University of Memphis University of Memphis Digital Commons

Electronic Theses and Dissertations

4-26-2023

Data-driven Methodologies and Applications in Urban Mobility

Huan H. Ngo

Follow this and additional works at: https://digitalcommons.memphis.edu/etd

Recommended Citation

Ngo, Huan H., "Data-driven Methodologies and Applications in Urban Mobility" (2023). *Electronic Theses and Dissertations*. 3133. https://digitalcommons.memphis.edu/etd/3133

This Dissertation is brought to you for free and open access by University of Memphis Digital Commons. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of University of Memphis Digital Commons. For more information, please contact khggerty@memphis.edu.

DATA-DRIVEN METHODOLOGIES AND APPLICATIONS IN URBAN MOBILITY

by

Huan Ngo

A Dissertation

Submitted in Partial Fulfillment of the

Requirement of the Degree of

Doctor of Philosophy

Major: Civil Engineering

The University of Memphis

May 2023

Copyright © Huan Ngo

All rights reserved

I dedicate this dissertation to my family from Viet Nam, for providing me with relentless support, security, and strong belief.

ACKNOWLEDGEMENTS

When eating fruit, remember the who

planted the tree.

Vietnamese Proverb

First, I wholeheartedly give my utmost gratitude to my advisor, Dr. Sabyasachee Mishra. He first introduced me to academic while I was a sophomore still and ignites my spark for exploring the possibility of science. During this journey, I continuously received the necessary guidance, support, encouragement, and patience from Dr. Mishra, all of which contribute greatly to the completion of this dissertation. I see and learn from him the patience, relentlessness, extraordinary work ethic, and his never-quit attitude and these characteristics pull me through the finish line of my Ph.D. journey. I am extremely thankful for the opportunity to work on realworld projects that would then have a significant impact on improving the mobility and safety of the community we serve, and this gives meaning to my Ph.D. work.

Besides my supervisor, I want to give Dr. Mihalis Golias a warm and extended thank you. Dr. Golias has provided me a strong background on optimization from which I rocketed it forward into interest applications in urban mobility. His energetic attitude helps me navigate the ups and downs, especially in the pandemic years. I am extremely grateful for Dr. Charles Camp. I am totally inspired for his passionate in teaching and the course he teaches such as Engineering Analysis 2 and Structural Analysis are filled with hands-on projects and from which I can hone in my writing and presenting skill to the professional level that is expected of a Ph.D. To Dr. Weizi Li, your feedback on my later research, especially in Reinforcement Learning, is proven to be invaluable and from which I am able to head in the right direction and thus I am extremely grateful for. I would also like to thank Dr. Rohan Shah and Dr. Amit Kumar, both of which are the co-author of my research and have help me tremendously in publishing our great ideas.

To fellow lab mates but most importantly friends, Diwas, Ali, Ishant, Avani, Suvin, Dimitris, and Vasilleios, there is no number of words that can express my gratitude. Pursuing a Ph.D. degree at such a young age is a tough ask and, in the field, where experience and seniority play an important role, I have much to learn. My friends help me through the beginning phase and bridges that gap. I am humble by your advice, sharing of experiences, guidance, and much encouragement, empathy, and compassion. I consider my friends as my second family because they have brought much intimacy, joy, and never-ending humor and make me feel less like a foreigner. I always believe that my labmates are the only one who can truly and fully understand the hardship of the Ph.D. journey and for which I am deeply grateful of.

I leave the last paragraph to my family because their contribution is the most important and immeasurable. To my younger brother, your passion in academic, especially in physics, and sequentially your success in university years have re-ignited and rediscovered in me the curiosity in science that I had when I was your age. Thanks to you, I am now as energetic as and ready to take on new challenges again and I am happy that we get to do that together. To my father, your words are limited but your action and love for me are limitless. Without your help, none of this would be possible. To my mother, your love and unwavering support for me are endless. You provide me with security that the world is for me to explore and there will always be a home that I can come back to. This gives me relentless power and from which I am able to walk through the gate of Ph.D. and continuingly and confidently step into new challenge.

iv

PREFACE

This dissertation includes three research papers as the main chapters. The topic and information on these chapters are listed as follows:

Chapter 2: This work is published in Transportation Research Part D: Transport and Environment. The paper is as follows:

Ngo, H., Kumar, A., & Mishra, S. (2020). Optimal positioning of dynamic wireless charging infrastructure in a road network for battery electric vehicles. Transportation Research Part D: Transport and Environment, 85, 102385. <u>https://doi.org/10.1016/j.trd.2020.102385</u>

Chapter 3: This work is published in Network and Spatial Economic.

The paper is as follows:

Ngo, H., and Mishra, S. (2022). Traffic Graph Convolutional Network for Dynamic Urban Travel Speed Estimation. Networks and Spatial Economics.

https://doi.org/10.1007/s11067-022-09582-9

Chapter 4: This work is submitted and under-review in Transportation.

The paper is as follows:

Ngo, H., Mishra, S. (2023). Carpooling and Repositioning Strategy for Mixed Autonomous Electric Taxi using Reinforcement Learning. (Under Review).

Because of this layout, our equations and citations are numbered and referenced within their own section only. On the other hand, chapters and sub-sections, figures, and tables follow the normal convention of numbering and referencing.

ABSTRACT

The world is urbanizing at an unprecedented rate where urbanization goes from 39% in 1980 to 58% in 2019 (World Bank, 2019). This poses more and more transportation demand and pressure on the already at or over-capacity old transport infrastructure, especially in urban areas. Along the same timeline, more data generated as a byproduct of daily activity are being collected via the advancement of the internet of things, and computers are getting more and more powerful. These are shown by the statistics such as 90% of the world's data is generated within the last two years and IBM's computer is now processing at the speed of 120,000 GPS points per second. Thus, this dissertation discusses the challenges and opportunities arising from the growing demand for urban mobility, particularly in cities with outdated infrastructure, and how to capitalize on the unprecedented growth in data in solving these problems by ways of data-driven transportation-specific methodologies. The dissertation identifies three primary challenges and/or opportunities, which are (1) optimally locating dynamic wireless charging to promote the adoption of electric vehicles, (2) predicting dynamic traffic state using an enormously large dataset of taxi trips, and (3) improving the ride-hailing system with carpooling, smart dispatching, and preemptive repositioning. The dissertation presents potential solutions/methodologies that have become available only recently thanks to the extraordinary growth of data and computers with explosive power, and these methodologies are (1) bi-level optimization planning frameworks for locating dynamic wireless charging facilities, (2) Traffic Graph Convolutional Network for dynamic urban traffic state estimation, and (3) Graph Matching and Reinforcement Learning for the operation and management of mixed autonomous electric taxi fleets. These methodologies are then carefully calibrated, methodically scrutinized under various performance metrics and procedures, and validated with previous research and

ground truth data, which is gathered directly from the real world. In order to bridge the gap between scientific discoveries and practical applications, the three methodologies are applied to the case study of (1) Montgomery County, MD, (2) the City of New York, and (3) the City of Chicago and from which, real-world implementation are suggested. This dissertation's contribution via the provided methodologies, along with the continual increase in data, have the potential to significantly benefit urban mobility and work toward a sustainable transportation system.

TABLE OF CONTENTS

Acknowledgements	iii
Preface	v
Abstract	vi
Table of Contents	viii
List of Tables	xi
List of Figures	xii
1. Introduction	
2. Optimal Positioning of Dynamice Wireless Charging Infrastructure in	n a Road Network
for Battery Electric Vehicle	5
2.1. Overview	6
2.2. Literature Review	9
2.2.1. Determination of BEV Charging Locations	9
2.2.2. BEV Driver Behavior in a Road Network with DWC Facilities	12
2.2.3. Contribution and Significance of this Study	13
2.3. Methodology	15
2.3.1. Modeling Approach	15
2.3.2. Upper-Level of Government Agency Decision Making	20
2.3.3. Lower-Level Network User Equilibrium	25
2.3.4. Solution Algorithm	
2.4. Numerical Experiments	
2.4.1. Small Test Network	

	2.4.2. Numerical Results and Insights	35
	2.4.3. Budget Sensitivity Analysis	39
	2.5. Case Study	43
	2.5.1. Montgomery County Network	43
	2.5.2. Equity in Transportation Funding	44
	2.5.3. DWC Implementation Plan for the Study Area	45
	2.6. Conclusion	48
3	. Traffic Graph Convolutional Network for Dynamic Urban Travel Speed Estimation	ı 63
	3.1. Overview	64
	3.2. Literature Review	67
	3.3. Methodology	73
	3.3.1. Step 1: Path Choice Prediction (PCP)	76
	3.3.2. Step 2: Partial Link Travel Time Prediction (PLTT)	78
	3.3.3. Step 3: Dynamic Urban Link Travel Speed Estimation	81
	3.3.4. Encoding link travel speed as node feature	81
	3.3.5. Modified Directional Adjacency Matrix	83
	3.3.6. Traffic Graph Convolution Network (TGCN)	85
	3.3.7. TGCN _{1stm} for capturing Temporal Dependencies	88
	3.3.8. Computational Graph	91
	3.4. New York City Case Study	93
	3.4.1. New York Taxi Dataset	93
	3.4.2. Results	96
	3.5. Model Evaluation and Validation	102
	3.5.1. Model Evaluation	102
	3.5.2. Model Validation	109

3.6. Conclusion and Future Work	116
4. Dynamic Carpooling and Repositioning for Mixed Autonomous Electric	Taxi Fleet based
on Reinforcement Learning	130
4.1. Overview	131
4.2. Literature Review	132
4.2.1. Operation Research on Autonomous Taxi	132
4.2.2. Research Gap and Contributions	134
4.3. Methodology	
4.3.1. Problem Settings	
4.3.2. Model Architecture	140
4.3.3. Dispatching and Repositioning Process	142
4.3.4. Solution Algorithm	
4.3.5. Baseline Model	164
4.4. Case Study	167
4.4.1. Simulated Environment and Data	167
4.4.2. Training Performance	171
4.5. Model Performance and Comparison	177
4.5.1. Performance Metrics Comparison	177
4.5.2. Optimal Carpooling Trips Generation	179
4.5.3. Repositioning	
4.6. Day in the Life of an Autonomous Electric Taxi	
4.7. Conclusion	
5. Conclusion	

LIST OF TABLES

Table 2.1 Past Studies on BEV Charging Infrastructure	10
Table 2.2. Link Properties of Test Network	
Table 2.3. Benefit from DWC implementation at Different Route Choice Scenario	
Table 2.4 Result of Sensitivity Analysis with Respect to Budget	39
Table 3.1. Node-Link Embedding Principal	83
Table 3.2. Evaluation of Step 1 Path Choice Prediction	105
Table 3.3. Validation between Observed and Predicted Trip Travel Time	111
Table 4.1. Summary of Literature on Operational Strategy for Autonomous Taxi	137
Table 4.2. Model Training Time	174
Table 4.3. Comparison Between Manual Allocation and Autonomous Taxi	179

LIST OF FIGURES

Figure 2.1 Schematic representation of the interaction between upper and lower level
Figure 2.2 Modified ConstrLMSRS Algorithm Flowchart
Figure 2.3. Small 18 Link Test Network
Figure 2.4 TSTT and TSNEC convergence with iterations
Figure 2.5 Values of TSTT and TSNEC via RBF and Traffic Assignment Approaches
Figure 2.6 DWC Plan under Different Budgets
Figure 2.7 Montgomery County Transportation Network
Figure 2.8 Transportation Policy Region by Funding Level
Figure 2.9 Final DWC Implementation Strategy for Montgomery County
Figure 3.1 Schematic Representation of The Sequential Three Steps Framework
Figure 3.2. Data Mapping
Figure 3.3. Visualization of Link Travel Speed Embedding
Figure 3.4. Node Message Passing
Figure 3.5. Convolution on Graph
Figure 3.6. TGCN _{lstm} Architecture
Figure 3.7. Computational Graph
Figure 3.8. Descriptive Statistics of the Dataset
Figure 3.9. Heatmap Showing Input and Output Speed of Selected Links
Figure 3.10. Output Link Speed of Major and Minor Links at Three Time Periods 101
Figure 3.11. Absolute Error of Step 1 and Step 2 at Selected Time Period
Figure 3.12. Step 2's Mean Absolute Percentage Error by Spatial Regions

Figure 3.13. Evaluation of Step 3: RMSE by Neighborhood and Time Period	. 108
Figure 3.14. Absolute Error in 6 Instances	. 110
Figure 3.15. Speed Camera Location	. 112
Figure 3.16. Boxplot of Absolute Error between Framework's Result and Ground Truth	. 114
Figure 3.17. Framework's MAPE and MAE by Time Period	. 115
Figure 4.1. Schematic View of Model Architecture	. 141
Figure 4.2. Taxi Request Shareability Graph	. 147
Figure 4.3. Equivalent Bipartite Graph	. 150
Figure 4.4. Policy Function $at = \pi(st, \theta *)$. 161
Figure 4.5. Baseline Model Architecture	. 164
Figure 4.6. Chicago Network	. 168
Figure 4.7. Anaheim Simulated Demand	. 170
Figure 4.8. Total Reward by Episode	. 173
Figure 4.9. Total Weekly Performance Metrics vs Dispatching and Repositioning	. 175
Figure 4.10. Comparison Between Baseline Model and Autonomous Taxi over 50 episodes	. 178
Figure 4.11. Illustration of Process 1 for Combining Demands into Trips	. 181
Figure 4.12. Total Number of Dispatch 30 minutes before Morning Peak Hour	. 183
Figure 4.13. Pickup density between two frameworks	. 184
Figure 4.14. Major Repositioning Route	. 186
Figure 4.15. A Typical Vehicle Trajectory in a Day	. 189
Figure 4.16. Taxi Status Throughout the Day	. 191
Figure 4.17. Fleet Usage Throughout the Day	. 193

This page is intentionally left blank.

1. INTRODUCTION

Urbanization is arising rapidly around the world in the 21st century, as shown by the increase in the percentage of urban population from 39% in 1980 to 58% in 2019 (World Bank, 2019). This speedy urbanization will introduce new mobility demands and place additional pressure on transportation systems, especially in dense urban areas with at-capacity and outdated infrastructure that cannot keep up with current demand. Along with this swift urbanization, there is an unparalleled growth in the amout of data generated just through daily activities and transactions. According to a Ralph (2013), 90% of the world's data at that time was generated during 2011-2012 alone. This unlocks the potential of methodologies in optimization and datadriven models such as deep learning, which rely on massive amounts of data and uncover hidden patterns via the adjustment of model weights and biases. The extensive and vast source of data serves as a wealth of knowledge from which deep learning models are able to learn and improve their accuracy to the point of matching or surpassing human capabilities. In addition, continuous developments in computer's processing power have made it feasible to process such large data, for example IBM's Infosphere processing up to 120,000 GPS points per second. We have identified three primary challenges or opportunities arising from the growing demand for urban mobility and suggested potential methodologies that have only recently become available thanks to this extraordinary growth of data and computing power.

Challenge in Consumer Electric Vehicles Adoption and How Strategic Positioning of Dynamic Wireless Charging Can Promote such Adoption.

The transportation sector is a major contributor to greenhouse gas emissions due to the process of burning fossil fuel of Internal Combustion Engine Vehicles (ICEV) and dense urban areas has experienced even more pollution due to the concentration of these vehicles. Battery Electric Vehicles (EV) offer a cleaner and more efficient transportation alteranative, but these EVs face challenges such as longer recharging times and limited driving range. Induction-based dynamic wireless charging (DWC) has been developed and introduced recently as an alternative to conventional plug-in charging methods. DWC can be installed under the road and allows EVs to be dynamically charged while driving above it. Thus, DWC directly addresses the two previously mentioned disadvantages of longer recharging times and limited driving range. Proper implementation of DWC will alleviate range anxiety among EV drivers and ultimately increases EV adoption rate and help them penetrate into the current car industry. This research aims to develop a planning framework for locating DWC facilities, taking into account system-level network costs, individual travel patterns, range augmentation, resource distribution, and budget availability. By doing so, DWC facilities can be effectively integrated into the transportation infrastructure and promote the use of BEVs and electric autonomous vehicles.

Opportunity in Leveraging Enormous Data to Predict Link-level Network Traffic

Cities are looking for new ways to improve urban mobility, either by building more road network infrastructure (e.g., adding more lanes and highways) or using Intelligent Transportation Systems (ITS) that leverage state-of-the-art technology and data to improve the urban mobility without investing in new infrastructure. While cities can request funding to build more infrastructure, the effectiveness of this investment is often questionable, as new infrastructure does not always result in improved mobility. In contrast, ITS has gained tractions in recent years, with the United States Department of Transportation (DOT) already implementing various ITS applications, such as Ramp Metering and Traffic Signal Coordination to improve urban mobility. However, one of the main challenges in ITS is traffic state estimation (TSE), which involves estimating traffic state variables with partially observed traffic data. Previous studies estimated a selected set of road segments whereas others focused on short-term prediction. The Dynamic Urban Link Travel Speed Estimation (DU-LSE) problem specifically aims to esimate link travel speed for every link in the road network (i.e., both major and minor arterial) at different times of the day and week. DU-LSE has potential benefits in various ITS applications such as monitoring traffic jams, estimating time of arrival, and route planning. While private companies offer TSE services, there are concerns about their coverage, accuracy, and reliability. Historically. DU-LSE has been studied where approaches ranging from model-driven to data-driven, with the latter gaining traction due to the abundance of data and computing power. However, data-driven studies often require sensing infrastructure, which can be costly for cities, particularly in developing economies. Thus, publicly available data, such as taxi trip datasets, is more preferable. This research explores the New York City Taxi dataset and develops a three-step framework on how to use this excellent source of data for solving the complete DU-LSE problem.

Opportunity in Improving the Traditional Taxi and Ride-hailing system with Carpooling, Smart Dispatching, and Preemptive Repositioning

Compare to the traditional taxi system, Transportation Network Companies (TNC), such as Uber and Lyft, are leveraging technology and connectivity to improve the efficiency of taxi services such as lowering customer wait time, increasing matching rate, and decreasing idling activities. However, TNCs still face challenges in supply and demand imbalance on both spatial and temporal dimensions, which they attempt to address through economic incentives such as surge pricing. While surge pricing has some positive impact on driver decision-making, it has strong negative effects on customer decisions on cancelling the trip and can create discrepancies in pricing for customers, who are close in proximity but separated by zone or time periods. One potential solution to this imbalance is the use of autonomous electric taxis (AET), which are fully compliant with a central strategy and can be preemptively and strategically repositioned from low demand to high demand areas. This research offers an end-to-end solution for the operation and management of a mixed AET fleet where the framework will combine taxi customer demands into carpooling trips, dispatching both HV and AET, and repositioning and recharging AET. The objective is to minimize wait time, cancellation penalties, repositioning costs, and undercharged penalties over the operating day.

2. OPTIMAL POSITIONING OF DYNAMIC WIRELESS CHARGING INFRASTRUCTURE IN A ROAD NETWORK FOR BATTERY ELECTRIC VEHICLES

ABSTRACT

Dynamic wireless charging (DWC) offers a plausible solution to extending Battery Electric Vehicle (BEV) driving range. DWC is costly to deploy and thus its locations need to be optimized. This raises a question often encountered in practice for infrastructure investment: how to determine the optimal locations of DWC facilities in a network. In this paper, we propose a sequential two-level planning approach considering the objectives of both the public infrastructure planning agency and the BEV users. Two different planners' objectives namely, total system travel time and total system net energy consumption are considered. Besides these objectives, constraints such as agency budget, range reassurance, and equity in resource distribution are also addressed at the planner's level. For each objective, BEV drivers respond by choosing their preferred route based on the location of DWC facilities implemented by the planner. An effective solution algorithm is utilized that has the capability of solving relatively large-scale real-world networks within a reasonable computational time. The numerical experiment and case study results provide useful insights on optimally positioning DWC infrastructure to minimize societal cost and energy.

Keywords: Battery electric vehicle; dynamic wireless charging; travel time; equity in resource distribution.

2.1. OVERVIEW

The transportation sector is the largest contributor of greenhouse gas emissions in the United States (U.S). In 2018, the sector contributed approximately 29% of the total energy consumption in the U.S and 92% of which was directly related to fossil fuel (USDOE, 2018). Besides major emission, ICEV puts a burden on the depleting fossil fuel reserve and adds to the national trade deficit by increasing imports. On the other hand, Battery Electric Vehicles (BEVs) provide a cleaner and more energy-efficient transportation option, as well as energy independence from fossil fuel. In the U.S, over the last five years, more than 100,000 BEVs were sold which represents nearly 1% of the total sales and the growth in market penetration is expected to increase in the future (Bomey, 2018). This market penetration of BEV is a result of multiple factors including higher operating efficiency, lower maintenance cost (compared to ICEV), and Federal and State tax incentives. The rising number of BEV has also led to an increase in demand for charging stations. As of July 2018, there are over 20,944 electric stations and 55,487 charging outlets in the U.S (USDOE, 2018). Given the rising market share of BEVs combined with the anticipation that the connected autonomous vehicles (CAVs) are most likely to be electric, the electricity demand of the electrified transportation system of the future would be enormous. However, BEVs still face certain disadvantages compared to ICEVs including extended recharging time and limited driving range.

BEVs' onboard energy-storage systems are primarily high capacity batteries that are typically charged by being plugged into the grid, either at a public charging station or a home outlet. However, it is widely known that the conventional plug-in charging method for BEV has several drawbacks. First, it prevents a BEV from operating while the battery is being charged, as the vehicle must remain physically connected to the grid through the cable. The incapability of a BEV to drive during charging is referred to as recharging downtime (Hwang et al., 2018). Refueling an ICEV with an average 15 gallons gas tank and a gas dispense rate at 5-10 gallons per minute would only take 2-3 minutes. According to Fuller (2016), to satisfy the 5 minutes charging time, a 200-mile (or 80 kWh) BEV would require a charging station with a power up to 960 kW which is infeasible given the current technology of battery charging.

Despite recent developments in battery technology, the driving range, which is the furthest distance Battery Electric Vehicles (BEVs) can travel without the need for refueling, is substantially small compared to Internal Combustion Engine Vehicles (ICEVs). Given the average fuel capacity of 20 gallons and fuel economy of 23 miles per gallon, an ICEV can drive up to 460 miles without the need for refueling. On the other hand, the average driving range of BEV can only reach to 190 miles (Bomey, 2018; Hwang et al., 2018; USDOE, 2018). This limitation can lead to range anxiety for BEV drivers where they are worried that whether they can reach their destination with the battery's remaining in the state of charge (Agrawal et al., 2016).

To overcome these disadvantages, researchers have developed induction based dynamic wireless charging (DWC). Although this technology is still evolving recent research in this domain indicates it has an edge over the conventional plug-in charging (Lukic and Pantic, 2013; Panchal et al., 2018). DWC facility can be embedded under a road and it will dynamically charge the BEV moving above. Due to this feature, DWC does not require BEV to experience charging downtime. This concept is followed by a number of studies focusing on the technical aspects of DWC (Budhia et al., 2013; Miller et al., 2015a, 2015b; Pelletier et al., 2016). Recently, researchers have also discussed the development of wireless charging BEV in relation to the commercialization of BEV (Jang et al., 2016, 2015; Ko et al., 2015; Ko and Jang, 2013). If

implemented properly, DWC can extend the driving range of a large fraction of BEV trips. This would satisfy the range requirement of benefiting BEVs and help to relieve the range anxiety of BEV drivers. Lin et al., (2014) found a significant increase in BEV adoption even if only 5% of the network is implemented with DWC since the technology can address the customer range anxiety problem. Therefore, the DWC Facility Location Problem (FLP) needs to be planned adequately to reap the maximum benefit of this evolving technology. DWC Facility will constitute an important piece of the infrastructure required to allow and promote the use of BEVs and pave the way for electric autonomous vehicles.

There have been several studies in the past devoted to locating refueling facilities, and in particular, recharging infrastructure for BEV. A review of these studies will be presented deliberately in Section 2.2 and from which, we identify the gaps and features that distinguish our research from others (see Section 2.2.3 for details). The aim of this paper is to extend the research on DWC-FLP by including five important considerations, which to the best of our knowledge have not been considered simultaneously in past studies. In particular, we develop an enhanced planning framework for optimally locating dynamic wireless charging facility considering comprehensively and simultaneously the system level network user costs, travel patterns of individuals, a reassurance that network users have enough range augmentation from DWC to get to their final destination, equity in resource distribution between sub-regions, and total budget availability from the public agency to support the needs of BEVs. The remainder of the paper is organized as follows: Section 2.2 presents a summary of related literature in the domain of optimal location for refueling facilities and BEV driver behaviors under the DWC implemented network. In Section 2.3 we present the modeling approach and solution algorithm proposed in this study. This is followed by a numerical experiment as proof of concept in Section

8

2.4. The case study in Section 2.5 presents the DWC-FLP considering the traffic network dataset in Montgomery County, Maryland USA. Section 2.0 presents conclusions, limitations of this study, and avenues for future research.

2.2. LITERATURE REVIEW

We next present a summary of past research in this domain. Past studies are summarized into two sub-sections. First, we present past efforts towards the determination of charging locations, and then we summarize the literature on BEV drivers' behavior in a road network with recharging facilities. Next, the contribution of this study in light of existing literature is highlighted.

2.2.1. Determination of BEV Charging Locations

Considering several studies in literature on addressing the problem of charging (either static or dynamic) FLP for BEVs, the review presented herein is not intended to include all the past research but to provide a review of selected researches in the domain of planning framework for BEV charging infrastructure for BEVs. In Table 2.1, we present a summary of these studies on five main elements which are Study Aspect, Objective Function, Constraints, Approach, and Additional Features. Some important methods considered were the flow-refueling location model (Kuby and Lim, 2007, 2005; Lim and Kuby, 2010), flow-based set covering model (Wang and Lin, 2009) and maximal covering location model (Farahani et al., 2013, 2012).

Authors	Study Aspect	Objective	Selected	Approach	Additional Features
		Function	Constraints		
Fuller	Locating DWC in a	Minimizing the	Range Constraint	Linear	Sensitivity analysis of
(2016)	network consist of	capital cost of		Programming	vehicle starting range
	selected highways in	implementing			and charging power
	California to support	DWC			
	tour-based between				
	cities trip				
Liu and	Locating multiple types	Minimizing	Budget Constraint:	Tri-Level	MSA solution algorithm
Wang	of charging facilities	Weighted sum of	Users' route	Programming	for lower-level user
(2017)	considering public social	travel cost and	choices follow		equilibrium
	cost, users' car	penalty fee for	Wardrop's first		
	ownership choice, and	failed trips	principle		
	users' route choice				
Chen et al.	Determining the optimal	Minimizing total	Budget Constraint	Active-Set	New User Equilibrium
(2016)	location of DWC	social cost		Based	Model for a DWC
				Approach	implemented network
Sathaye	Determining the location	Minimizing total	Budget Constraint;	Continuous	Considers both existing
and Kelley	of publicly-funded static	cost	selected highway	facility	charging station built by
(2013)	charging stations in the		corridors only	location	private institutes and
	Texas Triangle			models	demand uncertainty
	Megaregion				
Dong et al.	Locating multi-level of	Minimizing user	Budget Constraint	Genetic	Activity-based approach
(2014)	static recharging stations	range anxiety as		algorithm-	for simulating driver
	in the greater Seattle	measured by the		based	travel and recharging
	area	number of		optimization	pattern based on GPS
		interrupted trips			travel survey
		and missed			
		vehicle miles			
Riemann et	A Bi-Level approach to	Maximizing the	Covering	Mixed-	Lower-level network
al. (2015)	optimally locate DWC	amount of traffic	constraint	integer	flow problem solved by a
	from a set of selected	flow re-fueled by	formulated for the	nonlinear	Multinomial Logit model
	facilities	the facilities	AC-PC flow	program	based on Stochastic User
			refueling location		Equilibrium principle
			model		
Liu and	Sequentially	Minimizing the	Power transfer,	Deterministic	Considers both (1) a
Song	determining the optimal	capital cost of	supply and demand	and robust	deterministic model
(2017)	location of DWC and	implementing		optimization	ignoring uncertainty in
	optimal battery sizes for	DWC			energy consumption and
	electric buses				travel time and (2) an
					affinely adjustable robust
					counterpart model

 Table 2.1 Past Studies on BEV Charging Infrastructure

Authors	Study Aspect	Objective	Selected	Approach	Additional Features
		Function	Constraints		
Chen et al.	Studying the	Minimizing	Trip completion	Mathematical	Explores the
(2017)	implementation of	social cost as	assurance;	Formulation	competitiveness of DWC
	different types of	measured by the			over Static Charging
	charging considering	normalized sum			under both public and
	driver's choice of	in terms of			private provision
	charging facilities	monetary units of			scenarios; Charging
		capital cost,			prices follow either Nash
		charging time,			equilibrium in private
		electricity cost,			provisions or revenue-
		and total driving			neutral in public
		time			provision
Xi et al.	Locating static charging	Maximizing the	Mutually exclusive	Linear	Overall service levels are
(2013)	of either Level 1 or 2 for	summation of	charging location;	Integer	less sensitive to
	BEVs in the central	energy recharged	Fixed tour	Programming	optimization criterion as
	Ohio region	of the entire	schedule		in contrast to optimal
		system			DWC location
Xu et al.	Studying the factors	Maximum	None	Mixed Logit	Battery capacity,
(2017)	affecting user choice of	likelihood		Model	midnight indicator, the
	charging mode				initial state of charge,
	(normal/fast;				and the number of past
	home/public) and the				fast charging events are
	location of charging				important factors in the
	facilities in Japan by				users' decision-making
	using users' preference				process
	data				_
Huang et	Deployment of	Minimizing the	Charging	Mixed-	Utilizes multiple
al. (2015)	alternative refueling	capital cost of	characteristics	integer	deviations of paths
	stations in the	implementing		programming	between O-D pairs
	transportation network	DWC			instead of the shortest
					path
He et al.	Locating refueling	Minimizing the	Range Constraint	Bi-Level	Considers drivers'
(2015)	stations for BEV in a	capital cost of		Programming	spontaneous adjustments,
	road network using a	implementing			the relation between
	tour-based approach	DWC			travel and recharging
					decisions, and risk-taking
					behavior
Zhang et	Locating static	Maximizing total	Capacitated Flow	Arc Cover-	Includes demand
al. (2017)	supercharging for BEVs	flow coverage	-	Path Cover	dynamics resulting from
					newly implemented
					DWC

Table 2.1 Continued

2.2.2. BEV Driver Behavior in a Road Network with DWC Facilities

The public agency decides where to implement refueling facilities and in response to that plan, the drivers choose the route that maximizes their utility (or minimizes disutility). However, in the case of a DWC implemented network, a BEV driver may also account for an increase in driving range for his vehicle and hence it should be considered in disutility or cost function. A driver's disutility is also impacted by others' route choice decision as well. These decisions collectively affect the traffic flow as well as travel times in a network. In literature, these decisions are often attributed to the lower level problem representing network user perspective and several studies have been devoted to network flow estimation under given charging facility and range constraint. Past studies have proposed both deterministic approach and stochastic approach under range uncertainty for estimating flows of BEVs. Kitthamkesorn and Chen (2017) solved the combined modal split and traffic assignment problem by using a nested weibit model on the mode choice level and suggest that path-size weibit model on the route choice level since it performs better than the traditional logit model. Liu et al. (2016) developed a model for better fuel economy estimation for electric vehicles by customizing a realistic driving cycle based on the GPS data of drivers in California. Strehler et al. (2017) determined the shortest path for battery electric and hybrid vehicles by creating a model that accounts for several factors that are not usually recognized in the ICEV shortest path problem such as extended recharging time, the balance between speed and range, and regenerative braking. Xie et al. (2017) developed a pathconstrained traffic assignment for electric vehicles subject to stochastic driving ranges. Their research focuses on the tour or trip chain rather than the normal trip level where customer range anxiety is more likely to occur. When recharging time is concerned, electric vehicles' batterycharge level may be a non-linear function of recharging time in contrast to ICEV gasoline level

which is a linear function of fueling time. To solve this problem, Montoya et al. (2017) proposed a hybrid metaheuristic for solving electric vehicle routing problem that takes into account components considered in ICEV studies and specifically designed components reflecting the non-linear behavior of BEV recharging.

