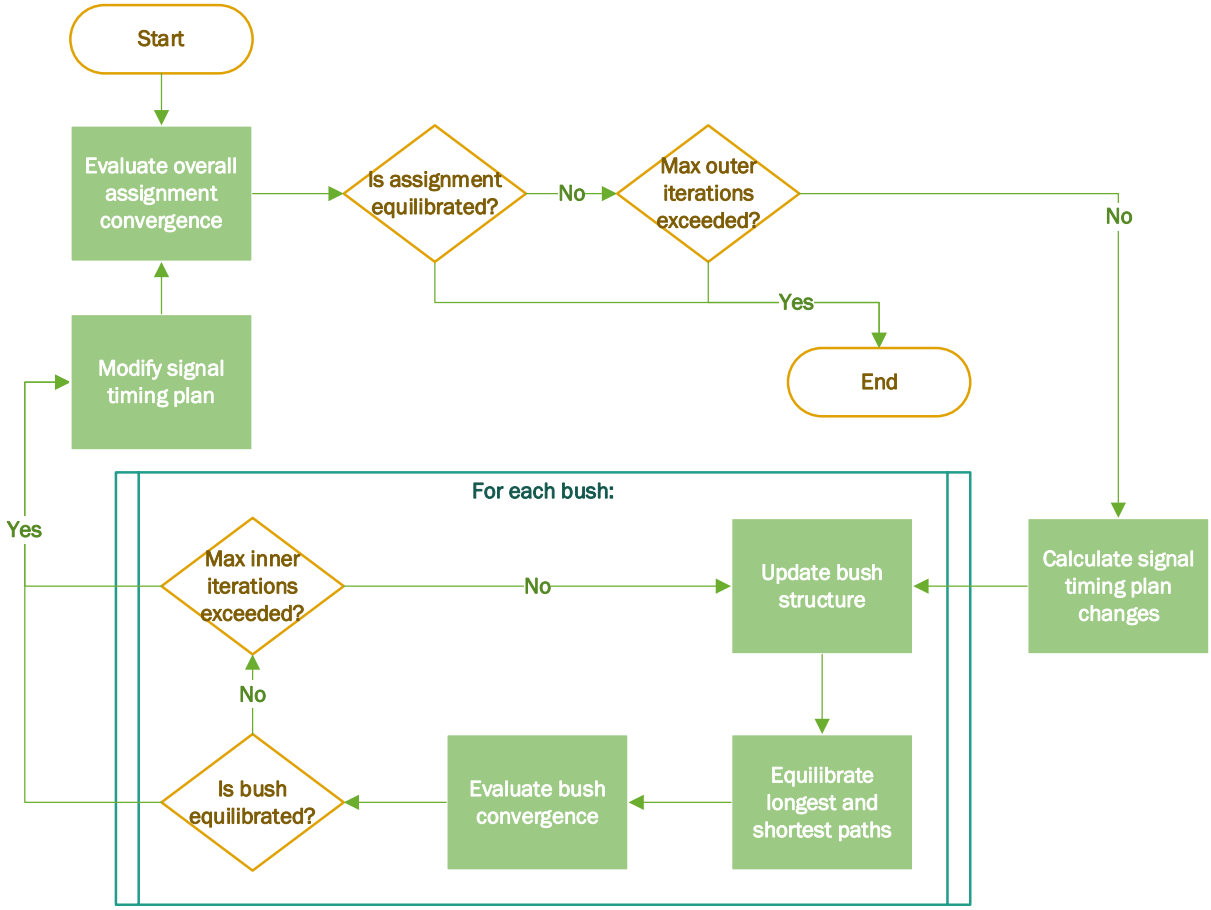

Figure 2: Illustration of modified Algorithm B implementation

File Input and Output

Multiple packages in the *wrap* framework contain relevant modules for file input and output, especially in regard to reading in model inputs such as network definitions, cost skims, generation rates, and more. Outputting results from various modules is also important and is implemented for most demand containers. In addition, because *wrap* is so extensible, groundwork has been laid for implementing database-backed storage instead of file-based storage, which may improve both speed of computations and resiliency of data storage.

This groundwork comes in the form of a defined interface for model inputs, dubbed simply `ModelInput`. While each model is unique in the actions performed beyond the traditional four steps of the UTMS, the interface defined for a *wrap* model will provide methods for one of the most complicated scenarios, and the simpler ones may use only a fraction of the defined functions.

Given the sporadic development timeline for the file IO modules, it should come as no surprise that several modules vary in their methods for performing IO operations. As a result, further development is possible to improve the timely performance of these modules. This is particularly desirable for the output of OD matrices, which require large computation time and disk usage for file output. Nonetheless, functional implementations exist for most every IO requirement in a UTMS model.

Comparing Performance for a Large-Scale Model

In developing and testing the *wrap* framework, several advantages became apparent when comparing a pre-existing implementation of the Dallas-Fort Worth Regional Travel Model to the newly implemented model. The biggest performance improvement comes from the implementation of parallelism and lazy evaluation in the *wrap* modules. Whereas the earlier model ran individual steps one after another in TransCAD, many of these steps can be run simultaneously on a multi-core processor, dramatically cutting down on computation time.

In addition, a large savings is achieved by reducing the amount of file IO performed. Consider that the prior model would read inputs to each step of the model, perform calculations, and write the outputs to files, then repeat for the next step. The new model improves upon this by only performing file IO when absolutely necessary, delaying intermediate calculations until their results are necessary for future computations. This is to say that files are only written once the model is completely built and calculated, rather than performing intermediate writes as each computation is performed, then requiring redundant reading of these files in the next step of the model.

Because several steps are evaluated lazily, there is a negligible amount of computation time associated with these steps (for example, primary production-attraction balancing takes under 50 milliseconds). By contrast, these steps would take a considerable amount of time to read inputs, perform calculations, and write outputs in the prior model. The greatest success of this model is that the computations involved in the full model are reduced from taking hours to taking minutes.

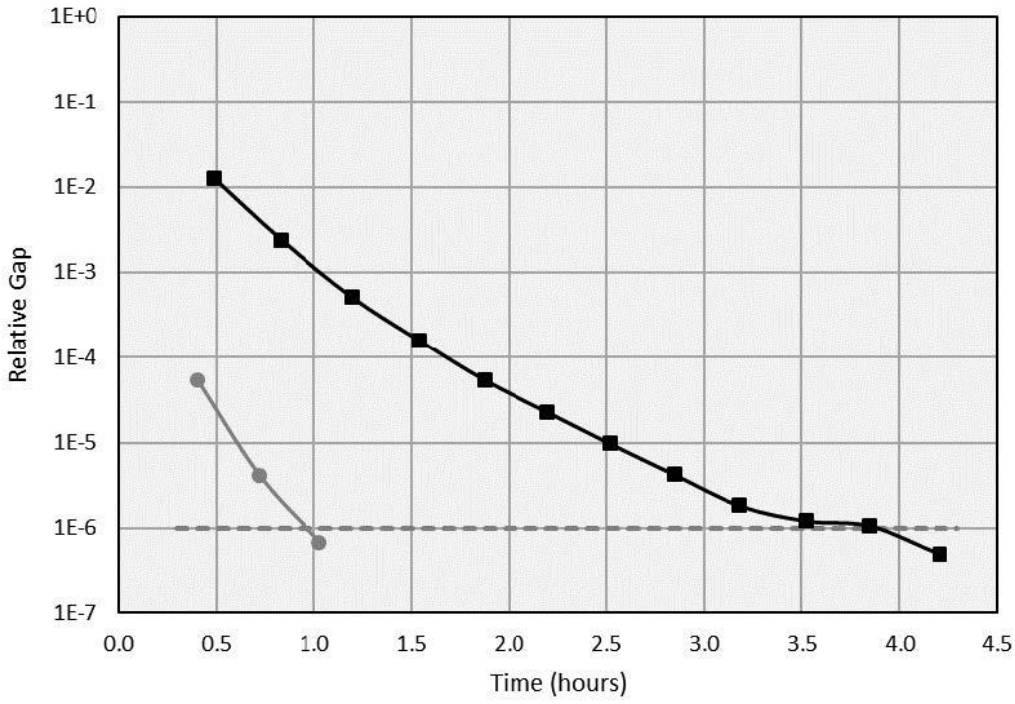

Figure 3: Algorithm B performance for initial (black) and warm-started (gray) OD matrices

Warm-Starting

One other major accomplishment of this new model is the improvements over TransCAD's assignment model. Using the wrap Algorithm B implementation, the new model is not only able to handle multiple value-of-time classes in performing general cost traffic assignment but is able to provide a "warm-start" solution which dramatically reduces calculation time for assignment. To do this, at the conclusion of each iteration, the bush structure is stored for later retrieval. In implementation, this involves determining, at each node in the bush, the fraction of flow into that node from each inbound link. While this does not provide an exact description of the state of a bush and its route choices in that it lacks a direct representation of total flow, this is easily calculated using the demand on the bush. Additionally, this allows for reusing bush structures even though the underlying demand may vary significantly. In this way, we provide a simplified representation of the bush that can be stored efficiently, minimizing the quantity of data to be stored and removing redundancies, and which can easily accommodate changes in demand such as is necessitated in experiments we present in Chapter 3. An added benefit of this simplified structure is to be able to incorporate a representation of the bush's topological order, thus maintaining an important graph descriptor for ready use.

By providing a close approximation of what the correct solution should be, Algorithm B can solve the traffic assignment problem dramatically faster than solving the problem from scratch. Figure 3 illustrates this impact, where the relative gap for two different scenarios is shown as a function of time. The line which begins from a higher gap illustrates the scenario which was loaded from scratch. This solution was then stored, then used as a warm start for the second scenario, which varied one of the input OD matrices and which was solved significantly faster.

The following two chapters present work which lean heavily on *wrap*'s operational efficiency. By leveraging the warm-starting capabilities and the bush structure of our implementation, we can quickly re-evaluate the most computationally complex facet of travel demand models (traffic assignment). This in turn is utilized to accelerate the process of gradient approximation used in Chapter 3 to improve estimates of trip generation rates. Additionally, the improvements we detail to Algorithm B's bush equilibration technique provide an efficient means to iteratively balance the competing ideals of travelers seeking to minimizing their individual travel costs and transportation system managers seeking to maximize the performance of the collective traffic signal system. The pressure functions we present in Chapter 4 vary in their speed of convergence to equilibrium, so the efficiency of *wrap* is of great benefit in these experiments.

3. Calibrating Travel Demand Models Using Field Data

Origin-destination matrix estimation (ODME) is widely used to improve transportation network models, and more broadly in transportation planning. By combining planning data from surveys and observed traffic data from the field, estimates are developed for the demand between OD pairs in a network. There are many methods for improving these estimates based on observable data, such as link flows.

However, in the network modeling community, most literature on this topic adjusts the OD matrix directly and feeds it into traffic assignment to better match observed link flows. This process has dimensionality issues – the size of the OD matrix is quadratic in network size, while the number of links is linear – and many proposed techniques fail to consider the earlier steps in the UTMS. This can lead to OD matrix estimates that do not correspond to plausible trip-making behavior. These issues contribute to overfitting, matching observed link counts without actually replicating the underlying behavior, and therefore producing bad models.

To remedy these issues, we propose to estimate OD matrices by tuning the parameters of trip generation, the earliest phase of the UTMS. This maintains the behavioral explanation for the OD matrix and dramatically reduces dimensionality. We describe herein a method for tuning these parameters to output a calibrated OD matrix minimizing the difference between assigned (predicted) and true (observed) link flows, as a proxy for reducing error between estimated and true (but unobservable) trip generation rates.

We test our method using two reference scenarios. Our experiments show that, given a noisy but reasonable estimate of trip generation rates, our search algorithm can approach the true rates using only observed network flows, using a test for overfitting. We also provide suggestions on how to adjust the demand model structure to prioritize performance on certain trip generation rates.

The remainder of this chapter is organized as follows: we detail the proposed novel optimization technique for ODME as well as the experimental framework, and then present results of initial tests. We conclude with a synopsis of our findings and a plan for future research on this topic.

Method

Let $\vec{t}$ be the vector of trip generation model parameters. These parameters cannot be directly observed, but must be calibrated based on observable metrics, such as total vehicle-miles traveled, trip length distributions, or link flows. In this chapter, we focus on the latter, assuming that a vector of link flows $\vec{f}^*$ is available from field observations.

Suppose we have a UTMS model F that describes the effect of trip generation parameters $\vec{t}$ to ultimately produce the link flows $\vec{f} = F(\vec{t})$. Given an initial estimate of the true trip generation parameters $\vec{t}_0$ from surveys or other data sources, our objective is to produce an improved estimate of the true parameters $\vec{t}^*$ using F and the observed link flows $\vec{f}^*$; that is, we seek $\vec{t}^*$ such that $F(\vec{t}^*) = \vec{f}^*$. This process is easier if $\vec{t}$ is of comparable dimension to $\vec{f}^*$ or smaller; using the final OD matrix in such a mapping introduces significant risk of overfitting. As described below, we apply a quasi-Newton local search to minimize the error between our best-guess parameters $\vec{t}_i$ and their true values $\vec{t}^*$.

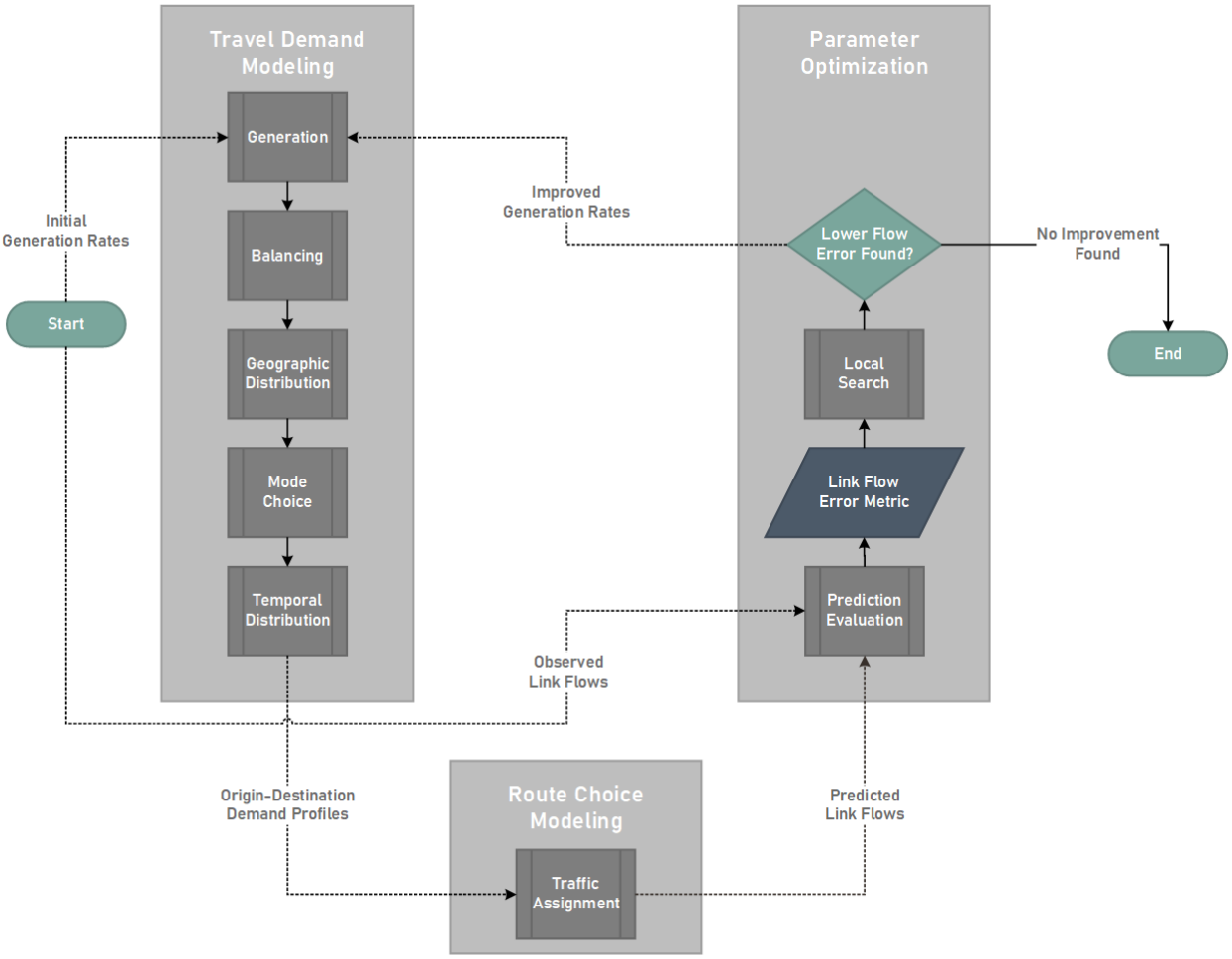

Figure 4: Model progression flowchart

Practically, this process refines an estimate of the true trip generation parameters found using a travel demand survey by incorporating observed "true" link flows into the estimation process. In this section, we first describe the particulars of our travel demand model before detailing the local search strategy we implement to accomplish this objective. An overview of our process is shown in Figure 4.

Travel Demand Model

In developing this strategy for ODME, we expanded on the feature set of *wrap*, our urban transportation modeling software, including a variety of features that are present in many simple models, as well as some extensions. This included implementing at least one form of each of the four traditional steps of the UTMS. For generality, and consistency with complex UTMS models in practice today, we developed an extended structure defining markets and trip purposes to address heterogeneity in travel behavior. By developing distinct models for each market, a variety of travel-making behaviors and their interaction effects can be modeled simultaneously. We are also able to address chained trips or tours in a limited way using the multiple-purpose approach.

In addition to interdependency of trips, markets' trip purposes define the full OD demand profile for all trips made by a market for the given purpose. Practically, this means that purposes must define the procedure and parameters associated with the traditional four steps of the UTMS. We also implement two extra steps in addition to the traditional four: trip balancing, which is the process of manipulating generated trips to ensure the number of trips produced equals the number of trips attracted across the network; and trip time distribution, as described in the literature review.

Trip generation, the step we focus on most in our experiments, involves two vector calculations per purpose, one to determine the number of productions in each zone, and one for the number of attractions. Since each market may feature one or more demographics with distinct behaviors for a given trip purpose (e.g., households with/without dogs having different behaviors for trips made for pet healthcare), multiple production and attraction vectors for a given purpose could be calculated separately and combined. We denote the set of demographics and their corresponding parameters as $\mathcal{G}$.

In our experiments, each trip purpose uses a basic scalar-vector multiplication to generate trips. Each trip purpose featured two demographic vectors d_P for trip production and two more d_A for attraction, as well as corresponding scalar trip generation parameters t_P and t_A (it is these parameters which we will ultimately adjust in calibration). The total productions p_z and the total attractions a_z generated at TAZ z is then given, respectively, by the following two equations:

$$p_z = \sum_{(d_P, t_P) \in \mathcal{G}} d_P(z) \cdot t_P \quad \{6\}$$

$$a_z = \sum_{(d_A, t_A) \in \mathcal{G}} d_A(z) \cdot t_A \quad \{7\}$$

Balancing, traditionally incorporated in the generation step of the UTMS but separated for clarity in our experiments, is defined in order to guarantee an equal number of total productions and attractions. We divide trip purposes into two types: production-proportional and attraction-proportional. For production-proportional trip purposes, a multiplier is calculated as the ratio of total attractions to total productions, and the number of productions per zone is multiplied by this value. The reverse is true for attraction-proportional trip purposes: the multiplier is the ratio of productions to attractions, and each zone's attraction value is multiplied by the value instead. In both cases, the total number of trips generated across all zones is calculated across all demographics. The result is a vector of productions and a vector of attractions, with equal sum.

Geographic distribution of trips is completed using a standard gravity model which iteratively derives balancing factors a and b for each zone, according to an impedance function corresponding to the cost of traveling between any two zones. There are multiple solutions to this standard gravity model, so to avoid complications in our experiments, the impedance function is held constant across all trip purposes in our experiments. This distribution mechanism then transforms the two vectors from the balancing step into a matrix of productions and attractions between any zone pair.

Mode choice splits the aggregate production-attraction matrix that results from geographic distribution into a set of matrices for each applicable mode. For our implementation, we divide all trips into single-occupancy and high-occupancy modes according to a fixed percentage consistent across all trips made by each market for each purpose.

The mode choice step must also consider that high-occupancy vehicle trips account for more than one person's trip. Therefore, the resulting matrix values are divided by the mode's occupancy to convert the person-trips from a production-attraction matrix into the vehicle-trips found in an origin-destination matrix.

Temporal trip distribution further splits trips into time-of-day segments. Trip chaining complicates UTMS models, as some trips are return trips (e.g., a traveler goes to and from a restaurant for lunch), while others are one-way (e.g., a trip to the grocery store after work before continuing home). To model this behavior, each trip purpose defines the time period in which trips' departing and returning legs (if applicable) are made. This step transforms an OD matrix into an OD profile with the added time dimension.

For our experiments, we compare our model outputs against "true" link flow values for a single time period. Therefore, we make the simplifying assumption of providing a fixed multiplier across all zone pairs for each trip purpose. The practical impact of this constraint is that people of a given market taking trips for a common purpose are modeled as taking trips in the same time period consistently across geography. The temporal distribution step concludes with the final collection of OD profiles that are used in network route choice. For our experiments, we focus only on a single static time period's OD matrix.

Route choice is performed on OD matrices that result from earlier steps of the UTMS. Our model uses the STA problem framework to perform route choice, minimizing the Beckmann function shown in Equation {1}. The resulting link flow pattern is a user equilibrium $\vec{f}^*$ that is known to be unique if the cost functions are increasing.

We solve the traffic assignment problem using our implementation of Algorithm B, extended to include multiple modes and multiple values of time. The former extension allows for restricting certain links which may only be used by high-occupancy vehicles, and the latter allows for different cost valuations for routes based on the trade-off of a link's toll and travel time.

The network traffic assignment problem thus completes the mapping F, which begins with trip generation parameters $\vec{t}$ and ends with link flows $\vec{f}$ corresponding to user equilibrium. The objective of our framework is to use the observed values of $\vec{f}$ to improve estimates of $\vec{t}$ without overfitting.

Parameter Optimization

Building on the demand model described above, we now describe our framework for improving initial estimates of the trip generation parameters using local search.