2.2.3. Contribution and Significance of this Study

There is a rich literature on the refueling FLP for ICEV but only a few focusing on BEV and even less concerning DWC instead of static charging. Furthermore, in the context of DWC-FLP, the majority of the studies focus on only one level of the bi-level problem which is either optimal DWC facility plan or BEV traffic assignment. Studies considering these two levels simultaneously are minimal and thus are preferable because of the strong interdependency between two levels. However, typically these studies have restricted to small size networks owing to the expense of computational complexity. In addition, past studies using the bi-level approach generally choose their decision variable in the form of a binary variable representing whether or not to implement DWC on the entire length of a link under consideration (Chen et al., 2016; Liu and Wang, 2017). However, this choice makes the model less flexible, and it would lead to a sub-optimal result under budget constraint especially in the case of a network containing long links e.g. highways. To be more specific, implementing DWC on an entire length of a highway would be an inefficient use of resources. The problem can be partially addressed by considering the highway as a collection of multiple smaller links. However, even this solution may not be optimal because it raises the question about how to segment the highway and what should be the optimal segment length. Therefore, a continuous variable representing a fraction of link length would be more appropriate for the optimization model. Besides the choice

of decision variables, in past studies, the network is treated as a whole which raises problems in practical implementation especially when there are variations in funding priority among subregions raising equity concerns.

This study endeavors to bridge these gaps in the literature stated above. We propose a sequential two-level planning approach considering the objectives of both the planner and road users. In the Upper-Level, two different planner objectives namely, total system travel time and total system net energy consumption are considered along with three distinctive elements. First, a trip completion reassurance constraint is used in the planner level to avoid costly failed trips which are important to overcome the range anxiety problem. Second, the proposed approach divides the network into sub-regions (different from traffic analysis zones or TAZ) and adds a constraint representing equity in resource distribution in the upper level to address the differences in funding priority between regions. Third, the model formulation adopts continuous decision variables to provide flexibility in DWC implementation as opposed to binary variables used in past studies. In the Lower-Level, we present a mathematical programming (MP) formulation for a single class BEV static deterministic user equilibrium problem representing users' route choices. The user route cost function takes into account the normalized negative cost incurred due to the recharging of BEV's battery through DWC as the user travels along their preferred path. For solving the BEV user equilibrium, an effective algorithm using a slope-based path shift propensity approach is deployed because of its capability to solve large-scale network in reasonable time.

2.3. METHODOLOGY

2.3.1. Modeling Approach

In general, a government agency decides FLP under a macro perspective such as maximizing the social benefits resulting from the facilities (e.g., implementation of DWC) while ensuring that required resources for the implementation would not exceed the agency budget. Hence, one may argue that FLP can be decided based on link flows to benefits a large fraction of network users. The government agency can get information on link flows under the current condition by a variety of methods such as using sensors (e.g. microwave or infrared sensors), traffic cameras, loop detectors, or through the four-step transportation planning. Based on the existing data of traffic flow, a typical approach for implementing DWC facilities would be to locate it on links having higher flow so that more cars can be recharged. However, this approach cannot encapsulate the likely micro interpretation of the network users for whom the facility is planned. In particular, the BEV drivers are likely to choose path by factoring in both DWC implementation and travel time. Given the range constraint of BEVs and range anxiety of BEV drivers, they may prefer the DWC implemented roads. Therefore the roads with high volume will likely be loaded with more traffic and result in high congestion and extended travel time, which is not ideal. Therefore, the approach based on the existing link volume, which does not account for the changes in traffic flow in the network due to DWC, does not yield the optimal result as initially intended by the planning agency. Therefore, for selecting an optimal DWC plan, an analytical framework is warranted that takes into account the network users' response to the DWC plan.

Thinking about a single network user's perspective, he/she chooses the best possible route that minimizes his/her disutility. It is practical to assume that the route which yields the

minimum generalized cost (computed by factoring in both travel time and DWC charging) would be selected. The aggregate responses of BEV drivers leading to an equilibrium traffic flow after a DWC plan need to be determined. Therefore, one-level mathematical programming is not appropriate for solving this DWC-FLP since there are two interdependent levels of decision making that is difficult to be modeled seperately. We define these two levels of optimization as an Upper-Level (UL) government agency's DWC implementation decision-making process and a Lower-Level (LL) network users' choosing route process.

The UL and LL problems will interact through a feedback mechanism. The relationship between the UL and LL are shown in Figure 2.1. In the UL, with information on current travel time and traffic flow data, government agency defines the length (what fraction of link length) and location (on which links) of DWC implementation with the objective of minimizing total societal cost. The DWC facility implementation plan will consequently affect the network's user path choices leading to changes in the traffic flow pattern and hence travel time of the links which are estimated in the LL.



Figure 2.1 Schematic representation of the interaction between upper and lower level

We next introduce the notations used in this paper, then UL and LL formulations are presented. Following are the notations used in the paper:

Notations

Set	
A	Set of links
W	Set of Origin-Destination pairs
P_w	Set of used paths for O-D pair w
D	Set of regions
Param	eters
b	Cost of implementing dynamic wireless charging facilities (in \$/mile)
θ	Agency budget
r	Additional recharging miles per miles traveled on DWC charging facilities (in
	mile/mile)
ψ	Power transfer rate (in kWh/mile)
η	Cost of one unit of electricity (in \$/kWh)
τ	Value of time (in \$/h)
Ε	Upper Limit for equity in resource distribution among sub-regions constraint (Unitless)
la	Length of link <i>a</i>
	Negative cost experienced by the driver due to DWC recharging along link a (in travel

- μ_a Negative cost experienced by the driver due to DWC recharging along link *a* (in travel time units, i.e. minutes)
- l_a^e Length of link *a* having DWC charging facility (in mile)

 cap_a Capacity of link a

- t_a^0 Free flow travel time on link *a*
- g_a Generalized cost for traveling on link a
- G_p^w Generalized cost for traveling on path p between an O-D pair w
- α_a Coefficient for link *a* for the link cost function
- β_a Coefficient for link *a* for the link cost function
- ζ Average fuel efficiency (in kWh per miles)
- s_p Vehicle initial range at the start of the trip using path p
- δ_{ap}^{w} Link-path incident parameter, which takes the value 1 if link *a* belongs to path *p* of O-D pair *w* and 0 otherwise
- q_w Travel demand for O-D pair w
- f_p^w Flow on path p of the O-D pair w
- ε_d Predefined constant for representing funding priority in area d
- C_d Preferred resource allocated to area d, reflecting the area's funding priority and road miles

Decision Variables

- y_a Length of DWC facility on link *a* as a percentage of the length of link *a*
- v_a Flow on link *a*

t_a Travel time on link a

 γ_a Energy consumption of traveling on link *a* (in kWh)

2.3.2. Upper-Level of Government Agency Decision Making

While deciding the location of DWC facilities or network improvement, the government agency typically has an objective to minimize the total societal cost. We propose two different metrics to quantify the societal cost. The first metric is Total System Travel Time (TSTT) addressing traffic condition and the second metric is Total System Net Energy Consumption (TSNEC) addressing energy efficiency. The first term TSTT can be calculated by taking the aggregate sum among all links within the network of its flow multiplied by its travel time. TSTT is an important metric to evaluate transportation network performance and thus, it is selected as an objective in the domain of network infrastructure investment in many studies (Marcotte, 1983; Abdulaal and LeBlanc, 1979; Mathew and Sharma, 2009; Chiou, 2005; Konur and Geunes, 2011; Chow et al., 2011; Gao et al., 2011; Hajibabai et al., 2014; FHWA, 2015; Chen et al., 2016, 2017; Jing et al., 2017; Liu and Wang, 2017). Consistent with past literature this study also uses TSTT as an objective at The Upper-Level. The second term TSNEC is determined by taking the aggregate sum over all links in the network of the product between link flow and its corresponding average energy consumption by BEVs traversing on that link. These two objectives namely, TSTT and TSNEC are incorporated in Model 1 and Model 2 respectively.

Model 1: Minimizing total system travel time (TSTT)

Objective Function:

$$\min z_1 = \sum_{a \in A} v_a t_a \tag{1}$$
Subject to:

$$b\sum_{a\in A} y_a l_a \le \theta \tag{2}$$

$$s_p + \sum_{a \in p} (ry_a l_a - l_a) \ge 0 \quad \forall p \in P_w, w \in W$$
(3)

$$0 \le y_a \le 1 \quad \forall a \in A \tag{4}$$

$$t_a = t_a^0 \left[1 + \alpha_a \left(\frac{\nu_a}{cap_a} \right)^{\beta_a} \right] \quad \forall a \in A$$
(5)

$$v_a = f(y_a) \tag{6}$$

$$g_a = t_a + \mu_a \ge 0 \qquad \forall a \in A \tag{7}$$

$$y_a, v_a, t_a, \gamma_a \ge 0 \qquad \forall a \in A \tag{8}$$

Equation (1) represents the objective function of Model 1, which minimizes the total system travel time. The value of the first term, traffic flow (v_a) depends on the UL decision variable, which is the DWC plan represented by the vector (y_a) . The second term, travel time (t_a) depends on the traffic flow (v_a) . Although the UL decision variable, which is the DWC plan (y_a) , is not explicitly present in the objective function, it fundamentally affects the objective function value. Equation (2) states that the accumulation of the cost of implementing DWC within the network must not exceed the agency budget. Equation (3) is a trip completion reassurance constraint designed to avoid a costly failed trip and to overcome range anxiety. We make a simplifying assumption that all vehicles selecting a path between an O-D pair have the same starting range s_p . Based on this assumption, Equation (3) ensures that DWC facilities are implemented in such a way that every vehicle can get sufficient additional range to complete

their trip by traveling over the DWC facilities implemented on the links along their path. The second term in Equation (3) represents the additional range obtained from recharging through the DWC facilities. In the commercial market, DWC facility power transfer is measured in kW and by multiplying it with the traveling time of a BEV over the facility, we get the amount of energy in terms of electricity (kWh) transferred to the vehicle's battery. However, in Equation (3), other parameters' units are in terms of distance or miles. We divide the electricity energy by average BEV electricity consumption rate (Wh/mile) to convert it to equivalent range. To simplify the process, we introduce a coefficient r representing the additional range (in mile) per miles of travel over the DWC facility. With a DWC facility power transfer rate of 4 kWh recharged per miles traveled and a 400 Wh/Mile average fuel economy of BEV, the value of r is 10 miles recharged/mile traveled. It implies (based on this example) that if a BEV travels one mile over the DWC facilitated part of a link, it will gain 10 miles of range while losing a single mile of range in traversing that part of the link, hence resulting in net 9 miles of gain in range. Equation (4) implies that the decision variable is a continuous variable representing the length of the DWC facility of a link as a fraction of that link length (hence a value between 0 and 1). Equation (5) is the link cost function (we use BPR function developed by Bureau of Public Roads) where link travel time (t_a) is a monotonically increasing function of link flow (v_a) . In Equation (5), t_a^0 is the free flow travel time and α_a , β_a , cap_a are the parameters of link cost function specific to link a. Equation (6) signifies that v_a is the function of the DWC plan (y_a) and given the input (y_a) , the LL traffic assignment task generates the output of link flows in the state of user equilibrium. Equation (7) ensures that no link has negative generalized cost that can otherwise promote circular paths by vehicles to gain extra driving range (it is also required for the feasibility of LL problem). The quantity μ_a in Equation (7) is defined by Equation (17) and (18) presented later.

Equation (8) represents the non-negativity characteristic of the following decision variables: traffic flow (v_a), DWC plan (y_a), travel time (t_a), and energy consumption (γ_a).

We add an equity in resource distribution constraint to address the differences in funding priority between sub-regions. Typically, a larger geographic region (i.e., state or county) consists of smaller sub-regions and available capital budget in practice is distributed based on funding priority of the region. In addition to funding priority, sub-regions also vary greatly in their area and specifically in the context of DWC-FLP, road land miles. Therefore, the money distributed to a sub-region must reflect both its funding priority and total road lane miles. To illustrate this constraint, let D_d , represents a sub-region, of study area D having d sub-regions, and these sub-regions are mutually exclusive to each other:

$$D_1 \cup D_2 \cup D_3 \dots D_m \cup D_n \dots \cup D_d = D, \quad D_m \cap D_n = \emptyset \quad \forall D_m, D_n$$

The equity constraint is described as follows:

$$\sum_{d \in D} \left(b \sum_{a \in D_d} y_a l_a - C_d \right)^2 \le E \tag{9}$$

$$C_d = \theta \frac{\varepsilon_d \sum_{a \in D_d} l_a}{\sum_{d \in D} (\varepsilon_d \sum_{a \in D_d} l_a)} \quad \forall d \in D$$
(10)

Equation (9) works on the basis of the sum of square of the differences between two terms. The first term is the total DWC implementation cost in a sub-region D_d and the second term C_d is reflecting the preferred resource allocated to sub-region D_d . This sum of square must be less than a predefined constant E, which is empirically determined. Equation (10) states that C_d reflects both the funding priority coefficient ε_d and total road lane miles of sub-region D_d and all C_d sum up to the total budget θ over the entire area D. Equation (9) is based on the assumption that a link is part of only one sub-region C_d . However, this is not a restrictive assumption, and if there are long links in a network that extends to more than one sub-region, then those links can be divided into multiple links each spanning in one sub-region. This will be typically the case for long arterials and interstate highways. Equation (9) is geared toward being an incentive constraint rather than a restricted one. Sub-regions can exceed its preferred budget allocated C_d in ways of improving the UL objective function value, as long as the sum of square of the difference between the preferred budget and the actual DWC cost over the entire region *D* does not exceed *E*.

Model 2: Minimizing total system net energy consumption (TSNEC)

Objective Function:

$$\min z_2 = \sum_{a \in A} v_a \gamma_a \tag{11}$$

Subject to:

$$\gamma_a = l_a \zeta - r y_a l_a \zeta \quad \forall a \in A \tag{12}$$

and constraints represented by Equations (2)-(10)

Equation (11) represents the objective function of the Model 2 which minimizes the total system net energy consumption by BEVs. The objective function value (in terms of Vehicle.kWh) is the summation among all links of the product between traffic flow and net energy consumption. Equation (12) represents the calculation of net energy consumption. The net energy consumption is computed as the required electricity (in kWh) to traverse link *a* minus the energy recharged through DWC (in kWh) while traveling along link *a*. Model 2 is also subject to the constraints (Equation (2) through (10)) listed in Model 1.

2.3.3. Lower-Level Network User Equilibrium

The Lower-Level problem aims to estimate the network flows resulting from the network users (BEV drivers) route choices in response to the government's DWC Implementation Plan. With an assumption that drivers are rational in their decision-making process, they will choose the path, among a set of available paths for their trip, which yields a minimum value for the normalized travel time. The generalized travel cost represents the aggregate of the following elements: (1) summation of travel times along the links included in the chosen path; (2) the aggregate benefit derived from DWC facility by BEV as they are driven along the chosen path. In this study, we assume that link costs (travel times) are separable and link travel time of a link depends on the flow of that link only.

The task of deciding the flows of paths/links based on the aggregate of network users path choice decisions is often referred to as a traffic assignment problem. Traffic assignment can be categorized as either static traffic assignment (STA) or dynamic traffic assignment (DTA). Both STA (Jiang et al., 2012; Xie and Jiang, 2016) and DTA (Agrawal et al., 2016) models have been used to characterize the route choice behavior of BEV drivers in the past. STA assumes that traffic is in a steady-state and hence flows and travel times of links can be represented using average conditions. The DTA models can capture the traffic flow dynamics more accurately compared to STA models due to the presence of temporal dimension in the model. Therefore, DTA can be utilized to accurately estimate the energy consumed by BEV, analyze the effectiveness of an operational strategy (e.g., signal coordination), and improve the traffic flow estimation. However, DTA models are characterized by inherent mathematical intractability (Peeta and Ziliaskopoulos, 2001) and deploying them in practical context entails simulation of the time-dependent traffic flow which is computationally expensive. It is difficult to design an efficient solution algorithm for a network design problem (e.g., optimal DWC location problem) that requires estimation of network flows numerous times using a DTA model. Therefore, due to these limitations of DTA, the STA is preferred in transportation planning context and is usually applied for network design problems (see e.g., Kumar and Mishra, 2018; Mishra et al., 2016; Kumar et al., 2019; Sharma and Mishra 2013). Taking the above factors into consideration, we seek to develop a static user equilibrium traffic assignment model to characterize the route choice behavior of BEV drivers in a network with DWC facility.

Wardrop's User equilibrium (UE) principle is mostly used for finding the network flows in a transportation network. It states that the journey times in all routes actually used are equal and less than those that would be experienced by a single vehicle on any unused route. UE is achieved when drivers cannot improve their travel time (cost) unilaterally by switching routes. According to Sheffi (1985), under non-negative monotonically increasing separable link cost function and non-negative demand, the UE-STA problem can be formulated as a convex optimization problem. In the context of this study, we present an MP formulation for single class BEV static deterministic user equilibrium (BEV-UE) problem in a network with DWC facility. The BEV-UE needs to incorporate changes in link cost functions due to DWC investment decisions. The BEV-UE problem is formulated as follows:

BEV-UE:

Objective Function:

$$\min z_{3} = \sum_{a \in A} \left(\int_{0}^{v_{a}} t_{a}(x_{a}) \, dx + \mu_{a} v_{a} \right) \tag{13}$$

Subject to:

$$\sum_{p} f_{p}^{w} = q_{w}, \qquad \forall w \in W$$
(14)

$$f_p^w \ge 0, \quad \forall \ p \in P_w, w \in W$$
 (15)

The definitional constraints:

$$v_a = \sum_{w \in W} \sum_{p \in P_w} \delta^w_{ap} f^w_p, \quad \forall \ a \in A$$
(16)

$$\mu_a = -l_a^e \psi \eta \left(\frac{60}{\tau}\right), \forall \ a \in A \tag{17}$$

$$l_a^e = y_a l_a, \forall \ a \in A \tag{18}$$

Equations (13)-(18) represents the BEV-UE formulation under the DWC facility proposed in this study. Equation (13) represents the minimization of the objective function. Equation (14) is the flow conservation constraint. Equation (15) ensures that path flows are nonnegative. Equations (16)-(18) are definitional constraints. Equation (16) defines the relationship between link and path flows. Equation (17) defines the negative cost experienced by BEV drivers due to DWC charging. Equation (18) determines the length of the link covered with DWC facility and connects UL decision variables to the LL problem. Next, we prove the equivalency of above MP formulation with user equilibrium of BEVs in a DWC facilitated network. **Proposition.** Under the assumption of monotonically increasing separable link cost function, the MP formulation presented by Equations (13)-(18) is equivalent to Wardrop User Equilibrium of BEV drivers defined as below:

BEV-UE in a DWC facilitated network is achieved when generalized cost of all used paths between an O-D pair are equal which is less than or equal to generalized cost of any unused paths.

Proof: The Lagrangian of the minimization problem represented by Equations (13)-(18) can be formulated as:

$$\mathcal{L}(\boldsymbol{f},\boldsymbol{\sigma}) = z_3(\boldsymbol{f}) + \sum_{w \in \boldsymbol{W}} \sigma_w \left[q_w - \sum_{p \in \boldsymbol{P}_w} f_p^w \right]$$
(19)

where, σ_w is the Lagrange multiplier associated with equality (flow conservation) constraint represented by Equation (14). Note that definitional constraints do not enter in the Lagrange function $\mathcal{L}(.)$. At the stationary point of Lagrangian, the following conditions need to hold:

$$f_p^w \frac{\partial \mathcal{L}}{\partial f_p^w} = 0, \forall p \in \mathbf{P}^w, w \in \mathbf{W}$$
(20)

$$\frac{\partial \mathcal{L}}{\partial f_p^w} \ge 0, \forall p \in \mathbf{P}^w, w \in \mathbf{W}$$
(21)

$$\frac{\partial \mathcal{L}}{\partial \sigma_w} = 0, \forall w \in \boldsymbol{W}$$
(22)

In addition to conditions (20)-(22), non-negativity constraints (15) of path flows need to be satisfied. Condition (22) simply states that the flow conservation condition needs to hold. Now for notational simplicity, we focus on a single O-D pair $w \in W$. However, the derived results will be valid for all O-D pairs. The partial derivatives of Lagrangian $\mathcal{L}(.)$ with respect to path flow variable is given as:

$$\frac{\partial \mathcal{L}}{\partial f_p^w} = \frac{\partial}{\partial f_p^w} z_3(f) - \sigma_w \tag{23}$$

Using the diagonal rule, the partial derivatives of z_3 with respect to f_p^w is given as:

$$\frac{\partial z_3}{\partial f_p^w} = \frac{\partial z_3}{\partial x_a} \frac{\partial x_a}{\partial f_p^w}$$
(24)

Noting the fact that the partial derivate of x_a with respect to f_p^w is δ_{ap}^w , the partial derivatives of z_3 with respect to f_p^w is given as:

$$\frac{\partial z_3}{\partial f_p^w} = \sum_{a \in A} (t_a + \mu_a) \delta_{ap}^w$$
(25)

Note that $t_a + \mu_a = g_a$ is the generalized cost of traveling on a link *a*. Using Equation (25) partial derivatives of z_3 with respect to f_p^w is the generalized cost of path *p* represented as G_p^w :

$$\frac{\partial z_3}{\partial f_p^w} = G_p^w \tag{26}$$

Therefore, using Equation (23), the partial derivatives of $\mathcal{L}(.)$ with respect to f_p^w is given as:

$$\frac{\partial \mathcal{L}}{\partial f_p^w} = G_p^w - \sigma_w \tag{27}$$

Now, using Equations (20), (21) and (27) we get:

$$f_p^w(G_p^w - \sigma_w) = 0, \forall p \in \mathbf{P}^w, w \in \mathbf{W}$$
(28)

$$G_p^w - \sigma_w \ge 0, \forall p \in \mathbf{P}^w, w \in \mathbf{W}$$
⁽²⁹⁾

The Equations (28) and (29) together imply that either flow on a path f_p^w is zero or its generalized cost G_p^w is equal to Lagrange multiplier σ_w . In addition, the Equation (29) implies that Lagrange multiplier σ_w of a given O-D pair is less than or equal to the generalized cost of all paths connecting this O-D pair. Now considering σ_w as the minimum generalized path cost for the O-D pair *w*, this is equivalent to the condition of Wardopian User Equilibrium. This proves that the MP formulation presented by Equations (13)-(18) is equivalent to Wardrop's User Equilibrium for BEV drivers in a network with DWC facility.

2.3.4. Solution Algorithm

The DWC-FLP problem is modeled as a Bi-Level Programming with a computationally heavy objective function. There are two main reasons for this difficulty as follows: (1) the value of the objective function cannot be explicitly calculated by the UL decision variable (i.e. DWC plan) alone; and (2) it requires an additional sub-level optimization model (i.e. BEV-UE) to compute the components (i.e. travel time and energy consumption) and ultimately the objective function value (OFV) and thus demands heavy computational time. In order to solve this problem, we utilize and extend an algorithm called Constrained Local Metric Stochastic Response Surface (ConstrLMSRS) developed by Regis (2011) to solve the Bi-level problem.



Figure 2.2 Modified ConstrLMSRS Algorithm Flowchart

The algorithm works as a feedback loop until the termination criteria are met. It consists of three stages: initialization, iteration, and conclusion as shown in Figure 2.2. We have presented the pseudo-code of the UL solution algorithm in the appendix.

At each iteration, a large number of candidate feasible solutions (we choose 20,000 in our numerical experiment and case study) are generated. Each candidate solution requires running the traffic assignment task for the LL BEV-UE to get the objective function value. Thus, the process of performing this task for the set of candidate solutions is an expensive task. In order to address this problem, while estimating the objective function value for each candidate solution, the Radial Basis Function (RBF) Interpolation method is utilized in substitution of performing the BEV-UE traffic assignment. The following section discusses briefly describes this method.

Radial Basis Function (RBF) Interpolation Method

The RBF interpolation was introduced by Powell (1992) and used by Regis (2011) in solving an optimization problem with an expensive objective function. The method can be processed with small computational cost. Here we present a brief overview of this method. For the full description of the radial basis function interpolation, please refer to (Powell, 1992).

Given a set of T training points, of which OFV are known : $T = \{y_t, Z(y_t)\}$ we can construct a response surface model: $S(y) = \sum_t \omega_t \phi ||y - y_t|| + l(y)$ to interpolate the objective function value. Note here that y_t is a variable with *a* dimensions.

Where:

- $\phi(r)$: A cubic form function $\phi(r) = r^3$
- **||**. **||**: Euclidean norm

l(y): A linear polynomial function in *a* variables to be determined, with a coefficient *c* which has (a + 1) dimensions

 ω_t : A coefficient to be determined which has t dimensions

The two coefficients ω_t and *c* can be calculated as follows:

$$\begin{pmatrix} \lambda & H \\ H^T & \mathbf{0}_{(a+1)\times(a+1)} \end{pmatrix} \begin{pmatrix} \omega_t \\ c \end{pmatrix} = \begin{pmatrix} Z^T(y_t) \\ \mathbf{0}_{a+1} \end{pmatrix}$$

Where:

- λ: A matrix with $t \times t$ dimension, calculated as: $\lambda_{ij} = \phi ||y_i y_j||$ with i, j = 1, 2, ..., t
- *H*: A matrix with $t \times (a + 1)$ dimension, where the t^{th} row is $[1, y_t^T]$.

By solving this set of equations, we acquire the value of the coefficient $\omega_t = (\omega_1, \omega_2, ..., \omega_t)^T$ and $c = (c_1, c_2, ..., c_{a+1})^T$. By plugging these coefficients back, the response surface model S(y)is constructed and utilized to interpolate the OFV of a candidate solution y.

The main advantages of ConstrLMSRS are wider searching range, faster evaluation of the objective function. For the first advantage, in each iteration, a large set of candidate points is generated via perturbing the current best solution. The perturbation step size is selected as a continuous variable to sufficiently cover all the possible solutions. Other approaches such as Active-Set Based Algorithm (Chen et al., 2016) only consider one feasible solution at a time. As a result, the searching region can cover a wide range of possible solutions without compensating computational time and power by evaluating them via the RBF method. For the second advantage, instead of running a traffic assignment task for each candidate point generated within each iteration, only one traffic assignment task is computed for the best candidate point per iteration, which results in a significantly fewer computing step, complexity, and time. It is important to mention that the study uses ConstrLMSRS method as it is able to deal with realworld size network with moderate computational time but other heuristic algorithms can also be used for this purpose such as Memetic Algorithm (Pishvaee et al., 2010), Differential Evolution (Koh, 2007), Evolutionary Algorithms (Lau et al., 2009) and Hill climbing (Los and Lardinois, 1982).

Lower Level BEV-UE Solution Algorithm

The LL problem (BEV-UE) is solved by customizing the SPSA algorithm developed by Kumar and Peeta (Kumar and Peeta, 2014). The SPSA flow update mechanism was used with the modified cost function and has been implemented in this study through a C++ script. Modified cost function includes the travel time and negative cost due to DWC charging. The SPSA yields a UE link flows and link travel times which is feedback to the UL. The SPSA implementation steps are not presented here for brevity (The readers can refer to Kumar and Peeta, (2014) for SPSA implementation details).

2.4. NUMERICAL EXPERIMENTS

2.4.1. Small Test Network

Numerical experiments are first conducted using a small size test network to obtain insights before conducting detailed analysis. The topology of the test network is shown in Figure 2.3. The network consists of 15 nodes, 18 links, three origins, and three destinations. Three origins are represented as nodes 1, 2, and 3. Similarly, three destinations are nodes 12, 13, and 14. The number inside the circle represents node number and the number beside the link represents the link number. The travel demand for various O-D pairs and paths in the form of a sequence of links is also shown in Figure 2.3. There are six O-D pairs with non-zero travel demand. In

addition, we divide the network into 8 sub-regions, each has its own set of links and funding priority coefficient ε_d as shown in Figure 2.3. Table 2.2 presents the links parameters of the test network.



Sub-Regions	Links	\mathcal{E}_d
1	1,2	1
2	3,4	1
3	5,6	1
4	7,8	1
5	9,10	1
6	11,12	1
7	13,14	1
8	15,16,17,18	1

Figure 2.3. Small 18 Link Test Network

Link Number	From Node	To Node	Ca	t_a^0	α_a	β_a	la
1	1	4	3000	1.25	0.15	4	1.3
2	1	5	4000	1.25	0.13	4.1	1.3
3	2	5	5000	1.25	0.1	3.9	1.3
4	2	6	3000	1.25	0.12	3.8	1.3
5	3	6	7000	1.25	0.13	3.5	1.3
6	3	7	6000	1.25	0.125	3.2	1.3
7	4	8	3500	1.25	0.128	3.3	1.3
8	5	9	8000	1.25	0.127	3.4	1.3
9	6	10	9000	1.25	0.13	3.9	1.3
10	7	11	2500	1.25	0.132	4.2	1.3
11	8	12	3500	1.25	0.133	4.6	1.3
12	9	12	4000	1.25	0.134	4.2	1.3
13	9	13	4500	1.25	0.136	3.3	1.3
14	10	13	5000	1.25	0.139	3.8	1.3
15	10	14	4000	1.25	0.138	3.2	1.3
16	11	14	3800	1.25	0.14	3.6	1.3
17	11	15	3800	1.00	0.14	3.6	1.1
18	15	14	3800	0.25	0.15	3.2	0.3

Assumptions

The study makes some assumptions for conducting numerical experiments which include: (1) the cost of implementing DWC is \$4 million per lane per mile, (2) all vehicles using the network are BEVs and have the capability to be charged with DWC, (3) % DWC refers to inductive charging available as a percentage of the link and in case of multiple lanes, only one lane is implemented with DWC facility, (4) the problem considered is an un-capacitated refueling model which indicates that there is no limitation on the number of vehicles being charged at the same facility (in our case a given section of link) at the same time, and (5) the public agency has \$3.6 million in budget and this budget scenario is herein referred to as the Base Scenario to distinguish itself with other budget scenario mentioned in Section 2.4.3 Budget Sensitivity Analysis.

2.4.2. Numerical Results and Insights

To assess the model convergence, the UL objective functions value within each iteration are stored for performance assessment purpose. Figure 2.4(a) shows the TSTT objective function value with the progress of iterations. The objective function value starts at 374,519 and decreases further with iterations. There are significant drops in the objective function value at the 8th, 15th, and 25th iteration, and it reaches the minimum value 374,258 after the 25th iteration. The algorithm terminates at the 40th iteration of. We see that the objective function value is not improving after the 25th iterations. This is due to the fact that the set of training points already covered most of the "peaks" and the iteration does not need to "search" any further. At the end of iterations, the TSTT value represents the objective function corresponding to the final best solution for the DWC plan.



(a) TSTT objective function convergence with iterations



(b) TSNEC objective function convergence with iterations Figure 2.4 TSTT and TSNEC convergence with iterations

Figure 2.4(b) shows the TSNEC objective function value with increasing iterations. In iteration 1, the TSNEC value was 29,690 which reduced to 27,894 in the 10th iteration and reached the minimum value of 25,662 after the 14th iteration. The TSNEC model reaches convergence sooner than the TSTT model. The result DWC plans for both TSTT and TSNEC in the Base Scenario are presented later in Section 2.4.3.

To validate the benefit of DWC in Model 2, we calculate the changes in TSNEC as compared to the Do-Nothing scenario and total energy recharged under various user route choice scenarios. The scenarios are developed by modifying the convergence criteria of the SPSA algorithm (Kumar and Peeta, 2014). The percentage of difference (nue_frac) between a route choice scenario and the base user equilibrium is computed as follow:

$$nue_frac = \frac{1}{|A|} \sum_{a \in A} \frac{|v_a^{ue} - v_a^{nue}|}{v_a^{ue}}$$

Where:

|*A*| Cardinality of set *A*

nue_frac:Percentage difference in route choice from UE v_a^{ue} :Flow of link *a* resulting from drivers on UE path

 v_a^{nue} : Flow of link *a* resulting from drivers on non-UE paths

A used path is considered as non-UE if its generalized cost is higher than minimum cost path of the O-D pair by more than 1% margin. Algorithm at the lower level was terminated as the value of *nue_frac* falls below various threshold levels (e.g. 10%, 20%). Table 3 shows the percentage of decrease in TSNEC as compared to the Do-Nothing scenario and the total energy recharged for various user route choice scenarios (represented by percentage difference from UE, *nue_frac*). The DWC plan is taken from the result of Model 2 minimizing TSNEC under a budget of \$3.6 million. The benefit from DWC drops as BEV users deviate from the user equilibrium state but only by a marginal margin. At the base user equilibrium, the percentage decrease in TSNEC and total energy recharged are 62% and 41,489 (Vehicle.kWh) respectively. The numbers drop noticeably in the 10% Difference scenario at only 55% and 36,636 (Vehicle.kWh). However, the decreasing rate flattens out as the percentage of difference increases. At the 60% Difference scenario, the percentage decrease in TSNEC and total energy recharged remains at a high level of 47.14% and 31,631 (Vehicle.kWh) respectively.

User route	% Decrease in TSNEC	Total energy recharged		
choice scenario	(compared to Do-Nothing)			
Base (UE)	61.88%	41,489		
10% Difference	54.69%	36,636		
20% Difference	53.92%	36,149		
30% Difference	53.72%	36,168		
40% Difference	53.45%	35,973		
50% Difference	47.32%	31,872		
60% Difference	47.14%	31,631		

Table 2.3. Benefit from DWC implementation at Different Route Choice Scenario

We also attempted to validate the RBF Interpolation performance since a poor estimation of the objective function would result in an incorrect best candidate point. Figure 2.5 shows the objective function value in both TSTT and TSNEC (plotted on the secondary axis) as estimated by the RBF Interpolation method (shown dotted) and by using modified SPSA (BEV-UE solution) method (shown as a solid line). The performance of RBF Interpolation is positive with a root mean square error between the predicted value and the actual value for the TSTT and TSNEC models as 465.17 and 51.62 respectively.



Figure 2.5 Values of TSTT and TSNEC via RBF and Traffic Assignment Approaches

2.4.3. Budget Sensitivity Analysis

A sensitivity analysis with respect to budget was performed. The results of budget sensitivity analysis in terms of optimal values of TSTT and TSNEC are shown in Table 2.4. For a budget of \$3.6 million, Model 1 of TSTT minimization results in an optimal TSTT value of 374,258 and its TSNEC is computed as 28,828. Similarly, Model 2 of TSNEC minimization results in an optimal TSNEC value of 25,612 and its TSTT is computed as 375,714. Two other budget scenarios were considered to assess model performance. One lower budget of \$3.4 million (i.e., 5% less than the base budget of \$3.6 million), and one higher budget of \$3.8 million (i.e., 5% more than the base budget) are used as two more budget scenarios. Optimal and computed values for TSTT and TSNEC respectively for a budget of \$3.4 and \$3.8 million are also presented in Table 2.4.

Figure 2.6(a)-(c) shows the percentage of DWC implemented in the 18-link network with a budget of \$3.4, \$3.6, and \$3.8 million respectively with the objective function TSTT. Similarly, Figure 2.6 (d)-(f) shows the percentage of DWC implemented in the 18-link network with a budget of \$3.4, \$3.6, and \$3.8 million respectively when the objective function is TSNEC. Overall, six scenarios were analyzed considering three budget levels for each objective function TSTT and TSNEC as summarized in Table 2.4. These numerical experiments provide some useful insights and are presented next.

Objective Function	Budget	Budget Scenario	TSTT Value	TSNEC Value
TSTT	\$3.6 million	Base	374,258*	28,828
TSNEC	\$3.6 million	Base	375,714	25,612*
TSTT	\$3.8 million	1.05 x Base	374,455*	32,695
TSNEC	\$3.8 million	1.05 x Base	375,607	24,308*
TSTT	\$3.4 million	0.95 x Base	374,250*	31,021
TSNEC	\$3.4 million	0.95 x Base	375,550	27,582*

 Table 2.4 Result of Sensitivity Analysis with Respect to Budget

Note: * shows optimal objective function value

The first observation from Figure 2.6 is that if two links are in the same sub-region (please refer to Figure 2.6 for sub-region layout), both Model 1 and 2 tend to apply DWC on only one link and the other would not receive any. However, the net gain in range due to DWC recharging among various path for a given OD pair do not suffer too much from this since each path has at least one link covered with DWC. This conforms to the constraint of equity in resource distribution, and in this case, all sub-regions are treated equally with the same funding coefficient ε_d .

The second observation is that links 8 and 9, which are in the middle of the network, receive DWC treatment in all budget scenarios and models. In general, links which are traversed by multiple paths (or shared link) are prioritized for DWC. Both links 8 and 9 are part of three used paths, which is higher than any other links. As a result, an investment on DWC on links 8 and 9 can be considered more cost-effective than others since those links provide services to multiple paths. The rationale behind this prioritization is in a tight budget scenario, if DWC facilities are implemented on links serving only one path, there would not be sufficient facilities to ensure all vehicles completing their trip without battery depletion. However, the amount of DWC implemented on links 8 and 9 should take into account the planner objective (i.e. TSTT and TSNEC). In contrast, links 16, 17, and 18 do not have any DWC treatment since those links constitute only one path.





The third observation from Figure 2.6, is that even under the same budget scenario, two different objective functions lead to two distinctly different results (set of y_a) implying that the optimal location of DWC facility will differ based on agency's objective (TSTT versus TSNEC). The TSNEC model favors the centralized approach which is shown in Figure 2.6(d) and Figure 2.6(f) where only 4 links out of 18 links are selected for DWC. The TSNEC objective function incentivizes the DWC plan to recharge as much vehicle as possible. One can expect higher traffic flows on links implemented with longer length of DWC which ultimately enhances the objective function value of TSNEC. In the example of links 8 and 9, the amount of DWC implemented is considerably higher than the others in all budget scenarios. In contrast, the TSTT model prefers a disperse approach toward DWC Implementation because in this case, travel time is a concern in the objective function. One disadvantage in the viewpoint of DWC facilities concentrating on selected links is that those links will attract more users leading to an increase in traffic flow and ultimately to higher travel time. If DWC facilities are implemented in a sprawling approach, users will have multiple choices for travel while gaining DWC benefits and network flow will be distributed more evenly to suit the planner's objectives of TSTT. In particular, looking at coverage (what percentage of link length) of DWC suggested by proposed models on links 8 and 9, we observe that for every budget scenario, TSTT based model tend to suggest smaller coverage of DWC on these two links compared to that suggested by TSNEC based model. This is due to the fact that TSTT favors the sprawling approach for DWC to avoid congestion and thereby provides multiple DWC enabled routes to BEV drivers. The difference between these two models is further reinforced numerically by Table 2.4, which indicates that for each budget scenario, the TSTT model results in a lower TSTT value and higher TSNEC value compared to the TSNEC model and vice versa.

2.5. CASE STUDY

2.5.1. Montgomery County Network

The proposed framework is applied to the Montgomery County network in Maryland as the case study to attest to the applicability of the proposed approach for real-size networks in practice. Montgomery is the most populous county in the state with a population close to one million, 400,000 households, and 600,000 employments. The County boundary and transportation network are presented in Figure 2.7. The County contains parts of the heavily traveled roadways in the Washington DC-Baltimore region (Washington DC is referred to as Washington in the remainder of the paper). The County has an extensive highway network with the Capital Beltway (or Interstate-495), which surrounds Washington, passing through Montgomery County. Interstate-270 forms one leg of an interstate triangle between Washington DC, Baltimore City, and Frederick city. The County also contains a portion of route 29, one of the major state routes, which traverses the Washington and Baltimore beltways. The Montgomery network consists of 4,420 links, 1,752 nodes, 225 of which are Origin-Destination nodes, and 34,187 Origin-Destination pairs with non-zero demand. The demand in the morning peak hour period is 3,564,993 vehicles. Montgomery County has an extensive continuous emission monitoring (CEM) program, and the mission is to examine emission reduction strategies. This paper is geared towards this mission by proposing an emission reduction strategy using the proposed DWC implementation model to facilitate the adoption of BEVs. However, the proposed methodology can be extended to other regions as well.



Figure 2.7 Montgomery County Transportation Network

2.5.2. Equity in Transportation Funding

The Montgomery County area is divided into smaller areas for the purpose of allocating resources which are called Transportation Policy Regions. The resources of each Policy Region are meant for the investment into the road exclusively confined within that area. In addition, the Montgomery County planning commission defines four levels of funding priority for each Transportation Policy Region. These four levels of funding raise the problem of equitable distribution of resources, which restrains the optimization model. The division of The Transportation Policy Region and its level of funding priority are shown in Figure 2.8.



Figure 2.8 Transportation Policy Region by Funding Level

The higher-ranking Policy Region tends to be closer to District of Columbia and along Interstate 270 such as Silver Spring CBD or Bethesda CBD. These areas tend to be quite small. In contrast, other areas that are on the lower side of ranking are located in remote areas. They are characterized by a larger area and are responsible for longer road land miles. In the model, these Transportation Policy Regions and funding priority are treated as sub-regions while incorporating the equity in resource distribution constraint in Equation (9).

2.5.3. DWC Implementation Plan for the Study Area

Model 1 and Model 2 have been implemented to the Montgomery County network for deciding the DWC implementation plan for two objectives namely, minimizing the TSTT and TSNEC. The optimization model is implemented based on the assumption of 100 million dollars budget and 60 minutes of initial recharging time or 30 miles in initial range for BEVs in all routes. The results of DWC facility location plan for TSTT and TSNEC minimization scenarios are presented in Figure 2.9(a) and Figure 2.9(b) respectively.

Insights from the numerical experiments of the two scenarios presented in Figure 2.9(a) and Figure 2.9(b) emphasize the importance of Interstate 270 as the models suggest a high percentage of DWC implementation along this highway. However, only several intermittent segments of Interstate 270 are suggested for DWC in both scenarios, which indicates that both scenarios prefer the non-contiguous segments for DWC. In particular, suggested DWC implemented segments on this highway are located where on and off-ramp movements of several traffic paths coincide with each other rather than on the non-weaving portion of Interstate highway. The scattered approach of implementing DWC can be more efficient compared to the continuous approach as it leads to larger network coverage under a restricted budget. Other important roads are Georgia Avenue, which connects Silver Spring to the Capital Beltway and Wheaton CBD, and Maryland State Route 200. The area on the southwest of the County such as Potomac does not receive the same treatment of DWC facilities compared to other areas which can be explained by its smaller number of trips (generated from or attracted to) as well as the shorter traveling distance.



(a) DWC Plan for TSTT Model



(b) DWC Plan for TSNEC Model Figure 2.9 Final DWC Implementation Strategy for Montgomery County

The objective function values of the two scenarios were compared with the Do-Nothing scenario (no DWC Implementation). Model 1 (TSTT minimization) experiences a 0.0055% decrease in total system travel time and Model 2 (TSNEC minimization) experiences a 28% decrease in total system net energy consumption. By using the optimal plan from TSTT and TSNEC model, the total system travel time is lowered by 998 million (Vehicles-Minutes) and 400 million (Vehicle-Minutes) respectively. In addition, the total energy recharged by BEVs through DWC under TSTT and TSNEC models are 5.35 million and 5.44 million (Vehicles-kWh) respectively. This indicates that the power requirement for electrifying the county's transportation system will be huge and may demand adequate planning for power generation and distribution.

2.6. CONCLUSIONS

In this paper, we propose a modeling framework for optimally positioning induction-based DWC facilities in a transportation network for BEVs. The framework aims to support transportation planners and engineers in local agencies. A bi-level modeling framework is proposed considering both the different objectives of the planner and network users. In the Upper Level, Total System Travel Time (TSTT) and Total System Net Energy Consumption (TSNEC) are two objectives of the planner considered. In the Lower Level (LL), the user's route choice is modeled subjected to the DWC infrastructure provided by the planner. As a proof-of-concept, an example 18 link network is tested under different budget scenarios to demonstrate the model performance in the TSTT and TSNEC minimization. Results showed that suggested DWC infrastructure investment is different for TSTT and TSNEC minimization, even though there is some commonality between two cases. Upon successful implementation of the 18 link network, the

model is applied to a real-world Montgomery County network from Maryland, USA. The results of the real-world network were intuitive, as model results suggest DWC on major highways and arterials in an intermittent fashion.

The insights from this research will enable planners and policymakers in making informed decisions and for devising plans and policies that are not only optimal from road network perspective but also from the perspective of power grids, transmission losses, and energy efficiency. The analysis results show that for the Montgomery County Case Study, with an assumption of 60 minutes recharging time yielding 30 miles of initial range for the user, a 100 million dollar expense in DWC is required to sufficiently recharge all BEV within the network. The optimal DWC plan of the TSTT model can lower the total system travel time by 0.0055% and the TSNEC model can lower the total system net energy consumption by 28%. Future avenues of research include analysis of DWC network in a mixed environment of conventional vehicles and BEVs; consideration or estimation of power availability from neighborhood electric grids; and induced demand because of DWC implementation.

APPENDIX 2.A

Pseudocode: Modified Constrained Local Metric Stochastic Response Surface

Stage 1. Initialization

- Step 1.1. Generate a set of initial *t* training points (DWC implementation plan) which satisfies all constraints of the optimization problem $Y_0 = \{y_1, y_2 \dots y_t\}$. These points do not necessarily yield the optimal results of the optimization. Each training points y_t has a dimension of *d*.
- Step 1.2. Evaluate the objective function of each training points using the expensive objective function $Z = \{f(y_1), f(y_2) \dots f(y_n)\}$. Sort for the minimum value of the: $Z_{best} = \min(Z)$ at y_{best} .
- Set 1.3. Setup the initial step size $\sigma_n = \sigma_{initial}$; Consecutive Success and Failure: $C_{succ} = 0$; $C_{fail} = 0$; and global successive failure $C_{gsf} = 0$

Stage 2. Iteration. While the termination condition $(n > N_{max} \text{ or } C_{gsf} > C_{gsfmax})$ is not satisfied

Step 2.1. Using the training points $T = \{(y_1, f(y_1)), (y_2, f(y_2)) \dots (y_t, f(y_t))\}$ create or update the response surface $S_n(y)$

Step 2.2. Generate *q* candidates points for each iteration *n*: $C_n = \{y_{n,1}, \dots, y_{n,q}\}$ as follow: For j =1...q:

Generate d uniform random numbers $w_1, w_2 \dots w_n$ in the range [0,1]. Let $I_{pert} = \{i: w_i < i\}$

 p_{sltc} }. If $I_{pert} = \emptyset$, then select j from the set [1,...,d] and set $I_{pert} = \{j\}$

Generate *j*-th candidate solution by: $y_{n,1} = y_{best} + \Delta_{n,j}$ where $\Delta_{n,j}^i = 0$ for all $i \notin I_{pert}$ and

 $\Delta_{n,j}^{i}$ is a normal random variable with mean 0 and standard deviation σ_{n} for all $i \in I_{pert}$

Step 2.3. For each $y_{n,j} \in C_n$

If the candidate point $y_{n,i}$ satisfy all constraints within the optimization,

Evaluate the objective function $S_n(y_{n,j})$ by using the response surface model.

Let
$$S^{min} = \min \{S_n(y_{n,j}), y_{n,j} \in C_n\}$$
 and $S^{max} = \max \{S_n(y_{n,j}), y_{n,j} \in C_n\}$. Compute the score for each $y_{n,j} \in C_n$ for the response surface: if $S^{max} \neq S^{min}$ then $V_n^S = (S_n(y_{n,j}) - S^{min})/(S^{max} - S^{min})$, else $V_n^S = 1$.

Evaluate the minimum distance from the candidate $y_{n,j}$ to training points by

$$D_n(y_{n,j}) = \min_{1 \le i \le n} ||y_{n,j} - y_i||, y_i \in \mathbb{Z}. \text{ The symbol } ||.|| \text{ represents the Euclidean}$$

norm. Let $D^{min} = \min \{D_n(y_{n,j}), y_{n,j} \in C_n\}$ and $D^{max} = \max \{D_n(y_{n,j}), y_{n,j} \in C_n\}.$
Compute the score for distance criterion score for each candidate: if $D^{max} \neq D^{min}$ then
 $V_n^D = (D_n(y_{n,j}) - D^{min})/(D^{max} - D^{min})$ else $D_n^S = 1.$

Step 2.4. Determine the weighted score for each candidate points: $V_n = w_n^S V_n^S + w_n^D V_n^D$. The coefficient w_n^S , w_n^D can be determined as follow:

$$w_n^{S} \begin{cases} v_{mod(n-n_0,k)} & \text{if } mod(n-n_0,k) \neq 0\\ v_k & \text{otherwise} \end{cases} \text{ and } w_n^{D} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{D} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{D} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{D} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{D} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{D} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{D} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an integer and } w_n^{S} = 1 - w_n^{S} \text{ where } k \text{ is an in$$

 v_k is a series of weights in ascending order within the range of [0,1]. Select y^* within the set of candidates points C_n that yields the highest weighted score V_n .

- Step 2.5. Evaluate the expensive objective function for the solution y^* to get the value $Z_{n=}f(y^*)$ and add the point $\{y^*, f(y^*)\}$ to the training points poll *T*.
- Step 2.6. If $Z_n < Z_{best}$ update the current best solution $Z_{best} = Z_n$, update the consecutive success and failures: $C_{succ} = C_{succ} + 1$; $C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = C_{fail} + C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = 0$; $C_{gsf} = 0$ otherwise $C_{fail} = 0$; $C_{fail} =$
 - 1; $C_{succ} = 0$.

Step 2.7. Adjusting the step size and counters:

If C_{succ} exceeds the maximum number of success C_{succ}^{max} , set $\sigma_{n+1} = 2\sigma_n$ and reset $C_{succ} = 0$ If C_{fail} exceeds the maximum number of success C_{fail}^{max} , set $\sigma_{n+1} = \sigma_n/2$, reset $C_{fail} = 0$,

and set $C_{gsf} = C_{gsf} + 1$

Set
$$n = n+1$$

End the while iteration.

Step 3. Conclusion

Return the optimal objective function value Z_{best} and the vector of decision variable $\{y_{best}\}$ when stopping criterion is met. The stopping criterion adopted in the Montgomery Case Study is either the iteration reaches 40 iterations or global successive failure reaches 10.

REFERENCES

- Abdulaal, M., LeBlanc, L.J., 1979. Continuous equilibrium network design models.
 Transportation Research Part B: Methodological 13, 19–32. https://doi.org/10.1016/0191-2615(79)90004-3
- Agrawal, S., Zheng, H., Peeta, S., Kumar, A., 2016. Routing aspects of electric vehicle drivers and their effects on network performance. Transportation Research Part D: Transport and Environment 46, 246–266.
- Bomey, N., 2018. Americans more likely to buy electric cars, AAA study finds [WWW Document]. USA TODAY. URL https://www.usatoday.com/story/money/cars/2018/05/08/electric-cars-aaa/586987002/

(accessed 9.8.19).

- Budhia, M., Boys, J.T., Covic, G.A., Huang, C.-Y., 2013. Development of a Single-Sided Flux
 Magnetic Coupler for Electric Vehicle IPT Charging Systems. IEEE Transactions on
 Industrial Electronics 60, 318–328. https://doi.org/10.1109/TIE.2011.2179274
- Chen, Z., He, F., Yin, Y., 2016. Optimal deployment of charging lanes for electric vehicles in transportation networks. Transportation Research Part B: Methodological 91, 344–365. https://doi.org/10.1016/j.trb.2016.05.018
- Chen, Z., Liu, W., Yin, Y., 2017. Deployment of stationary and dynamic charging infrastructure for electric vehicles along traffic corridors. Transportation Research Part C: Emerging Technologies 77, 185–206. https://doi.org/10.1016/j.trc.2017.01.021
- Chiou, S.-W., 2005. Bilevel programming for the continuous transport network design problem. Transportation Research Part B: Methodological 39, 361–383. https://doi.org/10.1016/j.trb.2004.05.001

- Chow, J.Y.J., Regan, A.C., Ranaiefar, F., Arkhipov, D.I., 2011. A network option portfolio management framework for adaptive transportation planning. Transportation Research Part A: Policy and Practice 45, 765–778. https://doi.org/10.1016/j.tra.2011.06.004
- Dong, J., Liu, C., Lin, Z., 2014. Charging infrastructure planning for promoting battery electric vehicles: An activity-based approach using multiday travel data. Transportation Research Part C: Emerging Technologies 38, 44–55. https://doi.org/10.1016/j.trc.2013.11.001
- Farahani, R.Z., Asgari, N., Heidari, N., Hosseininia, M., Goh, M., 2012. Covering problems in facility location: A review. Computers & Industrial Engineering 62, 368–407. https://doi.org/10.1016/j.cie.2011.08.020
- Farahani, R.Z., Miandoabchi, E., Szeto, W.Y., Rashidi, H., 2013. A review of urban transportation network design problems. European Journal of Operational Research 229, 281–302.
- FHWA, 2015. Appendix A: Highway Investment Analysis Methodology 2015 Conditions and Performance - Policy | Federal Highway Administration [WWW Document]. URL https://www.fhwa.dot.gov/policy/2015cpr/appendixa.cfm#_Toc464549617 (accessed 1.7.20).
- Fuller, M., 2016. Wireless charging in California: Range, recharge, and vehicle electrification. Transportation Research Part C: Emerging Technologies 67, 343–356. https://doi.org/10.1016/j.trc.2016.02.013
- Gao, L., Xie, C., Zhang, Z., Waller, S.T., 2011. Integrated Maintenance and Expansion Planning for Transportation Network Infrastructure. Transportation Research Record: Journal of the Transportation Research Board 2225, 56–64. https://doi.org/10.3141/2225-07

- Hajibabai, L., Bai, Y., Ouyang, Y., 2014. Joint optimization of freight facility location and pavement infrastructure rehabilitation under network traffic equilibrium. Transportation Research Part B: Methodological 63, 38–52. https://doi.org/10.1016/j.trb.2014.02.003
- He, F., Yin, Y., Zhou, J., 2015. Deploying public charging stations for electric vehicles on urban road networks. Transportation Research Part C: Emerging Technologies 60, 227–240. https://doi.org/10.1016/j.trc.2015.08.018
- Huang, Y., Li, S., 2015. Optimal deployment of alternative refueling stations on transportation networks considering deviation paths 27.
- Hwang, I., Jang, Y.J., Ko, Y.D., Lee, M.S., 2018. System Optimization for Dynamic Wireless Charging Electric Vehicles Operating in a Multiple-Route Environment. IEEE Transactions on Intelligent Transportation Systems 19, 1709–1726. https://doi.org/10.1109/TITS.2017.2731787
- Jang, Y.J., Jeong, S., Ko, Y.D., 2015. System optimization of the On-Line Electric Vehicle operating in a closed environment. Computers & Industrial Engineering 80, 222–235. https://doi.org/10.1016/j.cie.2014.12.004
- Jang, Y.J., Jeong, S., Lee, M.S., 2016. Initial Energy Logistics Cost Analysis for Stationary, Quasi-Dynamic, and Dynamic Wireless Charging Public Transportation Systems. Energies 9, 483. https://doi.org/10.3390/en9070483
- Jiang, N., Xie, C., Waller, S.T., 2012. Path-constrained traffic assignment: model and algorithm. Transportation Research Record 2283, 25–33.
- Jing, W., An, K., Ramezani, M., Kim, I., 2017. Location Design of Electric Vehicle Charging Facilities: A Path-Distance Constrained Stochastic User Equilibrium Approach. Journal of Advanced Transportation 2017, 1–15. https://doi.org/10.1155/2017/4252946

- Kitthamkesorn, S., Chen, A., 2017. Alternate weibit-based model for assessing green transport systems with combined mode and route travel choices. Transportation Research Part B: Methodological 103, 291–310. https://doi.org/10.1016/j.trb.2017.04.011
- Ko, Y.D., Jang, Y.J., 2013. The Optimal System Design of the Online Electric Vehicle Utilizing Wireless Power Transmission Technology. IEEE Transactions on Intelligent Transportation Systems 14, 1255–1265. https://doi.org/10.1109/TITS.2013.2259159
- Ko, Y.D., Jang, Y.J., Lee, M.S., 2015. The optimal economic design of the wireless powered intelligent transportation system using genetic algorithm considering nonlinear cost function. Computers & Industrial Engineering, Maritime logistics and transportation intelligence 89, 67–79. https://doi.org/10.1016/j.cie.2015.04.022
- Koh, A., 2007. Solving transportation bi-level programs with Differential Evolution, in: 2007
 IEEE Congress on Evolutionary Computation. Presented at the 2007 IEEE Congress on
 Evolutionary Computation, pp. 2243–2250. https://doi.org/10.1109/CEC.2007.4424750
- Konur, D., Geunes, J., 2011. Analysis of traffic congestion costs in a competitive supply chain. Transportation Research Part E: Logistics and Transportation Review 47, 1–17. https://doi.org/10.1016/j.tre.2010.07.005
- Kuby, M., Lim, S., 2007. Location of Alternative-Fuel Stations Using the Flow-Refueling Location Model and Dispersion of Candidate Sites on Arcs. Netw Spat Econ 7, 129–152. https://doi.org/10.1007/s11067-006-9003-6
- Kuby, M., Lim, S., 2005. The flow-refueling location problem for alternative-fuel vehicles. Socio-Economic Planning Sciences 39, 125–145. https://doi.org/10.1016/j.seps.2004.03.001
- Kumar, A., Haque, K., Mishra, S., & Golias, M. M. (2019). Multi-criteria based approach to identify critical links in a transportation network. Case Studies on Transport Policy, 7(3), 519-530.
- Kumar, A., Mishra, S., 2018. A simplified framework for sequencing of transportation projects considering user costs and benefits. Transportmetrica A: Transport Science 14, 346–371.
- Kumar, A., Peeta, S., 2014. Slope-based path shift propensity algorithm for the static traffic assignment problem. International Journal for Traffic and Transport Engineering 4, 297– 319.
- Lau, T.W., Chung, C.Y., Wong, K.P., Chung, T.S., Ho, S.L., 2009. Quantum-Inspired Evolutionary Algorithm Approach for Unit Commitment. IEEE Transactions on Power Systems 24, 1503–1512. https://doi.org/10.1109/TPWRS.2009.2021220
- Lim, S., Kuby, M., 2010. Heuristic algorithms for siting alternative-fuel stations using the Flow-Refueling Location Model. European Journal of Operational Research 204, 51–61. https://doi.org/10.1016/j.ejor.2009.09.032
- Lin, Z., Li, J.-M., Dong, J., 2014. Dynamic Wireless Power Transfer: Potential Impact on Plugin Electric Vehicle Adoption (SAE Technical Paper No. 2014- 01–1965). SAE International, Warrendale, PA. https://doi.org/10.4271/2014-01-1965
- Liu, H., Wang, D.Z.W., 2017. Locating multiple types of charging facilities for battery electric vehicles. Transportation Research Part B: Methodological 103, 30–55. https://doi.org/10.1016/j.trb.2017.01.005
- Liu, J., Wang, X., Khattak, A., 2016. Customizing driving cycles to support vehicle purchase and use decisions: Fuel economy estimation for alternative fuel vehicle users. Transportation

Research Part C: Emerging Technologies 67, 280–298. https://doi.org/10.1016/j.trc.2016.02.016

- Liu, Z., Song, Z., 2017. Robust planning of dynamic wireless charging infrastructure for battery electric buses. Transportation Research Part C: Emerging Technologies 83, 77–103. https://doi.org/10.1016/j.trc.2017.07.013
- Los, M., Lardinois, C., 1982. Combinatorial programming, statistical optimization and the optimal transportation network problem. Transportation Research Part B: Methodological 16, 89–124.
- Lukic, S.M., Pantic, Z., 2013. Cutting the Cord: Static and Dynamic Inductive Wireless Charging of Electric Vehicles. IEEE Electrification Magazine 1, 57–64. https://doi.org/10.1109/mele.2013.2273228
- Marcotte, P., 1983. Network design problem with congestion effects: A case of bilevel programming. Mathematical Programming 34, 142–162. https://doi.org/10.1007/BF01580580
- Mathew, T.V., Sharma, S., 2009. Capacity Expansion Problem for Large Urban Transportation Networks. Journal of Transportation Engineering 135, 406–415. https://doi.org/10.1061/(ASCE)0733-947X(2009)135:7(406)
- Miller, J.M., Jones, P.T., Li, J.-M., Onar, O.C., 2015a. ORNL Experience and Challenges Facing Dynamic Wireless Power Charging of EV's. IEEE Circuits and Systems Magazine 15, 40–53. https://doi.org/10.1109/MCAS.2015.2419012
- Miller, J.M., Onar, O.C., Chinthavali, M., 2015b. Primary-Side Power Flow Control of Wireless Power Transfer for Electric Vehicle Charging. IEEE Journal of Emerging and Selected Topics in Power Electronics 3, 147–162. https://doi.org/10.1109/JESTPE.2014.2382569

- Mishra, S., Kumar, A., Golias, M.M., Welch, T., Hossein, T., Haque, K., 2016. Transportation Investment Decision Making for Medium to Large Transportation Networks.
 Transportation in Developing Economies, A Journal of the Transportation Research Group of India (TRG) 2, 1–9.
- Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. Transportation Research Part B: Methodological 103, 87–110. https://doi.org/10.1016/j.trb.2017.02.004
- Panchal, C., Stegen, S., Lu, J., 2018. Review of static and dynamic wireless electric vehicle charging system. Engineering Science and Technology, an International Journal 21, 922– 937. https://doi.org/10.1016/j.jestch.2018.06.015
- Peeta, S., Ziliaskopoulos, A.K., 2001. Foundations of dynamic traffic assignment: The past, the present and the future. Networks and spatial economics 1, 233–265.
- Pelletier, S., Jabali, O., Laporte, G., 2016. 50th Anniversary Invited Article—Goods Distribution with Electric Vehicles: Review and Research Perspectives. Transportation Science 50, 3– 22. https://doi.org/10.1287/trsc.2015.0646
- Pishvaee, M.S., Farahani, R.Z., Dullaert, W., 2010. A memetic algorithm for bi-objective integrated forward/reverse logistics network design. Computers & Operations Research 37, 1100–1112. https://doi.org/10.1016/j.cor.2009.09.018
- Powell, M.J.D., 1992. The theory of radial basis function approximation in 1990. Advances in numerical analysis 105–210.
- Regis, R.G., 2011. Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. Computers & Operations Research 38, 837–853. https://doi.org/10.1016/j.cor.2010.09.013

- Riemann, R., Wang, D.Z.W., Busch, F., 2015. Optimal location of wireless charging facilities for electric vehicles: Flow-capturing location model with stochastic user equilibrium.
 Transportation Research Part C: Emerging Technologies 58, 1–12.
 https://doi.org/10.1016/j.trc.2015.06.022
- Sathaye, N., Kelley, S., 2013. An approach for the optimal planning of electric vehicle infrastructure for highway corridors. Transportation Research Part E: Logistics and Transportation Review 59, 15–33. https://doi.org/10.1016/j.tre.2013.08.003
- Sharma, S., & Mishra, S. (2013). Intelligent transportation systems-enabled optimal emission pricing models for reducing carbon footprints in a bimodal network. Journal of Intelligent Transportation Systems, 17(1), 54-64.
- Sheffi, Y., 1985. Urban Transportation Networks: Equilibrium Analysis With Mathematical Programming Methods. Prentice Hall.
- Strehler, M., Merting, S., Schwan, C., 2017. Energy-efficient shortest routes for electric and hybrid vehicles. Transportation Research Part B: Methodological 103, 111–135. https://doi.org/10.1016/j.trb.2017.03.007
- USDOE, 2018. Alternative Fuels Data Center: Electric Vehicle Charging Station Locations [WWW Document]. URL https://www.afdc.energy.gov/fuels/electricity_locations.html#/analyze?show_private=tru e&show_planned=true (accessed 7.25.18).
- Wang, Y.-W., Lin, C.-C., 2009. Locating road-vehicle refueling stations. Transportation Research Part E: Logistics and Transportation Review 45, 821–829. https://doi.org/10.1016/j.tre.2009.03.002

- Xi, X., Sioshansi, R., Marano, V., 2013. Simulation–optimization model for location of a public electric vehicle charging infrastructure. Transportation Research Part D: Transport and Environment 22, 60–69. https://doi.org/10.1016/j.trd.2013.02.014
- Xie, C., Jiang, N., 2016. Relay requirement and traffic assignment of electric vehicles. Computer-Aided Civil and Infrastructure Engineering 31, 580–598.
- Xie, C., Wang, T.-G., Pu, X., Karoonsoontawong, A., 2017. Path-constrained traffic assignment: Modeling and computing network impacts of stochastic range anxiety. Transportation Research Part B: Methodological 103, 136–157. https://doi.org/10.1016/j.trb.2017.04.018
- Xu, M., Meng, Q., Liu, K., Yamamoto, T., 2017. Joint charging mode and location choice model for battery electric vehicle users. Transportation Research Part B: Methodological 103, 68–86. https://doi.org/10.1016/j.trb.2017.03.004
- Zhang, A., Kang, J.E., Kwon, C., 2017. Incorporating demand dynamics in multi-period capacitated fast-charging location planning for electric vehicles. Transportation Research Part B: Methodological, Green Urban Transportation 103, 5–29.

https://doi.org/10.1016/j.trb.2017.04.016

This page is intentionally left blank.