Prediction Evaluation

We will first explain how we evaluate the error between our predicted link flows and the true values. We define our problem we seek to minimize as:

$$\epsilon_{RMSE}(\vec{f}) = \sqrt{\frac{\sum_i (\vec{f}_i - \vec{f}_i^*)^2}{n_f}} \qquad \{8\}$$

where n_f is the number of entries in $\vec{f} = F(\vec{t})$ and $\vec{f}^*$, i.e., the number of links in the network. A similar formulation, the parameter RMSE, reflects the error in model parameters, with the f terms reflecting

individual model parameters and the n_f term reflecting the total number of parameters in the calibration process.

Local Search

Provided an ϵ function, our local search successively modifies an initial parameter set $\vec{t}$ to produce progressively better flow estimates, in hopes that this will also produce better estimates of $\vec{t}^*$. We will content ourselves with a local optimum. Since our travel demand model is continuous in the trip generation parameters, we use Newton's method to find this local optimum.

However, the model's complexity in practice restrains us from exactly calculating its gradient and Hessian matrix. Therefore, we numerically approximate the gradient using finite differences, which requires us to run the demand model once per input dimension, and settle for an estimate of the Hessian matrix, restricting ourselves to *quasi*-Newton methods.

This differs from prior ODME research that treats the entire OD matrix as its inputs; due to the OD matrix's size, finite differences scales poorly, necessitating the use of algorithms less precise than quasi-Newton methods like W-SPSA to more slowly approach an optimal OD matrix (Antoniou, Azevedo, Lu, Pereira, & Ben-Akiva, 2015). By using a travel demand model, we dramatically reduce the dimensionality of our input space, allowing us to afford methods that use a finite difference gradient to find a local optimum.

To avoid calculating a Hessian in addition to the gradient, we use L-BFGS-B, a popular quasi-Newton method that approximates the Hessian using the gradient while enforcing boundary conditions (in our case, non-negativity) (Byrd, Lu, Nocedal, & Zhu, 1995).

Recall that $\epsilon\left(F\left(\vec{t}\right)\right)$ is our link flow error objective function. Let $\vec{t}_i$ be the estimated trip generation rate parameters and $\widehat{H}_i$ the estimate of the Hessian of the objective function at step i. Furthermore, let n_t be the length of $\vec{t}_i$, the number of trip generation parameters, and $\Delta_{ftol}, \Delta_{gtol} > 0$ be two scalar conditions for terminating the local search.

Initialize $\widehat{H}_0$ as an identity matrix $\mathbb{I}$ of size $n_t \times n_t$ and i as zero. Given an initial guess $\vec{t}_0$ and its corresponding objective function value x_0 and finite difference gradient $\vec{g}_0$, each L-BGFS-B search step proceeds as follows:

1. Determine if any entries of $\vec{t}_i$ are at their boundary, zero. If so, fix their values at zero and do not include them in the step calculations, effectively reducing the size of our input parameters. Remove their associated dimensions from $\widehat{H}_i$ and other values to match.
2. Find the descent direction $\vec{p} = -\widehat{H}_i^{-1}\vec{g}$.
3. Find an optimal step size $\alpha = \arg\min_{\alpha} \epsilon\left(\vec{t}_i + \alpha\vec{p}_i\right)$ using backtracking line search (for more details, see (Byrd, Lu, Nocedal, & Zhu, 1995)). Set our new estimate $\vec{t}_{i+1} = \vec{t}_i + \alpha\vec{p}_i$.
4. Find the error $x_{i+1} = \epsilon\left(F\left(\vec{t}_{i+1}\right)\right)$. Terminate the search here if the normalized improvement in the objective value $|x_{i+1} - x_i| < \Delta_{ftol}$.
5. Find the gradient at $\vec{t}_{i+1}, \vec{g}_{i+1} = \nabla\epsilon(\vec{t}_{i+1})$ using finite differences. Terminate the search here if every entry in $\vec{g}_{i+1} < \Delta_{gtol}$.
6. Let $\vec{s} = \alpha\vec{p}$ and $\vec{y} = g_{i+1} - g_i$. Update our Hessian estimate $\widehat{H}_{i+1} = U\widehat{H}_i U + \frac{\vec{s}\vec{s}^{\mathsf{T}}}{\vec{y}^{\mathsf{T}}\vec{s}}$, where $U = \mathbb{I} - \frac{\vec{y}\vec{s}^{\mathsf{T}}}{\vec{y}^{\mathsf{T}}\vec{s}}$ and $\mathbb{I}$ is the identity matrix.
7. Increment i and repeat.

Experimental Framework

To confirm the performance of the proposed ODME technique, we developed two scenarios based on networks from the Transportation Test Networks repository (Transportation Networks for Research Core Team, 2007), with light modifications to link tolls for experimentation. The network statistics for these scenarios are provided in Table 1.

Table 1: ODME scenario network details

Scenario	Zones	Links	Nodes
Sioux Falls	24	76	24
Chicago-Sketch	387	2,590	933

Each scenario models a single market with two trip purposes, each with two demographic vectors for generating productions and two others generating attractions. The purposes define the parameters of the individual UTMS model phases. Table 2 shows the properties of the trip purposes used in each scenario.

Table 2: Trip purpose UTMS properties for each ODME scenario

Purpose ID	Balancing Proportionality	Mode Choice		Temporal Distribution		Value of Time
		Single Occupancy	High Occupancy	Departure	Arrival	
Sioux Falls						
i	Production	88%	12%	0.70	0.44	0.69
ii	Production	85%	15%	0.65	0.35	0.45
Chicago Sketch						
iii	Attraction	77%	23%	0.58	0.32	0.678
iv	Attraction	78%	22%	0.70	0.25	0.54

Each trip purpose includes two demographics generating productions and two for attractions, necessitating a total of four rate parameters per purpose. Across each scenario's two trip purposes, this gives a total of eight input parameters per scenario. Notation-wise, we identify these parameters by (1) their purpose ID, (2) whether this is a production or attraction rate, and (3) an index corresponding to the demographic vector. For example, $ivP1$ represents the first production parameter for the second Chicago Sketch trip purpose (ID iv).

For these scenarios, there is no practical distinction between departure rates and arrival rates, as they are consistent across all zones, so the sum of the two values is more relevant than their individual values. For the L-BFGS-B local search, we used the default values for Δ_{ftol} and Δ_{gtol} provided by the SciPy implementation of the algorithm (2.22×10^{-9} and 10^{-5}, respectively). These lenient cutoff values keep each search running at very small model steps, giving us the chance to explore overfitting behavior and devise our own search cutoff criterion.

We conducted blind experiments in which an initial "ground truth" model for the scenario is prepared, fixing a set of input trip generation rate parameters as the true values. The flows output by our UTMS model using those rates are fixed as the true link flows. We then create a noisy set of trip generation rate parameters by drawing from a normal distribution centered at the true value $\bar{t}$ with a standard deviation of $\sigma = \frac{0.2\bar{t}}{1.96}$ for each parameter. Given the true link flows and the noisy input set, the ODME process was applied, and the improved inputs and link flow errors were recorded at each step of local search. Finally, the step-by-step inputs were evaluated against the true input set.

Because of the randomness involved in generating noisy input parameters, we repeated the process ten times per scenario in order to develop a representative batch of samples from which to draw conclusions. All experiments were run on an Intel Xeon 3.60 GHz CPU with 64 GB of RAM.

Results

Our analysis focuses on two main questions: whether it is possible to produce accurate estimates of $\vec{t}^*$ from the above procedure; and whether we can determine which trip generation parameters are easier to fit than others.

The local search algorithm decreases the link flow RMSE at each iteration, as it was designed to do. For Sioux Falls, our results show that, while starting with a mean and median link flow RMSE of 360.42 and 307.51, respectively, the search was able to reduce these by 83.7% and 82.3% to values of 58.69 and 54.32 at search termination. The Chicago Sketch trials began with a mean and median link flow RMSE of 85.08 and 63.97, respectively, and at termination, the local search had reduced this by 48.0% and 33.8% to values of 44.22 and 42.39, respectively.

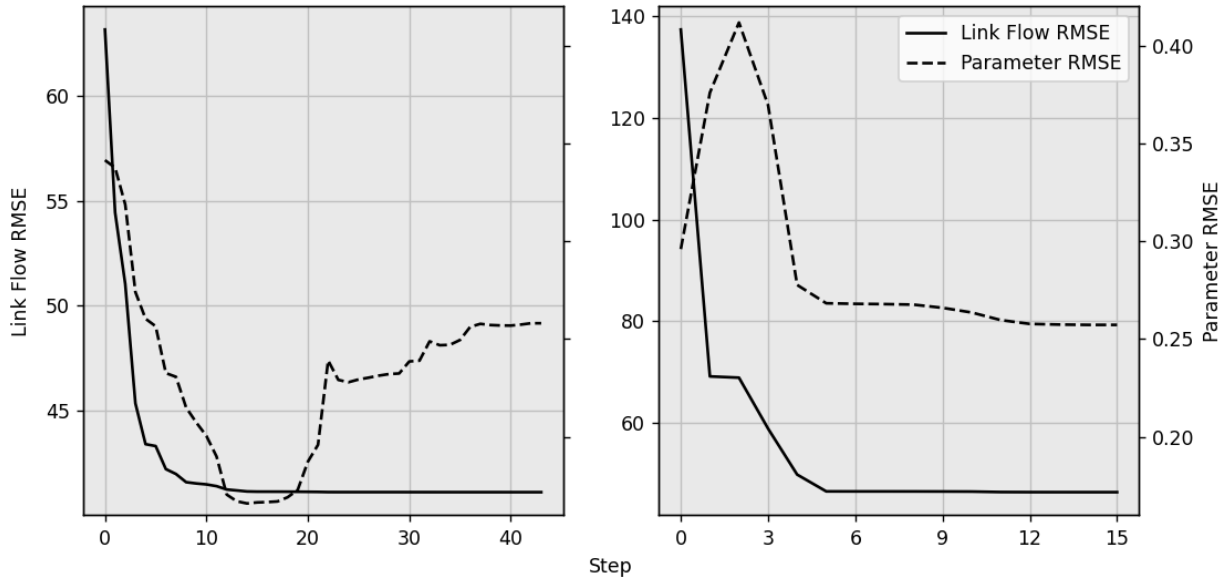

Figure 5: Parameter and Link Flow RMSE at each time step for a selected overfitting (left) and non-overfitting (right) Chicago Sketch trial

The true test of success, however, is whether it was able to produce good estimates of the unobserved parameters $\vec{t}^*$. In some cases, without further correction (described below), our method has a tendency to diverge from some parameters' true values, while attempting to drive link flow RMSE to a local minimum. This is overfitting. In Figure 5, two Chicago Sketch trials are selected to illustrate this behavior. The parameter RMSE increase observed at later iterations in the left subplot indicate that the search algorithm varies parameter values significantly while failing to make substantial improvements in link flow

RMSE, whereas this behavior is absent in the right subplot. Further analysis is presented in the next section.

Parameter Error Analysis

We compared our parameter results against the "true" values withheld in the blind test. The mean and median percent errors at the start and end of a local search are shown in Table 3, with color coding indicating the relative performance of a metric to those of other parameters (lower percent error, shown in greener cells, is better). Table 3 also details the unbalanced trips generated by each parameter, with shading to indicate the largest generators.

The parameters closest to their true values upon termination tend to be those that make up the most trips; the method we propose fits parameters that generate more trips better than those that produce less. Furthermore, there seems to be a better fit for parameters whose trips are left unchanged by trip balancing (see Table 2). Intuitively, this makes sense, as the influence these parameters have on the link flow RMSE is not reduced by interference from other parameters in the balancing step, so the algorithm estimates the more direct relationship of these parameters better.

Table 3: Demand model parameter percent error statistics

		% Error At Search Termination		% Error At Early Termination		% Error At Start	
Parameter ID	Trips	Mean	Median	Mean	Median	Mean	Median
Sioux Falls							
iP1	6,959.82	6.04%	6.46%	5.89%	5.68%	5.17%	4.82%
iP2	9,317.02	8.72%	6.82%	8.05%	6.38%	7.80%	7.09%
iA1	50,449.11	1.46%	1.44%	1.71%	1.58%	4.82%	3.50%
iA2	101,877.16	0.51%	0.53%	1.03%	0.90%	7.05%	4.85%
iiP1	16,912.80	7.61%	6.68%	8.17%	7.27%	8.37%	7.27%
iiP2	9,320.80	7.47%	5.63%	6.03%	3.12%	6.77%	3.98%
iiA1	27,362.40	10.51%	10.47%	4.31%	2.73%	7.99%	5.45%
iiA2	42,026.82	5.54%	4.53%	2.75%	2.42%	4.27%	2.48%
Chicago Sketch							
iiiP1	150,502.33	1.72%	0.55%	3.43%	3.00%	9.75%	10.15%
iiiP2	29,149.52	3.04%	1.37%	10.77%	9.63%	8.74%	6.54%
iiiA1	17,172.37	9.90%	8.30%	9.99%	9.89%	10.08%	8.67%
iiiA2	22,970.61	8.07%	6.67%	12.16%	9.16%	13.75%	12.17%
ivP1	49,432.37	2.24%	0.19%	4.38%	1.54%	8.28%	4.36%
ivP2	229,367.75	0.29%	0.19%	0.58%	0.37%	6.94%	4.64%
ivA1	388,644.23	14.08%	14.00%	7.64%	7.17%	7.72%	7.12%
ivA2	63,052.41	8.56%	8.63%	6.17%	5.97%	6.27%	6.22%

There are exceptions to both trends. Parameters $iiA1$ and $ivA1$ both make up large shares of their scenarios' total trips but show high error when the search terminates. Furthermore, while the productions in the Chicago Sketch scenario have low error, as do the attractions for trip purpose i, the attractions for purpose ii do not fit nearly as well, yielding a percent error of more than 14% on average. We assert that both of these exceptions are symptoms of overfitting.

These overfitting trends are visible in Figure 6 and Figure 7. In these plots, each line corresponds to one experiment run and shows how the link flow and parameter RMSE vary together. As the local search proceeds, the link flow RMSE decreases (hence the reverse horizontal axis on these plots). Initially, the parameter RMSE decreases along with the link flow RMSE. In some cases, reductions in link flow RMSE are accompanied by increases in parameter RMSE, indicating overfitting. In a few cases, the final parameter RMSE was in fact higher than the initial parameter RMSE due to this effect. The next subsection discusses a proposed remedy.

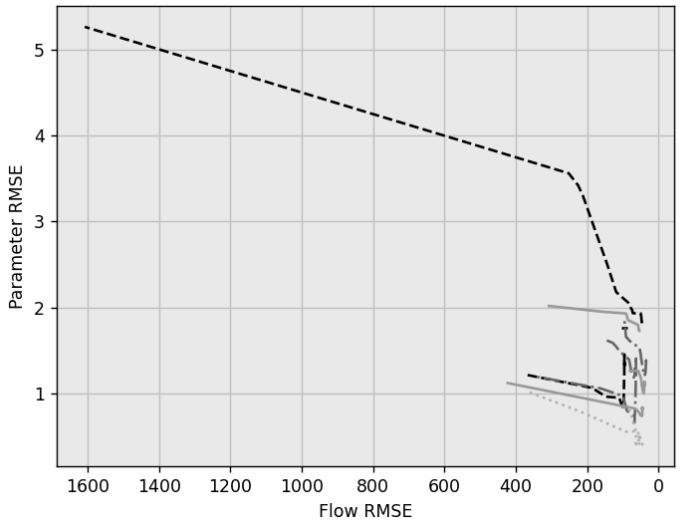

Figure 6: Link flow and parameter RMSE trends for Sioux Falls scenario trials, including all local search steps

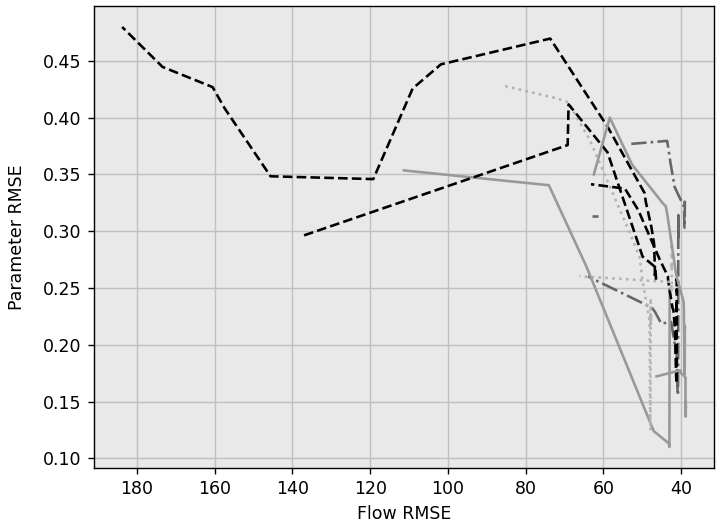

Figure 7: Link flow and parameter RMSE trends for Chicago Sketch scenario trials, including all local search steps

Early Termination Analysis

To investigate how to identify when overfitting occurs, we first turn to the change in parameter and link flow RMSE at each step. Of course, in practice it is not possible to know when the parameter RMSE is actually decreasing; but we can examine our experiments retroactively to look for correlations with observable measures to devise a stopping criterion. Because of this distinction in unobservability, we evaluate the two RMSE values differently.

For parameter RMSE, we use a simple percent change to quantify whether a step's parameter values $\vec{t}$ improve or deteriorate over the prior step. By contrast, we evaluate each step's link flow RMSE improvements according to a moving average of the link flow RMSE over the last five steps (or, if fewer than five steps have been completed, all prior steps). This serves to prevent over-criticizing when a single minor improvement is found for link flow RMSE before larger ones are found in later steps.

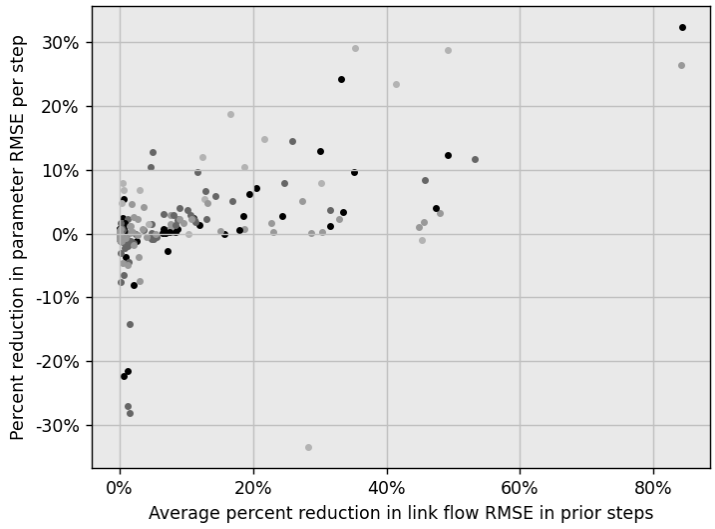

Figure 8: Averaged percent reduction in link flow RMSE vs. per-step reduction in parameter RMSE for Sioux Falls trials

These percent changes for each step of every trial are illustrated as scatter plots in Figure 8 and Figure 9. From these, we see that the bulk of steps improve parameter RMSE and provide significant reduction in link flow RMSE as well. However, there are several data points in these figures in which very little improvement is made in link flow RMSE while parameter RMSE actually deteriorates (points in the bottom half of the figures).

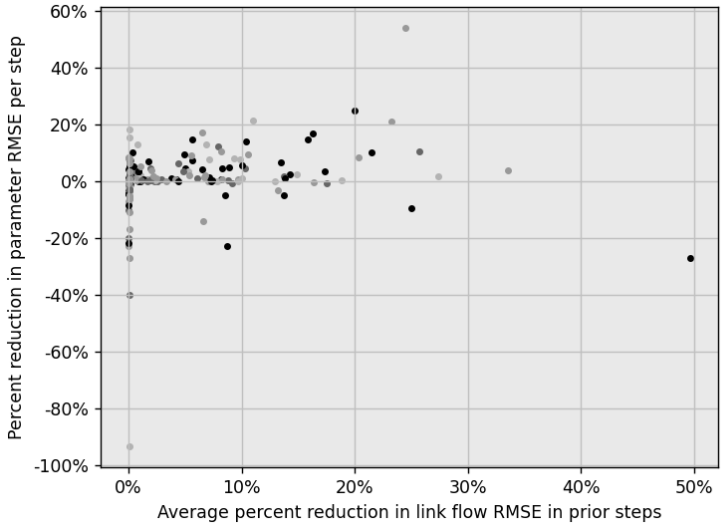

Figure 9: Averaged percent reduction in link flow RMSE vs. per-step reduction in parameter RMSE for Chicago Sketch trials

Based on this pattern, we propose to terminate the local search when the average improvement in link flow RMSE over the lookback period falls below a threshold value. We examined a variety of options for this cutoff value, determining that a good rule-of-thumb option is to use 2.5%. This value serves as a good

tradeoff point between the two scenarios we examined, leading to only 7.8% and 15.9% of steps which (wrongly) increase parameter RMSE for the Sioux Falls and Chicago Sketch scenarios, respectively. This represents a significant reduction from the 38.8% and 40.0% of steps for these two scenarios without early termination.

This early termination, by definition, will have higher link flow RMSE than if the local search algorithm were allowed to continue to its natural termination. Nonetheless, we see that, when implementing early termination, for Sioux Falls, our mean and median flow RMSEs rise to 66.39 and 64.08, respectively, an increase of only 2.1% and 3.2% over their full search termination values. Furthermore, we see for the Chicago Sketch scenario a mean and median flow RMSE of 45.02 and 42.76, respectively, for an increase of 0.9% and 0.6% over the respective values found when the local search terminates. We consider this an acceptable reduction in link flow matching performance, in light of the far greater improvement in parameter error (the actual goal).

As shown in Table 3, there is variation among different parameters in their mean and median performance relative to the values found at local search termination. In the Sioux Falls scenario, we see that the top two parameters in terms of generation have slightly worse performance in matching their true values, but the next two show dramatic improvements, indicating elimination of overfitting.

On the other hand, in Chicago Sketch, we see that the top parameter for trip generation, $ivA1$, has a significant, if not overwhelming, reduction in its error; yet this comes at the cost of significant increases in error for the $iiiP2$ and $iiiA2$ parameters. However, given that these two parameters make up the smallest generators of trips for this scenario, this may be acceptable.

In Figure 10 and Figure 11, we see the visual impact the early termination criterion has: on the left, a closer look at the final steps of the original local search results shows dramatic variations in several trials' parameter RMSE values; on the right, a much cleaner chart that is largely removed of sudden increases in parameter RMSE. This clearly illustrates the impact this technique has on reducing overfitting in our method.