3. TRAFFIC GRAPH CONVOLUTIONAL NETWORK FOR DYNAMIC URBAN TRAVEL SPEED ESTIMATION

ABSTRACT

The dynamic urban link travel speed estimation (DU-LSE) problem has been studied extensively with approaches ranging from model to data driven since it benefits multiple applications in transport mobility, especially in dense cities. However, with drawbacks such as heavy assumption in model-driven and not being capable for big city network in data-driven, there has not been a consensus on the most effective method. This study aims to develop a Sequential Three Step framework to solve the DU-LSE problem using only the passively collected taxi trip data. The framework makes use of two deep learning models namely Traffic Graph Convolution (TGCN) and its recurrent variant TGCN_{lstm} to capture both spatial and temporal correlation between road segments. The proposed framework has three advantages over similar approaches: (1) it uses only the affordable taxi data and overcomes the data's incompleteness both in spatial (full GPS trajectory is not available) and temporal (incomplete historic time-series) domain, (2) it is specifically designed to preserve the directionality nature of traffic flow, and (3) it is capable for large networks. The model results and validations suggest the framework can achieve high enough accuracy and will provide valuable mobility data for cities especially those without traffic sensing infrastructure already in place.

Keywords: Dynamic Traffic States Estimation; Link-level; Graph Convolution Network; Taxi Trip Data;

3.1. OVERVIEW

In the 21st century, urbanizations are happening in countries around the globe at an extraordinary speed, which is reflected by the increase in percentage of urban population from 39% in 1980 to 58% in 2019 (World Bank, 2019). This rapid urbanization will pose new mobility needs and lay more stress in the transportation system especially in cities with old infrastructure that could not keep up with even the current demand. City planner can ask funds for rehabilitation and expansion of the current infrastructure such as opening more lanes, building new roads, or transit lines. However, the cost effectiveness of these investments is quite often questionable since new infrastructure does not necessarily translate into better mobility as shown in the example of the Braess Paradox (Frank, 1981). On the other hand, there has been rising attention in improving urban mobility through Intelligent Transportation System (ITS), which leverages recent state-of-the-art technologies to increase the effectiveness of the system without major investment in the current infrastructure.

In the United States, the States' Department of Transportation (DOT) have already implemented many applications of ITS such as Ramp Metering, which limits the number of vehicle entering highway during peak hour to avoid a costly congestion, or Traffic Signal Coordination, which synchronizes multiple adjacent intersections to enhance a selected directional flow (DOT, 2019). One main challenge that has been consistently debated in ITS is traffic state estimation (TSE), which is the process of inferring traffic state variables (e.g., flow, travel time, density, etc..) with partially observed traffic data (Seo et al., 2017). However, studies in TSE vary greatly in the scope of estimation. In spatial scope, certain studies estimate a selected set of road segments such as major roadways (e.g., highways) and/or those implemented with traffic volume sensors (e.g., Inductive Loop). In temporal scope, papers are more focused on short-term prediction (Ermagun and Levinson, 2018; Thapa et al., 2022) since the estimation is more reliable. The research area of estimating all road segments in a dense urban network at every time of the day is new and unexplored. To this end, we introduce the Dynamic Urban Link Travel Speed Estimation (DU-LSE) problem which specifically aims at computing link travel speed for every link within a network, at different time of the day and day of the week. The outcome of DU-LSE can be beneficial to a wide variety of application in ITS such as monitoring traffic jam, estimating time of arrival, route planning (Kumar et al., 2019; Nantes et al., 2016; Papageorgiou et al., 2003; Seo et al., 2017; Xu et al., 2020), and even for the emerging technology of autonomous vehicle operations (Fountoulakis et al., 2017; Khan et al., 2017). There are private companies such as INRIX, HERE, or TOMTOM offering traffic estimation services but there are three main concerns for city planner who wishes to adopt this method. First, the estimation coverage of these services may be limited to major segments of the road network such as interstate and in urban mobility, knowledge of both major and minor segments (i.e., Central Business District) is significantly more beneficial. Second, since these services require data collected either from probe vehicle travel program or stationary traffic sensor, cities that do not have these infrastructures simply cannot utilize this method. Third, estimation for commercial companies is still not as accurate and reliable since there are multiple observations of the original data, and it is not clear on the sampling method. Therefore, there should be an independent alternative framework to compare such information.

DU-LSE has been studied extensively with methods ranging from model-driven to datadriven approaches (Seo et al., 2017). Conventional model-driven approach relies on theoretical principles represented by mathematical formulation to describe the physical traffic flow. On the other hand, data-driven approach such as deep learning relies on a vast amount of data and learns its hidden pattern through the optimization of model's weights and biases. This approach has recently gained traction due to two main factors. *First*, there is an unprecedented growth rate in the amount of data generated even just through passive daily actions. A study in 2013 found that 90% of the world data at that time was created during 2011-2012 alone (Ralph, 2013). This abundant source of data functions as a fuel to improve the accuracy of deep learning models. Second, continual advancement in computing power has paved the way for processing these large data such as IBM's Infosphere processing at a rate up to 120,000 GPS points per second (Biem et al., 2010). However, in the ITS field and especially in solving DU-LSE, studies using data-driven approach often require data from sensing infrastructure such as inductive loop detectors, license plate recognition devices, or 360° cameras. Cities that wish to take advantage of these studies for their ITS system either need to have these infrastructures already in place or invest in a new sensing infrastructure which can be costly. This poses a problem for cities with emerging population and economy especially those in developing countries. Thus, the use of data that is a byproduct of daily activities and publicly available is desirable. In the case of urban mobility data, taxi trip dataset has great potentials because not only it meets all these criteria but also is abundantly available, especially in dense urban areas. For example, as a result of the Open Data Law signed into effect as of 2012, the New York City Taxi & Limousine Commission (NYC-TLC) has released an astonishing record of 1.1 billion taxi trips from 2009 to 2015 (City of New York, 2019). Due to its enormous size, this dataset is perfectly suitable to fuel a deep learning model aiming to solve the DU-LSE problem. Furthermore, the taxi data has extensive both spatial and temporal coverage, which is demonstrated later in Section 3.4 case study.

Although there have been several papers devoted to DU-LSE (Yu et al., 2019; Wu et al., 2015; Liu et al., 2019; Sekuła et al., 2018; Zhan et al., 2013), the literature has not yet reached a

consensus on the most effective method because of drawbacks in the methodology such as heavy assumption in model-driven and lack of scalability in data-driven approaches. We shall present a more extensive review of the literature on DU-LSE and identify these drawbacks in Section 3.2. Therefore, this objective of this paper is to *develop a sequential three step framework that leverages a single dataset of taxi trip to estimate historical complete network link travel speed, disaggregated by time of the day.*

The remainder of the paper is organized as follows: Section 2 reviews the related literature in the domain of deep learning model in ITS and specifically, the DU-LSE problem. In Section 3.3, we present in detail each step in the sequential three step framework. This framework is then applied on the New York City Taxi dataset in Section 3.4. Section 3.5 discusses the evaluation validation of each step in the framework and finally, Section 3.6 concludes the paper with the discussion on findings, model performances, and avenues of future research.

3.2. LITERATURE REVIEW

The number of studies using deep learning model in ITS is increasing with applications ranging from ridesharing services (Geng et al., 2019; Ke et al., 2017; Yao et al., 2018), bikesharing services (Lin et al., 2018), to car parking demand prediction (Yang et al., 2019). These studies usually utilize a combination of two methods to capture both spatial and temporal relation. For spatial relations, studies utilize variations of Convolutional Neural Network (CNN) which have already achieved tremendous success in the field of image recognition and video classification. The main idea of CNN is to aggregate information of pixels located inside a predefined kernel filter and this filter is then transported through the rows and columns of the image's pixel to learn and identify common patterns. However, CNN has difficulty in

implementation for road network. Unlike image dataset, which is an Euclidean type data, road network is a graph structure data and there is no notion of direction but only notion of node connectivity. One possible solution is using Graph Convolutional Network proposed by Kipf and Welling (2017). The approach's main idea is that a host node would gather information from its neighbor one "hop" away from itself. The procedure can be repeated multiple times to reach to further neighbors. One thing to remind from this study is it is node-based, which means only node information can be processed whereas information of the links connecting these nodes are ignored.

For temporal relation, variations of Recurrent Neural Network (RNN) especially those in form of Long-Short Term Memory (LSTM) are utilized. The main idea of RNN is, it would take both the current and previous observations as input and the operation is repeated at each state of time and hence the name "Recurrent". However, RNN suffers from the exploding or vanishing gradient problem where changes of model's weights and biases during training are either too small or too big that it could not achieve convergence in the loss function. LSTM addresses this problem by introducing an internal state value at which the gradient flow is uninterrupted and thus avoids the exploding/vanishing gradient problem (Hochreiter and Schmidhuber, 1997). This approach has been the state-of-the-art model for capturing temporal relation in situation such as bike-sharing and car parking demand prediction (Lin et al., 2018; Yang et al., 2019). However, most studies using LSTM rely on a complete historic time-series dataset to effectively train the model. Traditionally, LSTM was often implemented in a local fashion where a node would only look back at its historical data and ignores its neighbor's, but more recent papers are starting to embed connectivity into the formulation and promote spatial message passing. For an in-depth review of deep learning model in ITS, we recommend the survey by Wang et al. (2019) where

the authors goes through various techniques and applications. One main take away from this survey is deep learning lack the interpretability power if not formulated appropriately.

In the DU-LSE problem, the common practice for achieving the link-level travel speed often involves a theoretical four step planning process, which are (1) trip generation, (2) trip distribution, (3) modal split, and (4) traffic assignment (Sheffi, 1975). However, the four-step planning process requires the planner to issue a Household Travel Survey, which is costly and conducted approximately once every 10 years thus results in a low reliability when being implemented in later period of the collecting cycle. In addition, the process relies on stringent assumptions such as Wardrop User Equilibrium that do not necessarily hold true in the real world (Yu et al., 2017). More recent research can be categorized into either model-based or datadriven approach. In model-based, (Yeon et al., 2008) used probabilistic breakdown for freeway segments and Discrete Time Markov Chain to estimate travel time with data from microwave sensors and CCTV on US202 in Philadephia, PA. With New York taxi data, (Zhan et al., 2013) proposed a two steps framework where route choice is first estimated by multinomial logit model and travel time is calculated by an optimization model minimizing the expected and observed path travel time. Another high-performance algorithm was introduced by (Wu et al., 2015) where the author used convex optimization coupled with dimensionality reduction scheme and projected gradient algorithm to estimate traffic. The data is a fusion between vehicle count via sensors and cellular network along I-210 region of Los Angeles. Other notable approach in model-based are Tucker decomposition-based imputation (Tan et al., 2013) for PeMS data in Sacramento County and maximum likelihood (Jenelius and Koutsopoulos, 2013) for GPS probes in Stockholm, Sweden. In data-driven approach, early work includes a three-layer neural network for low-pooling frequencies probe vehicle data (Zheng and Van Zuylen, 2013),

denoising stacked autoencoders for Caltrans PeMS (Duan et al., 2016), neural network with linear and hyperbolic layers to capture sharp non-linearity of traffic flow in case of special event such as a Chicago Football Game (Polson and Sokolov, 2017), and (Sekuła et al., 2018)'s neural network with a profiling model for ATR station and vehicle probe data in Maryland. A more contemporary approach with high performance is the 3D-TGCN by (Yu et al., 2019). The model makes use of Graph Convolutional Network and Dynamic Time Wraping, and it is applied to the PeMSD7 (2012) and PEMS-BAY (2017) data from Caltrans. One major contribution of this method is DTW introduces less training parameters compared to RNN-based model and the training process is more efficient.

Research gaps

After reviewing related literature, we identify four research gaps as follows:

First, most studies rely on well-established and dense dataset that provide a complete historic time-series collected from traffic sensors such as Inductive Loop Detector or License Plate Recognition (Cui et al., 2018; Diao et al., 2019; Wu et al., 2015; Yu et al., 2019; Zhu et al., 2018), floating probe vehicle GPS data (Cui et al., 2018; Yu et al., 2017; Zhu et al., 2018), mobile phone data (Wu et al., 2015; Zhu et al., 2018) or even a combination of all of it. However, these datasets require a supporting infrastructure already being implemented in the first place (e.g., sensors) and thus is not applicable to city that does not have such hardware or software system for the entire network. Even if a city decides to implement new sensing facilities, not only it would cost more but also pose a question of where to locate these devices. Other data sources such as GPS or mobile data is not readily available to most researchers and pose concerns about personal security.

- 2. Second, studies using variation of Graph Convolutional Network (Cui et al., 2018; Diao et al., 2019; Kawasaki et al., 2019; Kipf and Welling, 2017) are mostly node-based and link information (i.e., road length and number of lanes) are ignored. This also results in an undirected graph where there is only a notion of two nodes being connected by a link but no notion of direction. Therefore, these studies have limitation in capturing the directional flow nature which is inherent in traffic behavior. An example would be, the same highway connecting the suburb and the downtown area, north-bound traffic would differ greatly compared to south-bound traffic at a specific point of time. In addition, the north and south-bound road are represented as two separate links connecting the same origin-destination pair, which cannot be reflected in an undirected graph commonly used in GCN.
- 3. *Third*, studies aiming to capture temporal relation by using variations of LSTM (Cui et al., 2018; Lin et al., 2018; Liu et al., 2019) only apply the technique locally, which is a road segment will only look at its own historical data and not its neighbor.
- 4. *Finally*, most papers only consider network with moderate size (i.e., up to 500 links) and aim to capture only part of the city transportation network (i.e., highway) especially at segments where traffic sensors are located. In dense urban area, traffic estimation for all links is exponentially more useful than that of for selected highway segments.

Paper Contribution:

First, we use only the publicly available Taxi Trip data, which is a byproduct of daily
activities and requires minimal investment for City Planners who wish to implement this
framework. Taxi data is different than that of probe vehicle data or location-based
services data for which the city needs to purchase the data. But for taxi data, many cities

have memorandum of understanding with the taxi companies to share the data if they would like to operate in the city with assurance in user privacy. However, the dataset has two main disadvantages that prevents it from being widely used in the DU-LSE field. These are (i) each trip does not have full GPS traces but only the pickup and drop-off coordinates and even these data are completely random, sparse, and far between each other in a network and (ii) no complete historic time-series are provided. The sequential three step framework presented in this paper helps to address this problem of incomplete information so that city planner can use this readily available data for their task of determining DU-LSE. To the best of our knowledge, this is the first paper to estimate network-wide travel speed throughout the day with only one single taxi trip data.

- 2. Second, our paper captures the directional flow, which is to estimate travel speed on both directions of a road segment. This is challenging because conventional GCN works with undirected graph and there is only notions of node-connectivity and neighbor's message passing. We also want to preserve directional flow nature of traffic as explained in the following illustration. At an intersection, there are two traffic flows going in and out of the intersection in the East directions such as North, West, and South are not related. This notion applies to the remaining directions and traffic flow as well. To this end, our paper contributes Traffic Graph Convolutional Network to capture this directional flow nature by using a node-link embedding technique in conjunction with a modified directional adjacency matrix.
- Third, for capturing temporal relation, our paper introduces TGCN_{1stm} along with an appropriate model architecture to allow the node to look back at not only of its historical

data but also its neighbor's. TGCN_{1stm} makes uses of the core idea from TGCN, such as modified directional adjacency matrix, and LSTM architecture (Hochreiter and Schmidhuber, 1997).

4. Fourth, this paper makes use weight sharing to reduce training parameters and increase training efficiency. At a lower level of TGCN, each node in the network has its own neural network of which computational graph is created by branching from the host node out to its neighbor. By repeating this operation multiple times, the host node can gather information multiple "hops" away. If two nodes *i*,*j* are connected together via a real-world road segment, then only one weight *w*_{ij} is assigned to this pair and *w*_{ij} is shared across multiple computational graphs. We will explain this notion further in Section 3.3.5 of Computational Graph. This facilitates our case study to estimate dynamic urban link travel speed for large network within reasonable computational time. The number of training parameters is independent of the size of the taxi dataset, but the model accuracy benefits greatly from this taxi dataset size.

3.3. METHODOLOGY

Problem Definition:

We first describe the problem as follows. The framework takes input of a taxi trip dataset where each taxi $i \in I$ contains the information about (1) the pickup and dropoff coordinates, pickup and dropoff time, and travel distance. This information is represented as a tuple:

$$\{x_i^{start}, y_i^{start}, x_i^{end}, y_i^{end}, t_i^{start}, t_i^{end}, d\}, \quad \forall i \in I$$

The output of the model is the travel speed of every link in the network at every time of the day as follows: y_{at} , $\forall a \in A$, $t \in T$ where A is the set of links and T is the set of Time period.

All notations for this paper are summarized and presented in Appendix A. In addition, physical meaning of every notation will be mentioned and explained in text after it is introduced.

Overall Framework

In order to solve the DU-LSE using taxi trip data, we use a Sequential Three Step (S3S) framework which consists of (1) Path Choice Prediction, (2) Partial Link Travel Time Prediction, and (3) Dynamic Urban Link Travel Speed Estimation. The overall interactions between these steps are as follows. Step 1 takes the raw input which is taxi trip data and produces an output of predicted path choice for all taxi trips. This output is fed as an input to step 2 which then produces an output of partial link travel time. Step 3 then takes the step 2 output as input and produces the ultimate result which is the DU-LSE. Our study uses link-level travel speed as representation of traffic state instead of travel time for reasons discussed later in step 3. It's important to note that one can readily compute link travel time by dividing link length with travel speed. A schematic view of S3S framework's architecture and details of each step are provided in Figure 3.1. For the analysis of step 1 and 2, we first subset the taxi trip from the master dataset by a 30-minute time window (i.e., 01/01/2014 06:00-06:30). Steps 1 and 2 are then independently processed for each time window and repeated until every time windows within the analysis period (i.e., from the date 01/01/2014 to 12/31/2015) are executed. Therefore, we do not include the subscript of time window t in the description and formulation of step 1 and 2. The S3S framework is the key component of overcoming the incompleteness of taxi trip data as mentioned in Section 2.

Framework's Application on Similar Data Types

Besides a single taxi dataset with only the pickup and dropoff coordinates available, our research also works well or even better with similar but denser spatio-temporal dataset. One such example is the extension of taxi dataset including GPS traces. GPS traces provide two fundamental advantages over the original taxi dataset which are (1) the path chosen by the taxi between the Origin-Destination pair is known and (2) partial link travel time can be derived directly by comparing the timestamp between the start and end of a road segments for a taxi trip. Therefore, our framework can bypass step 1 and 2 completely because the path choice and partial link travel time derived from GPS traces is theoretically more accurate than the estimation from step 1 and 2. GPS traces can then be introduced directly to the input of step 3. The disadvantages of using GPS traces are (1) such data might not be available in some city due to privacy reason; and (2) the GPS-implemented fleet is biased within certain areas and does not cover the entire network. Our framework provides flexibility and robustness by including step 1 and 2 and practitioner can make a choice based on the data they have.



Figure 3.1 Schematic Representation of The Sequential Three Steps Framework

3.3.1. Step 1: Path Choice Prediction (PCP)

Given the input of the pickup and drop-off GPS coordinates and the observed travel distance of a taxi trip, this step aims to infer the path taken by that taxi trip. PCP step includes the following procedure. First, the pickup and drop-off coordinates are projected to the nearest link via a perpendicular line. This projected point is called the mapped point. Then the mapped point is projected to the nearest intersection which is then named as an intermediate node. We assume that the taxi would not make a U-turn for either picking up or dropping off customer which results in only one unique pair of intermediate nodes for pickup or drop-off. Figure 3.2 shows

both a schematic view and real-world application of the data mapping step. We also calculate the distances between the map points and intermediate nodes, which are named D_1 and D_2 respectively. Second, for each pair of intermediate nodes, we generate k-shortest-paths using Yen's Algorithm (Yen, 1971) and calculate the traveling distance for each path. We need to add this travel distance, D_1 , and D_2 together in order to get the predicted path distance. A pseudocode for Yen's Algorithm is provided in Appendix B. One drawback of Yen's Algorithm is the paths generated only vary slightly between each other and there are a lot of overlap links. In the real world, drivers are often presented with a diverse set of alternative paths with limited overlapping and the Yen's Algorithm cannot easily captures this notion. There are other algorithms aiming at improving speed or increasing diversity such as Hoffman's Algorithm (Hoffman and Pavley, 1959), Multi-pass (Chondrogiannis et al., 2020), Constrained Time-Dependent KSP Algorithm (Hu and Chiu, 2015), or Greedy Framework (Liu et al., 2018). However, these algorithms are complex and only being tested in moderate size network. Our case study features the New York City network with more than 9,500 links and 4,500 nodes, which can pose difficulties for those algorithms. In addition, the k-shortest-path problem need to be solved for every single taxi trip and the computational time and power for such algorithm is simply not applicable. The Yen's Algorithm is the only algorithm that can practically be applied in this situation. To alleviate this drawback, we exclude taxi trip records with too high of an error between the observed and predicted travel distance from the training set. The choice of k is a model's hyperparameter and in our New York case study, after several trials, we choose k=5 since it balances between computational time and model's accuracy. Finally, the taxi trip's chosen path is the path minimizing the absolute difference between the predicted path distance and the observed distance collected from the taxi trip record. An example of Path Choice Prediction is also shown

in Figure 3.2. Step 1 is executed independently and repeated for each taxi trip and each time period.



3.3.2. Step 2: Partial Link Travel Time Prediction (PLTT)

Given the input taxi trips including its predicted chosen path and observed travel time, this stage aims to infer the travel time of only selected links. Step 2 is also executed independently and repeated for each time period. In the main formulation of PLTT, the decision variable is the partial link travel time, and the constraint is the predicted trip travel time, which is computed directly from partial link travel time, is close to the observed trip travel time. Therefore, the number of decision variable and constraints are number of links included in the taxi set and number of trips respectively. However, if we apply PLTT directly to a set of taxi trips, the number of decision variables will be much greater than the number of constraints and the resulting partial link travel time is not as accurate. Therefore, we first create a preprocessing Taxi Trip Sub-setting (TTS) model to address this problem. TTS aims to maximize the number of trips selected for subset *S* while guarantees that the number of selected trips divided by the number of involved links is greater than a certain threshold. The decision whether a taxi trip record *i* is selected for subset *S* is represented by the decision variable x_i of the TTS model and the formulation is as follows:

TTS Model:

Objective Function

$$Max: \eta = \sum_{i \in I} x_i \tag{30}$$

Subject to:

$$L = \sum_{a \in A} \left[1 - \prod_{i \in I} (1 - P_i x_i) \right]$$
(31)

$$\eta/L \ge \beta \tag{32}$$

Equation (1) is maximizing the number of selected trips for the subset *S*. Equation (2) shows the total number of involving links *L* corresponding to the subset *S*. P_i is a vector of size $A \times 1$ representing the sequence of link constituting the travel path for taxi trip *i*. This means that if link *a* is presence in the path sequence of trip *i*, the *a*th value of vector P_i is equal to 1 and 0 for otherwise. The decision variable x_i is binary showing whether trip *i* is selected for the subset *S*.

The product \prod is an element-wise vector multiplication across every trip *i* of the vector $(1 - P_i x_i)$. The \sum is the sum of all elements in the resulting vector from the previous calculation and it is equal to the number of links included in subset *S*. Equation (3) ensures that the ratio between the number of trips selected and its corresponding number of participating links is larger than a hyperparameter $\beta \epsilon(0,1)$. A higher β and thus, a higher ratio of η/L would yield a more accurate result for PLTT because the number of constraints would be close to the number of decision variables. However, higher β would also narrow down the feasible space and ultimately, fewer links would be selected for prediction and step 3 will receive fewer inputs.

After the subset *S* is created, the second procedure of step 2 is to predict travel time of the link included in subset *S* based on the Partial Link Travel Time (PLTT) model. The decision variable tt is a vector of size A × 1 where the a^{th} element represents the travel time of link *a*. The model aims to minimize sum of square error between the predicted travel time and the observed travel time among all taxi trips in subset *S*.

PLTT Model:

Objective Function:

$$Min\sum_{i\in S} \left(t_i^{pred} + \Delta_i - t_i^{obs}\right)^2 \tag{33}$$

Subject to:

$$t_i^{pred} = \sum_{a \in A} tt \cdot P_i \quad \forall i \in S$$
(34)

Equation (4) is the objective function minimizing the sum of square error between the predicted travel time and the observed travel time over all taxi trips *i* belong to subset *S*. The

predicted travel time only accounts for the summation of link travel time along the selected path and thus, we introduce the term Δ_i to represent the total intersection delay for taxi trip *i*. Since only the full path travel time is known, we currently set this Δ_i to 0 and the intersection delay is incorporated into link travel time. However, in future research where taxi GPS traces are available, this Δ_i can be accurately determined and improve the accuracy of step 2. Equation (5) shows the predicted travel time of taxi trip *i* is equal to the sum among all links *a* the elementwise product $tt \cdot P_i$. This entire step is repeated for each time period.

3.3.3. Step 3: Dynamic Urban Link Travel Speed Estimation

Step 3 takes the input of partial link travel time from step 2 to produce an output of dynamic traffic state which is represented by link level travel speed. Step 3 procedure is as follows:

- 1. Encoding link speed as node features: this step aims to transfer link feature, which in this case is travel speed, as node features to facilitate message passing between nodes.
- 2. Creating a modified directional adjacency matrix: this step creates a matrix that represent simultaneously node connectivity and the relation between different directional flows.
- 3. Traffic Graph Convolution Network (TGCN): this step develops a convolution operation on graph structure to learn the spatial relation between nodes.
- 4. Recurrent TGCN: this step joins hidden layers at different time period together to learn the temporal relation between nodes. The computational graph of step 3 would be shown after the discussion of each individual component.

3.3.4. Encoding link travel speed as node feature

First, we convert the link travel time to travel speed as the input for the deep learning model since it has the following advantages over travel time:

- Link travel speed are highly corelated between road segments. Thus, it makes the process
 of message passing between node more efficient.
- In case the link is untraversable, the link travel speed can be set to zero and effectively stop the process of message passing between node whereas for link travel time, one must set an arbitrary large number, which can be subjective.

Recent development of graph convolution network only allows for message passing between node whereas link features such as the distinction between directional movement (i.e., northbound vs south-bound traffic) are usually neglected. In addition, the graph considered tends to be undirected. This paper aims to bridge this gap by embedding link features (travel speed) as node features. For a node *i*, there is a maximum of 4 outward link going from node *i* to its neighbor. The speed on these outward links would be embedded to the node features vector as shown in the example in Figure 3.3. Here, travel speed on outward links of node *i*, which are v_1, v_2, v_3 and v_4 , are embedded to node *i* feature representation vector:

$$X_i = [x_i^1, x_i^2, x_i^3, x_i^4]$$
 where $x_i^f = v_f, \ \forall f \in (1:4).$

The inward link to node i should not be embedded because it would be repetitive since node i's neighbors have already embedded it in their feature representation vector.



Figure 3.3. Visualization of Link Travel Speed Embedding

The order in which these link travel speed goes into the node feature vector depends on its north bearing angle α_a^{link} and this node-link embedding principal is presented in Table 3.1. The principal along with the modified directional adjacency matrix help our framework preserving the notion of direction which is inherent in traffic flow.

Table 5.1. Houe-Emix Embedding Fineipar	
Embedding Principle	
Link travel speed embedded to the first element	
Link travel speed embedded to the second element	
Link travel speed embedded to the third element	
Link travel speed embedded to the fourth element	

Table 3.1. Node-Link Embedding Principal

3.3.5. Modified Directional Adjacency Matrix

In the normal Graph Convolution Network by Kipf and Welling (2017), the adjacency matrix can only represent the node connectivity and the graph is undirected. This is not applicable for road network because the direction at which these nodes are connected is important. This can be shown by the following example:



(a) Example of Node Message Passing (b) Modified Adjacency Matrix Figure 3.4. Node Message Passing

We consider node 2 in Figure 3.4a as the node of interest or host and it is looking to gain information from its neighbor, node 1. We can say that the value of x_1^2 is highly related to x_2^2 while none other pair is related. Therefore, only the pair of x_i^f that has f = 2 can pass message between each other. This is because the direction of the link from neighbor 1 to host 2 lies in the f = 2 direction. This conforms with the node-link embedding principle mentioned in Table 3.1. Therefore, we construct a $[N \times N \times F]$ modified adjacency matrix A as follows:

A:
$$a_{ijf} = \begin{cases} 1 \text{ if the link from neighbor } j \text{ to host } i \text{ is in the } f \text{ direction} \\ 0 \text{ for otherwise} \end{cases}$$

The adjacency matrix for the example can be visualized in Figure 3.4b. Here, every links in the network is embedded separately even for those with the same OD pair and $a_{ijf} \neq a_{jif}$. The notion of direction is preserved with the use of the third-dimension *f*. The adjacency matrix, node-link embedding principle, and the traffic graph convolution network are specifically designed to be compatible with each other and capture the real-world network traffic as close as possible. In the GCN introduced by Kipf & Welling (2017), the adjacency matrix is normalized using the following operation to ensure the node features after convolution are not scaled up:

$$\bar{A} = \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2}$$

Where: $\hat{A} = A + I$, and I is an identity matrix and \hat{D} is diagonal node degree matrix of \hat{A}

This normalization ensures that the row sum of \overline{A} is equal to 1. However, this normalization is not necessary for our Modified Directional Adjacency Matrix. Within a certain direction *f*, we only want the host node *i* to look at one particular relevant neighboring node *j* and thus, only this pair a_{ijf} value is 1 whereas the remaining is 0. Therefore, this guarantee:

$$\sum_{j} a_{ijf} = 1, \forall i \in I, f \in F$$

To help explain this feature, we refer to back to Figure 4 and assume the host node is node 2 and the direction of interest is f = 2 (or East). Here, we only allow x_1^2 of node 1 to pass information to node 2's x_2^2 . Suppose there are other nodes neighboring node 2 to the East, North, and South. However, none of these neighboring nodes can pass information about x_2^2 and thus, their a_{ijf} value is 0. Therefore, for a particular direction, this guarantees the row summing to 1 for our Modified Adjacency Directional Matrix.