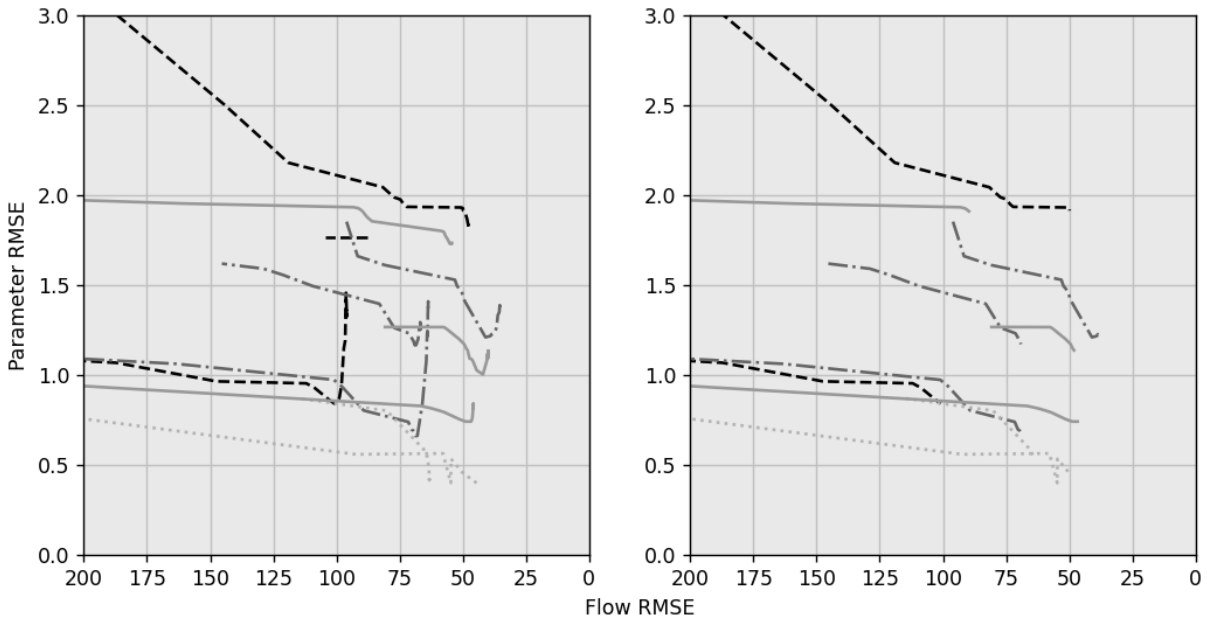

Figure 10: Link flow and parameter RMSE trends for the final steps of Sioux Falls trials, including the full local search (left) and truncating using the proposed early termination criterion (right)

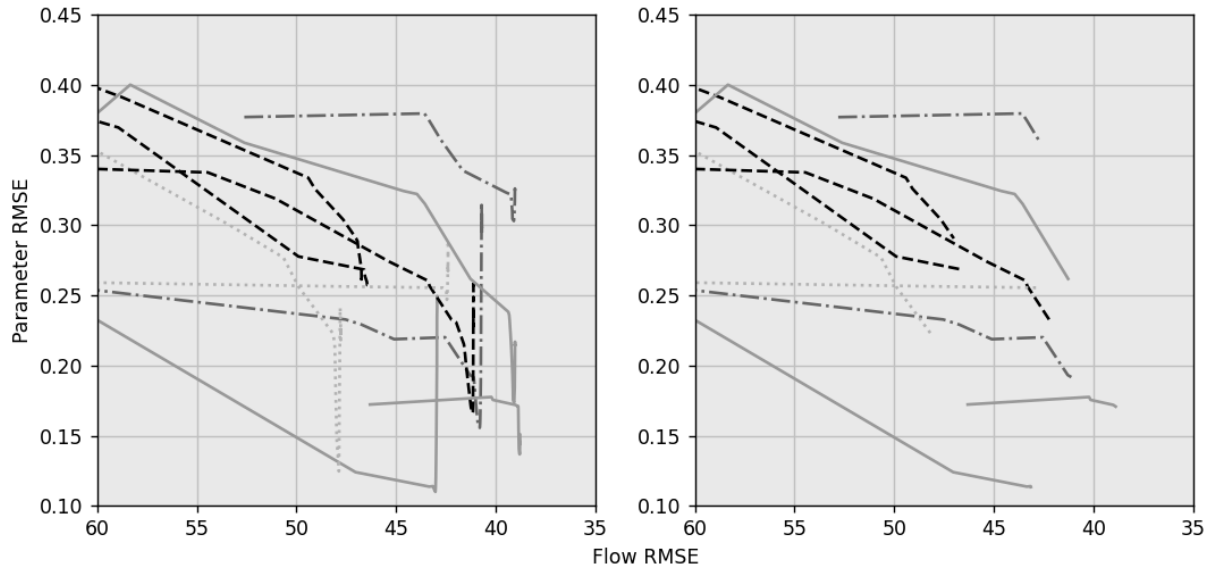

Figure 11: Link flow and parameter RMSE trends for the final steps of Chicago Sketch trials, including the full local search (left) and truncating using the proposed early termination criterion (right)

Discussion and Future Work

We described a procedure that improves a noisy initial estimate of trip generation rates given only observed link flows. Provided an accurate demand model for a study network, we can tune the generation

rates given by a demand survey within their confidence intervals, bringing in flow information from loop detectors and other sensors to further refine our demand model parameters and resulting OD matrix.

Our method is capable of reducing link flow RMSE and generation rate parameter RMSE by an average of 81.6% and 35.6%, respectively, in our Sioux Falls scenario and by 47.1% and 26.2% in the Chicago Sketch scenario. This performance is contingent on terminating the search if the average improvement over the last five step iterations falls below 2.5%; failing to do so may lead to overfitting the starting noisy observation and observed link flows, nullifying many of the improvements made by our method. Furthermore, although in a typical demand model, deciding how trip purposes are balanced is a design decision, our method performs much better on parameters that aren't obfuscated by scaling during the trip balancing step of the UTMS. We recommend that the trip generation rates a modeler deems more important (e.g., categories responsible for more trips) should have their rates fixed during the balancing step where possible. Rates the modeler deems as less important should be balanced against those fixed purposes.

Finally, we note that, although our method was able to improve on every noisy trip generation rate it was provided, we found that the quality of the rates output at search termination is dependent on the initial estimate, instead of always converging to the true parameters. This suggests that further research should explore ways to escape local optima without overfitting.

Our procedure can be easily extended to tune input parameters of the other steps of the UTMS. Future work can be undertaken to see if we can perform as well on those parameters as we do with trip generation rates. More broadly, the preliminary experiments outlined in this chapter reflect a best-case scenario where:

- Our travel demand model isn't too large in inputs,
- Our travel demand model doesn't take very long to run,
- We have true ("observed") flows for 100% of links, and
- Our demand model is perfectly accurate in describing the relationship between trip generation rates and link flows.

In practice, we will want to (a) explore input parameters outside of trip generation rates, (b) study networks much larger than Sioux Falls or Chicago Sketch, (c) use real (i.e., noisy) data which won't have true flows for every link, and (d) relax the assumption that our demand model is perfectly accurate. We

will need to adapt our approach to account for these practical limitations before it will be ready for use in the field.

In improving the abilities of travel demand modelers to predict travelers' trip generation rates, we thus provide a more accurate model overall, as the impacts of this improved accuracy will propagate to later stages of the traditional four-step travel demand modeling process (Zhao & Kockelman, 2002). In the work presented in the following chapter, we presume that a reasonably accurate travel demand model has been developed and/or calibrated, and we show that, by better reflecting the delays caused by traffic signals, we can leverage this model to more accurately anticipate travelers' responsive re-routing. By doing so, we can "head them off at the pass" – we can plan a long-range strategy for signal timings that improves network performance and limits travelers' abilities to gain an advantage by seeking out shortcuts.

4. An Improved Pressure Function for Iterative Traffic Signal Timing Optimization in a Static Route Choice Environment

Consider anew our introductory example city of Springfield, which recently retimed their traffic signals according to the observed traffic volumes in town. However, the citizens of Springfield are cunning, noticing nearly immediately the change in signal timings and, therefore, the delays on their routes. This led to many drivers altering their routing behavior to take advantage of new shortcuts created by the new timings, believing their behavior to be perfectly cromulent and without negative recourse. Quickly, a new traffic pattern develops as the roadway network reaches a new route choice equilibrium. Thus, the recent retiming is rendered stale rather quickly relative to the frequency with which municipalities update traffic signal timings. In this chapter, we will document an approach to iteratively improve signal timings with the expectation that drivers will adapt their routes, and we will show a novel improvement which improves network performance relative to state-of-the-art implementations of this tactic.

Route Choice Modeling

Travelers are, by their nature as human beings, learners – we adapt our route choices based on our perceptions of the choice values. The traffic assignment problem (that is, the search for an equilibrium in which no driver can unilaterally change their behavior to improve their outcome by finding a shortcut) is well-studied in the literature, as we outline in Chapter 1. The current practice of solving this problem is broadly split into two distinct modeling practices: static and dynamic traffic assignment. While the latter aims to maximize the realism of models in their ability to reflect traffic dynamics over a specified period of time and alter traveler behavior based on the performance of individual simulations, the former approach, more relevant to this work, aims to optimize route choice on a more long-term basis.

That is to say, a static assignment approach simplifies the assumptions of traffic dynamics to reach a long-run Nash equilibrium based on long-run travel costs (Bliemer, Raadsen, Brederode, Bell, & Wismans, 2014), guaranteeing an equilibrium solution which cannot be guaranteed by dynamic approaches (Boyles, Melson, Rambha, & Duthie, 2013). By remaining in the realm of static assignment, we are able to describe generalized human behavior – if a shorter route to their destination can reasonably be expected, as potentially evinced by repeated observation of traffic, drivers will naturally take it. The reliability of travel time of individual trips is less important in this context than the generalized long-term routing behavior, therefore we can utilize simple vehicle propagation models such as the BPR function we detail in Chapter 1 which provide a monotonically increasing delay as a function of the volume on a roadway. This

formulation depends, however, on the "long-run" concept by allowing links to serve more flow than their capacity. The intuition behind this variation from standard network optimization practice is that, once a trip has begun, it will *eventually* be served. With this relaxation in place, link capacities no longer need be a binding constraint.

Traffic Signal Timing Optimization

As detailed in the literature review, there is a bevy of research which investigates techniques to optimize traffic signal timings according to myriad performance metrics. However, a noticeably underdeveloped component of this body of work is that of the considerations of drivers' responsive route choices. The foundational assumption that travelers adjust their route choices in the long run should not be treated as independent of signal timing configurations at intersections in the roadway network. Over time, travelers will certainly notice time savings offered by routes with improved timings, and the capacity benefit of the timings will be consumed rather quickly. Therefore, we must consider that the route choice equilibrium is one in which transportation managers play a key role. By improving our modeling of the impacts of traffic signal delay on travel costs (most notably delay), we aim in this chapter to present a mechanism for driving this equilibrium to a more efficient state through anticipatory signal timing.

Smith details in several works, most notably (Smith, 1979; Smith, 2015), the prospect of iterative balancing of the signal timings and routing choices, suggesting an algorithm and control policy to do so which results in this extended form of route choice equilibrium – one in which not only can travelers not find a shortcut in their route, but that no improvement can be made by transportation systems managers to the signal timings to reduce the delay caused by traffic signal timings themselves. These results depend on the concept of a "stage pressure" function, that is, one which details the importance of an intersection approach to attaining equilibrium. The stage pressure is taken in practice as the product of the per-vehicle delay b_i encountered on a link and the saturation flow rate (i.e., the capacity) s_i of the link. That is, for all links i in the network,

$$p_i = b_i s_i \qquad \{9\}$$

This per-vehicle delay representation is of key importance to our contributions. We retain a traditional volume-delay function as a means to reflect the impact of demand on travel delay as a consequence of congestion $h(\cdot)$, but we also wish to incorporate a separate delay which we can optimize through modification of traffic signal timings $b(\cdot)$. Therefore, we separate the traditional travel time function utilized in modeling network propagation to be a sum of the two components, i.e. $t(\cdot) = h(\cdot) + b(\cdot)$. In

the next section, we describe a pair of pressure functions' associated per-vehicle delay functions that are established in the current literature.

Pressure Functions of Focus

One of the first formulations of the pressure representation of traffic, Smith's foundational P0 pressure function is fairly straightforward. The intersection delay for approach i in Smith's model is expressed as:

$$b_i(C, \boldsymbol{g}, \boldsymbol{Q}) = \frac{Q_i}{s_i g_i C} \quad \{10\}$$

In Equation {10}, Q_i is the queue length, s_i is the capacity of the roadway (also known as the saturation flow rate), C is the cycle length of the intersection, and g_i is the fraction of the signal cycle that is allocated to the approach. P0's dependency on queue length rather than overall demand is of note, as it does not meld well with a travel model that utilizes vertical queueing, as is common in static traffic assignment model. These vertical queues have a theoretical queue length of zero regardless of volume on the roadway. However, as this is a rather dull measuring stick against which to hold our contributions, we detail shortly the modifications made to the P0 pressure function to better accommodate these modeling assumptions.

While relaxing this constraint by modeling a physical queue is merited, in practice, as Wang et al. note, the queue length of a roadway is difficult to estimate (Wang, Li, Yue, & Mao, 2019), thus in attempting to maximize network capacity, they propose an alternate intersection delay function in which delay is expressed as:

$$b_i(C, \boldsymbol{g}, \boldsymbol{x}) = x_i \times \frac{(1 - g_i)C}{s_i - x_i} \quad \{11\}$$

In Equation {11}, x_i is the flow on the link, and C, s_i, and g_i are as above. They claim that this modified pressure function creates an equilibrium which maximizes network throughput given fixed demand. Throughout the remainder of this chapter, we refer to the pressure function Wang et al. propose by the shorthand initialism WLYM. One notable drawback to the WLYM delay function is that it is discontinuous at saturation flow and is undefined in over-saturated conditions. While we adjust in our experiments the WLYM delay function (detailed later) to mitigate this, it is once again a notable design incompatibility with the vehicle propagation assumptions of static traffic assignment models. Nonetheless, in an undersaturated network, the promise of throughput maximization is enticing.

However, throughput (or capacity) maximization is not necessarily the ideal metric by which to measure network performance, as it is not something the average driver would consider in their routing. Current practice for evaluating a travel demand model often utilizes metrics including total system travel time (that is, the sum of all trips' travel times) and total vehicle miles traveled, a metric which becomes increasingly important as our transportation system continues to move towards electric vehicles and away from internal combustion engines. Additionally, signal timings can be adjudged by the total delay caused by the signal operation – moving forward, we refer to this as "signalized delay". In the remainder of this chapter, we will compare the P0 and WLYM control policies and their associated intersection delay functions through these three lenses. We will also detail a novel queueing theory-based control policy and delay function and illustrate their superior performance in these three metrics relative to the P0 and WLYM policies.

Method

In the context of Smith and WLYM's work, it is fitting to model a roadway network using static traffic assignment, reflecting a long-term behavior pattern rather than a microscopic simulation of a single instance in time. As we detailed in Chapter 1, static assignment guarantees a route choice equilibrium given a monotonic delay function but does not respect capacity constraints. As such, we note again that WLYM's delay function is discontinuous at saturation flow and undefined in oversaturated conditions. To mitigate this, we adapt for our experiments the WLYM delay function: if the flow is within some small ε of the saturation flow s, we linearly extrapolate the delay using the first derivative at the point $(1-\varepsilon)\times s$. That is, the modified WLYM per-vehicle delay function can be expressed as:

$$b_i(C,\boldsymbol{g},\boldsymbol{x}) = \max(x_i,\quad (1-\varepsilon)s_i) \times \frac{(1-g_i)C}{s_i - \max(x_i,\ (1-\varepsilon)s_i)} + \max(0,\quad x_i-(1-\varepsilon)s_i)\cdot \left.\frac{db_i}{dx_i}\right|_{x_i=s_i} \quad \{12\}$$

To address WLYM's critique of P0, namely that it involves the difficult-to-measure queue length, in our experiments we substitute for queue length Q the volume on the link x, as the units are compatible, and this forms a natural analog to the queue length. Therefore, the implemented P0 per-vehicle delay function can be expressed as:

$$b_i(C,\boldsymbol{g},\boldsymbol{x}) = \frac{x_i}{s_i g_i C} \quad \{13\}$$

While this formulation is simplistic, it is a natural modification to make to accommodate the framework of static traffic assignment with vertical queueing. However, it still does not incorporate the spatial component of queueing.

Though queue spillback caused by oversaturated links is of paramount importance to dynamic traffic assignment, by obviating this facet of traffic propagation, we omit the potential to incorporate pressure caused by a link approaching a spillback condition. This is a promising area of future research, as shown by (Levin & Boyles, 2017). Future work may investigate the applicability of these "back-pressure" approaches to these signal optimization tasks. Nonetheless, in the scope of this work, we lean on using a simple analog by substituting overall demand for queue length.

A1: The Alexander 1 Pressure Function

Now consider that, in the context of long-term behavior, travelers are presumed to be equally likely to arrive at any given point in a traffic signal's cycle[§§]. Thus, the flow on the link can be treated as uniform arrivals. If each vehicle takes equal time to traverse the intersection, this opens the opportunity to borrow a principle from queueing theory: that of a D/D/1 queue (i.e., one with deterministic arrivals and service times). Thus, we can express the average per-vehicle delay as a function solely of the cycle length C and the fraction of the cycle, c_i, that is allocated to the roadway:

$$b_i(C, \boldsymbol{g}, \boldsymbol{x}) = (1 - g_i)^2 \times \frac{C}{2} \quad \{14\}$$

This expression, which we dub as the Alexander 1 (A1) delay function, considers first that a traveler arrives to a signal that is red with probability $1 - g_i$, and that the average delay encountered by a vehicle arriving on red would be half the amount of time the signal is red, i.e. $(1 - g_i) \times \frac{C}{2}$.

Note that, in contrast to the WLYM and P0 per-vehicle delay functions, the Alexander function is flow-invariant. While this would seem contrary to our expectations, consider that there are two components to the amount of delay on the road – one which is caused by the signals themselves, and the other caused by the volume of cars. In low-volume periods such as the dead of night, arriving at a stoplight will cause signal delay while the congestion delay will be negligible. In contrast, the same intersection in the peak hour would face high congestion delay. However, we can consider that any additional impact caused by

[§§] This assumption is, of course, not realistic in the context of coordinated signal control or highly imbalanced flow on a link's upstream intersection, but given the long-term nature of static assignment scenarios, it will suffice.

the signal beyond that of the low-volume scenario can be captured in the congestion delay of the roadway, expressed in a volume-delay function.

A2: The Alexander 2 Pressure Function

Nonetheless, an argument can be made that the signal delay should be modeled as a function of volume, separated from the congestion delay. To address this, we initially alter slightly the A1 delay function to scale linearly with the demand-to-capacity ratio, i.e.:

$$b_i(C, \boldsymbol{g}, \boldsymbol{x}) = (1 - g_i)^2 \times \frac{C}{2} \times \frac{x_i}{s_i} \quad \{15\}$$

We dub this expression of signal delay the Alexander 2 (A2) delay function. The intuition behind this addition is simplistic: assuming the bottleneck on the roadway is the signal, if a vehicle is on a roadway which is at capacity, the average delay would be exactly equal to that of the Alexander delay function above, but in low-volume scenarios, this would be diminished. Additionally, in a static traffic assignment context with vertical queueing, volumes can, in fact, *exceed* capacity. This formulation of signal delay allows for the following consideration: if the roadway has three times the saturation flow rate present, the average vehicle will wait for three cycles before they are served.

However, one drawback of the A2 pressure function is that it leans heavily on the static traffic assignment assumption that all flow will eventually be served, allowing links to be over-capacity. This, in turn, fails to account for the fact that the total delay caused by signals will grow as a function of the magnitude by which flow exceeds capacity. While the A1 function is a true-to-form reflection of the long-run average signal delay only if a *link* is not oversaturated, A2 depends on the assumption that the *network writ large* is not oversaturated to reflect the delay accurately. Scaling the overall demand on the network up or down, then, would heavily impact the pressure A2 exerts if capacities are held constant. As a result, while the A2 function can be used to generate an equilibrium, the A2 pressure function is not a fair candidate as a baseline to compare the signal delays in each of the various equilibria, as simply increasing demand uniformly would drive the A2 equilibrium towards system optimality.

A3: The Alexander 3 Pressure Function

To better reflect the signalized delay in an oversaturated network, we can break down the amount of signal delay into two components – one is the uniform delay, essentially the A1 pressure function, which reflects the delay caused by where in a signal cycle a arrives in the queue building up at the intersection; and the other is caused by oversaturation, that is, the number of cycles a driver must wait before being

served. However, as, up until now, we have been focusing on long-run delay, we must now consider that the oversaturation delay is infinite in an over-capacity facility in the long run. Yet consider that a rush hour does not last forever – the peak period eventually subsides, and the demand is, in fact, eventually served.

Suppose that we know the duration of time T for which demand will be modeled, e.g. the peak hour. Then the number of signal cycles modeled is simply $k = \frac{T}{C}$, and the queue length at the end of the modeled period will be $Q_i = T \times (x_i - s_i)^+$, where the bracketed term is constrained to non-negative values. The final driver in the queue, then, will wait $\frac{Q_i}{s_i}$ cycles to cross the intersection. If we presume that no further demand is added to the network after the modeled period, then the oversaturation delay can be expressed as $\frac{Q_i}{s_i} \times \frac{C}{2}$. Thus, we combine this with the A1 uniform delay to create our third iteration on the Alexander pressure function, A3:

$$b_i(T, C, \boldsymbol{g}, \boldsymbol{x}) = \left[(1 - g_i)^2 \times \frac{C}{2}\right] + \left[T \times \frac{C}{2} \times \frac{(x_i - s_i)^+}{s_i}\right] \quad \{16\}$$

Since the A3 method is a true reflection, then, of the average signal delay encountered in the long run by traffic during the modeled period, we can now depend on this as a solid measuring stick by which to compare the overall signal delay of each of the various equilibria.