3.3.6. Traffic Graph Convolution Network (TGCN)

After embedding link to node and constructing the directional adjacency matrix, TGCN will apply "convolution" over the road network. GCN can be described as a neural network where each node of the graph is a neuron itself and the propagation rule is the "convolution" operation. In CNN, "convolution" refers to aggregating information from nearby pixels. Therefore, the idea of "convolution" on a node can be regard as aggregating the information from all of its neighbor. This can be visualized in Figure 3.5a where the host node *u* is looking to gain information from its immediate neighbor v_1 , v_2 , v_3 , and v_4 .



(a) Convolution on Graph Domain



(b) Graph Convolution on Matrix Domain Figure 3.5. Convolution on Graph

Therefore, the Graph Convolution on graph domain can be formulated as follows:

$$h_{u,f}^{k} = relu\left(\sum_{v \in N(u)} w_{uv}h_{v,f}^{k-1} + b_{u}\right), \quad \forall u \in N, f \in F$$
(35)

Where relu(x) is the rectified linear activation function (ReLu) as follows:

$$relu(x) = \begin{cases} x \text{ if } x \ge 0\\ 0 \text{ if } x < 0 \end{cases}$$

Equation (6) represents the convolution on graph where for every neighbor v of u, we take the previous hidden layer of neighbor $h_{v,f}^{k-1}$, multiply it with the weight w_{uv} , and aggregate it to the current layer of host u. In addition, a bias b_u is added and the entire operation is transformed using the ReLu function. Here we do not need to add the subscript f for the weight w_{uv} because for a pair of host u and neighbor v, there is only one link in a specific f direction (e.g., f = 2 in the example of Figure 4) that can facilitate node message passing. This operation is repeated for each node u within the network and for each traffic directional flow f. This repetition can pose a problem in computational efficiency and Equation (6) is very difficult to be programmed. Therefore, we modify and migrate the operation from graph domain to matrix domain. First, instead of using the individual node representation $h_{u,f}^k$, we use a $[N \times F]$ matrix H_k where each row represents the node feature. In similar fashion, a $[N \times N]$ matrix W and a $[N \times F]$ matrix B would be utilized instead of the set of individual weights w_{uv} and b_u . The modified directional adjacency matrix A would facilitate the transition from graph domain to matrix domain. Equation (6) can be rewritten in matrix domain shown in Equation (7) below.

$$H^{f}_{k,pre_transformed} = a^{f} \odot W \otimes H^{f}_{k-1}, \quad \forall f \in (1:4)$$
(36)

$$H_k = relu(H_{k,pre_{transformed}} + B)$$
(37)

Here, for a specific value f = f, we first subset a matrix a^f from the modified adjacency matrix A and perform an element-wise matrix multiplication with W. The resulting matrix is then used as the first component in the matrix multiplication with the vector H_{k-1}^f subsetted from the previous layer network state H_{k-1} at f = f. A visualization of Equation (7) is shown in Figure 5b. This operation is repeated for every $f \in (1: 4)$. Equation (8) adds the bias *B* and the entire operation is transformed using the ReLu function and the resulting matrix is the current layer network state H_k . By performing Equation (7) and (8), a node will effectively learn the information from its neighbor one hop away. TGCN can be repeated multiple times to allow the node learning from neighbors further away. Equation (7) and (8) address the research gaps in Section 2 and it has the following three advantages: (1) It preserves the notion of directional flow, (2) It transform a complex convolution on graph domain to matrix domain and make it readily for scaling to larger network, and (3) the weights *W* can be shared across multiple execution of TGCN and thus greatly reduce the number of weights to be trained. This weight sharing feature will be discussed further in the computational graph section below.

3.3.7. TGCN_{lstm} for capturing Temporal Dependencies

After the TGCN operation is applied at each time period to capture spatial relation, we introduce the TGCN_{lstm} cell to join the traffic states at different time periods together. We modify the wellknown LSTM cell to a TGCN_{lstm} cell to capture both temporal and spatial relation in one operation and apply it to the final two layers K - 1 and K. The TGCN_{lstm} cell would take three inputs which are (1) the current layer at previous time period H_k^{t-1} , (2) previous layer at current time period H_{k-1}^t , and (3) the internal state at previous time period s^{t-1} . There are two outputs of the TGCN_{lstm} cell namely (1) the current layer at current time period H_k^t and the internal state at current time period s^t . The computation graph of LSTM can be shown in Figure 3.6a as follows:



(b) *tgcn*_{lstm} () operation **Figure 3.6. TGCN**_{lstm} **Architecture**

In the spatial aspect of TGCN_{lstm}, it functions in similar fashion as TGCN with the only difference being TGCN_{lstm} has two inputs H_k^{t-1} and H_{k-1}^t . Therefore, we would have to modify

the adjacency matrix a_{lstm}^{f} , the weight matrix W_{lstm} , and the convolution operation as shown in Figure 3.6b. First, the previous layer at current time period H_{k-1}^{t} and the current layer at previous time period H_{k}^{t-1} are row-wise concatenated resulting in a $[2N \times F]$ matrix. We column-wise concatenate the adjacency matrix a^{f} at f = f with an $[N \times N]$ identity matrix to get the adjacency matrix a_{lstm}^{f} for TGCN_{lstm}. The weight matrix is also in $[2N \times N]$ dimension where the first $[N \times N]$ part is the same as TGCN and the second $[N \times N]$ part only has weights in the diagonal location and the remaining location are 0. We then perform the dot product between a_{lstm}^{f} and W_{lstm} . The resulting matrix is the first component of the matrix multiplication, and the second component is the subset at f = f of the concatenated input $[H_{k-1}^{t}; H_{k}^{t-1}]$. The process is repeated for every f. A schematic representation of which is shown in Figure 6b. Despite the concatenation, the resulting matrix will have N rows thus make it easily compatible with other operation. This adjustment encourages the node to not only takes temporal information from itself but also from its neighbor. We name the entire operation as $tgcn_{lstm}()$ as follows to better describe the calculation of the internal values occurs within the TGCN_{lstm} cell.

$$H_{k-1}^{t,f} = tgcn_{lstm} (W_{lstm}, [H_{k-1}^{t,f}, H_{k}^{t-1,f}]) = a_{lstm}^{f} \odot W_{lstm} \otimes [H_{k-1}^{t,f}, H_{k}^{t-1,f}], \ \forall f \in (1:4)$$

From the computational graph shown in Figure 3.6, we have 4 internal parameters namely for_t , inp_t , act_t , and out_t and these parameters are calculated by the following equations:

$$for_t = \sigma[tgcn_{lstm}(W^{for}, [H_{k-1}^t, H_k^{t-1}]) + B^{for}]$$
(38)

$$inp = \sigma \left[tgcn_{lstm}(W^{inp}, [H_{k-1}^{t}, H_{k}^{t-1}]) + B^{inp} \right]$$
(39)

$$act_{t} = tanh[tgcn_{lstm}(W^{act}, [H^{t}_{k-1}, H^{t-1}_{k}]) + B^{act}]$$
(40)

$$out_{t} = \sigma[tgcn_{lstm}(W^{out}, [H_{k-1}^{t}, H_{k}^{t-1}]) + B^{out}]$$
(41)

$$s_t = for_t \cdot s_{t-1} + inp_t \cdot act_t \tag{42}$$

$$H_k^t = out_t \cdot \tanh(s_t) \tag{43}$$

Where σ () is a sigmoid function and tanh () is a hyperbolic tangent function as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Equation (9) – (12) represent the calculation of the internal parameters for_t , inp_t , act_t , and out_t respectively. In each of these equations, a new set of trainable weight and bias are introduced. These weights and biases are shared across time periods. Equation (13) determines the internal state value at current time period s_t . This ensures the gradient flow is uninterrupted across time periods. Equation (14) shows the final network link travel speed at current time period H_k^t .

3.3.8. Computational Graph

In this section, we present the computational graph stating the flow of information from the input to the output. We choose the number of layers as k = [0:4] and t = [1:28] for our New York City Case study. A training sample would represent a typical day (i.e., Monday 01/06/2014). First, the input of a typical sample is disaggregated into multiple states H_0^t where each state represents a specific time period (i.e., 06:00 – 06:30). In the example of our New York City network, our analysis period ranges from 06:00 in the morning to 20:00 in the evening at 30 minutes interval which results in 28 states of H_0^t . A combination of these 28 H_0^t states will form one training sample. The input would be partial link travel speed from step 2. If the speed of the link is not

available, the value is set to 0 thus disrupts the message passing from the beginning. Then, the TGCN are applied three time to H_0^t at each time period resulting in H_3^t . The TGCN's weight W is shared from H_0^t to H_3^t but not across different time period. This means that the TGCN's weight is distinctive for each time period. Therefore, we differentiate it by the subscript *TGCN*_t. At the last two layers, the TGCN_{1stm} takes the input of H_3^t , H_4^{t-1} , and s^{t-1} and produces the outputs of H_4^t and s^{t-1} . Here, the set of weights and biases for TGCN_{1stm} are shared across time periods. The final layer H_4^t corresponds to the final output of network wide dynamic link travel speed. In this computational graph, a node can gather information from neighbors as far as 5 "hops" away. Figure 3.7 shows the full computational graph of the DU-LSE model.



Figure 3.7. Computational Graph
The loss function is a sum of square error (MSE) between the predicted travel speed and the observed travel speed, and it is accumulated over all H_4^t . The loss is computed on the observed values only and it is computed in Equation 15 shown below:

$$Loss = \sum_{t \in T} (H_4^{t,obs} - H_4^{t,pred})^2$$
(44)

3.4. NEW YORK CITY CASE STUDY

3.4.1. New York Taxi Dataset

Our study's methodology can benefit cities possessing taxi trip data such as Chicago or San Francisco and we choose New York City to demonstrate our framework. The Taxi Trip dataset used in this paper is obtained from the New York City Taxi and Limousine Commission (NYC-TLC) and it is publicly available on the City of New York's official website¹. Each observation of the dataset is a single taxi trips which includes several details of the trip and our paper mainly focuses on the following four: (1) Pickup and Dropoff Coordinates; (2) Pickup and Dropoff Timestamp; (3) Observed Travel Distance; and (4) Observed Travel Time. The coordinates are only available from 2015 backwards and thus we select January 1st, 2014, to December 31st, 2015 as our analysis period. We load the taxi trips dataset into a local database using PostgreSQL to facilitate the task of querying specific records falling into a time period. Our daytime analysis period ranges from 06:00 - 20:00 with a 30-minute interval resulting in 28 time period. Upon doing descriptive statistics, we recognize the difference in travel pattern between the weekdays is significant enough that we would need to create several models for each set of day. This is shown in Figure 3.8b below. Due to limited resources, we only consider Monday of each week for analysis, but the methodology can be readily applied to other weekdays as well.

¹ www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

We select the entire Manhattan Road network for analysis which has 9,617 road segments or links and 4,750 nodes or intersection. The network is obtained from the OpenStreetMap Database². Therefore, we select taxi trips of which both pickup and drop-off coordinates fall within the Manhattan network. In addition, we present several visualizations of the dataset from both a spatial and temporal perspective. Figure 3.8a shows the spatial aspect of the dataset where the Manhattan Network and the Taxi Trip Pickup and Dropoff location are presented. On the other hand, Figure 3.8b shows the temporal aspects where the average number of trips and travel speeds are presented.

² <u>https://www.openstreetmap.org/relation/175905</u>



(a) Spatial visualization of the dataset including: road network, pickup density, and dropoff density



(b) Temporal visualization of the dataset

Figure 3.8. Descriptive Statistics of the Dataset

3.4.2. Results

First, we choose the following values for the hyperparameters in the New York Taxi case study. We set k = 5 in the k^{th} shortest path algorithm; ratio threshold $\beta = 0.85$ in the TTS model; and intersection delay $\Delta_i = 0$ in the PLTT model of step 2.

To implement the methodology discussed in the previous section, we use multiple means of programming. First, the raw data in .csv format is stored in a PostgreSQL database for easy querying. For the first two steps of Path Choice Prediction and Partial Link Travel Time Prediction, we use R programming with built in library for Parallel Computing and Optimization Algorithm. For calculating k^{th} shortest path, we use a function from PostgreSQL which is pgrouting and it uses Yen algorithm for this computation. The first step is computationally heavy since the computer needs to do Data-mapping and finding k-shortest-path for every taxi trip records. However, the process is accumulative and the "divide and conquer" strategy is applicable. After finishing the first two steps, we use Python programming, specifically the library Tensorflow 2.0 (Google, 2020), to develop our deep learning model.

In Step 3, the model takes the input of Partial Link Travel Time from Step 2, converts it into input speed, and produces the output of network wide link travel speed. In our New York city network case study, the size of the network is 9,617 link and the number of time period is 2,912 (note that 28 time period would comprise a day). The output speed is available across all these links and time periods. However, it would be difficult to show all these values and we decide to illustrate our result for selected links and time periods only. We choose the time periods on Monday, 06/23/2014 to show our result and this date is chosen randomly. Figure 3.9 shows the heatmap of step 3 input and output. Each row represents a specific road segment, and each column represents each time period from 06:00 - 20:00 at 30 minutes increment. The road

segments are taken from the Midtown Area just below the Central Park. The heatmap color scale is provided on the right representing travel speed in miles per hour.

The color in each cell represents the link travel speed. For the input heatmap in Figure 3.9a, links, of which travel speed is not available, are colored in white. Cells tilting toward dark purple denote a lower link travel speed and those tilting toward brighter yellow denote a higher travel speed. From Figure 3.9b, the most links experience travel speed less than 15 mph in most time of the day and only a few links such as West 57th Street or West 43rd Street have travel time larger than 25 mph during the earlier period of the day (i.e., 06:00 - 07:30). For each link, we also see a gradual change in travel speed throughout time of the day where most link start at a higher travel speed, decrease and remain low during working hours, and increase back in later in the evening.



(a) Speed Input of Selected Links from 06:00 - 20:00



(b) Speed Output of Selected Links from 06:00 – 20:00 Figure 3.9. Heatmap Showing Input and Output Speed of Selected Links

Figure 3.10 shows the temporal aspect of our model's result where link travel speed at each time period is illustrated. To demonstrate the spatial aspect, we plot two subsets of the New York city network representing major arterial and minor arterial in the Midtown area respectively in Figure 3.10. For each subset, three time period namely 07:00 - 07:30, 12:00 - 12:30, 18:00 - 18:30 are selected since these time periods represent different trip purpose (i.e., commuting, lunch, and recreation) and vary greatly in the Origin-Destination trip demand. In the computational graph shown in Figure 3.7, the TGCN module's training weights vary between time period which gives the model flexibility in capturing different network speed patterns emerging throughout time of the day and this flexibility is reflected in Figure 3.10. The link travel speeds are color-coded with the same analogy as in Figure 3.9 and the color scale is provided on the right. In the subset for major arterial plots, the northern part of Manhattan has

considerably higher travel speed than the rest of the network whereas the Midtown and Downtown area experience consistent low travel speed with some exceptions such as the highways near the Brooklyn Bridge. In overall network performance, the 07:00 - 07:30 time period rank first in travel speed, followed by the 18:00 - 18:30 and the 12:00-12:30 time period as illustrated by the color tilting from the "brighter" side to the "darker" side. Figure 3.10b shows the subset of minor arterial in the Midtown area accompanied by its location with respect to the Manhattan Network as a whole. The majority of the link has travel speed less than 15 mph through all three time period. One small observation is the two long links in the bottom right of the network which has relatively high travel speed as compared to the remaining links. These two links are part of the Park Avenue Road at the segment near Grand Central Terminal and these links are on a different elevation compared to the rest. The curve link in the middle of the network is Broadway where the southern segment is utilized as walking avenue which explains the sudden stop in this link. The first northern segment before West 57th street of Broadway Avenue has higher travel speed compared to the rest of the segment. After West 57th and downward, there road geometry is expanded from a normal two-lane one direction road into multiple purpose road with two side parking and bike lane available. These segments also have a higher density of retail areas, restaurants, and offices too.



(a) Output Link Travel Speed of Major Arterial at Three Different Time Period



(b) Output Link Travel Speed of Selected Links in Midtown at Three Time Period Figure 3.10. Output Link Speed of Major and Minor Links at Three Time Periods

3.5. MODEL EVALUATION AND VALIDATION

The first part of this section evaluates the model's performance internally by comparing the prediction result from each step of the framework with the taxi trip data. The second part of this section validates the entire framework's result with the ground truth data.

3.5.1. Model Evaluation

In this section, we discuss the evaluation for every three step of the model's sequential framework. Step 1 aims to predict travel path by minimizing absolute difference between the predicted distance from the travel path generated from K-Shortest-Path algorithm and the observed travel path recorded by the taxi. Figure 3.11a shows the distribution of the absolute error between predicted and observed value of step 1 at three different time periods. We disaggregate the plot by trip distance because for longer trip, we expect a larger error compared to shorter one. The unit of y-axis is density and because the figure is a distribution plot, the total area under the curve sums to 1. In the trip distance 0-4 and 4-8 category, the distribution is highly centered around the mean value being near zero with a slight skew to the right. In the 8-12 and 12-16 category, the distribution has a longer tail on the right, which indicates some outliers with high absolute error. However, the mean remains at a low enough value. In general, the time period 12:00-12:30 has a better performance compared to the other two. In similar fashion, step 2 aims to predict link travel time at selected links by minimizing the error between observed and predicted trip travel time and the absolute error distribution of which is shown Figure 3.11b. We also see the same distribution as in Figure 3.11a for the first four trip distance category. Step 2 only selects a fraction of the observation for analysis and by keeping the ratio between number of trips selected and number of links in subset *S* high, the optimization model is able to achieve high accuracy.



(a) Absolute Error between predicted and observed trip distance



(b) Absolute Error between predicted and observed trip travel time Figure 3.11. Absolute Error of Step 1 and Step 2 at Selected Time Period

We introduce two metrics namely Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) to evaluate not only this step but also the other two remaining steps and the calculation of these two metrics are shown in Equation (16) and (17).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1:n} (X_i^{pred} - X_i^{obs})^2}$$
(45)

$$MAPE = \frac{1}{n} \sum_{i=1:n} \left| \frac{X_i^{pred} - X_i^{obs}}{X_i^{obs}} \right| \times 100\%$$
(46)

We then subset each taxi trip *i* by its Time Period and Travel Distance and calculate these subset's *RMSE* and *MAPE* instead of the entire dataset in order to identify any pattern of these two metrics across Time Period and Travel Distance. The resulting evaluation is shown in Table 2 where the rows represent time period, and the columns represent travel distance. As trip distance increases, the value of both *RMSE* and *MAPE* increase since the origin-destination are too far apart that the K-Shortest-Path cannot cover all available path. At 0-4 mile of distance, the *RMSE* and *MAPE* are around 0.19 mile and 7.5% respectively whereas at 12-16 mile of distance, the values are 3 mile and 19% respectively. In contrast to travel distance, the RMSE and MAPE do not greatly fluctuate which is reasonable since the formulation of step 1 does not involve time period.

	0-4 mile		4-8 mile		8-12 mile		12-16 mile	
Time Period	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
	(mile)	(%)	(mile)	(%)	(mile)	(%)	(mile)	(%)
6:00 - 6:30	0.194	7.5	0.928	11.7	1.766	15.9	3.046	18.4
6:30 - 7:00	0.198	7.6	0.929	11.9	1.913	17.9	3.031	19.3
7:00 - 7:30	0.198	7.4	0.968	12.3	2.009	19.3	2.977	18.7
7:30 - 8:00	0.197	7.5	1.011	13.1	1.982	19.0	2.797	15.4
8:00 - 8:30	0.195	7.6	1.039	13.7	2.012	19.3	2.954	19.0
8:30 - 9:00	0.192	7.6	1.038	13.9	1.940	18.2	3.342	20.8
9:00 - 9:30	0.192	7.7	1.051	14.2	1.985	18.6	3.036	18.2
9:30 - 10:00	0.193	7.7	1.070	14.4	1.961	18.3	3.542	23.0
10:00 - 10:30	0.192	7.6	1.079	14.5	1.984	18.5	3.347	22.0
10:30 - 11:00	0.191	7.6	1.093	14.7	1.947	18.0	2.757	17.2
11:00 - 11:30	0.192	7.6	1.101	14.7	1.937	17.9	2.842	17.9
11:30 - 12:00	0.192	7.7	1.107	14.7	1.891	17.4	2.731	16.6
16:00 - 16:30	0.188	7.5	1.035	13.6	1.891	17.3	2.735	17.1
12:00 - 12:30	0.191	7.8	1.107	14.7	1.901	17.3	2.631	16.4
12:30 - 13:00	0.189	7.7	1.085	14.5	1.924	17.7	2.955	18.9
13:00 - 13:30	0.191	7.7	1.111	14.7	1.890	17.3	2.508	15.1
13:30 - 14:00	0.189	7.7	1.102	14.6	1.948	17.9	2.506	16.6
14:00 - 14:30	0.190	7.7	1.093	14.4	1.919	17.5	2.819	17.2
14:30 - 15:00	0.189	7.7	1.089	14.4	1.933	17.7	3.150	18.0
15:00 - 15:30	0.189	7.6	1.074	14.1	1.909	17.4	2.577	15.9
15:30 - 16:00	0.189	7.6	1.052	13.9	1.921	17.6	2.560	14.3
16:30 - 17:00	0.187	7.3	1.014	13.4	1.893	17.5	3.166	19.3
17:00 - 17:30	0.188	7.3	1.001	13.1	1.826	16.6	2.523	15.6
17:30 - 18:00	0.187	7.3	0.988	12.9	1.819	16.5	2.456	14.4
18:00 - 18:30	0.186	7.2	0.958	12.4	1.849	16.9	2.606	14.6
18:30 - 19:00	0.185	7.1	0.953	12.3	1.783	16.0	2.095	12.4
19:00 - 19:30	0.184	7.0	0.946	12.2	1.726	15.4	2.502	14.8
19:30 - 20:00	0.185	7.0	0.934	11.9	1.698	15.0	2.103	12.0

Table 3.2. Evaluation of Step 1 Path Choice Prediction

We perform the evaluation of step 2 Partial Link Travel Time. In this step, we predict travel time of selected links by minimizing the sum of square of the error between the predicted travel time and the observed travel time of selected taxi trips in subset *S* and thus, this error is the metric to evaluate the performance of step 2. Unlike step 1, step 2's evaluation focuses more on spatial pattern. Since a taxi trip span across multiple spatial regions and the error is calculated on

a trip basis, the total trip error is distributed evenly among the spatial regions. Figure 3.12 shows the Mean Absolute Error (MAE) and the MAPE at each region. The takeaway of these figures is that we can estimate a taxi trip's error by summing the MAE value of all the regions that it passes by. This shows how certain areas might have lower accuracy in partial link travel time prediction.

On average across all taxi trips, step 2 reaches a performance of 2.6 minutes in MAE and 18% in MAPE. The spatial pattern of MAE is very similar to the RMSE's where higher error is located in the Midtown neighborhood near the Central Park and lower error is located at the Northern part of Manhattan such as Harlem, Hamilton, and Washington Heights neighborhood. This is because of two reasons. First, Midtown neighborhoods have considerably higher not only the number of taxi trips, as shown in Figure 3.8a, but also the number of taxi trips passing through this area and thus, there are more variations in the error. Second, in step 2 of PLTT's formulation, we model intersection delay, but the value is assumed to be 0 due to lack of data. The variation in travel time at each intersection between red, green, or even left turn waiting/signal and the accumulation of such variations throughout the trip results in an overestimation of the link travel time. Midtown neighborhoods have shorter road segments and significantly higher number of intersections compare to Northern Manhattan's which results in accumulation of intersection delay and ultimately higher error. However, the error value is still low at 0.6 minutes on MAE and 4% MAPE in these areas and it is not significantly more than the minimum error.

To evaluate the Step 3 of Dynamic Urban Link Travel Speed Estimation, we calculate RMSE between the output link travel speed and the partial input travel speed for the observed record of the input only. The RMSE is disaggregated by Manhattan neighborhood and time of the day and the values are shown in Figure 3.13. Here, we see a difference in RMSE pattern throughout time period. For time period from 06:00 - 10:00 the average of RMSE across neighborhoods are 1.8 mph. However, this value decreases considerably later in the day with an average of 0.7 mph at time period 10:00 - 15:00 and 0.2 at time period 15:00 - 22:00. This is mainly due to the number of taxi trips are much higher from late morning until evening compared to early morning as shown in Figure 3.8b.



A higher number of taxi trip results in more observed data and better estimation for step 3. The top 3 neighborhoods with the highest accuracy are East Harlem South, Upper East Side, and Gramercy of which average RMSE value are 0.53, 0.60, and 0.61. This is reasonable because it's in the middle of the network and a lot of taxi trip would travel through these areas.



Figure 3.13. Evaluation of Step 3: RMSE by Neighborhood and Time Period

3.5.2. Model Validation

In this section, we first validate our framework with a baseline framework from Zhan et al. (2013). Then, we obtain ground-truth travel speed data collected from traffic sensor by the New York Department of Transportation's Traffic Management Center and compare it with our framework.

Validation with Similar Methodology

We choose Zhan et al. (2013) as the baseline framework for comparison because their paper has the similar scope to our's, which is estimating link travel time using taxi dataset with only the origin and destination known (GPS traces are not available). Their methodology considered path taken by taxi trips as latent and proposed a multinomial logit model to estimate the probability of choosing a particular path. The decision variable is the link travel time, and the expected path travel time are based on this. By multiplying the path probability with its expected path travel time, the paper arrived at the expected trip travel time and the objective is to minimize the root mean square error between the expected and observed trip travel time. Since their methodology is applied to a small network near Central Park, we also apply their methodology to our taxi dataset on a small network that is identical to the Midtown network in Figure 10b. From the 2-year taxi dataset, we select 2 days and 3 time periods from each day resulting in 6 instances. The 2 days are April 7 and 20 of 2015 and the 3 time periods are 07:00 - 07:30, 12:00 - 12:30, and 18:00 - 18:30. After applying Zhan et al. (2013)'s methodology to these instances, we obtain the link travel time and compare with our three-step framework's result.

Figure 3.14 shows the absolute error between our result and Zhan et al. (2013)'s at these 6 instances. The links are color-coded based this absolute error with darker color indicating lower error and vice versa and we notice the following pattern. First, the absolute error is highest

in the first time period of 07:00 – 07:30 and it decreases in the noon and evening time periods. Since later time periods have more taxi trip records than the first time period, the accuracy of both methodologies are higher and there is less variance. However, the majority of the link are still within 2 minutes of absolute error indicating similar estimation. Second, links in the East-West direction has slightly more error than links in the North-South direction because of two reasons. First, North-South links are usually shorter than East-West and thus the error is smaller. Second, taxi demand going in the general North-South of Manhattan (e.g., from Central Park to Financial District) is proportionally higher than the demand going East-West. With more taxi trips record in the North-South direction, the estimation of North-South links for both methodologies will be more accurate and stable resulting in less error.



Figure 3.14. Absolute Error in 6 Instances

Next, we use the link travel time from both methodologies combined with the predicted path choice from step 1 to calculate the predicted trip travel time and validate with the observed trip travel time. Table 3.3 shows the Mean Absolute Error (MAE) and the Mean Absolute Percentage Error (MAPE) of these two methodologies at 6 instances. In terms of, our three-steps framework is slightly more accurate across all 6 instances, but two methods still have very low error at an average of 1.5 and 1.0 minutes respectively. In terms of MAPE, our framework also has lower error except for the fourth instance. On average, the MAPE for Zhan et al. (2013) and our three-steps framework are 30% and 26% respectively. Zhan et al. (2013)'s framework requires to subset the taxi trip record of which origin and destination is within the studied network only. Since our Midtown network is small, the taxi trip is relatively short resulting in a shorter travel time and a higher value of MAPE.

		Mean At	osolute Error	Mean Absolute		
		(m	inutes)	Percentage Error (%)		
		Zhan et	Three-steps	Zhan et	Three-steps	
Date	Time	al. (2013)	Framework	al. (2013)	Framework	
Day 1	07:00 - 07:30	1.4	1.0	27.9%	22.8%	
	12:00 - 12:30	1.6	1.1	27.1%	24.3%	
	18:00 - 18:30	1.1	1.0	29.1%	29.6%	
	07:00 - 07:30	1.0	1.0	35.3%	29.8%	
Day 2	12:00 - 12:30	2.2	1.1	23.7%	15.6%	
	18:00 - 18:30	1.6	1.2	38.1%	33.1%	

 Table 3.3. Validation between Observed and Predicted Trip Travel Time

Validation with Ground Truth Speed Sensor

We obtain the ground truth speed data from the City of New York³. The New York Department of Transportation's Traffic Management Center and other local agencies have installed multiple traffic speed detectors in a variety of form. One of which is camera detector which also provides

³ https://data.cityofnewyork.us/Transportation/Real-Time-Traffic-Speed-Data/qkm5-nuaq

live view of traffic and can be viewed via the website⁴. Therefore, we filter the data based on the following criteria to ensure it is compatible for comparison with the framework's result: (1) the years are between 2014 and 2015, (2) weekday is Monday and time of the day is between 06:00 – 20:00, (3) the traffic speed detectors fall within the Manhattan borough only. Ultimately, we get 19 traffic speed detectors and a total of 57,611 observations from 4/18/2015 - 11/30/2015. The location of these speed detectors is shown in Figure 3.15 below:



Figure 3.15. Speed Camera Location

⁴ https://webcams.nyctmc.org/

Each speed detector is assigned to the corresponding link based on its location and the traffic flow direction of which the camera is capturing. The timestamp of each observation is also translated into the equivalent time period so that the ground truth data and model's result are comparable. We compute the absolute error between the model's result and the ground truth data, aggregate by time period, and utilize boxplot to show the distribution of these errors as a box plot at each time period as shown in Figure 3.16. The majority of these boxplots have median typically falling under 5 which indicates the framework has high level of accuracy. The 25th percentile is close to zero whereas the 75th percentile is around 10 mph. However, the number of outliers are substantial and the distribution of which is wide. We observe that the time period 5-7 and 19-21 or correspondingly 08:00 - 09:30 and 15:00 - 16:30 have more outliers than the rest. During these hours, the network usually experiences traffic congestion, and the travel speed would vary greatly from minutes to minutes. Our model is estimating a 30-minutes average and these variations result in a high number of outliers. Beside congestion, some driver tends to drive leisurely slow and chooses longer but less traffic route in contrast to rushing driver choosing shortest path. These are heterogeneity in driving behavior, and it can increase the framework's error. Finally, as previously mentioned in step 2 evaluation section, the framework does not capture intersection delay and the result tends to overestimate the individual link travel time as, which ultimately contributes to the absolute error. However, giving only limited knowledge (i.e., sparse taxi trip), the framework is still able to estimate travel speed within a 5mph accuracy.



Figure 3.16. Boxplot of Absolute Error between Framework's Result and Ground Truth

In addition to Figure 3.17 showing the distribution error, we also provide the Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) by each time period in Figure 3.17. The MAPE averages around 22.3% and the values are slightly higher during midday compared to the early morning or late afternoon. The MAE plot also tells the same story where higher values are observed during midday. As shown in Figure 3.17b, the number of taxi trips during midday is significantly less than the morning peak hours or later in the evening where people tend to go out for recreational purposes. Therefore, the input for step 3 of during these hours are not as strong the others and as a result, the error is higher. However, the average MAE of all time period is still 5.8 mph indicating reasonable estimation accuracy.



Figure 3.17. Framework's MAPE and MAE by Time Period

3.6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a sequential three step framework to solve the problem of Network Wide Dynamic Link Travel Speed using only the Taxi Trip Dataset. In the first two steps, information from taxi trip record such as pickup and drop-off location, travel time, and distance are processed to give output of partial link travel speed. The third step makes use of a novel deep learning model which consists of two main components which are Traffic Graph Convolution (TGCN) for capturing spatial relation and TGCN_{1stm} for temporal relation. The result suggests that the framework can estimate dynamic link level travel speed in dense urban area and the accuracy is high as shown in the framework's application and evaluation in the New York case study. The significance of our study are (1) it only requires one open source (or low-cost) and passively collected dataset (i.e., Taxi); (2) unlike other normal GCN, the TGCN is specifically designed for traffic network and is able to capture directional flow of traffic through the Node-Link Embedding and Modified Adjacency Matrix; (3) in TGCN-LSTM, the model encourages node to take both historical data from itself and its neighbor; and (4) the model is capable of obtaining network wide travel speed for bigger networks with up to 9,500 links.

However, the three-step framework has several drawbacks. First, the computational efficiency in the first step is low since for each taxi trip record, the following two tasks must be executed: coordinate mapping and kth-shortest-path generation. We make full use of our computer's capability through parallel computing and efficient algorithm design. The process is accumulative and can be solved independently, but it still demands time due to the shear amount of data (i.e., up to one week). Second, the Yen's Algorithm generates an alternative path set that lacks diversity and cannot capture real-world driver's route choice. Third, we cannot accurately evaluate step 1 and 2. In step 1 the ground truth data on the taxi's path and itinerary are unknown

whereas step 2 can overestimate link travel time due to unknown in intersection delay. For future research, all of these drawbacks can be fully addressed if taxi trips GPS traces (i.e., not just origin and destination but location at different timestamp) is made available.

A possible approach to deal with the estimation of path travel times from pick-up and drop-off time stamps lies with consideration of delay in the presence of traffic control devices such as signalized or unsignalized intersections, is to have detailed attributes of type of signal (signalized, stop or yield sign), and signal characteristics (timing of pre-timed signalized, actuated-signalized, semi-actuated signalized). The methodology proposed in the paper can incorporate the effect of signals in path travel time estimation, though in the case study in the absence of such data the delay accounted from signals are not considered. Considering dynamic state of traffic flow and signal timings are not always available on the urban road network and the goal of this paper is develop a model to estimate travel time as accurate as possible with least information and make the model more generic for planning applications for large networks, the results presented in the case study may associate with under or over estimation of path travel times.

We believe our research can be beneficial for urban planners in the ITS domain, especially those in developing economies without state-of-the-art infrastructure already in place. Future work for this study can concentrate on integrating the travel time estimation not only of road network but also other modes of travel especially public transportation including transit and bike providing urban dwellers complete information on travel cost between competitive travel modes for their decision making. In addition, further investigation could be emphasized on computational efficiency and intra-direction convolution, which allows for not only through movement but also turning movements.

ACKNOWLEDGMENT

This research is partly funded by the Center for Transportation Innovations in Education and Research (C-TIER) at the University of Memphis. The authors are grateful to associate editor and anonymous reviewers for their constructive feedback. Any opinions, findings, and recommendations expressed by the authors herein do not reflect the view of the agency funding and supporting this study.

Data Availability Statement

The data used in this study are publicly available. The New York Road Network dataset is provided by OpenStreetMap via the following link:

https://www.openstreetmap.org/relation/175905. The New York Taxi dataset is made available by the New York City Taxi and Limousine Commission at <u>www1.nyc.gov/site/tlc/about/tlc-trip-</u> record-data.page. The exact algorithm for the framework can be provided upon request at hhngo@memphis.edu.

Conflict of Interest Statement

All authors declare that they have no conflicts of interest.

APPENDIX 3.A. Sets, Parameters, and Decision Variables used in the paper

Sets

Ν	Set of nodes
A	Set of links
F	Set of node features or directions
Κ	Set of layers
Ι	Set of taxi trips
Т	Set of time periods

Parameters

η	Total number of trips selected for subset S in the TTS model of step 2
L	Total number of links included in subset S in the TTS model of step 2
P _i	$[A \times 1]$ vector representing links comprising the travel path of taxi trip <i>i</i>
β	Hyperparameter in the TTS model of step 2
length _a	Length of link a
t_i^{pred}	Predicted travel time for taxi trip <i>i</i>
t_i^{obs}	Observed travel time for taxi trip <i>i</i>
Δ_i	Intersection delay of trip <i>i</i>
α_a^{link}	North-bearing angle of link <i>a</i>
Α	Modified directional adjacency matrix with dimension $[N \times N \times F]$
a ^f	Subset of A at f with dimension $[N \times F]$
h_u^k	Feature representation of a node u at k^{th} hidden layer
H_k^t	$[N \times F]$ matrix contains information of all nodes in the network at k^{th}
	hidden layer and time period t
for _t	Forget value for TGCN _{1stm} at time period t
inp _t	Input value for TGCN _{lstm} at time period t
out _t	Output value for TGCN _{lstm} at time period t
act _t	Activation value for TGCN _{lstm} at time period t
s _t	Internal State value for TGCN _{lstm} at time period t
Loss	Overall loss value for the deep learning model in step 3

Decision Variables

$x_i = \{0, 1\}$	1 if trip i is selected for subset S in the TTS model of step 2, and 0
	otherwise
tt	$[A \times 1]$ vector representing link travel time from PLTT model
W^t	$[N \times N]$ weight matrix in TGCN for passing information between
	neighbors at time period t
W ^{for} , W ^{inp} ,	$[N \times 2N]$ weight matrices for passing information and calculating internal
W^{act} , W^{out}	state values in TGCN _{lstm} cell
$B^{for}, B^{inp},$	$[N \times F]$ Bias matrices in TGCN _{lstm} cell
B ^{act} , B ^{out}	

Models and Operations:

РСР	Path Choice Prediction. The first step in the framework
PLTT	Partial Link Travel Time Prediction. The second step in the framework
TTS	Taxi Trip Subsetting. An inner preprocessing model within step 2
TGCN	Traffic Graph Convolution Network. An operation for passing message
	between host node and its neighbor, specifically design to capture
	directional traffic flow.
TGCN _{lstm}	Traffic Graph Convolution Network with Long-Short Term Memory. An
	operation for passing message between host node's historical data as well
	as its neighbor's

APPENDIX 3.B.

Yen's Algorithm

Input: Graph G(V,E); Origin o; Destination d; number of shortest path K **Output:** A set of alternative paths $A = [p_1, \dots, p_k]$ 1 Find the first shortest path: A[0] = Dijkstra(G,o,d); 2 Initialize a container for storing potential paths: B = []3 For k = 1 to K do 4 A spur node ranges from the first to the node next to last of the previous k-shortest-path 5 For i = 0 to length(A[k - 1]) - 2 do Assign the spur node: $V_{spur} = A[k-1].node(i)$ 6 7 Create root path by taking nodes from origin to spur node: $P_{root} = A[k-1]$.nodes(0, i); 8 For every path p in A do: 9 If $P_{root} == p.nodes(0, i)$ 10 Remove the edges to ensure the spur path is different than the previous k-shortestpath: remove p.edge(i,i+1) from G 11 End p loop 12 Remove nodes from G that are the same as root path except for spur node: remove Proot[-V_{spur}] from G; 13 Create spur path from spur node to destination: P_{spur} = Dijkstra(G, V_{spur}, d); 14 A complete path is a combination of root and spur path: $P_{total} = P_{root} + P_{spur}$ 15 Add the potential path to the storage: B.append(Ptotal) End i loop 16 17 Sort the potential paths from smallest to largest cost: B.sort() The kth-shortest path is the path with lowest cost from B: A[k] = B[0]18 19 End k loop 20 Return A

REFERENCES

- Biem, A., Bouillet, E., Feng, H., 2010. IBM infosphere streams for scalable, real-time, intelligent transportation services | Proceedings of the 2010 ACM SIGMOD International Conference on Management of data [WWW Document]. URL https://dl.acm.org/doi/10.1145/1807167.1807291 (accessed 7.20.20).
- Chondrogiannis, T., Bouros, P., Gamper, J., Leser, U., Blumenthal, D.B., 2020. Finding kshortest paths with limited overlap. The VLDB Journal 29, 1023–1047. https://doi.org/10.1007/s00778-020-00604-x
- City of New York, 2019. Taxi Trips Record from the New York City Taxi & Limousine Commission [WWW Document]. URL https://www1.nyc.gov/site/tlc/about/tlc-triprecord-data.page (accessed 7.20.20).
- Cui, Z., Henrickson, K., Ke, R., Wang, Y., 2018. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting 11.
- Diao, Z., Wang, X., Zhang, D., Liu, Y., Xie, K., He, S., 2019. Dynamic Spatial-Temporal Graph Convolutional Neural Networks for Traffic Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 33, 890–897. https://doi.org/10.1609/aaai.v33i01.3301890
- DOT, 2019. Intelligent Transportation Systems Benefits of Intelligent Transportation Systems Fact Sheet [WWW Document]. URL

https://www.its.dot.gov/factsheets/benefits_factsheet.htm (accessed 7.20.20).

- Duan, Y., Lv, Y., Liu, Y.-L., Wang, F.-Y., 2016. An efficient realization of deep learning for traffic data imputation. Transportation Research Part C: Emerging Technologies 72, 168– 181. https://doi.org/10.1016/j.trc.2016.09.015
- Ermagun, A., Levinson, D., 2018. Spatiotemporal traffic forecasting: review and proposed directions. Transport Reviews 38, 786–814.
 - https://doi.org/10.1080/01441647.2018.1442887
- Fountoulakis, M., Bekiaris-Liberis, N., Roncoli, C., Papamichail, I., Papageorgiou, M., 2017.
 Highway traffic state estimation with mixed connected and conventional vehicles:
 Microscopic simulation-based testing. Transportation Research Part C: Emerging
 Technologies 78, 13–33. https://doi.org/10.1016/j.trc.2017.02.015
- Frank, M., 1981. The Braess paradox. Mathematical Programming 20, 283–302. https://doi.org/10.1007/BF01589354
- Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., Liu, Y., 2019. Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 33, 3656–3663. https://doi.org/10.1609/aaai.v33i01.33013656
- Google, 2020. TensorFlow [WWW Document]. TensorFlow. URL https://www.tensorflow.org/ (accessed 9.23.20).
- Hochreiter, S., Schmidhuber, J., 1997. LSTM can Solve Hard Long Time Lag Problems, in: Mozer, M.C., Jordan, M.I., Petsche, T. (Eds.), Advances in Neural Information Processing Systems 9. MIT Press, pp. 473–479.
- Hoffman, W., Pavley, R., 1959. A Method for the Solution of the Nth Best Path Problem. Journal of the ACM 6, 506–514. https://doi.org/10.1145/320998.321004

- Hu, X., Chiu, Y.-C., 2015. A Constrained Time-Dependent K Shortest Paths Algorithm
 Addressing Overlap and Travel Time Deviation. International Journal of Transportation
 Science and Technology 4, 371–394. https://doi.org/10.1016/S2046-0430(16)30169-1
- Jenelius, E., Koutsopoulos, H.N., 2013. Travel time estimation for urban road networks using low frequency probe vehicle data. Transportation Research Part B: Methodological 53, 64–81. https://doi.org/10.1016/j.trb.2013.03.008
- Kawasaki, Y., Hara, Y., Kuwahara, M., 2019. Traffic State Estimation on a Two-Dimensional Network by a State-Space Model. Transportation Research Procedia 38, 299–319. https://doi.org/10.1016/j.trpro.2019.05.017
- Ke, J., Zheng, H., Yang, H., Chen, X. (Michael), 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. Transportation Research Part C: Emerging Technologies 85, 591–608. https://doi.org/10.1016/j.trc.2017.10.016
- Khan, S.M., Dey, K.C., Chowdhury, M., 2017. Real-Time Traffic State Estimation With Connected Vehicles. IEEE Transactions on Intelligent Transportation Systems 18, 1687– 1699. https://doi.org/10.1109/TITS.2017.2658664
- Kipf, T.N., Welling, M., 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs, stat].
- Kumar, A., Haque, K., Mishra, S., & Golias, M. M., 2019. Multi-criteria based approach to identify critical links in a transportation network. Case Studies on Transport Policy, 7(3), 519-530.
- Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bikesharing network: A graph convolutional neural network approach. Transportation

Research Part C: Emerging Technologies 97, 258–276. https://doi.org/10.1016/j.trc.2018.10.011

- Liu, H., Jin, C., Yang, B., Zhou, A., 2018. Finding Top-k Shortest Paths with Diversity. IEEE Transactions on Knowledge and Data Engineering 30, 488–502. https://doi.org/10.1109/TKDE.2017.2773492
- Liu, Z., Liu, Y., Meng, Q., Cheng, Q., 2019. A tailored machine learning approach for urban transport network flow estimation. Transportation Research Part C: Emerging Technologies 108, 130–150. https://doi.org/10.1016/j.trc.2019.09.006
- Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., Chung, E., 2016. Real-time traffic state estimation in urban corridors from heterogeneous data. Transportation Research Part C: Emerging Technologies, Advanced Network Traffic Management: From dynamic state estimation to traffic control 66, 99–118. https://doi.org/10.1016/j.trc.2015.07.005
- Papageorgiou, M., Kiakaki, C., Dinopoulou, V., Kotsialos, A., Yibing Wang, 2003. Review of road traffic control strategies. Proc. IEEE 91, 2043–2067. https://doi.org/10.1109/JPROC.2003.819610
- Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transportation Research Part C: Emerging Technologies 79, 1–17. https://doi.org/10.1016/j.trc.2017.02.024

Ralph, J., 2013. 2.5 quintillion bytes of data created every day. How does CPG & Retail manage it? [WWW Document]. IBM Consumer Products Industry Blog. URL https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/ (accessed 12.29.19).

- Sekuła, P., Marković, N., Vander Laan, Z., Sadabadi, K.F., 2018. Estimating historical hourly traffic volumes via machine learning and vehicle probe data: A Maryland case study.
 Transportation Research Part C: Emerging Technologies 97, 147–158.
 https://doi.org/10.1016/j.trc.2018.10.012
- Seo, T., Bayen, A.M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: A comprehensive survey. Annual Reviews in Control 43, 128–151. https://doi.org/10.1016/j.arcontrol.2017.03.005
- Sheffi, Y., 1975. Urban Transportation Networks | Professor Yossi Sheffi [WWW Document]. URL https://sheffi.mit.edu/book/urban-transportation-networks (accessed 3.2.20).
- Tan, H., Feng, G., Feng, J., Wang, W., Zhang, Y.-J., Li, F., 2013. A tensor-based method for missing traffic data completion. Transportation Research Part C: Emerging Technologies 28, 15–27. https://doi.org/10.1016/j.trc.2012.12.007
- Thapa, D., Paleti, R., & Mishra, S., 2022. Overcoming challenges in crash prediction modeling using discretized duration approach: An investigation of sampling approaches. Accident; analysis and prevention, 169, 106639.
- Wang, Y., Zhang, D., Liu, Y., Dai, B., Lee, L.H., 2019. Enhancing transportation systems via deep learning: A survey. Transportation Research Part C: Emerging Technologies 99, 144–163. https://doi.org/10.1016/j.trc.2018.12.004
- World Bank, 2019. Urban population (% of total population) | Data [WWW Document]. URL https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS (accessed 7.20.20).
- Wu, C., Thai, J., Yadlowsky, S., Pozdnoukhov, A., Bayen, A., 2015. Cellpath: Fusion of cellular and traffic sensor data for route flow estimation via convex optimization. Transportation

Research Part C: Emerging Technologies 59, 111–128. https://doi.org/10.1016/j.trc.2015.05.004

- Xu, D., Wei, C., Peng, P., Xuan, Q., Guo, H., 2020. GE-GAN: A novel deep learning framework for road traffic state estimation. Transportation Research Part C: Emerging Technologies 117, 102635. https://doi.org/10.1016/j.trc.2020.102635
- Yang, S., Ma, W., Pi, X., Qian, S., 2019. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. Transportation Research Part C: Emerging Technologies 107, 248–265. https://doi.org/10.1016/j.trc.2019.08.010
- Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z., 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction 8.
- Yen, J.Y., 1971. Finding the K Shortest Loopless Paths in a Network. Management Science 17, 712–716.
- Yeon, J., Elefteriadou, L., Lawphongpanich, S., 2008. Travel time estimation on a freeway using Discrete Time Markov Chains. Transportation Research Part B: Methodological 42, 325– 338. https://doi.org/10.1016/j.trb.2007.08.005
- Yu, B., Li, M., Zhang, J., Zhu, Z., 2019. 3D Graph Convolutional Networks with Temporal Graphs: A Spatial Information Free Framework For Traffic Forecasting. arXiv:1903.00919 [cs, stat].
- Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X., 2017. Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. Sensors 17, 1501. https://doi.org/10.3390/s17071501

- Zhan, X., Hasan, S., Ukkusuri, S.V., Kamga, C., 2013. Urban link travel time estimation using large-scale taxi data with partial information. Transportation Research Part C: Emerging Technologies 33, 37–49. https://doi.org/10.1016/j.trc.2013.04.001
- Zheng, F., Van Zuylen, H., 2013. Urban link travel time estimation based on sparse probe vehicle data. Transportation Research Part C: Emerging Technologies 31, 145–157. https://doi.org/10.1016/j.trc.2012.04.007
- Zhu, L., Guo, F., Polak, J.W., Krishnan, R., 2018. Urban link travel time estimation using traffic states-based data fusion. IET Intelligent Transport Systems 12, 651–663. https://doi.org/10.1049/iet-its.2017.0116
This page is intentionally left blank.

4. OPERATION STRATEGY FOR MIXED AUTONOMOUS ELECTRIC TAXI FLEET BASED ON REINFORCEMENT LEARNING

ABSTRACT

Current Transportation Network Company is hindered by the imbalance between supply and demand in spatial and temporal dimensions. One plausible solution is the introduction of autonomous electric taxi (AET) and the central controller can preemptively and optimally reposition taxis to high demand area. In addition, the controller needs to simultaneously consider the need of human driven taxi (HV) because the near future adoption of AET is likely to include both types. This paper develops a framework namely RL-AET for dispatching single and carpooling taxi trips, repositioning, and recharging a fleet of autonomous electric taxi with a multi-objective of minimizing customer and system-oriented costs. The framework makes use of optimization models and minimum weight perfect matching for dispatching single and carpooling trips for both AET and HV and a reinforcement learning model for repositioning AET between zones and recharging. An asynchronous solving algorithm is used to ensure the framework is computationally capable of real-world network applications. Our RL-AET framework is performed on the case study of Chicago Road network and compared with the Baseline Model strategy from traditional practices. The result shows that the proposed framework is able to reduce the total weekly wait time and number of cancelled trips by 75% and 72% respectively.

Keywords: Autonomous electric taxi; reinforcement learning; graph matching; repositioning; carpooling; recharging; asynchronous algorithm.

4.1. OVERVIEW

Urban areas continue to grow in population and the transportation system need to catch up with this demand either in infrastructure or in improved efficiency. One such improvement is the evolution of taxi services from traditional taxi to Transportation Network Companies (TNC) leveraging technology and connectivity such as Uber and Lyft. However, even TNC performance is still hindered by the imbalance between supply and demand on both spatial and temporal dimensions. TNCs employ economic incentives to address the supply-demand imbalance to a limited degree of success. Specifically, these incentives, or surge pricing, regulate the price based on the level of imbalance between supply and demand which is disaggregated in both spatial and temporal dimension. However, it is shown that surge pricing has a small positive impact on the driver's decision for accepting trips further away but considerable negative impact on the customer decision on whether to cancel the trips (Chen et al., 2015). Both strategies are based on discrete time or space interval, and this can create discrepancy where two customers are closely located but charged at different prices if they are separated by the border between zone or time period.

A potential solution to this supply demand imbalance is autonomous taxi and these vehicles are likely to be electric. Autonomous Electric Taxi (AET) is fully compliant to a central strategy and recent researches have been devoted into developing cooperative strategy between vehicle for promoting improved efficiency (Chen et al., 2016; Lin et al., 2018; Mao et al., 2020; Turan et al., 2020). The central strategy can include repositioning vehicles from over flocked areas to future potential high demand areas and more demand will be fulfilled. In addition, idling time in low demand area can be utilized for recharging. There is also another layer of complexity where TNC needs to continuingly offer service to human-driven vehicle (HV) taxi while simultaneously operates their AET. Researchers have developed several frameworks for operational strategy for AET with approaches ranging from model-based to data-driven (Liang et al., 2021; Mao et al., 2020; Shou et al., 2020; Tang et al., 2020; Yang et al., 2020). The literature still lacks research on making sequential decisions of single pickup and carpooling for both HV and AET, repositioning and recharging of AET on real-world network (see section 2.2 for detailed research gaps). Therefore, the objective of this paper is to *develop a framework for sequentially perform the following operation: (1) combining taxi demands into carpooling trips, (2) matching and dispatching both HV and AET with these trips, and (3) repositioning and recharging AET with the system-level objective of minimizing the customer wait time, cancellation penalty, repositioning cost, and undercharged penalty accumulated over the operating day.*

The remainder of the paper is organized as follows: Section 4.2 reviews the literature on operational strategy for ridesharing services. Section 4.3 provides the methodology which includes framework architecture, individual processes, and the solution algorithm. In Section 4.4, we apply the framework in the Chicago Road network and provide insights on the training process. Section 4.5 compares the RL-AET framework with the Baseline Model strategty. Section 4.6 concludes our finding in this paper and proposes new future avenue for improving the research.

4.2. LITERATURE REVIEW

4.2.1. Operation Research on Autonomous Taxi

Research in operational strategy for autonomous taxi often aims to determine the strategy for dispatching, repositioning, and recharging. The approaches of these studies can be sectioned into either model-based or data-driven. Model-based approaches use principles to represent real-

world strategy and the principle is mathematically formulated into either deterministic or optimization models. Yang et al. (2020) proposed a spatial probability model to adjust and optimize both matching time interval and matching radius. The paper utilized a multi-objective function to both minimize customer wait and pickup time and maximize matching rate. Based on queuing theoretic, Ma et al. (2019) developed a non-myopic model for dispatching strategy of an integrated system between ridesharing company and the transit system. However, model-based approach is limited by the formulation and ultimately, the principles that it relies on.

In contrast, data-driven does not rely on principle and is model-free and more flexible. In the field of operation research for AET, Reinforcement Learning (RL) is a new but promising approach. RL works on the principle where an agent (or agents) takes action in an environment. After an action is executed, the agent's state will be transitioned, and the agent receives a local reward. The objective is to maximize the cumulative reward at the end of the simulation. In AET operation, the environment usually includes the road network, the taxi fleet, and the customer demand orders. There can be a centralized single agent representing the central controller making all the decisions or decentralized multiple agents representing individual AET vehicles, which is referred as Multi-agent Reinforcement Learning (MARL). Another branch of RL is Inverse RL where the agent will observe from the real-world best practices and develops its policy accordingly. However, due to the lack of stringent constraints, RL model does not converge in a reasonable computing time.

So far, the literature in AET operation mainly concerns with three operations which are (1) combining demands into carpooling trips, (2) matching these trips with available taxis, and (3) repositioning and recharging AET. In the first operation, research commonly introduces certain carpooling criteria and the collection of demand pairs satisfying these criteria will form a

connectivity graph. Therefore, the problem of determining optimal carpooling trips can be solved as maximum matching in this connectivity graph (Alonso-Mora et al., 2017; Vazifeh et al., 2018; Zhan et al., 2016). In the trip-vehicle matching, the problem can be formulated as an integer linear programming since it provides better flexibility in including real-world constraints. The decision variable is often binary, and the objective function is linearly related to the decision variable which facilitates convergence. Finally, in repositioning, unlike the first two operations, the benefit of reposition is not immediately realized and directly related to the repositioning action. It may take several timestamps for a vehicle to be repositioned, arrive at the right zone, and realize the benefit. However, the effect of repositioning is significant and strategic repositioning will greatly benefit the long-term cumulative reward. Optimization is not equipped to formulate and capture this problem where the objective is latent to the decision variable and thus, it is not suitable for representing the repositioning process. In contrast, RL is perfectly suitable for repositioning because it makes use of the policy and value function, which is often configured as a neural network, to learn the latent relationship between repositioning action and the reward and executes a strategic sequence of actions to achieve maximum long-term cumulative reward. In Table 4.1, we provide a summary of papers in operational strategy for autonomous taxi.

4.2.2. Research Gap and Contributions

After reviewing the literature, we recognize four research gaps and provide contributions of this paper as follows:

 First, studies (Lin et al., 2018; Mao et al., 2020b; Shi et al., 2020; Shou et al., 2020; Tang et al., 2020; Yang et al., 2020) considered AET operation to be zone-based, resulting primarily in two drawbacks: (i) inaccuracy in actual travel time computation; and (ii) vehicle location, itinerary, and demand pickup and drop-off location cannot be accurately represented. The proposed framework utilizes both zone-based to accelerate the vehicle-demand matching and node-based road network to accurately record the vehicle location, itinerary, and demand pickup and drop-off location.

- 2. Second, studies (Liang et al., 2021; Lin et al., 2018; Ma et al., 2019; Shi et al., 2020; Yang et al., 2020; Zhou et al., 2019) develop their framework around single pickup, with additional surge pricing included in some studies, but the majority of which do not simultaneously include efficient strategies such as carpooling and repositioning. This inhibits the potential for more matching. Our framework considers not only AET taxi but also HV and provides both parties all dispatching option namely single pickup and carpooling. In addition, customers wait time and HV wait time are explicitly modeled as those with higher waiting times are prioritized.
- 3. Third, the literature (Mao et al., 2020) assumes the taxi supply is drawn from a statistical distribution based on historical data and this is a common current practice in TNC where HV user can go online at any time and location. However, for managing a fixed AET fleet, this approach is limited in keeping track of every AET because TNC needs to understand the temporal state and status of every vehicle. To the best of our knowledge, our framework is the first to allocates for each vehicle an itinerary matrix, which records, at both current and future timestamps, the AET's location, zone, status (e.g., idling or single pickup, etc), and state of charge. Our framework directly and methodically uses the matrix in the formulation of dispatching and repositioning. In particular, the matrix assists Process

3 in accessing AET's current location and scheduled drop-off location and time and Process 4 in calculating future available vehicles and undercharged AETs.

4. Finally, studies (Mao et al., 2020; Shi et al., 2020; Tang et al., 2020; Yang et al., 2020) apply their framework on small node-based example (2 link 2 node network) or an urban area but limited to zone-based (Manhattan with 8 taxi zones) partially due to limitation in the solving algorithm. This paper utilizes an asynchronous approach where multiple agents are experiencing and updating the policy concurrently. This accelerates the learning process since every agent will benefit from the shared learning experience. Therefore, the framework is applicable to real network as shown in the Case Study section.

AET Action Observation Authors Network Approach Case Study Scope Considered (Ma and Carpooling and Node-Heuristic Utilize request-in-advance to yield best performance for carpooling. Chengdu, China. Koutsopoulos, repositioning based Algorithm New York, USA 2022) (Liang et al., 2021) Single pickup, Does not consider carpooling. Node-MARL Manhattan, NY rebalancing based Single Pickup with RL (with (C. Chen et al., Node-Dynamic pricing strategy is zone-based which can create Hangzhou, China 2021) ABMS) discrepancy for customer nearing each other. dynamic pricing based (Jiao et al., RL SARSA solving algorithm has limited capability since the level of Didi Chuxing with Repositioning Node-2021) complexity is not linear with the variable range. three cities based (X. Chen et al., Single pickup Modified Monte Carlo tree search is significantly faster than Branch Hangzhou, China Node-RL 2021) & Bound and Brute Force algorithm. based (Turan et al., 2020) Single pickup with Node-Equity between customer (wait time) is not considered. Manhattan and San RL pricing and charging Francisco based Maximizing matching rate and minimizing wait time are included in (Yang et al., 2020) Single pickup Zone-Model-based A study area of 500 km² the multi-objective function making it lacks practical interpretation. based (Shi et al., 2020) Single pickup, Zone-RL Customer equity is not considered. The model formulation can 8x16 Simulated charging based match one order with two taxis and receives reward for both. Zones Prior data from real drivers is required and the model's optimality is Vehicle trajectories (Shou et al., 2020) Repositioning Zone-Inverse RL based limited to this dataset and not the theoretical optimum. from Didi Chuxing Model does not consider a fixed-size fleet of taxi but rather the (Mao et al., 2020) Single pickup, RL Manhattan, New Zone-Repositioning based supply is drawn from a distribution. York with 64 zones

 Table 4.1. Summary of Literature on Operational Strategy for Autonomous Taxi

(Tang et al., 2020)	Single pickup,	Zone-	RL	The model formulation requires discretizing the vehicle's remaining	Tongzhou, China
	rebalancing, charging	based		battery range and the bin can be subjective.	
(Ma et al., 2019)	Single pickup,	Node-	Model-based	The model is suitable for trips from suburban to downtown district	Long Island and
	repositioning	based		(Long Island to Manhattan) rather than daily intra urban commute.	Manhattan, NYC
(Zhou et al., 2019)	Repositioning	Zone-	MARL	No specifics about which individual taxis are matched with which	Didi Chuxing
		based		specific demand request.	
(Simonetto et al.,	Carpooling,	Node-	Federated	Total unimodular constraints cannot embed complex real-world	New York City
2019)	Repositioning	based	optimization	operation constraints.	
(Lin et al., 2018)	Repositioning	Zone-	MARL	Maximization of local agent's reward does not guarantee the	Chengdu, China
		based		maximization of system's performance.	
(Alonso-Mora et	Single pickup,	Node-	Graph	Repositioning is simply empty vehicle dispatching which might not	Manhattan, New
al., 2017)	carpooling,	based	Matching	guarantee long-term optimal system performance.	York
	rebalancing				
(Chen et al., 2016)	Matching,	Node-	Agent-based	The paper is based on simulation where deterministic algorithm is	Austin, Texas
	rebalancing, and	based	modeling	used for matching and repositioning. Missed opportunity in allowing	
	recharging			the model to learn from experience	
(Zhan et al., 2016)	Matching	Node-	Graph-	Cannot embed more real-world constraints such as zonal matching,	New York Taxi
		Based	matching	electric range, preference for vehicle type.	

4.3. METHODOLOGY

4.3.1. Problem Settings

This paper proposes a centralized approach where the taxi coordinator, herein referred to as the controller, will make decisions for every taxi in the fleet whether it is single pickup, carpooling, repositioning, recharging, or idling. The spatial setup of the framework is a road network where each intersection is a node and road segments are links connecting these nodes. If a taxi's location is in the middle of the segment, the algorithm will approximate it into the nearest node and this approximation is utilized to significantly reduce the computational complexity of the framework. The city is divided into taxi zones, and this is a common practice among both public and private taxi services. These taxi zones facilitate the algorithm in matching taxi and customer and repositioning taxi between zones. However, the rest of our framework is still node-based which includes vehicle location, itinerary, and demand pickup and drop-off location.

In temporal setup, our framework considers a continuous period during a day (i.e., 06:00 AM to 08:00 PM) which constitutes an episode. Note that our taxi demand profile will vary through time of the day and day of the week. Within an episode, we discretize it into intervals with equal duration Δt (i.e., $\Delta t = 5$ minutes) and the index of each interval constitutes a timestep. Therefore, an episode can be described as $T = \{t \in [1, 2, ..., T_{max}]\}$.

In terms of taxi demand request, a request will contain the following features namely (1) pickup location, (2) drop-off location, (3) pickup time, and (4) preference for carpooling (i.e., do/do not want to carpool). In request's pickup time, the customer can either request to be picked up at the current timestep or future timestep (reserving in advance). Without the reserving in advance option, carpooling strategy will be less effective.