Extending *wrap* Functionality

To conduct our comparison of these five policies, we first extended the *wrap* software detailed in Chapter 2 to incorporate emulated ring-barrier-style traffic signal timing plans. This involved modifying the typical link performance function to include two separate components of delay – the volume-delay function of the road itself and the intersection delay defined by one of the above pressure functions. That is, the resulting link performance function, as used in evaluating route choices and network equilibrium convergence, can be expressed as $LPF(T, C, \boldsymbol{g}, \boldsymbol{x}) = h(\boldsymbol{x}) + b(T, C, \boldsymbol{g}, \boldsymbol{x})$, where $h(\cdot)$ represents the non-signalized link performance function (in our experiments, the traditional BPR function (Bureau of Public Roads, 1964) with parameters defined for each of the individual network links). This combined link performance function is then used as the per-link travel cost, an input to the Beckmann function, the optimization problem's objective function expressed in Equation {1}, as well as to the flow shifting mechanism utilized in our implementation of Algorithm B (Dial, 2006; SPARTA Lab, 2017).

What's more, to improve the fidelity of the timing plan representation, the *wrap* network representation was expanded to encompass the concept of individual turning movements which constitute a given link's

output flow. However, this added level of modeling accuracy raises the question of link interactions, a confounding issue in the complexity of traffic assignment problems. This is because, on a given roadway with multiple turning movements, the delay on a link is, in fact, more closely related to the *worst-performing* turning movement, as we assume that movement will block the others from proceeding. Since, for the equilibrium guarantee to still be valid, we need a smooth delay function, the idea of incorporating the maximum delay of the constituent turning movements directly is problematic.

To address this, in calculating a link's intersection delay b_I, we elect to supplant a maximum with a soft maximum, calculated as the log sum exponent of the approach's component turning movement delays b_i:

$$b_I = \ln \sum_{i \in I} e^{b_i} \quad \{17\}$$

This allows us to maintain a differentiable delay function, the importance of which will soon be evident. The derivative with respect to the volume on each turning movement then simply becomes a value-weighted average of the exponentiated volume, i.e.:

$$\frac{\partial}{\partial x_i} b(x_i) = \frac{e^{b(x_i)}}{\sum_j e^{b(x_j)}} \quad \{18\}$$

For simplicity of implementation, the shifting of flow during route choice equilibration retains the previously implemented Algorithm B as discussed in Chapter 2. However, this algorithm requires calculation of the derivatives of our subject intersection delay functions. We detail these in Table 4. In our experiments, we will vary the green shares g_i of each approach, reallocating green shares in in each iteration; however, we hold the intersection cycle lengths as constant as a simplifying assumption which may be relaxed in future work.

Table 4: Derivatives of the focus pressure functions

P0	WLYM***	
$\frac{db_i}{dx_i} = \frac{\left(g_i - x_i \frac{dg_i}{d_{x_i}}\right)}{s_i g_i^2}$	$\frac{db_i}{dx_i} = C \times \frac{\left(x_i(x_i - s_i)\frac{dg_i}{dx_i} + s_i(1 - g_i)\right)}{(s_i - x_i)^2}$	
A1	**A2**	**A3**
$\frac{db_i}{dx_i} = (1 - g_i) \times \left(-C\frac{dg_i}{dx_i}\right)$	$\frac{db_i}{dx_i} = (1 - g_i)C \times \left(\left(-x\frac{dg_i}{dx_i}\right) + \frac{(1 - g_i)}{2}\right)$	$\frac{db_i}{dx_i} = (1 - g_i) \times \left(-C\frac{dg_i}{dx_i}\right) + kC^2$

Experimental Framework

For testing our proposed pressure functions, we turn once again to the Sioux Falls network as detailed in Chapter 3. The original network model does not include a signal timing plan component, so an initial one was generated from scratch. This involves determining the signal groupings, cycle lengths, and green proportions for each turning movement. Foundationally, for all intersections we assume that there are no permissive turning movements (ones in which traffic must yield to traffic from a separate turning movement, e.g. right turns on red or left turns with a flashing yellow arrow). The initial timing plan allocates cycle time to each signal group equally, and each compatible movement in a ring receives equal shares of the signal group's cycle time. The cycle length was set uniformly for all intersections to be 90 seconds in duration. We also assume that *all* intersections in the network are signalized as a simplification of our scenario.

Let us now consider the signal groupings – there are four types of intersection in the Sioux Falls network, ranging from two-approach intersections to a single five-approach intersection. For the two-approach intersections, each approach was treated as a separate signal group, with only one approach able to proceed at a time. For three-approach intersections, a timing plan was developed in which one signal group contains two of the approaches and a second contains the third. This allows for a protected left turn phase for one leg. Additionally, three-approach intersections also feature specialized turning movements for right turns that overlap both the corresponding through movement and the movement

*** Note that, in implementation, the flow value x_i is capped at some defined upper limit value, $(1 - \varepsilon) \times s_i$ when calculating this derivative.

opposite the corresponding outbound link. These overlap movements were also implemented for right turns at four-approach intersections, and all other movements are as would be expected in a typical four-way intersection timing plan with protected left turn phases. Each approach to the five-way intersection was allocated to its own signal group, with all turning movements from that approach being served simultaneously. Additionally, overlap movements were once again incorporated for the five-way intersection.

An initial flow assignment was generated corresponding to drivers utilizing the shortest path under free-flow conditions, and the initial assignment and timing plans serve as the initial solution to our bilevel optimization problem of balancing signal timings and routings. From this initial solution, the flows and signal timings were iteratively equilibrated using the Algorithm B and Smith shifting techniques, respectively.

As with Smith, we must provide a small scaling factor w_i at each iteration i. To improve convergence, we elect for the following monotonically decreasing function based on the method of successive averages:

$$w_i = \frac{2}{(i+2)B}, \qquad i \in \{0,1,2,\dots\} \tag{19}$$

In the above expression, B represents an upper bound on the pressure b_i any link may exert. In our experiments, we calculate this upper bound as a function of the capacity s_i and the cycle length C:

$$B = 100 \cdot \max_i(C_i \cdot s_i) \tag{20}$$

Future work may investigate improvements in convergence rate by modifying this step size function, as this simplistic approach is far from optimal. Nonetheless, as the resulting equilibria will be identical regardless of our choice of function for calculating w, we can be assured that this is an acceptable albeit imperfect implementation choice.

Pseudocode providing a high-level description of the implemented joint route choice/signal timing optimization procedure is presented in Figure 12. At the beginning of each iteration, we calculate the change in green shares Δg_i based on the method detailed in (Wang, Li, Yue, & Mao, 2019), based on the flows on the network as they are in that instant (i.e., the start of the iteration), thus allowing us to treat these values as constant. This is shown in Equation {21}**,** where w is the scaling factor, g_i is the green time allocated to each approach, s_i is the saturation flow on each link, and b_i is the pressure function selected.

```
BEGIN
Initialize assignment using warm-start if available, else free-flow shortest path
Evaluate assignment gap function
Calculate the stage pressure upper bound B
Set iteration index i = 0
While the assignment gap function exceeds the target threshold and the total iterations has not breached the threshold:
    Calculate a new scaling factor w:
        w = 2 / ((i + 2) * B)
    Calculate the amount to shift each intersection's cycle time allocation between the timing plan's constituent signal group, Delta_G[]:
        For each signal group sg_a:
            For each signal group sg_b:
                If rank(sg_a) < rank(sg_b):
                    Delta_P = pressureFunc(sg_a) – pressureFunc(sg_b)
                    // g_a is the fractional cycle share of sg_a
                    Delta_G[ sg_a ] += w * Delta_P * g_a
                    Delta_G[ sg_b ] -= w * Delta_P * g_b
    Calculate the amount to shift each signal group sg's green share allocation between the constituent phases of each ring, Delta_M[]:
        For each turning movement tm_a:
            For each turning movement tm_b:
                If rank(tm_a) < rank(tm_b):
                    Delta_P = pressureFunc(tm_a) – pressureFunc(tm_b)
                    // g_a is the fractional share of the shared ring
                    Delta_M[ tm_a ] += w * Delta_P * g_a
                    Delta_M[ tm_b ] -= w * Delta_P * g_b
    For each bush to assign:
        Calculate bush path costs utilizing the combined congestion delay-signalized delay link performance functions
        Update the bush by pruning unused links and adding shortcut links
        For each inner iteration allowed:
            Equilibrate the bush by shifting flow from the longest path to the shortest path
            If the bush's gap function is below the desired threshold, go to next bush
    For each intersection in the network:
        Shift the cycle time allocation between the component signal groups as per Delta_G
        Shift the ring time allocation between the component turning movements as per Delta_M
    Re-evaluate assignment equilibrium gap function
    i += 1
END
```

Figure 12: Combined route choice and signal timing optimization algorithm pseudocode

$$\Delta g_i = \sum_{\{i,j\}:\, i<j} w\{g_i \cdot (s_j b_j - s_i b_i)\} \quad \{21\}$$

Once these values are calculated, we equilibrate flow utilizing those derivatives before adjusting the signal timings at the end of the iteration. The iterative process was run for each of the pressure functions until the relative gap of the solution dropped below the level of 10^{-6}, indicating an equilibrium has been reached between signal timings and route choices. For the A3 pressure function, a modeling period of 2.5 hours was used, representing 100 signal cycles of demand. For the WLYM pressure function, an ε value of 0.9 was used, meaning that the derivative is capped at the value corresponding to 90% of capacity and that, above that flow, the pressure increases at a constant rate equal to that derivative. Future work may investigate the sensitivity of the various functions' performance to these parameter choices.

Results

To begin our comparison of the pressure functions, the total distance traveled, also referred to as vehicle-miles traveled or VMT, is listed in Table 5 and visualized in Figure 13. For each equilibrium, this was calculated as a sum of the product of each link i's flow x_i and its length l_i:

$$\text{VMT} = \sum_i x_i \cdot l_i \quad \{22\}$$

Table 5:Total vehicle miles traveled in the Sioux Falls network for each pressure function

Equilibrium	Total Distance Traveled (vehicle-miles, millions)
A1	3.414
A2	3.420
P0	3.460
A3	3.633
WLYM	3.650

As can be seen, the A1 and A2 functions improve over the existing P0 and WLYM methods in terms of reducing VMT, reducing P0 VMT by 1.33% and 1.14%, respectively; meanwhile, A3 performs only marginally better than WLYM (improving VMT by 0.46%) and significantly worse than P0 (a 5.02% increase). Nonetheless, VMT may not be of critical importance if congestion delay, signal delay, and overall total travel times can be improved.

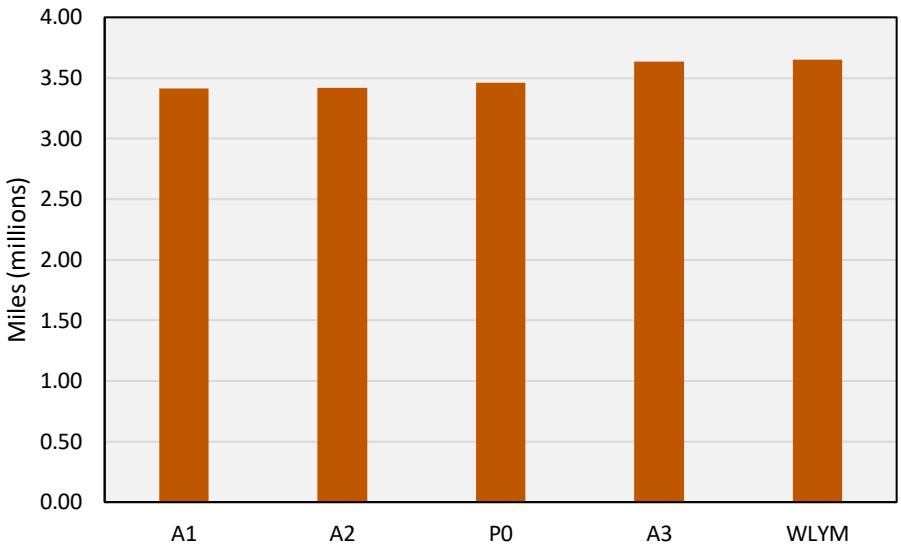

Figure 13: Total vehicle miles traveled in the Sioux Falls network for each pressure function

Let us next turn to the congestion delay on the network, that is, the component of link delay which is attributable to the volume of flow on the link and not the traffic signal. In these experiments, this is the delay reported by the BPR function, the chosen volume-delay function. As shown in Table 6 and Figure 14, A3 significantly improves in this metric, outperforming P0 by 0.50% and WLYM by 2.1%. By contrast, A1 and A2 perform marginally worse than P0, increasing its total congestion delay by 0.40% and 0.06% respectively, though they both outperform WLYM by 1.19% and 1.52%.

Table 6: Total delay attributable to link congestion under each pressure function

Equilibrium	Congestion Delay (minutes, millions)
A3	7.442
P0	7.480
A2	7.484
A1	7.509
WLYM	7.599

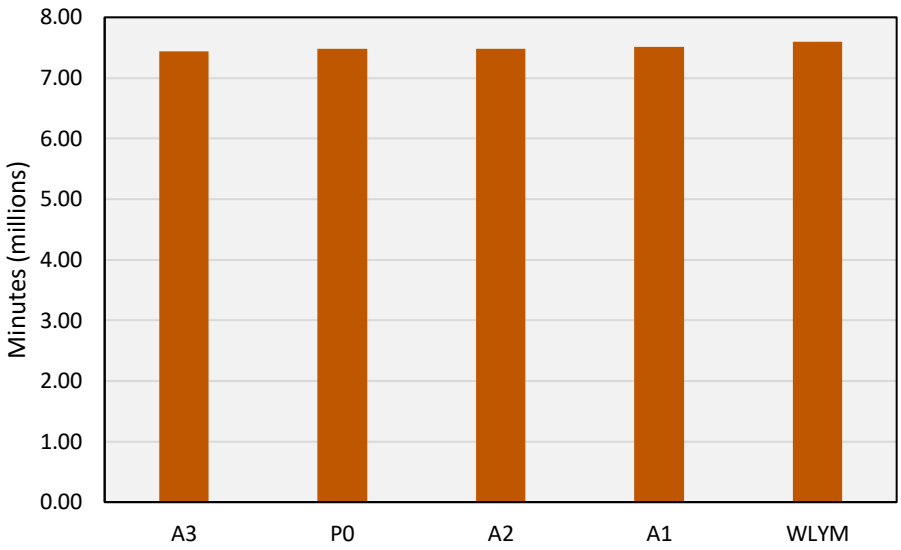

Figure 14: Total delay attributable to link congestion under each pressure function

Now, as we have previously discussed, the various pressure functions each measure signal delay differently, so we present in Table 7 the disparate signal delay metrics for each equilibrium. As this table illustrates, the metrics are of varied magnitude, so comparing the delay of each equilibrium according to its own delay function is not an apples-to-apples comparison.

Table 7: Per-pressure function signalized delay for each pressure function's equilibrium

Equilibrium	Delay Value (minutes, millions)				
	A1	A2	A3	P0	WLYM
A1	0.695	0.854	55.683	3.491	66.623
A2	0.695	0.853	55.346	3.486	66.308
A3	0.710	0.851	48.577	3.435	61.132
P0	0.693	0.845	54.128	3.439	64.784
WLYM	0.704	0.840	48.435	3.392	59.979

We select A3 as our baseline signal delay metric given that it serves as a best reflection of the uniform *and* oversaturation delay encountered on each link during the modeled period. As illustrated in the table and in Figure 15, WLYM is the champion equilibrium by A3 standards, improving over A3 itself by 0.29%. Additionally, A3 and WLYM each outperform P0 by more than 10%, while A1 and A2 create a 2.87% and 2.25% increase in signal delay over the P0 scenario.

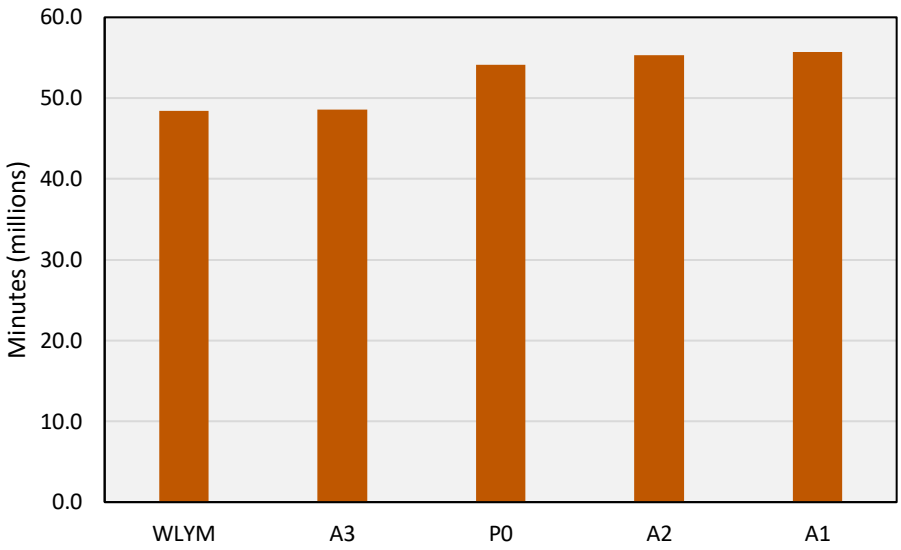

Figure 15: Signal delay for each equilibrium as measured with the A3 pressure function

However, let us consider that, while WLYM outperforms A3 in signal delay, it also falls far shorter in the congestion delay metric. Therefore, we present in Table 8 and Figure 16 the total travel time, i.e. the sum of congestion delay and A3 signal delay. By this metric, A3 narrowly improves over the WLYM method (an improvement of 0.03%). Combined with the improvement A3 provides over WLYM in total VMT, it appears clear that the A3 equilibrium is superior to that of WLYM.

Table 8: Total travel time for each equilibrium as measured with the A3 pressure function

Equilibrium	Total Travel Time (minutes, millions)
A3	56.02
WLYM	56.03
P0	61.61
A2	62.83
A1	63.19

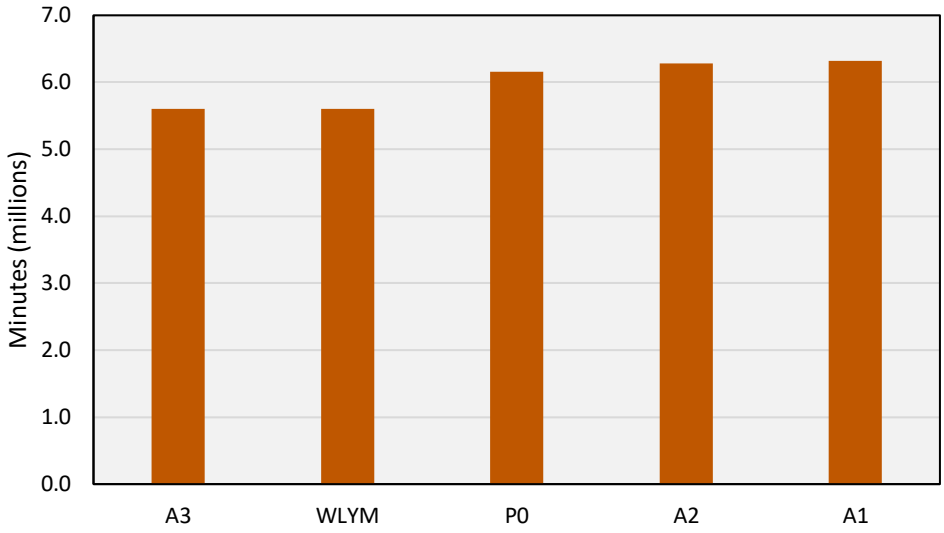

Figure 16: Total travel time for each equilibrium as measured with the A3 pressure function

Figure 17 illustrates the demand-to-capacity ratios on each link in the network across the various equilibria. As can be seen, not only do P0 and WLYM equilibria cause most of the links in the network to be oversaturated, but it is apparent that the Alexander pressure functions selectively route traffic to more pragmatically take advantage of available capacity on the fringes of the network while not suffering from the high levels of congestion in the core of the network as P0 and WLYM do. This can also be seen in Figure 18, showing the flows resulting from each pressure function's equilibrium, and Figure 19 which illustrates the amount of total signal delay encountered on each link in each equilibrium. Finally, Figure 20 shows the per-vehicle signal delay on each link. Given a cycle length of 90 seconds, these maps illustrate the effectiveness of the A3 pressure function in reducing the number of links on which drivers must wait multiple signal cycles to traverse an intersection.

Discussion and Future Work

We have detailed three new expressions for the amount of delay a vehicle will encounter in the long run due to a traffic signal, and experiments show that they each offer improvements over the current state of the art as represented by P0 and WLYM. While A1 and A2 both improve over these methods in VMT, this comes at the cost of higher signal delay, potentially counteracting any emissions improvements that may arise from the reduced VMT. By contrast, A3 improves over P0 and WLYM in overall travel time but fares less well in the VMT metric. This can be considered a tradeoff for policy-makers – signal timings may be optimized to minimize VMT or total travel time, but these goals do not necessarily align in all circumstances.

A handful of assumptions have been made for simplicity in these experiments, and future work may seek to relax these. First, the overall amount of demand is a key contributor to the signal delay in several of the pressure functions, so investigations of the sensitivity of the various functions to total demand are merited. Additionally, a fixed ε value was selected as the cutoff value for the WLYM pressure function to prevent numerical issues near capacity – it remains to be seen how the selection of this value impacts the performance of WLYM relative to the other pressure functions. Experiments should also be conducted on larger-scale networks and with varying signal timing plan structures to investigate the impact of network topology on pressure function performance. Finally, while cycle lengths were held uniform and unchanging across the network in our experiments, this is a common variable in setting signal timing plans, so further investigation which varies cycle lengths (either uniformly or per-intersection) is also warranted.

In addition to modifications to the experimental structure, we note that there are additional metrics by which the flow-signal timing equilibria resulting from these policies may be measured. One interesting prospect for further research is the invocation of stochastic microsimulation modeling approaches to better reflect traffic propagation characteristics, measuring the cost of travel as realized in a Monte Carlo simulation as a baseline that may match real-world conditions more accurately. Such a metric could not guarantee an equilibrium on its own due to the intrinsic drawbacks of dynamic traffic models (Boyles, Melson, Rambha, & Duthie, 2013) but could provide an interesting point of comparison of the equilibria generated using the approaches we present herein. Further model expansions which incorporate stochasticity in other manners are also of merit, with stochastic origin-destination demand and noisy route cost estimation being natural areas of future investigation.

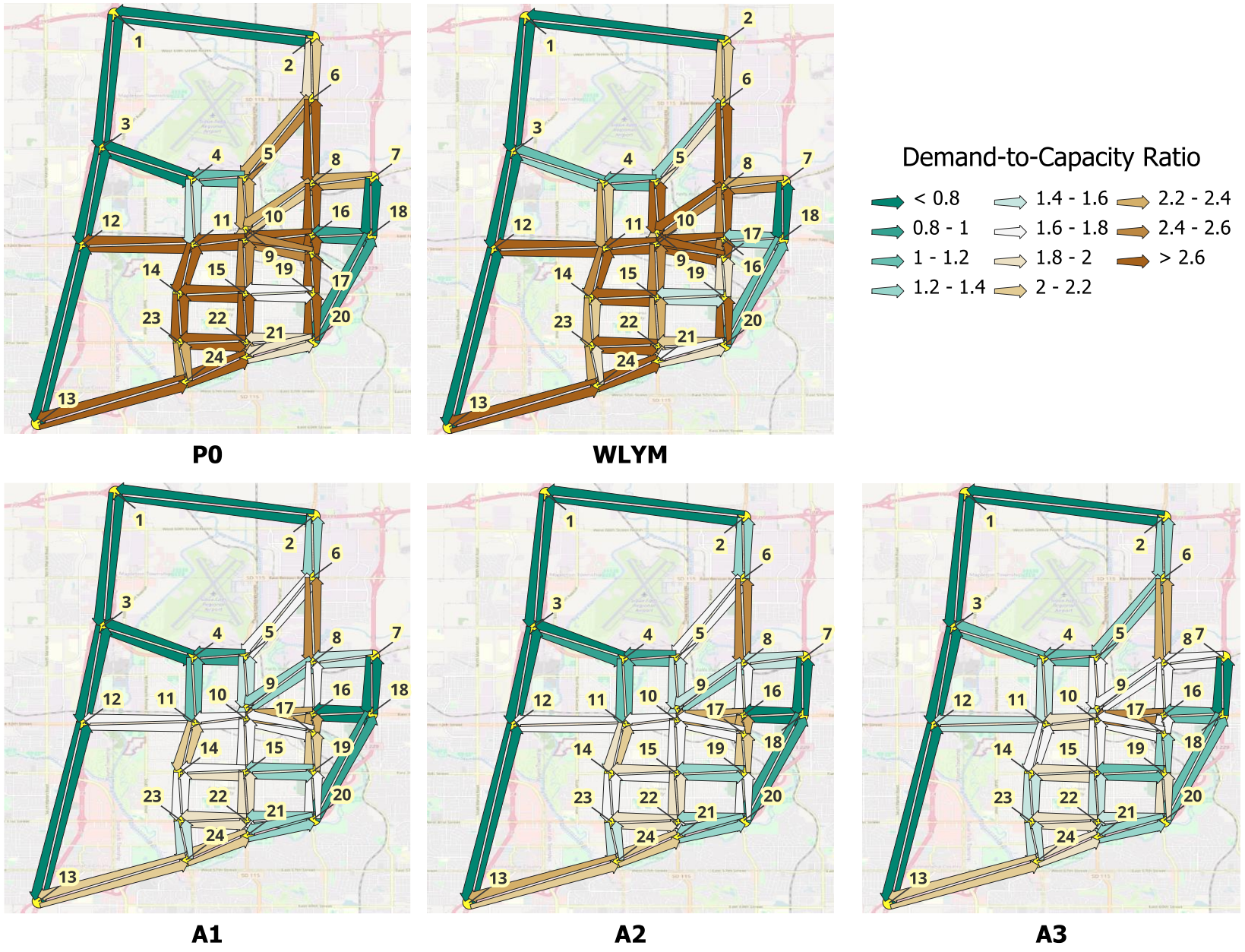

Figure 17: Demand-to-capacity ratio across each equilibrium

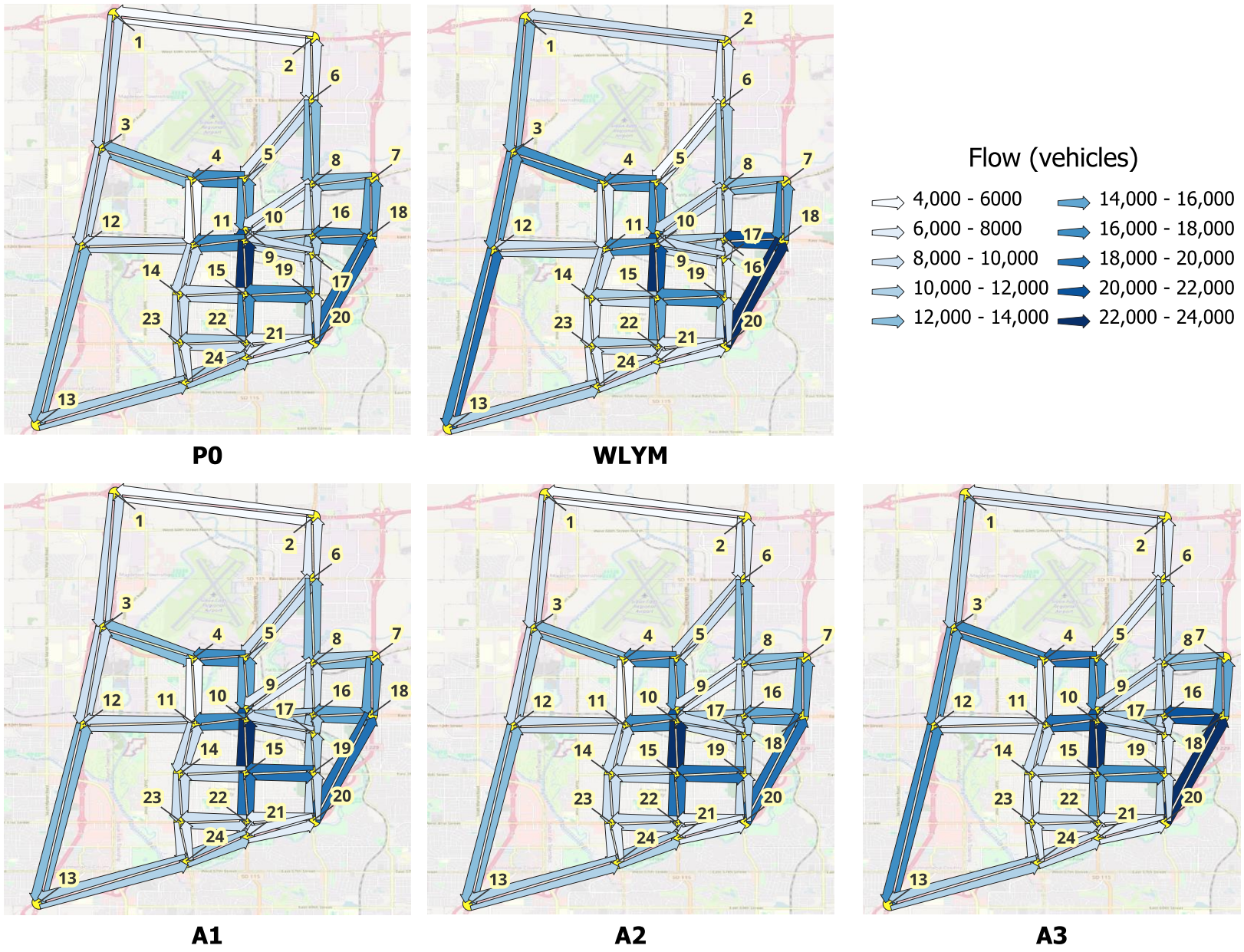

Figure 18: Maps of equilibrium flow resulting from each pressure function

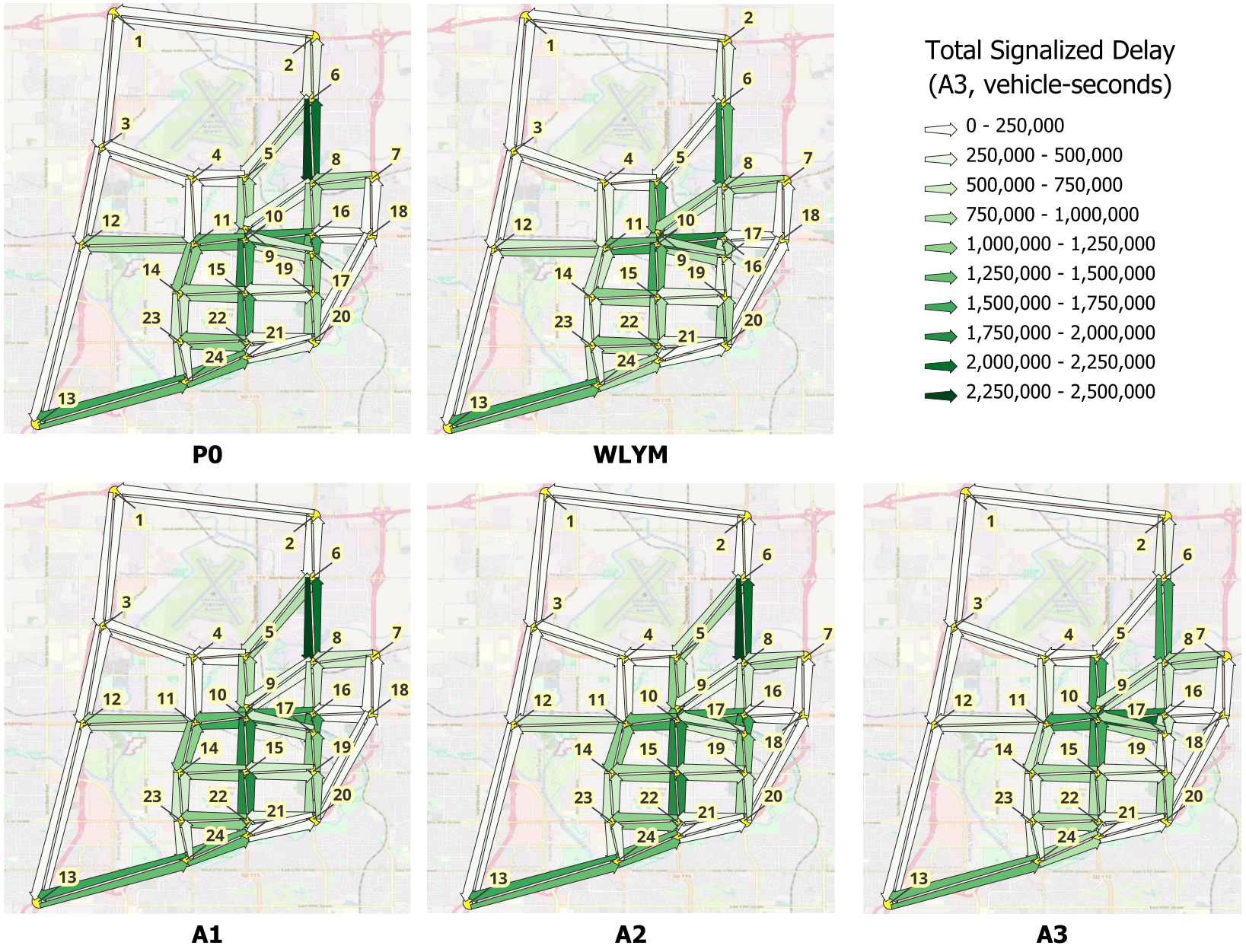

Figure 19: Maps of delay attributable to traffic signals (as determined by the A3 delay function) at each pressure function's equilibrium

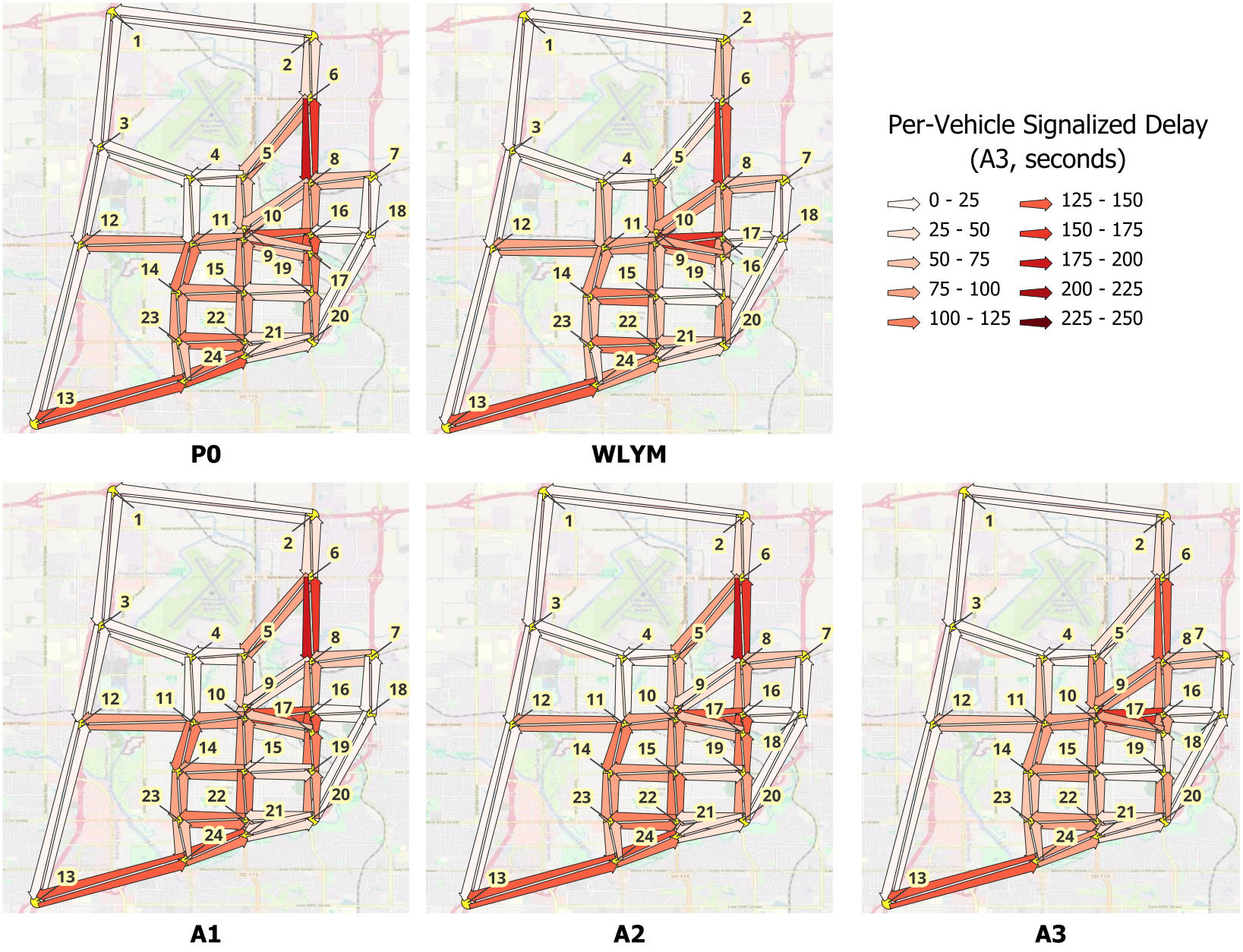

Figure 20: Maps of per-vehicle delay attributable to traffic signals (as determined by the A3 delay function) in each equilibrium

5. Corridor Travel Time Prediction Using Real-Time Probe Data: A Lone Star Case Study

This chapter tests machine-learning based models for travel time estimation in real-time on selected corridors in Austin and El Paso. Developed models compute dynamic travel times between origins and destinations within a corridor using real-time data. In comparison to current practice, which estimates travel times based solely on current conditions, the dynamic models we detail in the next section are expected to provide more accurate travel time estimates by incorporating forecasts of traffic conditions as travelers move down a corridor.

Two regions are to be considered in this in this effort. These include origin-destinations in both directions along two facilities in Austin (State Highway 130 and Interstate 35) and two facilities in El Paso (Interstate 10 and State Highway Loop 375). We will consider two routes on IH-35 corridor through Austin to account for the use of the upper or lower decks of the split-deck segment. In total, this work will analyze origins and destinations on ten corridors: two on IH-10 (one in each direction), two for SL-375, two for SH-130, and *four* for IH-35 (two per direction, including one for the upper and one for the lower deck routing).

The models to be tested on the field have already been studied in this effort for one path pair on IH-35 through Austin and led to promising results. Deploying these models at new sites requires training and calibration using site-specific traffic data. This chapter will summarize model training techniques, challenges faced, and resulting performance from the trained models.

The following sections summarize the trained models, the experimental design, and next steps in this project.

Model Description

Two types of models have been developed in this effort to forecast travel times on segments along a corridor and will be used in our field experiments. These models use segment-level speed data at five-minute intervals (as provided by INRIX Roadway Analytics) and traffic volume data at five-minute intervals collected by a limited number of smart work-zone trailers provided by TxDOT. Real-time speed and volume data retrieved every five minutes are used to forecast segment-level travel times on downstream segments for up to one hour into the future. Using such forecasts, origin-destination travel time estimates for time step t are computed using forecasted segment-level travel times that correspond to the arrival time at each segment along the corridor, which cannot be measured at time t.

The first tested model consists of a recurrent neural network (RNN) which can capture temporal dynamics. Using the standard `torch` Python library, our model creates a RNN for each roadway segment in a corridor for twelve timesteps into the future, each containing a single hidden layer with 200 nodes. Each of these separately trained models takes as input the per-segment INRIX travel time estimates for all segments in the same travel direction, a vector of volume data along the corridor, indicator variables for the availability/reliability of said sensor data, plus an embedded categorization of the hour of the week from which the data was drawn. These models are then trained on a subset of the available data to minimize the mean squared error (MSE) of the output predictions when compared to the ground truth, assumed to be the dynamic travel time as determined from INRIX historical data.

The second model is a linear time series model composed of a module which standardizes input data (i.e., shifts to a zero mean and unit variance) and a linear regression module which fits a multi-output regression based on the input time-shifted INRIX travel time data which outputs a vector of predictions for the next n timesteps.

The performance of the machine-learning-based dynamic travel time models will be compared against two simpler models: a dynamic model that forecasts downstream segment travel times using weakly mean values, and a static model that implicitly assumes that travel times in downstream segments will not change from the previous timestep (which is consistent with current practice). In this chapter, we focus on the development of the RNN-based models, as these have shown promising results and incorporate the features of the linear time-series models as well.

Experimental Design

The general procedure of developing predictions for the RNN and time series models is illustrated in Figure 21. The live stream of INRIX travel time data and TxDOT Lonestar C2C sensor data[†††] is stored in a PostgreSQL database for reference, and the catalogued data is used as training and evaluation data for model development. Additionally, after training has progressed to develop sufficiently-performing models, the most recent input data is fed to these models to generate predictions in real time.

For our experiments, we aim to train and test each corridor's models on a year-long collection of data to provide the models with a firm sense of typical traffic dynamics, as well as individual instances of traffic

[†††] In the form of measurements of volumes, occupancies, and speeds

anomalies. Segment-level speeds data will be provided by INRIX. C2C data is obtained in real time from the TxDOT Lonestar data feed.

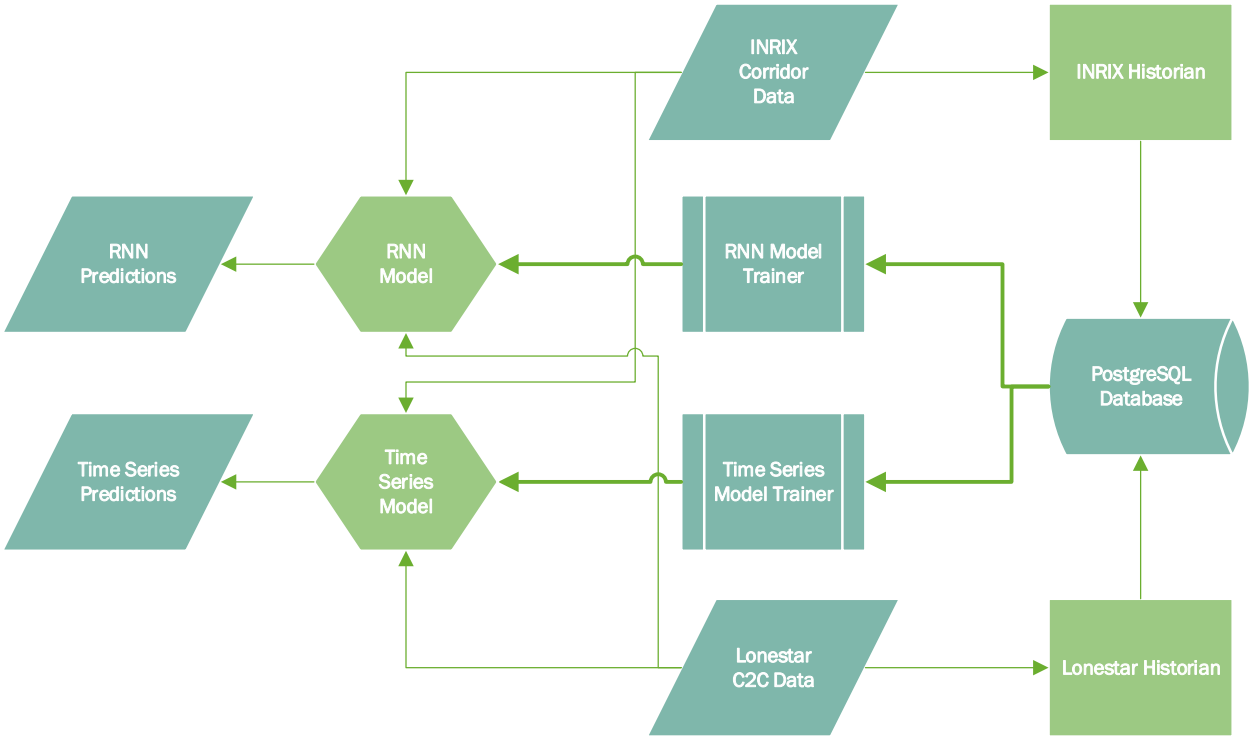

Figure 21: Model training and execution workflows

Trained models are evaluated by using live-stream data from the INRIX and C2C feeds as inputs to generate real-time origin-destination travel time estimates. Model outcomes will be archived in a database for model performance analysis, and also made available in real-time through a simple interface. Travel time estimates are evaluated against a posterior ground truth obtained from INRIX data. A comparison of the accuracy of the model predictions will be developed, and performance will be evaluated both for long-term accuracy as well as the ability to accurately predict the onset and dissipation of traffic anomalies. The evaluation process may include comparing model results to the predictions of both our naïve models and to the current estimates provided on dynamic message signs. The model sensitivity to input data quality and availability will also be reported as part of the evaluation process.