We make the following assumptions on HV as follows. First, HV supply is drawn from a historical supply distribution in both spatial and temporal dimension. In Process 2, 3, and 4, HVs are eligible to pick up any taxi trips and in some instances, it is mandatory for the trip to be completed by HV due customer's preference. HV with longer waiting time will be prioritized in these dispatching processes. If a HV is matched and completed a trip, it would log out of the system and the controller no longer needs to consider it. If a HV is not matched at this time period, it will be transferred to the next period with an updated waiting time. If the HV waiting time without a match is greater than a certain threshold, it will log out of the system and incurs a significant cost in the dispatching processes.

4.3.2. Model Architecture

The framework consists of multiple processes for matching AETs and HVs with taxi request, which also includes carpooling for optimal performance, and repositioning and recharging AETs if necessary. The model's architecture is shown in Figure 4.1.



Figure 4.1. Schematic View of Model Architecture

An episode starts with the first timestep and the controller gathers the vehicle state, which includes location, taxi zone, status, state of charge, and itinerary. Demand requests within a timestep will be aggregated and processed at the end of the timestep. These requests include both immediate request and reserving-in-advance request. First, the controller will use Process 1 to determine a set of trips that optimally pickup all demands. In this step, a trip can consist of more

than one taxi demand representing carpooling. Process 1 does not concern with the limit for number of demands in a trip because in the post-processing step, trips larger than the predetermined threshold Δ , which is usually the taxi's capacity, are separated into two new trips. After forming trips in Process 1, Process 2 assigns these trips to currently empty AETs and HVs within the same zone and prioritizes trips and HVs with higher waiting time. We make an assumption that HV supply will be drawn from a normal distribution based on historical data (Mao et al., 2020), but our framework is capable of any setting. Penalties for both unserved trips and HV waiting time are incurred in Process 2's objective function. If there are still unserved demand leftover, the controller will use Process 3 to match a single trip (only includes one customer) with AETs and HVs currently fulfilling a single customer within the same zone. Process 3 provides another alternative for carpooling where the carpooling trip do not need to be formed prior to matching but it can be formed while matching and the taxi is active. If there are still unserved trip, the controller will use Process 4 to match trips with empty AETs and HVs in the adjacent zones where pickup time is less than 10 minutes. The second condition is added to ensure the algorithm only searches for vehicle around the border between two zones. If there are still unserved trips, these will be transferred to the next timestep with an updated waiting time. Unmatched HVs will be stored and transferred to the next timestep with updated waiting time. After matching vehicles with taxi demand, the controller will use Process 5 to reposition AETs between zones and recharge. The timestep finishes by updating the vehicle and trip location, status, zone, and itinerary.

4.3.3. Dispatching and Repositioning Process

First, the notation used will be explained in text and in addition, we provide a summary table for all notations in Appendix 4.A.

Process 1: Determinizing the Set of Carpooling Trips

Process 1 consists of two main steps: (1) Constructing Taxi Request Shareability Graph and (2) Solving Minimum Path Cover of Directed Acyclic Graph.

Constructing Taxi Request Shareability Graph G_{TRS}

The purpose of this step is to explore how taxi demand request can be combined together to form a set of carpooling trips. Given a set of taxi demand request D, each request is formulated as a tuple $[pu_d, do_d, t_d^{dem}, c_d, \varphi_d]$, $\forall d \in D$ representing pickup and dropoff location, desired pickup time, preference for carpooling, and preference for type of service (AET or HV). First, we form a Taxi Request Shareability Graph $G_{TRS} = (V_{TRS}, E_{TRS})$, where each request $d \in D$ will form a node $v \in V_{TRS}$ and each edge $e(d_1, d_2) \in E_{TRS}$ states that the pair demand d_1 and d_2 is eligible for forming a carpooling trip. Process 1 scans through the batch of taxi demand and forms an edge in the shareability graph $e(d_1, d_2) \in E_{TRS}$ if and only if the pair of demands (d_1, d_2) concurrently satisfies the carpooling constraints as follows:

$$t_{d_1}^{dem} + \tau_{pu_{d_1} \to pu_{d_2}} \le t_{d_2}^{dem} \tag{47}$$

$$t_{d_2}^{dem} - t_{d_1}^{dem} \le \Delta_1 \tag{48}$$

$$\tau_{do_{d_1} \to do_{d_2}} \le \Delta_2 \tag{49}$$

$$c_{d_1}c_{d_2} = 1 \tag{50}$$

$$\varphi_{d_1} = \varphi_{d_2} \tag{51}$$

Constraint (1) indicates that after picking up customer d_1 at time $t_{d_1}^{dem}$, the taxi needs to travel to customer 2 pickup location and arrives no later than the requested pickup time $t_{d_2}^{dem}$. Constraint (2) ensures the transit time between picking up customer d_1 and d_2 is less than a certain

threshold Δ_1 . Constraint (3) ensures the travel time between the drop off location of customer d_1 and d_2 is less than a certain threshold Δ_2 . Constraint (4) states both customers allow carpooling option for their trips. Constraint (5) describes both customers have the same preference for type of vehicle. In addition to these constraints, without loss of generality, we assume d_1 and d_2 have different pickup location so that:

$$\tau_{pu_{d_1} \to pu_{d_2}} > 0 \tag{6}$$

In the case that d_1 and d_2 have the same pickup location, without loss of generality, Process 1 automatically combines these two demands together into one trip and this trip is represented by only one single vertex in the G_{TRS} . This assumption facilitates the acyclic feature for G_{TRS} and considerably accelerates the algorithm as will be discussed in the later section. However, these constraints alone cannot reflect how effective the carpooling trip is compared to two single trips. Therefore, Process 1 assigns a cost to these edges as calculated in Equation (7) as follows:

$$cost_{e_{(d_1,d_2)}} = \frac{\tau_{d_1|d_1+d_2} - \tau_{d_1}}{\tau_{d_1}} + \frac{\tau_{d_2|d_1+d_2} - \tau_{d_2}}{\tau_{d_2}}$$
(7)

Where:

 $\tau_{d_1|d_1+d_2}$ Travel time experienced by the customer d_1 in the carpooling trip $\tau_{d_2|d_1+d_2}$ Travel time experienced by the customer d_2 in the carpooling trip τ_{d_1} Travel time experienced by the customer d_1 in a single trip τ_{d_2} Travel time experienced by the customer d_2 in a single trip

This cost indicates the detouring time which is the increase in travel time experienced by both customer d_1 and d_2 due to carpooling instead of single trips. In addition, to ensure both customers would not experience a significant increase in their expected travel time, we limit this weight to a certain threshold Δ_3 as follows

$$cost_{e_{(d_1,d_2)}} \le \Delta_3$$
 (8)

This would eliminate such long detours. In our algorithm, instead of looping through every pair of d_1 and d_2 demands possible, we make use of matrix manipulation to arrive at the list of eligible edges. Below is the pseudocode of the Algorithm for feasibility checking.

Algorithm 1: Feasibility Checking for Carpooling Edges

Input: *d_pickup*, *d_dropoff*, *d_time*, *d_carpool*, *d_vehtype* are vectors representing the pickup and drop-off location, the desirable pickup time, preference for carpooling, and preference for vehicle type respectively. *node*_{traveltime} is a matrix representing the travel time between two nodes. The rows and columns represent the start and end of the trip.

1.	Let <i>cst</i> 1, <i>cst</i> 2, <i>cst</i> 3, <i>cst</i> 4, <i>and cst</i> 5 represents the matrix for constraints 1 through 5
	respectively. Each matrix has dimension of $[D \times D]$ where the row and column represent
	the first and second customer respectively. The edge cost also E_{cost} is also a $[D \times D]$
	matrix where the (d_1, d_2) cell value is equal to $cost_{e_{(d_1, d_2)}}$.
2.	D_{time}^{origin} is a $[D \times D]$ matrix where the first column is d_{time} and it is broadcasted to the
	remaining column. The same operation also applies to $D_{carpool}^{origin}$, $D_{vehtype}^{origin}$.
3.	D_{time}^{destin} is a $[D \times D]$ matrix where the first row is d_{time} and it is broadcasted to the
	remaining rows. The same operation also applies to $D_{carpool}^{destin}$, $D_{vehtype}^{destin}$.
4.	Let the [<i>row</i> , <i>col</i>] be the subsetting of a matrix where <i>row</i> , <i>col</i> is the set of selected rows
	and columns respectively.
5.	Calculate: $cst1 = D_{time}^{origin} + node_{traveltime}[d_pickup: d_pickup] - D_{time}^{destin}$
6.	Calculate: $cst2 = D_{time}^{destin} - D_{time}^{origin} - \Delta_1$
7.	Calculate: $cst3 = node_{traveltime}[d_dropoff:d_dropoff] - \Delta_2$
8.	Calculate: $cst4 = D_{carpool}^{origin} \cdot D_{vehtype}^{destin} - 1$
9.	Calculate: $cst5 = D_{vehtype}^{origin} - D_{vehtype}^{destin}$
10.	Calculate: $cst6 = E_{cost} - \Delta_3$

11. Calculate the matrix True/False for eligible links:
cst = (cst1 ≤ 0) & (cst2 ≤ 0) & (cst3 ≤ 0) & (cst4 = 0) & (cst5 = 0) & (cst6 ≤ 0) Note that the operation & is element wise.
Output: wherever a cell cst[d_i,d_j] has a True value, then the demand pair d_i,d_j can form an edge in the shareability graph.

Figure 4.2a gives an example of 6 pairs of taxi in a road network and there are potentially 30 edges that can be formed. Process 1 constructs the Taxi Request Shareability Graph, and the result is shown in Figure 4.2b. Demands are represented by nodes and because of the aforementioned constraints, there are only 6 edges indicating 6 potential carpooling pairs. It is important to identify that the graph is directed and $e_{(d_1,d_2)} \neq e_{(d_2,d_1)}$.

Definition 1: Given the Taxi Request Shareability Graph $G_{TRS} = (V_{TRS}, E_{TRS})$, the vertexdisjoint path cover of G_{TRS} is a set of directed paths $\{P_1, P_2, P_3, \dots, P_m\}$ with size *m* such that every vertex $v \in V_{TRS}$ belongs to exactly one path. Each path is represented by a sequence of edges $\{e_1, e_2, \dots, e_k\}$ and the path length can be 0 if the path only covers one vertex. The mathematical equivalent of this notion is:

$$N(P_1) \cup N(P_2) \dots N(P_m) = V_{TRS}$$
 (9)

$$N(P_i) \cap N(P_i) = \emptyset, \forall i, j \in M$$
⁽¹⁰⁾

Where $N(P_i)$ is the set of vertices covered by path P_i



Theorem 1: Let $L = \{P_1, P_2, P_3, \dots, P_m\}$ be a vertex-disjoint path cover of the request shareability graph G_{TRS} . Then all the demands in V_{TRS} can be served by *m* trips.

Proof: The merging of the sets of demands $N(P_i)$ along all path $P_i \in L$ is equal to the set of demands V_{TRS} . Since there are *m* paths and each path is served by one trip, the number of trips needed to serve V_{TRS} is *m*.

Corollary of Theorem 1: The minimum trips needed to served the set of demand V_{TRS} is equal to the minimum vertex-disjoint path cover of the request shareability network G_{TRS} .

In general, finding the minimum vertex-disjoint path cover in a graph is computationally hard and the problem is classified as NP-hard. However, due to the constraints in forming edges E_{TRS} , the G_{TRS} graph is directed and acylic and the problem can be solved in polynomial time. This is solved in the next step as follows.

Solving Minimum Vertex-Disjoint Path Cover as Minium Weight Perfect Matching in the Equivalent Bipartite Graph

Definition 2: A directed graph G = (V, E) is acylic when there is no directed cycles. This means that there is no directed path that starts at a node and ends at the same node.

Theorem 2: Because of the constraints (1), (2), and (3) in forming edges, the request shareability graph G_{TRS} is directed and acyclic.

Proof: This theorem can be proved by contradiction. Assume there exists a path P_{cycle} with a directed cycle. Without loss of generality, we can also assume the path has a length of two as follows: $P_{cycle} = \{(d_1, d_2), (d_2, d_1)\}$ where d_1 and d_2 are the demands. Since $e(d_1, d_2)$ and $e(d_2, d_1) \in E_{TRS}$, the following conditions must be satisfied concurrently:

$$t_{d_1}^{dem} + \tau_{pu_{d_1} \to pu_{d_2}} \le t_{d_2}^{dem}$$
 (a)

$$t_{d_2}^{dem} + \tau_{pu_{d_2} \rightarrow pu_{d_1}} \le t_{d_1}^{dem} \tag{b}$$

From Equation (a) and (b), we can reach:

$$\begin{split} t_{d_1}^{dem} + \tau_{pu_{d_1} \rightarrow pu_{d_2}} &\leq t_{d_1}^{dem} - \tau_{pu_{d_2} \rightarrow pu_{d_1}} \text{ or equivalently:} \\ \\ \tau_{pu_{d_1} \rightarrow pu_{d_2}} + \tau_{pu_{d_2} \rightarrow pu_{d_1}} &\leq 0 \end{split}$$

This contradicts Constraint (7) which states that: $\tau_{pu_{d_1} \rightarrow pu_{d_2}} > 0$ and $\tau_{pu_{d_2} \rightarrow pu_{d_1}} > 0$.

Since G_{TRS} is proven to be a Directed Acylic Graph, the minimum weight vertex-disjoint path cover problem for G_{TRS} is not NP-hard and can be solved in polynomial time. First, we need to transform the G_{TRS} to the equivalent Bipartite Graph $G_{TRS-Bipartite}$ as follows.

Definition 3: A bipartite graph is a graph where the nodes can be subsetted into two completely disjointed and independent group U and V. In addition, every possible links can only connect to a node from U to a node from V.

First, every vertex in G_{TRS} are duplicated and assigned to both group U and V, which represent the first and second taxi demand of the potential carpooling trip respectively. The link connecting between these two groups can be constructed from the edges of $e(d_1, d_2) \in E_{TRS}$ where d_1 belongs to the first set U and d_2 belongs to V. An example of which is shown in Figure 4.3 below. Two groups U and V are located on left and right respectively and the edges and its cost of $G_{TRS-Bipartite}$ graph corresponds to the edges and costs of G_{TRS} . Note that both subsets have the same number of vertices or |U| = |V|. In addition, we can also assume that $G_{TRS-Bipartite}$ is complete by assigning infinite costs to the edges not present.

Therefore, the problem of finding the minimum weighted vertex-disjoint path cover in graph G_{TRS} is equivalent to the well-known problem of minimum weighted perfect matching $G_{TRS-Bipartite}$. This minimum weighted perfect matching is also known as the assignment problem and has been studied extensively in the literature (Alonso-Mora et al., 2017; Zhan et al., 2016). Our model uses the Hungarian algorithm (Harold Kuhn, 1955) which has a computational complexity of $O(|V|^3)$. Due to the acylic nature of the original shareability graph G_{TRS} , the result matching will ensure timeline feasibility and there will be no trips containing any loop. Figure 4.3 shows an example on the result of this algorithm where the matched edges are colored in red. From this matching result, we can identify the two paths/trips covering the six taxi demands as follows: $P_1 = \{d_1, d_2, d_3, d_4\}$ and $P_2 = \{d_5, d_6\}$. Since the demands are formulated as d_1, d_2, d_3 , and d_4 , we formulate their respective pickup and drop-off points as $pu_1, pu_2, pu_3, pu_4, do_1, do_2, do_3, and do_4$. Thus route P_1 are equivalent to $P_1 = pu_1 \rightarrow pu_2 \rightarrow pu_3 \rightarrow pu_4 \rightarrow do_1 \rightarrow do_2 \rightarrow do_3 \rightarrow do_4$.



Figure 4.3. Equivalent Bipartite Graph

Therefore, the output of Process 1 will be a set of trips $L = \{P_1, P_2, P_3 \dots P_m\}$ where each trip is represented by a vector:

$$P_m = \{pu_m, do_m, t_m^{trip}, \varphi_m, N(P_m), p_m, num_m, w_m\}, \quad \forall m \in M$$

In this vector, pu_m , do_m , t_m^{trip} , and φ_m are the trip pickup, drop-off location, trip pickup time (derived from the first customer), and preference for type of vehicle. $N(P_m)$ is the set of demands covered by trip P_m and p_m is the itinerary of this trip. num_m , w_m are the number of demands and the combined waiting time of all demands in the trips respectively. The term trip exclusively refers to the set of demands combined together due to Process 1. From this point onward, the term demand will be used interchangeably with trip. A trip can have either one or multiple demands. The rest of the framework will only interact with trips. In Processes 2, 3, and 4, the model can view a trip as a demand since the formulation only concerns about the pickup, drop-off location, and the requested pickup time.

Process 2: Matching trip with empty idling HV and AET in the same zone

Objective Function:

Minimizing:

$$z_{2} = \alpha_{1} \sum_{d} \left[(1 - \sum_{v \in V} x_{v,m}) \rho_{m} \right] + \alpha_{2} \sum_{v \in V} x_{v,m} \tau_{l_{v},pu_{m}} + \alpha_{3} \sum_{v \in V_{HV}} \left(w_{v}^{hv} \sum_{m} x_{v,m} \right)$$
(11)

Subject to:

$$cst_{\nu,m}^{proc2} = \begin{cases} 1 \text{ if } \left\{ \begin{pmatrix} z_{\nu}^{veh} = z_{m}^{trip} \end{pmatrix} \text{ and } (s_{\nu} \in [0,1]) \\ and (rg_{\nu} \ge \varepsilon) \text{ and } (\varphi_{\nu} = \varphi_{m}) \\ 0 \text{ for otherwise} \end{cases} \right\}, \forall \nu \in V, m \in M$$

$$(12)$$

$$x_{v,m} \le cst_{v,m}^{proc2}, \forall v \in V, m \in D$$
(13)

$$\rho_m = \frac{\left[\sum_{\nu} (\tau_{l_{\nu}, o_m} cst_{\nu, m}) + w_m\right]}{\sum_{\nu} cst_{\nu, m}^{proc2}}, \forall m \in M$$
(14)

$$\sum_{m} x_{\nu,m} \le 1, \forall \nu \in V \tag{15}$$

$$\sum_{v} x_{v,m} \le 1, \forall m \in M \tag{16}$$

$$x_{\nu,m} \in \{0,1\}, \forall \nu \in V, m \in M$$

$$\tag{17}$$

Equation (11) represents the objective function of Process 1, which minimizes the system matching and dispatching cost. The first term shows that if a trip is not fulfilled by any vehicle, a penalty ρ_m is added to the system cost. The second term shows the total travel time between the taxi location to its customer's pickup location, which encourages the model to choose the nearest taxi among a set of candidates. In other deterministic approaches, taxi requests will be fulfilled by the nearest vehicle which can cause a problem if two requests share the same nearest taxi.

This paper's optimization approach overcomes this problem by being flexible (i.e., second taxi request can take the second closest taxi) while still ensuring satisfactory system level performance. The third term reflects the waiting time of HV and penalty if the waiting time is excessive. Each term is assigned with individual weight, and we set the weight as $\{\alpha_1, \alpha_2, \alpha_3\} =$ $\{0.3, 0.2, 0.2\}$ in our experiment. Equation (12) and (13) states a taxi v is allowed to fulfill trip m if both are in the same zone and taxi v is either idling or repositioning. The taxi's remaining range also must be greater than a predefined threshold to avoid the risk of running out of charge. This threshold is defined based on historic longest trip and nearest charging station. In addition, the vehicle type must match the trip preference for type of vehicle. Equation (14) shows the penalty ρ_m of trip m if it is not fulfilled, which is the average pickup time of a nearby permissible taxi. The penalty also includes the wait time w_m , which accounts for the combined wait time of all demands within the trip, so that the model will prioritize picking up trip which has been waiting for a longer time. Equation (15) says that each taxi can pick up at most one request. Equation (16) states that a maximum of one taxi is allowed to pick up request m. Equation (17) ensures the decision variable is binary where 1 shows taxi v is picking up request m and 0 for otherwise. Processes 3 and 4 (as will be described later) are formulated as a integer linear programming (ILP) and this term herein will be used in replacement of optimization for the rest of the studies.

After executing the ILP, the controller obtains the dispatching matrix showing exactly which trip request is fulfilled by which particular vehicle. The unserved trip can be represented by the trip status vector s_m^{tr} where element m^{th} gets value 1 if trip m is fulfilled and 0 for otherwise. s_m^{tr} is first initialized with all zeros and can be updated in Equation (18):

$$s_m^{tr} = s_m^{tr} + \sum_{\nu} x_{\nu,m} \quad , \forall m \in M$$
(18)

Process 3: Carpooling with currently serving HV and AET taxi

There are two options that a carpooling trip can be formed. The first option is to pre-determine which set of demands can form a trip and match empty vehicle with this trip. This method is performed by Processes 1 and 2. The second option is to assign currently single-serving HV and AET, which is not considered in the first option, with a single trip. Process 3 fulfills this gap and represents this second option, and thus, more trips can be matched.

Objective Function:

Minimizing:

$$z_{3} = \alpha_{1} \sum_{m} \left[(1 - \sum_{v \in V} x_{v,m}) \rho_{m} \right] + \alpha_{2} \sum_{v \in V} x_{v,m} \tau_{l_{v},pu_{m}} + \alpha_{3} \sum_{v \in V_{HV}} \left(w_{v}^{hv} \sum_{m} x_{v,m} \right)$$
(19)

Subject to:

If the following conditions are all concurrently satisfied, then $cst_{v,m}^{proc3} = 1$ and 0 for otherwise:

- 1. Vehicle *v* and trip *m* shares the same taxi zone.
- 2. Vehicle *v* is currently serving one customer (i.e., $s_v = 2$).
- 3. Vehicle v can pick up the trip m on or before the requested pickup time.
- 4. The current customer's wait time for pick up the second customer is less Δ_1 .
- 5. Travel time between destinations is less than Δ_2 .
- 6. Vehicle range is greater than a pre-defined threshold $(rg_v \ge \varepsilon)$.

(20)

7. Vehicle type matches the trip preference ($\varphi_v = \varphi_m$).

$$x_{\nu,m} \le cst_{\nu,m}^{proc3}, \forall \nu \in V, m \in M$$
(21)

$$\rho_m = \frac{\left[\sum_{v} (\tau_{l_v, o_m} cst_{v, m}^{proc3}) + w_m\right]}{\sum_{v} cst_{v, m}^{proc3}}, \forall m \in M$$

$$(22)$$

$$\sum_{m} x_{\nu,m} \le 1 - s_m^{tr}, \forall m \in M$$
(23)

And constraints represented by Equations (15) to (17)

The objective function of Process 2 is minimizing the system matching and dispatching cost as shown in Equation (19), which is similar to Equation (11). Equations (20) and (21) ensure only taxis concurrently satisfying all the listed conditions is eligible for carpooling. Equation (22) shows the penalty ρ_m for failed trip. Equation (23) states that the trip fulfilled in Process 2 will not be fulfilled again in Process 3. After the ILP is executed, the framework updates the trip status vector s_m^{tr} as shown in Equation (18).

Process 4: Matching trip with idling HV and AET taxi in adjacent zone

In Process 2, by introducing the concept of zonal matching, the original set of trips and vehicles are divided into batches based on their zone. Since each batch has smaller size than the original set, the algorithm time is reasonably accelerated. However, it leaves out the pair of trips and vehicles which are close to each other but separated by the zone border line. Process 4 fills in this gap by matching trip with idling HV and AET in adjacent zone. The formulation is as follows:

Objective Function:

Minimizing:

$$z_4 = \alpha_1 \sum_m \left[\left(1 - \sum_{v \in V} x_{v,m} \right) \rho_m \right] + \alpha_2 \sum_{v \in V} x_{v,m} \tau_{l_v, p u_m} + \alpha_3 \sum_{v \in V_{HV}} \left(w_v^{hv} \sum_m x_{v,m} \right) \right]$$
(24)

Subject to:

$$cst_{v,m}^{proc4} = \begin{cases} 1 \text{ if } \begin{cases} \left(z_{v}^{veh} \epsilon \ Z_{z_{m}^{dm}}^{adjacent} \right) \text{ and } \left(\tau_{l_{v},pu_{m}} \leq \Delta_{3} \right) \\ \left(rg_{v} \geq \varepsilon \right) \text{ and } \left(\varphi_{v} = \varphi_{m} \right) \text{ and } \left(s_{v} \in [0,1] \right) \end{cases}, \forall v \in V, m \in M \end{cases}$$

$$(25)$$

$$0 \text{ for otherwise}$$

$$x_{\nu,m} \le cst_{\nu,m}^{proc4}, \forall \nu \in V, m \in M$$
(26)

$$\rho_m = \frac{\left[\sum_{v} (\tau_{l_v, o_m} cst_{v, m}) + w_m\right]}{\sum_{v} cst_{v, m}}, \forall m \in M$$

$$(27)$$

$$\sum_{m} x_{\nu,m} \le 1 - s_m^{tr}, \forall \nu \in V, m \in M$$
(28)

And constraints represented by Equations (14) to (16)

The objective function (24) of Process 4 is also to minimize the system matching and dispatching cost similar to Equation (11) and (19). Equation (25) and (26) ensure that for each trip, only idling or repositioning taxis in the adjacent zones of which picking up time is less than 10 minutes is eligible for matching. Process 4 also follows the same range and vehicle type requirements as of Process 2's. Equation (28) shows that only trip not being fulfilled by Processes 2 and 3 is considered. The framework also updates the trip status vector s_m^{tr} as shown in Equation (29). After completing the dispatching Process (1), (2), (3), and (4), the number of cancelled trips can be calculated as follows:

$$f(w_m) = \begin{cases} 1; if \ w_m \ge \Delta_5\\ 0 \ otherwise \end{cases}$$
(29)

$$n_{cancelled} = \sum_{m} (1 - s_m^{tr}) f(w_m)$$
(30)

Where Δ_5 is the threshold for customer wait time. Equation (28) is a step function stating that if a customer wait time exceeds Δ_5 , the customer will cancel the trip. Equation (31) is calculating the number of cancelled trips by checking the demand status and whether the wait time exceeds Δ_5 .

Process 5: Repositioning Between Zones and Recharging Taxi

State: After taxi dispatching and matching is completed, the controller uses Process 4 to reposition vehicles between zones. We define the state space as follow:

$$S \coloneqq \{(t, s_t), t \in [1: T_{max}]\}$$

The set of state *S* consists of individual states s_t at each timestep *t*. At each state s_t , the Process records the following elements: unserved demand, idling vehicles, available future vehicles, future historical demand, and dispatching results. All of these elements will help Process 4 makes a better decision on how many taxis to dispatch between zones. The individual state is defined as:

$$s_t = (\boldsymbol{L}, \boldsymbol{N}^{idling}, \boldsymbol{F}, \boldsymbol{H}, \boldsymbol{U})$$

Where:

L: Unserved demand matrix where the rows represent taxi zones, columns represent waiting time discretized by 5-minutes bin, and each element values show how many unserved demands at each zone by waiting time.

N^{*idling*}: A vector representing the number of idling vehicles at each zone

- *F*: Future available vehicle matrix where the rows represent taxi zones, columns represent timestep in the future, and each element shows how many vehicles would be available at each zone and time bin. The future available vehicle is calculated based on the itinerary of occupied vehicles only.
- H: Historical demand matrix where the rows represent taxi zones, columns represent 5-minutes time bin in the future, and each element shows how many requests are expected at each zone and time bin. This is calculated based on the historical average.
- *U*: Undercharge vector representing the number of AET at each zone of which state of charge is less than a certain threshold

Action: The RL model's immediate action is the repositioning and recharging matrix X^{rep} of AET where the rows correspond to the start zone, columns represent the end zone plus one additional column dedicated to charging and the value of each element ranges from 0 to 1 as shown in Equation (31). Then the number of repositioning rep_{ij} and charging ch_j can be calculated via Equation (32) and (33):

$$a(s_t) = X^{rep} = \{ [x_{ij}^{rep}, x_j^{ch}] \mid i, j \in \mathbb{Z}; \ x_{ij}^{rep}, x_j^{ch} \in [0:1] \}$$
(31)

$$rep_{ij} = round \left\{ \frac{x_{ij}^{rep}}{\sum_{j} (x_{ij}^{rep} + x_{j}^{ch})} n_{j}^{idling} \right\}, \quad \forall i, j \in \mathbb{Z}$$

$$(32)$$

$$ch_j = n_j^{idling} - \sum_{i \in \mathbb{Z}} x_{ij}^{rep}, \forall j \in \mathbb{Z}$$
⁽³³⁾

Equation (32) and (33) are interpretable as the percentages of idling AET vehicles. The framework will randomly select rep_{ij} idling AETs to dispatch from zone *i* to *j* and the

destination node in zone *j* is randomly selected. In terms of charging, the framework keeps track of AET's remaining range rg_v as well as the closest charging station and the minimum distance $Dmin_v$ to that station. The state of charge soc_v can be calculated as the ratio between remaining range and the minimum distance as shown in Equation (33).

$$soc_v = \frac{rg_v}{Dmin_v}, \forall j \in Z$$
 (34)

If $soc_v < 1$, the AET vehicle does not have enough range to reach the nearest charging station and it will be defunct for the rest of operating day. This would incur a penalty in the reward function described later. The framework will select ch_j AET vehicles in ascending order of soc_v to go charging at theirs nearest charging station respectively. On the vehicle battery capacity, charge-depleting mode, and charging station power specifications, we present the detail numbers in Section 4 of Case Study.

Reward: In our framework, the controller aims to minimize the system cost, which is a combination of wait time, cancellation, repositioning, and undercharge AET penalty. Therefore, we use a negative sign to inverse the cost into reward as shown in Equation (21). All terms in the reward function are in time unit and the weights β_1 , β , β , and β are used to scale and represent the importance of each term and in our experiment the following is used: { β_1 , β_2 , β_3 , β_4 } = {0.25, 0.35, 0.15, 0.25}. Equation (34) is the local reward and Process 4's objective is to maximize the total reward accumulated throughout all timesteps:

$$r(s_{t}, a_{t}) = -\left[\beta_{1} \sum_{m} (1 - s_{m}^{tr}) w_{m} + \beta_{2} \sum_{m:w_{m} > \Delta_{5}} (1 - s_{d}^{tr}) + \beta_{3} \sum_{ij} rep_{ij} t_{ij}^{zone} + \beta_{4} \sum_{\nu:(soc_{\nu} < 1 \& s_{\nu}^{\nueh} \neq 2)} (1)\right]$$
(35)

After all processes are completed as shown in the order of Figure 4.1, the AET vehicle is dispatched either in single pickup, carpooling, repositioning, or recharging and an itinerary is created accordingly. After that, AET's location, status, and zone are updated. Both leftover demand and unmatched HV vehicle's wait time is updated and transferred to the next timestep. The environment state of next timestep will be calculated based on the updated information.

4.3.4. Solution Algorithm

In the ILP of Processes 1, 2, and 3, a Branch and Bound Algorithm is used. This approach is commonly used in discrete and combinatorial optimization in real-world practice where the problem can be non-convex. In our case study, we experiment with an AET fleet of 400 vehicles and an average demand of 80 requests per 5-minutes and the algorithm reaches global optima in less than one second.

In the reinforcement learning of Process 5, since our state and action are continuous variable, the policy-based method is the most suitable approach. A policy means given a certain state, the controller takes a particular action among several options. Therefore, one can surmise a policy as a function π () determining the best action given a state or $a_t = \pi(s_t)$. Policy-based method aims to find the optimal policy $\pi(s_t, \theta^*)$, of which if the agent follow will result in the maximum expected cumulated reward at the end of an episode. The policy is parametrized by θ

and therefore the problem becomes finding the θ that will maximize the cumulative expected reward throughout all timesteps of the episode as shown in Equation (36):

$$Maximizing: J(\theta) = \sum_{t} E_{a_t \sim \pi_{\theta}(s_t, \theta)} r(s_t, a_t)$$
(36)

We choose a neural network for our policy function and the architecture of which is shown in Figure 4.4. The neural network takes the input of the state as represented by five matrices discussed earlier and its architecture is tailored based on the nature of the problem. First, the number of outgoing/dispatching vehicle from each zone should depend on number of idling and future vehicle only. Therefore, these two matrices are combined and connected to the outgoing vector. With the same analogy, the unserved demand, historical demand, and undercharge matrix are combined and connected to the incoming vector. Note that an additional $(Z+1)^{th}$ element is added for charging need. By performing matrix multiplication between the outgoing and incoming vector, we get the dispatching matrix and this approach has two main advantages. First, it effectively reduces the number of weight and bias needed. Second, it ensures the neural network is interpretable since the sum of outgoing vehicle from each zone is proportionate to the outoing vectore, which is related to the number of future and idling vehicle. The same feature also applies to incoming vehicle where it is related to unserved, historical, and charging demand.