The input data received from INRIX and the Lonestar C2C feed require a degree of preprocessing, which we handle in bulk for the RNN model training datasets. Among the critical steps in this is the determination of the UTC and local time at which each data point was collected. This is essential to prevent errors caused by time zone variability across a given dataset, for example due to Daylight Savings Time. We further

embed the timesteps into five-minute bins which reflect the point of time *in the current week*[‡‡‡], thus incorporating the cyclic nature of traffic as well as its dependence on daily patterns. This, however, does not include an indicator for longer-term seasonality; thus, the models may not be able to effectively recognize patterns caused by, say, shorter daylight hours in winter.

Computational Design

While trained models can produce travel time estimates almost instantaneously, the training process is resource intensive. Given the high quantity of nodes not only in each individual RNN, but also the total number of nodes among all RNNs, it should come as no surprise that the computational complexity of the RNN training task is substantial. The number of parameters to be estimated for a single corridor pair approaches 60 million parameters, which, if trained on a CPU, would require months of wall clock time to complete. To improve training speed, we offload training to a GPU capable of performing floating point operations at a much higher rate. This reduces the training time for RNNs to a matter of minutes. We further note that the computational complexity of the training appears to scale approximately linearly with both the amount of training data used as well as the number of INRIX segments which make up the corridor for which the RNNs are being trained.

To keep an eye to automating the procedure of training and predicting travel times, we develop modularized source code which can be ran as a continuous process (a daemon or app) or on a schedule to further train the model as new datapoints become available, as well as a means to provide an updated travel time upon demand. These processes will communicate with a database to provide data reliability, and a front end will provide predictions to end-users.

Training Data

The training runs reported in this document considered two data types: segment-level travel times from INRIX and volume data provided by the TxDOT El Paso District from the C2C data feed. Models were trained using either twelve months of INRIX data or six months of both INRIX and C2C traffic volume data.

[‡‡‡] We term this binning “weekpoints.”

For training RNN models, we developed two bidirectional corridors of focus – Interstate Highway 10 through El Paso and State Highway 130[§§§] bypassing Austin. Maps showing these two initial corridors are shown in Figure 22 and Figure 23, respectively. These corridor endpoints were chosen due to their encompassing the divergence and re-entry of two alternate routes which will be studied in comparison – State Highway Loop 375 and Interstate Highway 35. The development of reliable models for predicting travel times across these four corridors will provide important information for drivers transiting the two studied areas and may influence their route choice significantly. The SH-130 corridor encompasses 109 discrete segments in each direction, with each segment providing independent travel time data. The IH-10 corridor provides the same for 60 and 55 segments for the east- and west-bound directions, respectively.

INRIX datasets feature incomplete data on occasion, but this is quite rare – less than 0.25% of datapoints are missing across all corridors. Nonetheless, this missing data must be accounted for to provide sufficient data for the RNN training. To do this, we make two data manipulations. First, we exclude dates for which a large period of data is missing for all segments in the corridor. These dates tend to involve major events such as the beginning or ending of Daylight Savings Time, or the February 2021 winter storm disaster and ensuing electricity crisis. Second, for data where only a select number of points are missing, we replace missing values with the mean value for the missing segment-weekpoint combination. That is, if a datapoint for Segment 100 is missing at Weekpoint 20 for a given day, we provide instead a replacement value calculated as the mean value for all records of Segment 100 at Weekpoint 20. We further provide to the RNN a Boolean variable (one which takes only true or false values) indicating that this datapoint is unreliable.

Additionally, the C2C data feed also provides volume, occupancy, and speed data at three detector locations, disaggregated across four lanes at each location, for both directions of the IH-10 corridor in El Paso. This dataset also includes missing datapoints, with per-lane data availability ranging from 84% to 95% of datapoints provided, depending on detector location and lane. Missing datapoints must also be accounted for, so we preprocess this dataset by replacing missing values with the mean value for that data source across available data for the same weekpoint.

To properly train the RNN models, the utilized dataset must be split into three distinct classes of records: training data, which is fed to the RNN models to develop their parameters in each epoch; validation data,

[§§§] This corridor also uses a small portion of State Highway 45, a facility of comparable quality. For brevity, we refer to the corridor as SH-130.

which is fed into the RNN models while holding the parameters constant and provides a scoring metric (in this case, mean squared error) which the RNN seeks to minimize; and testing data, which is only fed into the model upon completion of training as a means to confirm the generalizability of the models. For our models, 20% of dates were withheld from the model as the testing dataset; of the remaining data, 80% (64% of the whole dataset) was provided for training and the other 20% (16% of the whole dataset) was provided for validation. The dataset was split by date, with each day's data provided from 03:00 to 03:00 the next morning.

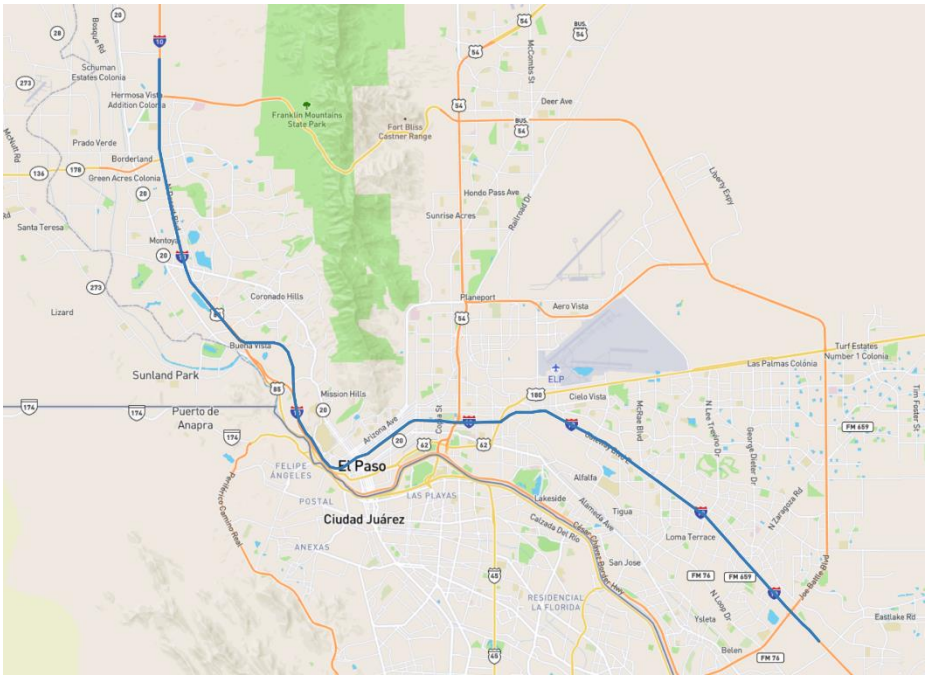

Figure 22: Westbound IH-10 INRIX Segment Map

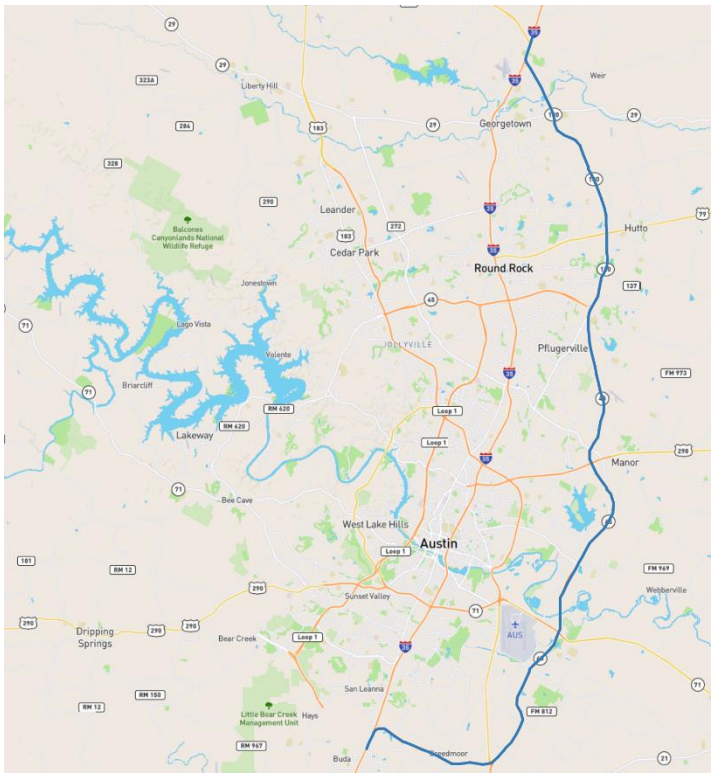

Figure 23: Northbound SH-130 INRIX Segment Map

The following scenarios were evaluated:

- IH-10 Eastbound with 12 months (2021-01 – 2021-12) of INRIX travel time data
- IH-10 Westbound with 12 months (2021-01 – 2021-12) of INRIX travel time data
- SH-130 Northbound with 12 months (2021-01 – 2021-12) of INRIX travel time data
- SH-130 Southbound with 12 months (2021-01 – 2021-12) of INRIX travel time data
- IH-10 Eastbound with 6 months (2021-12 – 2022-05) of INRIX travel time data and C2C volume and occupancy data
- IH-10 Westbound with 6 months (2021-12 – 2022-05) of INRIX travel time data and C2C volume and occupancy data

Results

The output of the RNN training process is a set of models which take in datapoints for the current timestep and output a prediction of segment-level travel times at a given timestep in the future. These models are iteratively improved by evaluating the change in prediction MSE for the validation dataset as a function of the input training data. The MSE acts as the objective function of the optimization problem which we seek to minimize, but we aim to avoid overfitting (the process of training a model to predict the input training data alone, without good generalizability). To do so, we follow the standard procedure of

withholding the validation and testing datasets discussed earlier and penalize the RNN parameter optimizer for models that predict these withheld datasets poorly – this avoids letting the RNN train over the full dataset so as to enforce the aim of developing a model that generalizes to all data rather than the training dataset alone.

We further evaluate the predictive strength of our models using the coefficient of determination (R^2) and mean absolute error (MAE) metrics. Equation {23} shows the formula for MSE, Equation {24} shows the formula for MAE, and Equation {25} shows the formula for R^2. In these formulae, Y_i represents the true segment travel time for a given timestep, $\bar{Y}$ represents the mean true segment travel time, and $\hat{Y}_i$ represents the predicted segment travel time for a given timestep.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad \{23\}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \quad \{24\}$$

$$R^2 = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})} \quad \{25\}$$

In general, we see these error metrics degrade as we forecast further into the future, but near-term results show promise. While they are not yet strong in their predictive power (e.g., R^2 does not exceed 0.6 for the studied corridors), we believe further training and model refinement may improve performance.

Figure 24 presents the R^2 as a function of the number of timesteps into the future for which predictions are generated. In general, results show a trend across all corridors in which predictive power drops considerably as we predict further and further into the future. While this is not unexpected, the predictive power of the models falls significantly after two timesteps. Potential issues for the observed fit include:

- Limitations in input data: Our models were trained using data for year 2021, and traveler behavior may have changed considerably throughout the year due to the impacts of the COVID-19 pandemic, leading to poorer results (given that we are not including data which could predict changes in behavior in response to local outbreaks). Additionally, annual seasonality has not been explicitly considered. Future work may address this by adding more years' data to the training dataset and/or providing an input variable indicating the week or day of the year at which the input datapoint was collected.

- RNN design: the training experiments described in this document used the same RNN design used in our initial protype for IH-35 through Austin. While in the Austin case the single-hidden-layer RNN led to very promising results, adding hidden layers is often better suited to approximating complex nonlinearities such as weekly or seasonal patterns in input data. Future work may investigate the benefits of adding both additional records (i.e., adding more than one year of data) and/or reshaping the structure of the RNN, as both would be expected to improve model generalizability.

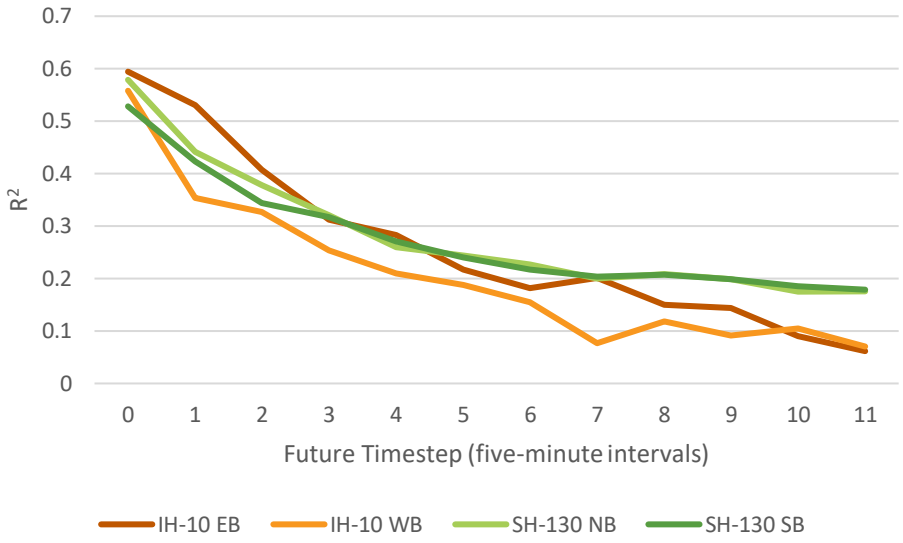

Figure 24: R^2 values for models trained on a full year's data

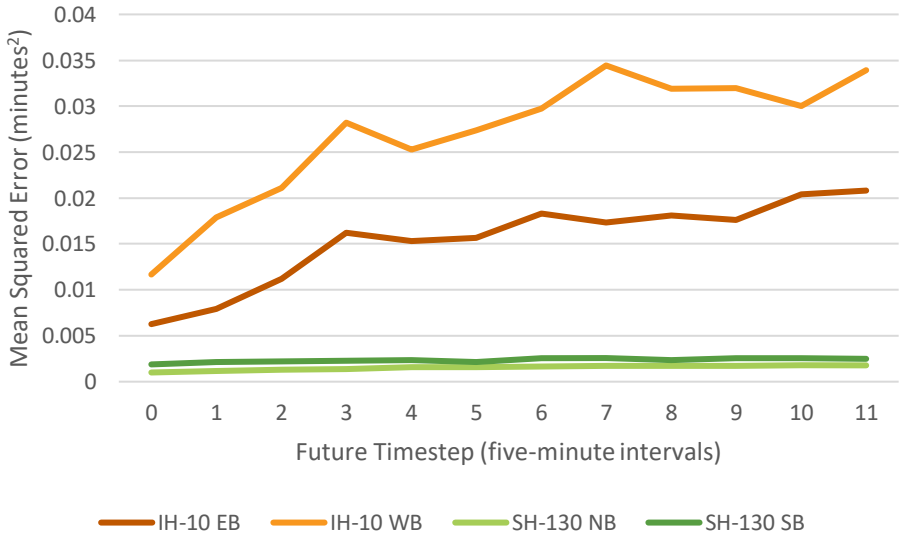

Figure 25: Mean Squared Error for models trained on a full year's data

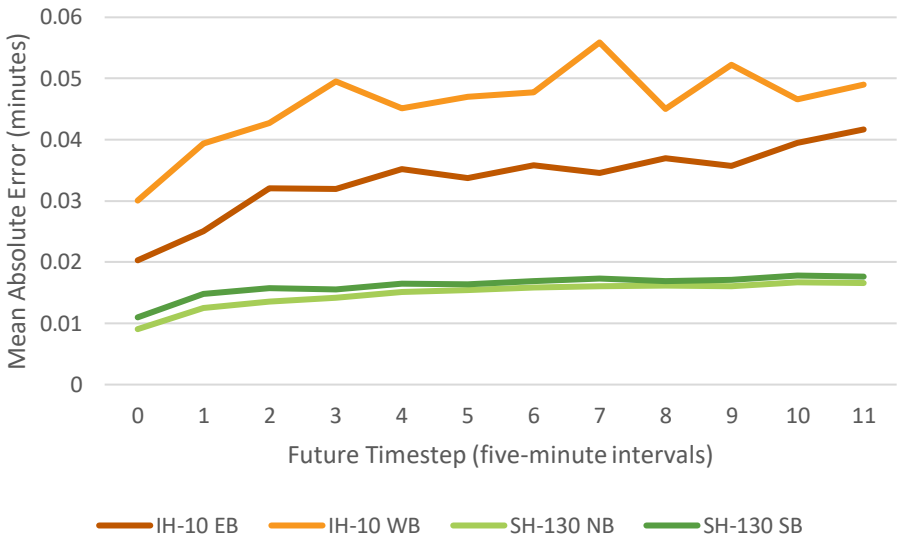

Figure 26: Mean Absolute Error for models trained on a full year's data

R^2 values on the testing dataset were in the same range as those observed in the training dataset, indicating relatively good generalizability of the model and a lack of overfitting.

Figure 27 provides an additional illustration of the model performance on the IH-10 eastbound corridor. In this figure, read left-to-right then top-to-bottom, each graph shows the true segment travel times (X axis) compared to the predictions from the models (Y axis) for the testing dataset. Each subsequent graph forecasts travel times a further five-minute timestep into the future. What we generally see is that the model has difficulty predicting conditions of high travel times, most likely caused by non-recurring congestion such as is caused by accidents. We also see a tendency to underestimate travel times, possibly due to the lack of volume data in these models to indicate how susceptible each segment is to being oversaturated.

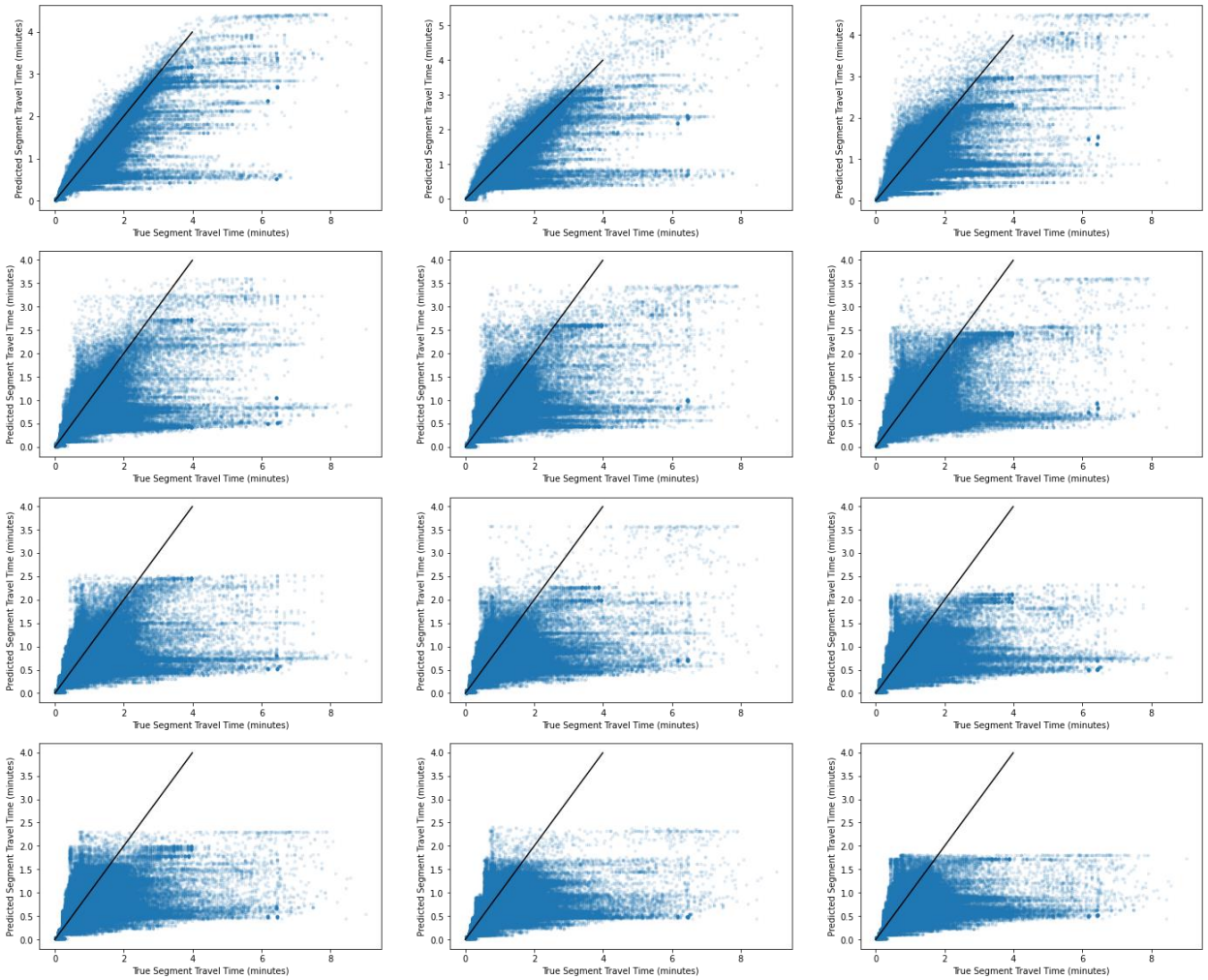

Figure 27: Correlation plot of testing data for IH-10 eastbound forecasts

Figure 28 through Figure 30 compare the performance of models trained with and without volume data using R^2, MSE, and MAE. All other experimental parameters being held constant, we see somewhat similar performance between the models in terms of predictive power. Results show a decline in predictive accuracy as we look further ahead into the future for both models. We also see that, in most cases, further into the future, the models with volume and occupancy data show an improvement over the model without the added data.

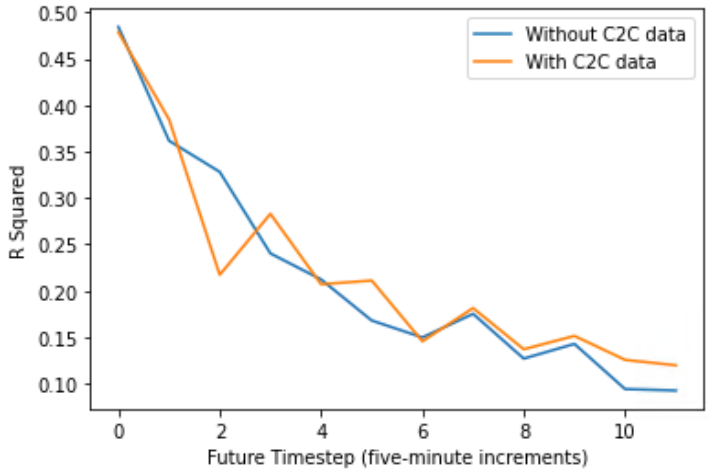

Figure 28: Testing R^2 values for models with (orange) and without (blue) C2C data

In the nearer-term predictions, this is not the case. We suspect this may be caused by a combination of factors. First, while the model is being trained using additional data sources, no compensation was made in model or training design to account for this data. Generally, it is advisable to increase the RNN size and/or prolong training when additional data dimensions are provided. As a result, the model without C2C data may have been able to discern more clearly the importance of the individual data points it received due to not expending computation time determining the importance of the added points as well. Second, in this experiment, we train the model only on data ranging from December 2021 to May 2021, due to limited data availability from the Lonestar C2C data source. This in turn may cause a severe impact on predictive power due to the seasonality of traveler behavior and lack of sufficient examples to train the model.

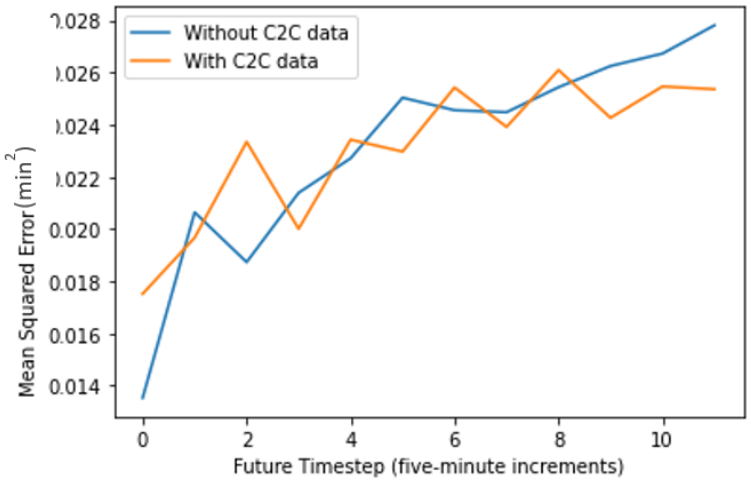

Figure 29: Testing MSE values for models with (orange) and without (blue) C2C data

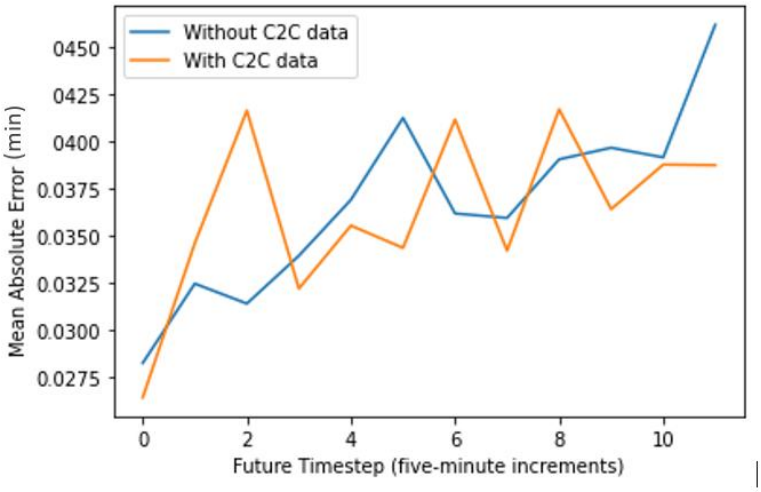

Figure 30: Testing MAE values for models with (orange) and without (blue) C2C data

Future Work

We have refined the code base used for model training and developed tools that may be more easily extended to the analysis of new sites and datasets. The results obtained using a simple RNN design are encouraging, given the low MAE values. However, the predictive power of the models may be improved in new iterations of the training process.

A series of next steps will be undertaken to further refine the models and develop better predictions for the focus corridors. We are encouraged by efforts to procure more C2C data for several corridors which will allow for more robust training of models using volume data. We also aim to provide these models with more training time as well as test a variety of alternate RNN structures. By adding training time and restructuring the neural networks, we anticipate an improved performance can be obtained, especially when sufficient data is provided for the model to estimate seasonal fluctuations.

We further aim to test this procedure on alternate corridors including Interstate Highway 35 in Austin and State Highway Loop 375 in El Paso. We aim to develop corridor-level travel time estimates for these and compare to logs of travel times announced to drivers on roadside DMSs. Finally, we may further develop our time series model to compare against our RNN model performance.

6. Reinforcement Learning Environments for Signal Control

This chapter investigates the application of Reinforcement Learning (RL) techniques to the traffic management scenario. In particular, the goal of integrated corridor management (ICM) is in focus, in hopes of enabling advanced control techniques that can respond to observed roadway conditions faster and more effectively than current practices which focus primarily on human design choices. Given the myriad data sources available to transportation management centers (TMCs), we assume that a plausibly accurate representation of a roadway network's congestion conditions can be developed. Furthermore, standardized mechanisms of traffic control exist, including the National Transportation Communications for Intelligent Transportation Systems Protocol (NTCIP), thus providing a unified framework for influencing roadway traffic levels.

We aim to leverage the available technology to develop an RL agent that can observe roadway conditions through the data sources identified in other sections of this project and respond appropriately once trained in a simulated environment. This arrangement allows for safe testing of potentially dangerous control policies in a sandboxed environment as the agent learns to function properly. We next discuss the common components of a RL agent model, including how each component ties to the experiment at hand. In the next section, we discuss the initial proof-of-concept model that was developed, including the scenario generation technique and development of a training environment. In addition to the results from this model, we further detail in a later section a series of alterations made to the simulated environment to decrease the artificiality associated with the training environment, as well as the implications of those changes. We conclude this chapter with a discussion of further steps we aim to undertake in improving the RL agent.

Initial RL Model

To prove the applicability of the RL mindset to the traffic control problem, a toy model was developed which gave an RL agent control over an intersection's operations. This section details the resources available to the RL agent, the training environment to which it was connected, and the results of the initial experiments therefrom. In the next section, we detail improvements made beyond the initial design and the implications these modifications had for agent performance.

Scenario Development

The initial test scenario models a pair of frontage roads along a freeway, as well as a cross street which intersects the frontage roads at a pair of signalized intersections. An example illustration is shown in Figure 31. Traffic flow was modeled using the cell transmission model (CTM), a micro-to-mesoscopic method for modeling traffic density using discretized sections of each modeled roadway. While CTM does not perfectly model traffic behavior, it provides a solid means for describing vehicle trajectories in a dynamic environment. In each training scenario, traffic was randomly generated on the roadways in the network, drawn from a normal distribution with mean conditions providing oversaturated demand levels for the intersections. Furthermore, a characteristic of each scenario was that the distributions for the frontage roads had higher mean volume than the distributions of the side street.

Every five simulated minutes of traffic, an updated state description is provided to the agent, detailing each link's CTM density[****] adjacent to the intersection, and the agent is then afforded an opportunity to modify its control scheme. The actuators available to the agent control what portion of each five-minute increment is dedicated to serving traffic from the frontage roads vs. traffic on the cross street, with no constraints on the time allocations. While these scenarios do not encompass the full spectrum of traffic conditions, nor do they provide a realistic set of policy constraints, the experiments conducted on these scenarios nonetheless provide strong takeaways as to further design considerations which we shall discuss later.

[****] Density on a roadway uniquely describes the traffic conditions present, so this provides an excellent representation of a given link's congestion level.

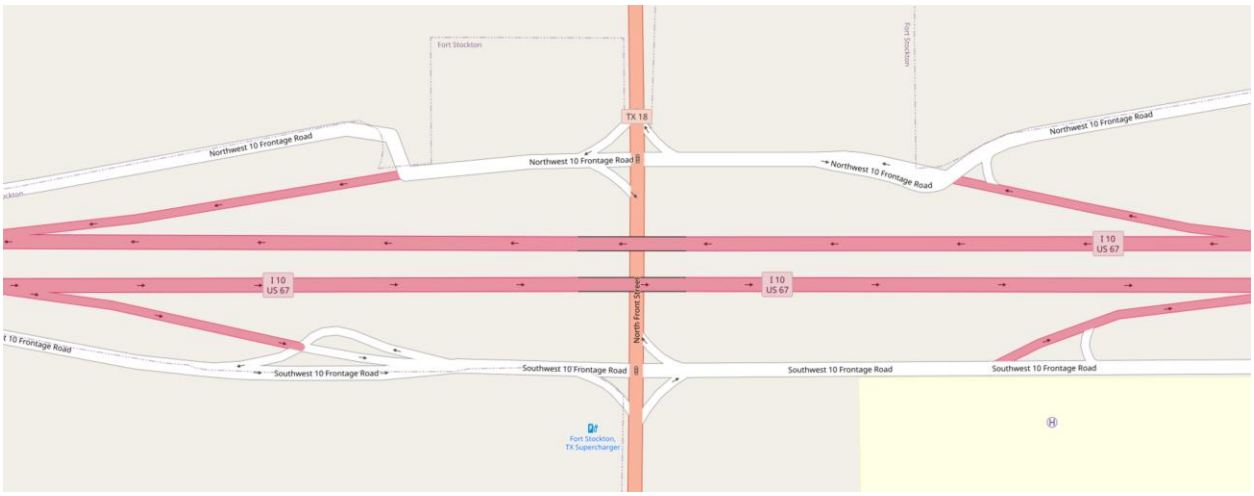

Figure 31: Example scenario illustration

Training Environment Development

To avail ourselves of common and well-researched RL algorithms, an interface was developed which allows our experiments to be conducted in the OpenAI framework. By adapting our scenarios to fit this interface, we can utilize prepackaged and validated algorithms, known as the *stable baselines*, in training an RL agent. Since our state space representation (i.e., the roadway densities) and the actuators available to the agent are both continuous in nature, we selected the soft actor-critic (SAC) learning algorithm as our performance and learning algorithm.

For our evaluation standard, we selected total intersection delay to measure the agent's performance. While such a metric would not be terribly practicable in real-world trials due to the difficulty in measuring exactly, it serves as a good intuitive goal for our agent to minimize the amount of delay it causes and is validated frequently in the literature. In the next subsection, we discuss why this goal, while superficially agreeable, induces a policy that is far from desirable in practice.

Outcomes and Discussion

Training of the initial RL agent developed a consistent and definitive strategy which the agent deemed optimal given its reward function – to simply allocate all time to the (higher-volume) frontage roads, and none to the cross street. The reasoning behind this is easy to see if we dig into it: given that all the agent cares about is total delay, it sees no reason to provide preferential treatment to the vehicles on the side street. The agent can minimize delay by simply focusing on which roadway has higher volume – the palpable unfairness of never serving the crossing traffic is lost on the agent.

These outcomes clearly illustrate the criticality of care in designing a reward mechanism. While an evaluation function such as total delay may seem, at a high level, like an obvious choice, the repercussions of such a policy are often severe and highly undesirable. This is due to the principle that an RL agent will, after sufficient training, develop a strategy to perform its task exceedingly well, subject to whatever may define performing the task "well." Equity considerations are necessary in the design of proper reward functions. In the next section, we discuss a variety of improvements which may alleviate the non-egalitarian difficulties imposed by a pure reward-driven agent.

Further RL Modeling

As described in the prior section, a variety of limitations exist on the initial model developed as proof of concept. This includes unrealistic state representation and actuation capabilities, as well as a reward function that provides perverse incentives leading to policies with serious equity concerns. In this section, we detail further steps undertaken to refine the RL agent environment to mitigate some of these shortcomings. In making these improvements, we transition our learning environment away from one which models traffic using the CTM and towards a state-of-the-art traffic simulation environment. We therefore developed an alternate environment using PTV VISSIM (shown in Figure 32) which models a singular intersection, but which provides far more realistic vehicle dynamics, control methods, and state measurements.

Figure 32: PTV VISSIM microsimulation software

State Improvements

In our initial model, the agent was provided a (near-)perfect representation of the roadway network's state via the roadway density adjacent to the intersections. However, an entire field of study exists which tries to quantify the density of a roadway, with levels of success which are far from perfect. Nonetheless, we can develop approximations for the state representation which depend solely on technology which is currently available to practitioners.

In our VISSIM environment, we have access to a variety of measurements which reflect real-world technological capabilities for quantifying traffic conditions. For our refinement of the model, we selected three key metrics to represent the state space – the number of vehicles on each roadway, plus the roadway's arithmetic and harmonic mean speeds. While the benefits of these metrics are apparent in their reflection of present technological capabilities, they also create a degree of detriment in our learning experiments. Because there are now three metrics for the state space, its dimension is increased, leading to a much larger space which may need to be explored by the RL agent. Further work will explore the viability and necessity of multiple state representation metrics in these scenarios.

Action Improvements

While the concept of determining how to modify traffic conditions via direct timeshare-allocation may be enticing and is somewhat common in the literature, this is entirely outside the realm of reality as defined by the relevant standards from NTCIP. Furthermore, as our experiments showed, such a level of authoritative control over intersection operations can lead to policies that would be intolerable to the traveling public, leading to unimaginable amounts of road rage. To account for this, we include in our next environment iteration a virtual Econolite ASC/3 traffic signal controller (TSC) which conforms to the relevant NTCIP standards. A screenshot of the emulated controller is shown in Figure 33.

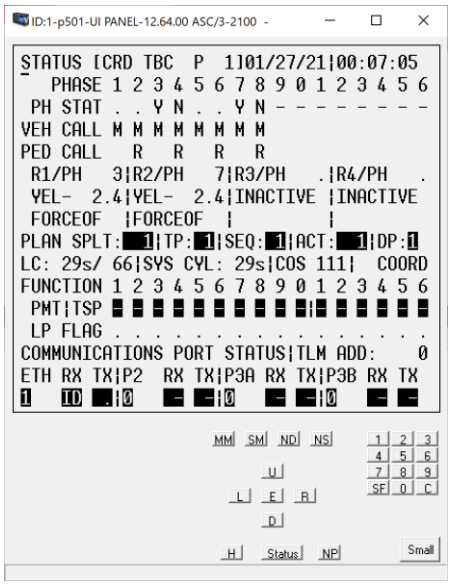

Figure 33: Econolite ASC/3 virtual traffic signal controller

We further provide for our agent a series of discrete actions which can be taken under the NCTIP constraints to modify the control schema of the TSC. Specifically, we aim to modify the cycle length of the signal, as well as the splits allocating shares of a cycle to different directions of travel. The former of these was successfully implemented, while the latter remains in active development. This is because, while cycle length is a single parameter, the splits associated with a standard eight-phase traffic signal timing plan in fact require four parameters, thus increasing the control complexity greatly. Future work will explore the value of these and other control mechanisms.

Reward Improvements

As described in the prior chapters, the reward mechanism which is used in training is of the utmost importance to the RL agent's success, and a poorly designed function can create unintended consequences which will be intolerable in the real world. The experiments described in the previous chapter showed clearly that total delay is not suitable as a metric, as it does not treat opposing traffic fairly, relative to busier roadways.

To account for this, we identify a few ways to address this inequity in designing a reward function. Consider the adage that "time is money" – under this lens, it makes sense that delay caused by traffic can be considered as a negative income. This, in turn, allows us to incorporate traditional income inequality metrics in our reward function. Some common metrics include the Hoover index, the Thiel index, and the Gini coefficient. Of these, the latter has an apparent advantage in its reflection of the entirety of traveler

delay equality, and it provides a coefficient between zero and one which can be used as a scaling factor for other metrics such as delay. However, the Gini coefficient has drawbacks in that it encompasses knowledge of an entire population, which would not be available in the transportation environment. Nonetheless, approximations can be developed which may provide practical application to such experiments.

Results and Discussion

In implementing the changes discussed in this section, a mixed bag of successes and failures was attained. On the one hand, utilizing a state-of-the-art simulation environment provided much faster simulations of environment scenarios, and the data much more closely reflected that which is available currently to practitioners. On the other hand, however, we see an increase in the theoretical complexity of the problem which indicates a much slower learning process may arise.

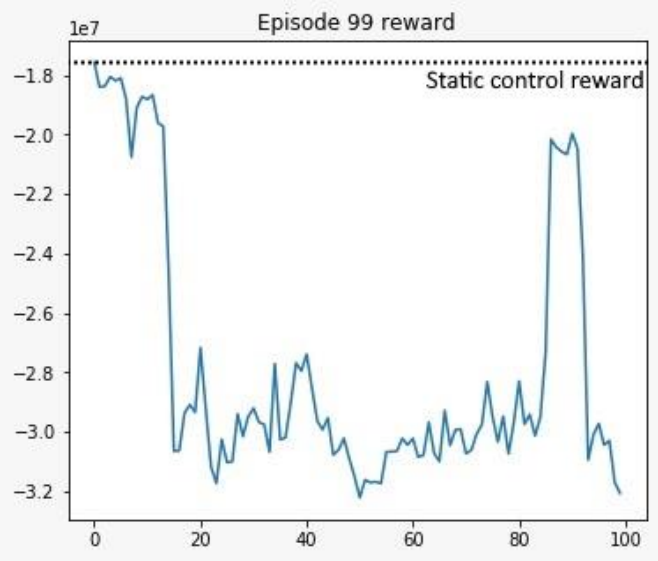

Figure 34: Reward function for a 100-episode trial

In experiments to date, however, this has yet to be seen, as, while in the prior section, a clear-cut optimal policy was developed, no such determination has been made in the modified environment. Recent investigation indicates this is due to a misconfiguration of the new reward mechanism, further highlighting the essential nature of a well-engineered performance measurement component. As a result of this problem, we see the RL agent unable to determine one policy as being better from any other, leading to a random exploration of the action space with no development of an intelligent policy, as illustrated in Figure 34. This glitch can be rectified in further work, which should then allow for cohesive policy learning by the RL agent.

Next Steps

As alluded to, a variety of weaknesses exist which can be addressed in future experiments. Most critical of these is the reward function. The performance evaluation criterion is essential to any RL agent, and the experiments performed to date validate this. By engineering a better reward function, we may be able to provide more meaningful feedback to enable an RL agent to make informed decisions, leading to better performance for the network.

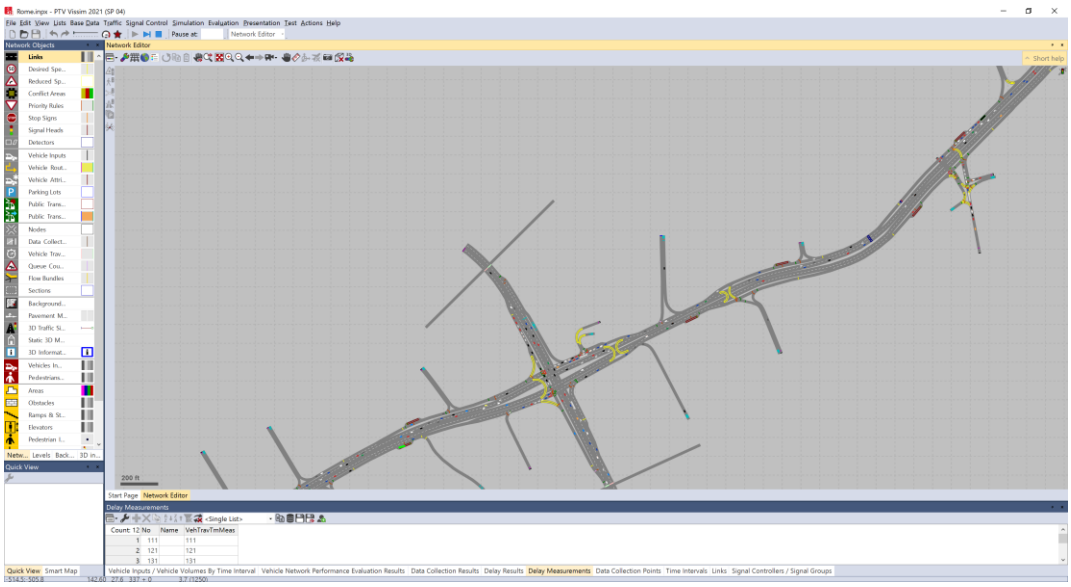

Figure 35: VISSIM demonstration corridor (Rome)

In addition, avenues for further exploration include modified state representations, additional traffic control mechanisms, and a wider variety of traffic scenarios which can train the agent to respond to various states of traffic rather than simply oversaturated conditions. By investigating these further, we aim to develop a versatile, efficient RL traffic management agent that may eventually perform on-line traffic management while simulating scenarios off-line to determine the best policy.

Finally, we aim to investigate the scalability of various learning algorithms in the transportation context, especially as the size of the action space is increased. An example network is shown in Figure 35 which models a much larger area and can provide an expanded control set for an agent to learn. This in turn may impact the rate at which agents train significantly.

7. Concluding Remarks

The guiding principle of this dissertation has been one which aims to remove the barriers to innovation between the transportation modeling and operations sectors. Despite strong independent research areas that improve these two public works areas, they remain siloed to an unpleasant extent. In daily operations, transportation systems managers have access to myriad data sources that describe in microscopic detail the state of a roadway network, yet the applications of these data sources towards improving travel demand models is lacking. Additionally, while cities must build travel demand models for long-range project planning efforts, their integration into daily operations is an area with strong potential for expansion. To extend these bodies of research, this dissertation identifies multiple practical contributions that improve the demand model calibration and traffic signal optimization capabilities of practitioners. It is our hope that these advancements will provide an exciting path forward to better synergize these two highly interrelated concepts.

After an overview of the current state of the transportation engineering as it pertains to our subject matter, we begin by detailing in Chapter 2 the largest *engineering* contribution of this work – the extensive development of *wrap*, a cross-platform, free-and-open-source travel demand modeling software that is efficient in speed and storage space and is scalable to even the largest networks in usage today. We present a detailed description of the software packages we have developed, including design choices that improve the performance of the static traffic assignment algorithm we have implemented based on Algorithm B. The employment of warm starting is an effective means of not only providing an approximate solution to the traffic assignment problem when roadway network characteristics change, but also allow for quick re-evaluation of performance subject to changes in travel demand. This software engineering feat is foundational to the scientific contributions we present as well, as our experiments depend on both these benefits for operational expediency.

In Chapter 3, we present the first major scientific contribution of this dissertation – a method to calibrate the trip generation rates utilized in the earliest stages of a travel demand model by comparing against samples of measurements of roadway volumes. Through this work, we illustrate a value proposition for travel demand modelers that has been overlooked in current research, one which economizes on the availability of roadway count data from transportation operations sources such as those collected by detectors connected to traffic signal controllers. While it is common practice in industry to calibrate various travel demand model parameters based on network link counts, the bulk of scientific literature

on this process focuses on calibrating the parameters of *choice models* rather than the basic trip generation rates. Furthermore, there is a large body of work investigating tuning origin-destination demand profiles using link counts, but a direct matrix manipulation approach is lacking in behavioral explainability that comes with tuning parameters in earlier stages in the travel demand modeling process. Additionally, while in current practice generation rates are calibrated at a global level through site- and purpose-specific studies, we find no evidence of the use of calibration techniques to vary these rates for specific models to match traffic flows across the entire network. Furthermore, given that there are myriad privacy concerns related to attaining a fully representative travel demand model in a real-world study, we demonstrate the effectiveness of our method on matching the "true" trip generation behavior of a synthetic population to avoid these issues. Our results show that an approach utilizing quasi-Newton methods and the L-BFGS-B search algorithm can accurately tune trip generation rates to match observed flows, providing a strong use case for model calibration using data from operations sensor sources.

The second key contribution of this dissertation is given in Chapter 4, in which we extend the current body of research on pressure-based traffic signal timing optimization by identifying a pressure function that more accurately reflects the delay posed to travelers caused by traffic signals. In practice, signal timings are set based on the flows forecasted from travel demand models or field measurements; then as time progresses, the traveling public will alter their route choices to seek shortcuts that may arise due to these new timings. While travelers make this behavioral shift over a period of weeks, the responsive process of re-timing the signals occurs on the order of years, thus creating an imbalance. The iterative equilibrium solution approach posed by Smith was implemented in the *wrap* software package, alongside basic ring-barrier traffic signal timing functionality, to balance the value of changes in traffic signal timings with the value of the route choice decisions travelers will make in the long run because of these changes. By "playing the game" to its logical extreme, we are able to short-circuit the drawn-out cycle of practitioners setting signal timings and travelers seeking shortcuts, thus providing improved timing routing choices out of the gate. We show that the novel A3 pressure function we propose is not only a sturdy benchmark metric for comparison of signal delay across multiple equilibria, but that it also provides competitive performance in all studied metrics. We further describe two more pressure functions that, subject to the values of policymakers and desirability of the various performance metrics, can be used to improve routing behavior beyond both the A3 equilibrium and those of the pressure functions common in the literature today. This, in turn, presents a valuable improvement to the state of traffic operations by leaning on the expertise of travel demand modelers.

Next, we address the practical applicability by presenting in Chapter 5 a framework and initial case study efforts which aim to improve corridor travel time predictions using machine learning models based on real-time data commonly available in practice. Utilizing both roadway sensor data from the Texas Department of Transportation as well as commercial probe data from INRIX as data sources, we present findings that show promise in utilizing recurrent neural networks to improve forecasting efforts. This is a valuable tool as we aim to better model link-level traffic dynamics, both in terms of predictive power as well as computational efficiency. While there remain several avenues for further investigation in this area, the potential to calibrate link performance models utilizing the bevy of data is an attractive one that we can further use to marry the operations and planning sectors.

Finally, we describe in Chapter 6 the efforts we have undertaken to implement a formal reinforcement learning environment for traffic signal control. By utilizing market-leading microscopic simulation software, common learning algorithms, and industry-standard control communications protocols, we have developed an extensible and readily implementable means to model the impacts of traffic signal timing policies on network performance in exceedingly fine detail and to optimize the signal timings to maximize a given performance metric. However, as we show, the definition of "optimized" signal timings is subject to various interpretations, and a poor definition of performance can lead to policies which would be considered impractical or downright unethical. These perverse incentives are, therefore, a cautionary tale for practitioners – the signal timing policies that a reinforcement learning agent will develop are capable of impressive performance, but they depend on a carefully considered reward function which must reflect the values of policymakers.

As we have described, the "game" of transportation is one with a variety of players, each with their own distinct motivations. The efforts of practitioners in both the travel demand modeling and transportation systems management are valiant, but we identify in this dissertation specific gaps which we aim to fill. In doing so, we aim to bring these two distinct groups closer together, in hopes that such collaborative efforts will make for a more efficient world.

8. References

Alegre, L. N., Ziemke, T., & Bazzan, A. L. (2021). Using Reinforcement Learning to Control Traffic Signals in a Real-World Scenario: An Approach Based on Linear Function Approximation. *IEEE Transactions on Intelligent Transportation Systems*, 9126-9135.

Antoniou, C., Azevedo, C. L., Lu, L., Pereira, F., & Ben-Akiva, M. (2015). W-SPSA in Practice: Approximation of Weight Matrices and Calibration of Traffic Simulation Models. *Transportation Research Procedia, 7*, 233-253.

Aslani, M., Mesgari, M. S., & Wiering, M. (2017). Adaptive Traffic Signal Control with Actor-Critic Methods in a Real-World Traffic Network with Different Traffic Disruption Events. *Transportation Research Part C: Emerging Technologies, 85*, 732-752.

Aycin, M. F., & Benekohal, R. F. (1999). Comparison of Car-Following Models for Simulation. *Transportation Research Record, 1678*, 116-127.

Bar-Gera, H., & Boyce, D. (2003). Origin-Based Algorithms for Combined Travel Forecasting Models. *Transportation Research Part B: Methodological, 37*(5), 405-422.

Beckmann, M. J., McGuire, C. B., & Winsten, C. B. (1956). *Studies in the Economics of Transportation.* New Haven: Yale University Press.

Bernardin Jr., V. L., Trevino, S., Slater, G., & Gliebe, J. (2015). Simultaneous Travel Model Estimation from Survey Data and Traffic Counts. *Transportation Research Record: Journal of the Transportation Research Board*, 69-76.

Bevis, H. W. (1959). A Model for Predicting Urban Travel Patterns. *Journal of the American Institute of Planners, 25*(2), 87-89.

Bhat, C. R., & Steed, J. L. (2002). A Continuous-Time Model of Departure Time Choice for Urban Shopping Trips. *Transportation Research Part B: Methodological, 36*(3), 207-224. doi:10.1016/S0191-2615(00)00047-3

Bhavsar, P., Safro, I., Bouaynaya, N., Polikar, R., & Dera, D. (2017). Machine Learning in Transportation Data Analytics. In *Data Analytics for Intelligent Transportation Systems* (pp. 283-307). Elsevier.

Bliemer, M. C., Raadsen, M. P., Brederode, L. J., Bell, M. G., & Wismans, L. J. (2014). *A Unified Framework for Traffic Assignment: Deriving Static and Quasi-Dynamic Models Consistent with General First Order Dynamic Traffic Assignment Models.* Institute of Transport and Logistics Studies.

Boyles, S. D., Lownes, N. E., & Unnikrishnan, A. (2022). *Transportation Network Analysis* (0.90 ed., Vol. 1).

Boyles, S. D., Melson, C., Rambha, T., & Duthie, J. C. (2013). Diverge Nodes and Dynamic Traffic Equilibria. *INFORMS Annual Meeting.*

Bricka, S., & Bhat, C. R. (2006). Comparative Analysis of Global Positioning System-Based and Travel Survey-Based Data. *Transportation Research Record, 1972*, 9-20. doi:10.1177/0361198106197200102

Bureau of Public Roads. (1964). *Traffic Assignment Manual.* Washington: United Stated Department of Commerce, Urban Planning Division.

Bwambale, A., Choudhury, C. F., Hess, S., & Iqbal, M. S. (2020). Getting the Best of Both Worlds: A Framework for Combining Disaggregate Travel Survey Data and Aggregate Mobile Phone Data for Trip Generation Modelling. *Transportation*, 2287-2314.

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing, 16*(5).

Cascetta, E. (2009). *Transportation Systems Analysis: Models and Applications.* New York: Springer. doi:10.1007/978-0-387-75857-2

Cascetta, E., & Russo, F. (1997). Calibrating Aggregate Travel Demand Models with Traffic Counts: Estimators and Statistical Performance. *Transportation*, 271-293.

Cascetta, E., Pagliara, F., & Papola, A. (2007). Alternative Approaches to Trip Distribution Modelling: A Retrospective Review and Suggestions for Combining Different Approaches. *Papers in Regional Science, 86*(4), 597-620. doi:10.1111/j.1435-5957.2007.00135.x

Cascetta, E., Papola, A., Marzano, V., Simonelli, F., & Vitiello, I. (2013). Quasi-Dynamic Estimation of O-D Flows from Traffic Counts: Formulation, Statistical Validation and Performance Analysis on Real Data. *Transportation Research Part B: Methodological*, 171-187.

Chen, S., Prakash, A. A., Lima de Azevedo, C., & Ben-Akiva, M. (2020). Formulation and Solution Approach for Calibrating Activity-Based Travel Demand Model-System via Microsimulation. *Transportation Research Part C: Emerging Technologies*.

Christos, M., & James, J. Q. (2020). Unsupervised Deep Learning for GPS-Based Transportation Mode Identification. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC).* Rhodes, Greece.

Chu, T., Wang, J., Codecà, L., & Li, Z. (2019). Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, 1086-1095.

Cipriani, E., Florian, M., Mahut, M., & Nigro, M. (2011). A Gradient Approximation Approach for Adjusting Temporal Origin-Destination Matrices. *Transportation Research Part C: Emerging Technologies*, 270-282.

Dasgupta, S., Rahman, M., Lidbe, A. D., Lu, W., & Jones, S. (2021). *A Transportation Digital-Twin Approach for Adaptive Traffic Control Systems.* arXiv.

Day, C. M., Bullock, D. M., Li, H., Remias, S. M., Hainen, A. M., Freije, R. S., . . . Brennan, T. M. (2014). *Performance Measures for Traffic Signal Systems: An Outcome-Oriented Approach.* West Lafayette, Indiana: Purdue University.

Dial, R. B. (2006). A Path-Based User-Equilibrium Traffic Assignment Algorithm that Obviates Path Storage and Enumeration. *Transportation Research Part B: Methodological, 40*(10), 917-936.

Dias, F. F., Nair, G. S., Ruíz-Juri, N., Bhat, C. R., & Mirzaei, A. (2020). Incorporating Autonomous Vehicles in the Traditional Four-Step Model. *Transportation Research Record, 2674*(7), 348-360. doi:10.1177/0361198120922544

Doorley, R., Alonso, L., Grignard, A., Maciá, N., & Larson, K. (2020). Travel Demand and Traffic Prediction with Cell Phone Data: Calibration by Mathematical Program with Equilibrium Constraints. *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-8). IEEE.

Easa, S. M. (1993). Urban Trip Distribution in Practice I: Conventional Analysis. *Journal of Transportation Engineering, 119*(6), 793-815.

Evans, S. P. (1976). Derivation and Analysis of Some Models for Combining Trip Distribution and Assignment. *Transportation Research, 10*(1), 37-57. doi:10.1016/0041-1647(76)90100-3

Fisk, C. S. (1989). Trip Matrix Estimation from Link Traffic Counts: The Congested Network Case. *Transportation Research Part B: Methodological, 23*(5), 331-336.

Fratar, T. J. (1954). Vehicular Trip Distribution by Successive Approximations. *Traffic Quarterly, 8*(1), 53.

Gace, I., Pevec, D., Vdovic, H., Babic, J., & Podobnik, V. (2021). Driving style Categorisation Based on Unsupervised Learning: A Step towards Sustainable Transportation. *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech).*

Gartner, N. H., & Al-Malik, M. (1996). Combined Model for Signal Control and Route Choice in Urban Traffic Networks. *Transportation Research Record: Journal of the Transportation Research Board*, 27-35.

Genders, W., & Razavi, S. (2018). Evaluating Reinforcement Learning State Representations for Adaptive Traffic Signal Control. *Procedia Computer Science, 130*, 26-33.

Gini, C. (1936). *On the Measure of Concentration with Special Reference to Income and Statistics.* Colorado College Publication.

Gregoire, J., Qian, X., Frazzoli, E., de La Fortelle, A., & Wongpiromsarn, T. (2015). Capacity-Aware Backpressure Traffic Signal Control. *IEEE Transactions on Control of Network Systems, 2*(2), 164-173.

Hajbabaie, A., & Benekohal, R. F. (2015). A Program for Simultaneous Network Signal Timing Optimization and Traffic Assignment. *IEEE Transactions on Intelligent Transportation Systems*, 2573-2586.

Harth, M., Langer, M., & Bogenberger, K. (2021). Automated Calibration of Traffic Demand and Traffic Lights in SUMO Using Real-World Observations. *SUMO Conference Proceedings*, (pp. 133-148).

Haydari, A., & Yilmaz, Y. (2020). Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems, 23*(1), 11-32.

Institute of Transportation Engineers. (2021). *Trip Generation Manual.* Washington: Institute of Transportation Engineers.

Jain, N. K., Saini, R. K., & Mittal, P. (2019). A Review on Traffic Monitoring System Techniques. In *Soft Computing: Theories and Applications* (pp. 569-577). Singapore: Springer.

Jaynes, E. T. (1957). Information Theory and Statistical Mechanics. *Physical Review, 106*(4), 620-630. doi:10.1103/PhysRev.106.620

Kővári, B., Szőke, L., Bécsi, T., Aradi, S., & Gáspár, P. (2021). Traffic Signal Control via Reinforcement Learning for Reducing Global Vehicle Emission. *Sustainability, 13*(20).

La, P., & Bhatnagar, S. (2010). Reinforcement Learning With Function Approximation for Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, 412-421.

Levin, M. W., & Boyles, S. D. (2017). Pressure-Based Policies for Reservation-Based Intersection Control. *Transportation Research Board 96th Annual Meeting.* Washington.

Li, Z.-C., & Ge, X.-Y. (2014). Traffic Signal Timing Problems with Environmental and Equity Considerations. *Journal of Advanced Transportation, 48*(8), 1066-1086.

Liu, H. X., Wu, X., Ma, W., & Hu, H. (2009). Real-Time Queue Length Estimation for Congested Signalized Intersections. *Transportation Research Part C: Emerging Technologies, 17*(4), 412-427.

Loce, R. P., Bernal, E. A., Wu, W., & Bala, R. (2013). Computer Vision in Roadway Transportation Systems: A Survey. *Journal of Electronic Imaging, 22*(4).

Marzano, V., Papola, A., & Simonelli, F. (2008). Effectiveness of Origin-Destination Matrix Correction Procedure Using Traffic Counts. *Transportation Research Record: Journal of the Transportation Research Board*, 57-66.

Model Development Group, NCTCOG Transportation Department. (2007). *Dallas-Fort Worth Regional Travel Model (DFWRTM): Model Description.* Arlington, Texas: North Central Texas Council of Governments.

Mukherjee, J., & Kadali, R. (2022). A Comprehensive Review of Trip Generation Models Based on Land Use Characteristics. *Transportation Research, Part D: Transport and Environment, 109*. doi:10.1016/j.trd.2022.103340

National Transportation Communications for Intelligent Transportation System Protocol. (n.d.). *Published Standards*. (NTCIP Joint Committee) Retrieved August 01, 2022, from https://www.ntcip.org/document-numbers-and-status/

Nguyen, S. (1976). A Unified Approach to Equilibrium Methods for Traffic Assignment. *Traffic Equilibrium Methods.* doi:10.1007/978-3-642-48123-9_8

Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Hossein Abad, Z. S., . . . Far, B. (2022). Reinforcement Learning in Urban Network Traffic Signal Control: A Systematic Literature Review. *Expert Systems with Applications*.

Office of the Law Revision Council. (2021). *49 U.S.C § 5303 - Metropolitan transportation planning.* Retrieved 08 01, 2022, from United States Code: https://uscode.house.gov/view.xhtml?path=/prelim@title49/subtitle3/chapter53&edition=prelim

Office of the Law Revision Counsel. (2021). *49 U.S.C § 5304 - Statewide and nonmetropolitan transportation planning.* Retrieved 08 01, 2022, from United States Code: https://uscode.house.gov/view.xhtml?path=/prelim@title49/subtitle3/chapter53&edition=prelim

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., & Wang, Y. (2003). Review of Road Traffic Control Strategies. *Proceedings of the IEEE.*

Papathanasopoulou, V., & Antoniou, C. (2015). Towards Data-Driven Car-Following Models. *Transportation Research Part C: Emerging Technologies, 55*, 496-509.

Patriksson, M. (2015). *The Traffic Assignment Problem: Models & Methods.* Mineola, New York: Dover Publications.

Pradhan, A., & Kockelman, K. M. (2002). Uncertainty Propagation in an Integrated Land Use-Transportation Modeling Framework: Output Variation via UrbanSim. *Transportation Research Record, 1805*(1), 128-135. doi:10.3141/1805-15

Robillard, P. (1975). Estimating the OD matrix from observed link volumes. *Transportation Research, 9*(2-3), 123-128.

Shi, Y., Gu, Z., Yang, X., Li, Y., & Chu, Z. (2023). An Adaptive Route Guidance Model Considering the Effect of Traffic Signals Based on Deep Reinforcement Learning. *IEEE Intelligent Transportation Systems Magazine*, 2-15.

Smith, M. J. (1979). A Local Traffic Control Policy Which Automatically Maximises the Overall Travel Capacity of an Urban Road Network. *Proceedings of the Symposium on Traffic Control Systems* (pp. 11-32). Berkeley: Traffic Engineering and Control.

Smith, M. J. (2015). Traffic Signal Control and Route Choice: A New Assignment and Control Model which Designs Signal Timings. *Transportation Research Part C: Emerging Technologies*, 451-473.

Smith, M. J., & Liu, R. (2015). Route Choice and Traffic Signal Control: A Study of the Stability and Instability of a New Dynamical Model of Route Choice and Traffic Signal Control. *Transportation Research Part B: Methodological*, 123-145.

Smith, M. J., & van Vuren, T. (1993). Traffic Equilibrium with Responsive Traffic Control. *Transportation Science*, 87-207.

Smith, M. J., Liu, R., & Mounce, R. (2015). Traffic Control and Route Choice: Capacity Maximisation and Stability. *Transportation Research Part B: Methodological*, 863-885.

Smith, M. J., Viti, F., Huang, W., & Mounce, R. (2023). With Spatial Queueing, the P0 Responsive Traffic Signal Control Policy May Fail to Maximise Network Capacity Even if Queue Storage Capacities are Very Large. *Transportation Research Part B: Methodological*.

SPARTA Lab. (2017). wrap: an Urban Transportation Modeling System Written in Java. Austin: The University of Texas at Austin.

Spiess, H. (1990). Conical Volume-Delay Functions. *Transportation Science, 24*(2), 153-158.

Stevanovic, A., Stevanovic, J., Zhang, K., & Batterman, S. (2009). Optimizing Traffic Control to Reduce Fuel Consumption and Vehicular Emissions: Integrated Approach with VISSIM, CMEM, and VISGAOST . *Transportation Research Record, 2128*, 105-113.

Stouffer, S. A. (1940). Intervening Opportunities: A Theory Relating Mobility and Distance. *American Sociological Review, 5*(6), 845-867. doi:10.2307/2084520

Sun, D., Benekohal, R. F., & Waller, S. T. (2006). Bi-level Programming Formulation and Heuristic Solution Approach for Dynamic Traffic Signal Optimization. *Computer-Aided Civil and Infrastructure Engineering*, 321-333.

Tavares, A. R., & Bazzan, A. L. (2012). *Reinforcement Learning for Route Choice in an Abstract Traffic Scenario.* ResearchGate.

Transportation Networks for Research Core Team. (2007). *Transportation Networks for Research*. Retrieved 07 31, 2020, from https://github.com/bstabler/TransportationNetworks

Urbanik, T., Tanaka, A., Lozner, B., Lindstrom, E., Lee, K., Quayle, S., . . . Bullock, D. (2015). *Signal Timing Manual, Second Edition.* Washington: National Academies of Sciences, Engineering, and Medicine.

Van Zuylen, H. J., & Willumsen, L. G. (1980). The Most Likely Trip Matrix Estimated From Traffic. *Transportation Research Part B: Methodological, 14*(3), 281-293.

Walthall, R. D., & Walton, C. M. (2019). *Incorporating Active Transport into the Regional Planning Process to Support First and Last Mile Travel.* Federal Transit Administration, Office of the Assistant Secretary for Research and Technology, UTC Program. Washington: U.S. Department of Transportation. Retrieved 08 01, 2022, from https://sites.utexas.edu/cm2/files/2019/10/Year-2_Walton_Incorporating-active-transport-into-the-regional-planning-process-to-support-first-and-last-mile-travel.pdf

Wang, S., Li, C., Yue, W., & Mao, G. (2019). Network Capacity Maximization Using Route Choice and Signal Control With Multiple OD Pairs. *IEEE Transactions on Intelligent Transportation Systems*, 1595-1611.

Wardrop, J. G. (1952). Some Theoretical Aspects of Road Traffic Research. *Proceedings of the Institute of Civil Engineers.* doi:10.1680/ipeds.1952.11259

Webster, F. V. (1958). *Traffic Signal Settings.* Road Research Lab Tech Papers.

Wei, H., Zheng, G., Gayah, V., & Li, Z. (2021). Recent Advances in Reinforcement Learning for Traffic Signal Control: A Survey of Models and Evaluation. *ACM SIGKDD Explorations Newsletter*, 12-18.

Wei, W., Wu, Q., Wu, J., Shen, J., & Li, T. (2021). Multi-Agent Deep Reinforcement Learning for Traffic Signal Control with Nash Equilibrium. *IEEE 23rd Int Conf on High Performance Computing & Communications* (pp. 1435-1442). Haikou: IEEE.

Willumsen, L. G. (1978). *Estimation of an O-D Matrix from Traffic Counts – A Review.* Leeds: University of Leeds.

Wilson, A. G. (1970). The Use of the Concept of Entropy in System Modelling. *Operations Research Quarterly, 21*(2), 247-265. doi:10.2307/3008157

Yan, S., Zhang, J., Büscher, D., & Burgard, W. (2020). Efficiency and Equity are Both Essential: A Generalized Traffic Signal Controller with Deep Reinforcement Learning. *IEEE/RSJ International Conference on Intelligent Robots and Systems.* Las Vegas.

Yau, K.-L. A., Qadir, J., Khoo, H. L., Ling, M. H., & Komisarczuk, P. (2017). A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control. *ACM Computing Surveys*.

Zhao, Y., & Kockelman, K. M. (2002). The Propagation of Uncertainty Through Travel Demand Models: An Exploratory Analysis. *The Annals of Regional Science, 36*(1), 145-163. doi:10.1007/s001680200072