Figure 4.4. Policy Function $a_t = \pi(s_t, \theta^*)$ After Process 4 is fully trained, we arrive at the optimal policy $\pi(s_t, \theta^*)$ which will show the optimal repositioning action $a_t = \pi(s_t, \theta^*)$ at any given state s_t . Policy-based method works well with continuous state and action variable, and it can avoid the curse of dimensionality in computing reward of state-action pair. Policy-based method will first let the agent play in the environment and generates several sample trajectories $\{(s_t, a_t), t \in T\}$ and observes the cumulative reward so that it can prioritize trajectories with higher reward and vice versa. From these sample trajectories, the gradient of Equation (36) can be approximated (Sutton and Barto, 2018) in Equation (37) and the parameter θ can be updated in Equation (38):

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1:N} \left[\left(\sum_{t} \nabla_{\theta} \log \pi_{\theta}(a_{t}|s_{t}) \right) \left(\sum_{t} r(s_{t}, a_{t}) \right) \right]$$

$$\theta \leftarrow \theta + \alpha \nabla J(\theta')$$
(38)

This is commonly known as the Vanilla Policy Gradient Methods (Sutton and Barto, 2018) but there is a main drawback in Equation (36) where the agent must complete the trajectory (play until the end of the episode) and repeat the process N times until the gradient $\nabla_{\theta} J(\theta)$ can be computed and parameter θ can be updated. This causes two problem which are (1) large variances in approximated gradients and (2) being computationally expensive in updating the parameter. In order to overcome these problems, we use the A3C approach as proposed by Mnih et al. (2016). This approach has three main features which are Asynchronous, Advantage, and Actor-critic. First, A3C introduces multiple local agents with their own copies of the environment to simulate. The local agent can then compute their gradient locally and report to a newly global agent asynchronously which then passes back the updated parameter to the local agent. Second, Advantage refers to the difference (or how good) an action a_t is compared to the average of other action in state s_t . Third, the architecture of each local agent follows an actorcritic framework. The actor is the policy function $\pi(s_t, \theta)$ with parameter θ to perform the action whereas the critic network, which also has the same neural network architecture as the actor network but parametrized with θ_{ν} judges the accuracy of the estimation. The pseudocode for our problem of Autonomous Taxi Dispatching and Repositioning with A3C is shown below:

Algo	Algorithm 2. Autonomous Taxi Dispatching and Repositioning with A3C				
1.	Initialize shared global parameters θ for policy network $\pi(a_t s_t, \theta)$, θ_v for the value function network (i.e.,				
	critic) $V(s_t, \theta_v)$, and episode counter $g \in [1: G_{max}]$				
2.	Assume local worker specific parameter θ' and θ_{v}'				
3.	Each local worker is assigned a set of episodes and it interacts with their own copies of the environment.				
	The union of all local worker's episode set is $[1: G_{max}]$				
4.	While $g < G_{max}$:				
5.	Assign global parameter to local worker: $\theta' = \theta$ and $\theta_{v}' = \theta_{v}$				
6.	For timestep $t = [1:T_{max}]$				
7.	Gather the 5-minutes bin taxi demand and vehicle state data				
8.	Perform matching and dispatching Process 1,2, and 3.				
9.	Take repositioning action using policy: $a_t \approx \pi(s_t, \theta')$ and from state s_t				
10.	Based on number of repositioning vehicles between zones a_t , randomly select taxi in each zone and				
	reposition to other zone until a_t is reached.				
11.	Update vehicle state (i.e., location, zone, status, and itinerary), demand waiting time, number of				
	cancelled trips, and transfer unserved demand to the next timestep's demand input.				
12.	Calculate the reward r based on waiting time, cancellation penalty, and repositioning cost, and				
	transition to the next state s_t'				
13.	Compute the actor loss:				
	$J(\theta') = \log \pi(a_t s_t, \theta') [r + \gamma V(s_t', \theta_v) - V(s_t, \theta_v)] + \beta H(\pi(s))$				
14.	Calculate actor gradient and asynchronously update global policy network:				
	$\theta \leftarrow \theta + \alpha \nabla J(\theta')$				
15.	Compute the loss of critic network:				
	$J(\theta_{\nu}') = r + \gamma V(s_t', \theta_{\nu}) - V(s_t, \theta_{\nu})$				
16.	Calculate critic gradient and asynchronously update global critic network.				
	$\theta_v \leftarrow \theta_v - \alpha \nabla J(\theta_v)$				
17.	End For Loop				
18.	End While				

4.3.5. Baseline Model

First, this paper framework for combining taxi demands, dispatching, repositioning, and recharging is named Reinforcement Learning-Based Operation for AET fleet (RL-AET). In addition to RL-AET, we develop a Baseline Model (BM) also for dispatching and repositioning as a benchmark to compare with RL-AET. The Baseline Model is developed by reviewing and combining the previous literature on this topic (Alonso-Mora 2017, Chen 2016, and Zhan 2016). The spatial and temporal setting is also the same as RL-AET's mentioned in Section 4.3.1. The assumptions for AET and HV in RL-AET is also true for BM. The BM model architecture is shown in Figure 4.5 as follows:



Figure 4.5. Baseline Model Architecture
BM first uses Process 1BM for combining taxi demand into taxi trips in the same manner as Process 1 in RL-AET. Process 2BM uses the greedy algorithm to match empty vehicle with trips within the same zone. The vehicle will search for the trip with minimum traveling pickup distance. In the case two or more vehicles share the same minimum trip, the process randomly selects one vehicle, and the remaining vehicle will need to go with the second-best option. The pseudocode for Process 2BM is as follows:

Alg	Algorithm 3: Process 2BM Dispatching					
Inp	Input: Set of Trips D; Set of Vehicles V					
1.	For vehicles v in V:					
2.	Select the trip <i>d</i> among set of trips within the same zone that minimize: $argmin(\tau_{v,d}, d \in D)$					
3.	If trip <i>d</i> is already selected:					
	Move to the second-best option. Repeat the if condition if necessary (i.e. second-best option is					
	selected too) until all options are exhausted.					
	If all options are exhausted and v is not matched:					
	Declare <i>v</i> as idling.					
4.	End For Loop					
Output:						
The unmatched set of trips $D_{unmatched}$						
Set of idling vehicles <i>V_{idling}</i>						
Set	of matching $M = \{(v_i, d_i), i \in M\}$					

After Process 2BM, BM first recharges any AET vehicle under the required minimum range (i.e., 30 miles). Then BM uses the remaining idling vehicle for repositioning through Process 3BM. This process is the simplified version of (Chen et 2016) in which the strategy is deterministic, and no optimization is involved. The repositioning algorithm is as follows.

Algorithm 4: Process 3BM Repositioning					
Input: Number	Input : Number of unmatched demands by zone $D_{unmatched} \in \mathbb{R}^{\mathbb{Z}}$; Set of available by zone $V_{idling} \in \mathbb{R}^{\mathbb{Z}}$				
1. Calculate	Calculate normalized unmatched demand by:				
	$d'_{i} = \frac{d_{i}}{\sum_{j \in D_{unmatched}} d_{j}}, \forall i \in Z; D_{norm} = [d'_{i}, i \in Z]$				
2. Calculate	the number of vehicles needed in each zone: $V_{needed} = D_{norm} * \left(\sum_{v \in} V_{idling} \right)$				
3. Calculate	the number of incoming vehicles in each zone: $V_{incoming} = V_{needed} - V_{idling}$;				
4. Reshape V	$V_{incoming} \in R^{[1 \times Z]}$; Reshape $V_{idling} \in R^{[Z \times 1]}$				
5. Calculate	the repositioning matrix as: $R = V_{idling} \times V_{incoming}$ (Matrix Multiplication)				
If sum (V_{in})	$_{ncoming}$) > sum (V_{idling}) :				
N	Normalize: $R = R / sum(V_{incoming})$				
Else:					
N	$Vormalize: R = R / sum(V_{idling})$				
Output:					
The repositioning matrix <i>R</i> .					

After the 3 processes, BM updates the vehicle and taxi trip itinerary and state in the same manner as RL-AET. The procedure is repeated for each timestep until the end of the episode.

4.4. CASE STUDY

4.4.1. Simulated Environment and Data

To demonstrate the applicability of the proposed framework, we use a dense urban network of Chicago metropolitan area. Chicago is home to a population of approximately 2.7 million and the roadway network spans across 240 square miles. There are 77 taxi zones which are represented by the polygon in Figure 4.6a. These taxi zones are the same traffic analysis zones defined by Chicago metropolitan agency for planning considering land use, socioeconomic data, and density of roadway network. For illustration of results, we aggregate these zones into nine Sides (i.e., neighborhoods) as shown in Figure 6a. In the traditional taxi system, drivers/dispatchers used these Sides as reference for spatial area to facilitate dispatching and repositioning. However, since these areas are quite large in size, it is difficult for drivers to immediately arrive at the desire location for passenger pickup. Therefore, the use of smaller taxi zones alleviates this problem and increases the accuracy and effectiveness of repositioning. The road network can be modeled as a graph where links represent road segments and nodes represent intersections. Chicago network consists of 7,393 links and 2,514 nodes as shown in Figure 4.6b. We assume charging station locations throughout the network as shown by the green dots in Figure 4.6b. In addition, we assume link travel time is available as an input as most cities can readily gather these data either from observed data, traffic surveys or by use of a combination of data and analytical methods (Ermagun and Levinson, 2018). The link travel time can vary temporally (both hourly and daily), and the trip travel time will reflect this. Uninterrupted traffic flow facilities such as Interstates 90, 55, 290, and 94 are represented in yellow. Our study considers a fleet size of 600 autonomous electric taxis. HV supply is drawn from a historical distribution.



(a) Chicago Taxi Zones and Sides



(b) Chicago Network Figure 4.6. Chicago Network

In regard to the electric specifications, we assume the battery capacity of 80kWh at a consumption rate of 3.75 miles per kWh to arrive at a 300-mile range. To simplify the formulation, we use miles as the metric to measure the remaining battery charge so if a vehicle starts at a 300-mile range and travels for 10 miles, the remaining battery charge is simply 290 miles. In terms of charging station power rate, we assume a 12 miles/min charging rate. These assumptions are in consistent with state-of-the-art battery technology and recent literature (Fuller, 2016).

In terms of taxi requests, the demand is simulated based on the real-world Chicago Taxi Data. The real-world data is not used here because this dataset is aggregated in both time and space which defeats the purpose of disaggregated analysis (i.e., consideration of individual trip, and taxi) presented in this paper. In this real dataset, only the pickup and drop-off census tracts are published whereas the exact location is not provided. The exact pickup and drop-off time are not available but instead these timestamps are aggregated into 15 minutes interval. Therefore, we simulated our taxi data based on the real data such that it preserves the relative difference in density in both space and time. Figure 4.7a and 4.7b shows the distribution of taxi demand in spatial and temporal dimensions, respectively.



(a) Taxi Origin Demand at Afternoon Peak Hour



(b) Demand Variation by Time of The Day and Day of The Week

Figure 4.7. Anaheim Simulated Demand

In Figure 4.7a, we can identify the variation in demand by taxi zone where the Northern part of the city has the most demand followed by the West and Central area. However, the Downtown area has less taxi demand as expected because of two main reasons. First, this map shows the pickup location and not drop-off so although there may be a number of customers going to the Downtown area, this pattern is not shown on the map. Second, the Downtown area has great concentration of access point to public transit. This area is known as "The Loop" where there are multiple Chicago Transit Authority, Subway Lines passing through such that it is more convenient for the passengers. The taxi demand also varies temporally, and we utilize the historical demand to construct the temporal variation by time of the day and day of the week as shown in Figure 4.7b. During the weekdays, there are two peaks in the morning and afternoon at 8:00 and 17:00 respectively and the afternoon peak is slightly higher. In the weekend, there's only one afternoon peak and the traffic are more stable. It is important to note that the demand varies by day of the week and since the supply of HV and AET remains relatively unchanged, the long-term reward will also vary by day of the week.

4.4.2. Training Performance

The Chicago Case Study is executed in a Dell Precision Tower which has an Intel Core i7-6700 CPU at 3.40 GHz (8 CPUs) and 16GB of RAM. The RL architecture is scripted in Python 3.9 whereas dispatching processes are solved via CPLEX. The script also uses the deep learning library PyTorch for training and enables parallel multi-core processing for the asynchronous algorithm. In our framework, an episode is equivalent to a full day of operation for the entire fleet. Since the demand varies greatly by day of the week as shown in Figure 4.7b, the episode cumulative reward at the end of the episode will also vary. Therefore, it is better to compare

episodes belonging to the same day of the week. We train the model for 182 episodes or 26 weeks and plot the episode cumulative reward in Figure 4.8. The x-axis is the training week, the y-axis is the reward, and each line represents day of the week. The episode cumulative reward is the accumulation of all local rewards at each timestep as described in Equation (34).

Overall, each day reward has an average increase by 15% by the end of the training. Sunday and Monday have significantly higher rewards since these days have much less demand. There are three reasons for this marginal increase in system reward. First, there is inadequate supply of taxi compared to the demand which results in a very high value for system cost and hence, negative rewards. In the later sections, we demonstrate that most taxi are serving to its capacity throughout the day and the fleet downtime is minimized. Second, the reward design includes weighted objective which scales up the value and thus, the improvement with respect to the total reward is marginal. However, with regards to the key performance metrics such as customer wait time and cancelled trips, we see significant improvement as demonstrated in Section 5. Third, the environment setting (number of AETs and demands) remains the same and any improvement in performance is achieved only by better repositioning. Due to the inadequate supply and Process 2, 3, and 4 using optimization, there is much less vehicle leftover for repositioning and smarter repositioning will have less impact on the reward compared to increasing fleet size. In addition, by using minimum matching and optimization in Process 2, 3, and 4, the untrained model already achieves a good enough reward and sets a high benchmark for the RL model to improve upon. This does not indicate the training is not useful where in later section, we demonstrate how the model is dispatching smarter by decreasing the number of repositioning and empty miles traveled and increasing the number of matchings simultaneously.



Figure 4.8. Total Reward by Episode

We report the training time of the model as follows. Since each day of the week has different demand, busy days such as Friday and Saturday will have longer training time. We summarize the value of average running time per episode (a day) and per timestep (5 minutes period) for each day of the week in Table 4.3. Since the model is trained on 182 episodes with an average of 1.3 hours running time per episode, the entire training would take 244 hours to complete. However, our A3C architecture enables multi-core training, and in our setting, we use

8 cores to run 8 episodes simultaneously and effectively reduces the training time to 30 hours. In real-world setting, practitioners only have to train the model for the first time in the backend and use the trained policy function as is in real time. In this case, for a timestep with 5-minute period, our model solves it (combining demand into trips, dispatching, and repositioning) in less than 30 seconds. Thus, the model is perfectly capable as an online model where optimal decisions canbe suggested in near real time.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Average Running Time per Episode (hrs)	1.0	1.3	1.4	1.5	1.5	1.6	1.1
Average Running Time per timestep (seconds)	20.8	28.7	30.5	31.6	33.1	33.3	23.4

 Table 4.2. Model Training Time

The main objective of our framework is to minimize customer wait time and number of cancelled trips accumulated at the end of the episode and these values are shown in Figure 9a. The line and columns show the total weekly waiting time and number of cancelled trips, respectively. The total waiting time decreases from approximately 59 to 48 million minutes while the number of cancelled trips is reduced from 109 to 88 thousand trips. These values steadily decrease which indicates the stability in the training process and over the course of the training week, the wait time and cancelled trips drops by 17% and 19%. Since the supply of taxi remains unchanged throughout the week and Process 1, 2, 3, and 4 for carpooling and dispatching are modeled as optimization and already optimized, this increase in performance can be mainly attributed to the repositioning process as the agent is constantly learning and updating the policy function parameter to yield better decision. This improvement can be attributed to (1) better understanding in spatial-temporal relationship between vehicle supply and demand, (2)

better timing in repositioning because taxi may take several timestamps to arrive at the destined zone, and (3) better routing of vehicle.



(c) Total Weekly Repositioning and Empty Miles Traveled Figure 4.9. Total Weekly Performance Metrics vs Dispatching and Repositioning

Another measurement of the framework's performance is the total weekly number of dispatching based on Process 2, 3, and 4. Figure 4.9b shows the value of these measurements across 26 weeks of training. The majority of the trips are matched via Process 2 which indicates the effectiveness of zonal dispatching in reducing computational time while maintaining high matching result. Process 2 matching increases from 80 to 100 thousand trips per week and since each trip on average serves 2.7 customers, the demand served is very high. Process 3 and 4 has less matching than Process 2 and the number is stable throughout the training. However, these processes are still an integral part of the model and are responsible for at least 20% of the matching. Overall, the framework is able to increase the number of dispatching by 20% between the start and end of the training.

Besides dispatching, Figure 4.9c shows the action in Process 5 as measured by total weekly repositioning and empty miles traveled. We can see that both metrics decrease considerably while the dispatching process gets more matching. This indicates the model is able to reposition more effectively by minimizing unnecessary repositioning that do not result in immediate dispatching. In addition, we make an observation that the number of vehicle repositioning remains stable after week 10 while the empty miles traveled continues to decrease. This indicates the model is limited in number of taxi available for repositioning due to the increase in matching resulted from smarter repositioning, and thus it needs to lower the system cost via other means such as routing AET to minimize empty vehicle travel distance.

4.5. MODEL PERFORMANCE AND COMPARISON

4.5.1. Performance Metrics Comparison

In this section, we present model performance of trained Reinforcement Learning for AET (RL-AET) framework and compare with the Baseline Model (BM) on 35 episodes of new simulated demand data for validation purposes. The performances of the BM and trained RL-AET framework is shown in Figure 4.10. First, we can notice a repeated pattern after 7 episodes in both wait time and cancelled trips for both the BM and RL-AET framework because of the demand variation between days of the week. Weekdays performance on average are about 32% worse than weekend with the highest cost commonly happens on Friday and lowest cost occurs on Sunday. For BM, as shown in Figure 10a, the daily wait time and cancelled trips during weekdays are very high at the average of 1.3 million minutes and 65 thousand cancelled trips per episode. Although BM has a repositioning process, it is based on deterministic approach and is not flexible to several external factors. Here, BM relies only on the current outstanding demands in each zone to determine the vector of incoming vehicle and it does not consider (1) future available vehicle based on current itinerary and (2) future demand based on historical average. While greedy dispatching guarantees each individual taxi achieves the minimum pickup distance, it does not guarantee system minimum quality of service. The algorithm only considers the TNC objective while customer metrics are not included, and these have an effect on wait time and cancelled trips.



(a) Baseline Model



⁽b) RL-AET

Figure 4.10. Comparison Between Baseline Model and Autonomous Taxi over 50 episodes

On the other hand, RL-AET's metrics are plotted on the same scale in Figure 10b to highlight the difference in performance. The wait time and cancelled trips are 790 thousand minutes and 19 thousand trips on average respectively for the RL-AET indicating significant improvement over the BM model. This is a decrease of nearly 40% and 70% for wait time and cancelled trips. Based on these numbers, we infer that RL-AET is prioritizing more in decreasing number of cancelled trips than wait time because trip cancellation is more detrimental to the TNC's quality of service and reputation. This prioritization is formulated in the reward function

in Equation 34 by the usage of weights $\{\beta_1, \beta_2, \beta_3, \beta_4\}$. In addition, we provide Table 4.3 showing the explicit details on wait time and cancelled trips variation by day of the week. There are more improvements from the RL-AET model during the weekend compared to the weekday. During weekday, demand is considerably higher than supply and there is little opportunity for improvement while on the weekend, with more balance in supply and demand, the RL-AET can plan ahead and utilize fully the AET fleet via smart carpooling and repositioning.

		Averag	ge Daily Wait	t Time	Average Daily Cancelled Trips			
		Baseline	RL-AET		Baseline	RL-AET		
	Day	Model	Model	%Improved	Model	Model	%Improved	
	Monday	783,686	1,324,562	41%	18,246	64,808	72%	
	Tuesday	878,214	1,404,631	37%	21,277	67,892	69%	
ſ	Wednesday	929,929	1,463,817	36%	22,177	70,513	69%	
ſ	Thursday	949,430	1,419,742	33%	23,451	74,119	68%	
ſ	Friday	982,775	1,482,832	34%	24,544	72,683	66%	
	Saturday	564,300	1,130,428	50%	13,482	54,846	75%	
ſ	Sunday	456,596	1,023,178	55%	9,866	48,551	80%	

Table 4.3. Comparison Between Manual Allocation and Autonomous Taxi

4.5.2. Optimal Carpooling Trips Generation

In this section, we provide the validation for Process 1 of combining demand into carpooling trips. Figure 11 provides the illustration for the result of Process 1. Figure 11a shows a random set of 12 demand cases from an episode during the morning peak hour and each demand path is color-coded to preserve uniqueness. As shown in the Figure 11, these demand origins and destinations are close to each other which brings about the opportunity for carpooling and this setting is very common for dense urban areas such as Chicago. Process 1 aims to bundle these demands into carpooling trip to have as few vehicles needed to serve all of these demands while

also minimizing detouring time. As listed in Equation (1) - (6) in Section 3.3, Process 1 also follows operational constraints namely (1) all demands are open to carpooling, (2) customers have same preference for type of vehicle, and (3) on-time performance. The result is shown in Figure 11b where these 12 demand cases can be served by 3 carpooling trips with 3 vehicles and the trip paths are color-coded to help distinguish between them. In addition to guarantee the demand chain satisfies all operational constraints, Process 1 also minimizes the summation of every demand's detouring time. Figure 11b confirms that close proximity origins and destinations being bundled together. In this example, demand is concentrated in 3 Chicago Sides which are Northwest, West, and Southwest and there are 3 trips serving each Sides. In addition, the carpooling trip's distance is also considerably shorter than if the demand within it are served individually. Figure 4.11c shows the macro perspective via the Taxi Request Shareability Graph. Edges in this graph, as denoted by the dotted blue line, represent the demand that potentially can be bundled together. Process 1's goal is to determine the minimum set of paths needed to cover all of the vertices/demand while satisfying there is no vertices belong to more than one path. A path can consist of only one vertex. This Optimal Path Cover result is shown in Figure 11c as the blue solid line. In this instance, there are 303 demands, and these are 19,186 potential demands pair that can be linked together. After executing the algorithm, we identify the minimum number of carpooling trips needed is 130 which yields on average 2.3 demands served per trip.



Figure 4.11. Illustration of Process 1 for Combining Demands into Trips

4.5.3. Repositioning

We investigate the repositioning matrix rep_{ij} at a particular episode (i.e., episode 5 or Thursday) to understand RL-AET's strategy for repositioning. In particular, we sum all repositioning vehicles within 30 minutes period prior to the morning peak hour of 08:00 and aggregate by Chicago Sides (refer to Figure 4.6b on this description) as shown in Figure 4.12. In the morning, since the demand prior to the peak hour is not as high, there are more available vehicles to reposition and the total number of incoming vehicles to this area is matching the peak hour demand. The North Side (3) has the highest demand at 53 followed by Far North Side (1) at 49 and West Side (5) at 47 and the total incoming vehicle from repositioning are 25, 52, and 31 respectively. We can see that RL-AET is not completely accurate where there is variation in demand and incoming vehicle. However, there variation is either very small or due to two reasons. First, the number of available vehicles is limited and second, this zone future's demand can step in later. The RL-AET framework also attempts to diversify the destination zone to avoid over flocking at a particular zone if not necessary. The majority of repositioning vehicles originates from the West Side (5). There are usually very few demands from here and this area is relatively large with more taxi supply. In addition, taxi completing a request at this zone can be relocated to other higher demand zones. In the repositioning matrix, if the origin zone and destination zone are the same, then the RL-AET model is simply directing those vehicles to remain in place and no repositioning cost is incurred to the system. There are still cases with considerable amount of over-repositioning such as in zone 8 and 9. The RL-AET usually compensates for it later on by reflecting on the immediate reward, previous action, the

transitioned state, and repositioning itinerary. The practical action is to decrease the number of incoming vehicles to this area in the future timestamp.

	Repositioning Matrix								
	1	2	3	4	5	6	7	8	9
1	7	0	1	0	0	4	0	1	0
2	0	0	0	0	0	0	0	0	0
3	7	1	1	0	2	1	6	1	4
4	0	0	0	0	0	0	0	0	0
5	29	22	22	6	27	15	22	0	3
6	0	0	0	0	0	0	0	0	0
7	9	1	1	0	2	2	3	0	5
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	5	0	2	0
	Total	Incon	ning V	ehicle	e				
	52	24	25	6	31	27	31	4	12
	Dema	and							
	49	26	53	23	47	26	29	2	6

Repositioning Matrix

Figure 4.12. Total Number of Dispatch 30 minutes before Morning Peak Hour

We investigate the pickup density for a typical day's operation between two frameworks. Figure 4.13 shows the pickup density by aggregating pickup activity of AET during a day into a hexagonal area and uses a color ramp from light yellow to dark purple for representing the total count of pickups in that area. Lighter color means the hexagon has less pickup activity and vice versa. With less down time, the RL-AET framework is able to serve more customer than the BM framework as shown by the darker color on Figure 4.13b compared to Figure 4.13a. Both models have similar spatial pattern for pickup location and the majority of pickup occurs in the North and West Side. In addition, RL-AET can also pickup demands on outer parts of the city especially in the airport area and Far Southeast Side thanks to smart repositioning. RL-AET also sees a more concentrated and discrete pickup density compared to BM, of which pickup distribution is more even across the cities. Since Figure 4.13 only shows AET activity only, it can be inferred that RL-AET model prioritizes certain road segments and areas for AET pickup since it can strategically and effectively use HV for the remaining areas. This works in conjunction with prioritized route for repositioning to yield less empty vehicle traveling distance. In summary, BM is able to pick up 4,576 trips on this particular day whereas in the same setting, RL-AET is able to pick up 13,895 trips which is an approximately three-fold increase.



Figure 4.13. Pickup density between two frameworks

Following pickup density, we explore major repositioning route to see how dispatching and repositioning strategies are working together. Figure 4.14 shows the major repositioning route used where the link color and size represent the number of times it being traversed by repositioning vehicle. First, we observe that road segments with high repositioning flow have the same spatial pattern with high density pickup area in Figure 4.13b. This confirms the benefit of smart repositioning by enabling the repositioned AET with an immediate pickup. The controller uses primarily highway to reposition because it can take on more vehicle because of higher capacity. The majority of repositioning routes use the Loop in Interstate 90, 55, and 290. There are a lot of connection between Sides 3, 4, and 5 since the demand density here is the highest. This conforms with Figure 4.12 where there is more dispatching action around these 3 Sides. The repositioning route is not evenly distributed throughout the network, and it follows an arterial approach so that it would not disturb the current city traffic too much. In this operating day, the entire fleet of 600 AETs traveled 2,455 miles in total with an average of 4 miles of repositioning per vehicle per day. The total empty vehicle traveling distance is approximately equal to one trip, but it enables the AET to match on average 6 trips per day more compared to the BM model.



Figure 4.14. Major Repositioning Route

4.6. Day in the Life of an Autonomous Electric Taxi

We investigate these frameworks further from a single taxi point of view. This can be done by examining a "Day in a Life" of one taxi or its itinerary as shown in Figure 4.15a and 4.15b for BM and RL-AET respectively. In each framework, the taxi's initial location at the start of the day is shown as a blue triangle. Any movement is recorded by the colored line where the blue, green, and red represent repositioning, single pickup, and carpooling, respectively. Anytime a road segment is overlapped several times; the latest taxi movement type will be visible. The customer pickup and drop-off locations are shown in black circle and square, respectively. Anytime the taxi stops for recharging, the location is denoted as a green square. The taxi finishes its day shift at the red triangle. In the BM framework, we see the vehicle initially starts at the southside at 6:00 and repositions to the central area where it picks up its first taxi trip at 6:45. The majority of trips have at least 2 customers or more which highlights the effectiveness of Process 1 which uses Minimum Weight Perfect Matching for bundling demands into carpooling trips. Repositioning is very limited, and throughout the day, the taxi activity remains around the Far North, Northwest, and West side and the area of services is limited. These sides have considerably higher demand than the others and since BM's repositioning relies on the immediate supply/demand imbalance score, it prioritizes these sides over others. This bias approach can have a negative impact on equity in pickup location (not equal opportunity) and affects the community interest.



(a) Baseline Model



(b) RL-AET Model Figure 4.15. A Typical Vehicle Trajectory in a Day

Figure 4.15b of the RL-AET framework shows a different story where the repositioning is now executed by the learned policy function. The vehicle initially starts at the south side and

picks up the first customer at 6:15. Throughout the day, the AET is able to pickup 32 trips and most of them have 2 or more customers. Due to the heavy activities and extended distance traveled, the vehicle stops around noon for 20 minutes of recharging. Compared to the BM model, RL-AET is more committed into repositioning where the AET is repositioned from the downtown area to area closed to the O' Hare airport via Interstate 90 and receives an immediate pickup after. Unlike some repositioning approaches (Chen et al., 2016; Fagnant and Kockelman, 2014), Process 5 allows for direct non-adjacent zone repositioning whenever necessary and have an immediate effect on the quality of service. The empty vehicle travel distance is accounted for and minimized as discussed in the reward function in Equation (34). The taxi covers more area than the Baseline Model and there is notable action in the South Sides. The pickup location is well spread and there is no preference in the taxi zone that the vehicle likes to operate in. Thus, RL-AET follows equity in serving area and provides satisfactory service to communities throughout the city.

We also examine the temporal strategy by plotting taxi status during the day. Figures 4.16a and 4.16b show the vehicle status for the BM and RL-AET framework, respectively. In the BM framework, the taxi is matched with adequate number of trips and most of these trips are carpooling. The average idling time in between dispatching is 13.4 minutes. We also notice an extended period from 10:15 to 13:55 where the taxi is idling. This can be either there is no demand request within the searching radius or there is a closer taxi fulfilling that demand instead. The taxi spends 15 minutes for repositioning, approximately 5.7 hours for idling, and 8 hours for fulfilling customer trip, which yields a usage percentage of 57%. At the end of the operation, this taxi can fulfill 20 trips. In contrast to the BM, the RL-AET framework has very little idling time in between with an average time between actions of 6.6 minutes. The vehicle utilizes its idling time to

reposition and despite the repositioning time being minimal (20 minutes for the entire day), the taxi can usually have an immediate pick up after the repositioning action. Due to the heavy activity, the vehicle also stops at 13:00 to recharge the battery. In this framework, the taxi spends 2.3 hours, 20 minutes, and 10.2 hours on idling, repositioning, and fulfilling trips respectively. This yields a higher usage percentage of 72% and thus, the taxi is able to complete 32 trips which is an 150% increase compared to the Baseline Model.





(a) Baseline Model

(b) RL-AET Model

Figure 4.16. Taxi Status Throughout the Day

In addition to providing the single taxi point of view, we look at the fleet usage throughout the day as shown in Figure 4.17a and 4.17b for BM and RL-AET, respectively. In the BM Model, on average, only 21% of the fleet are in the productive status of fulfilling customer. This is not due to lack of demand but because of the spatial-temporal imbalance between supply and demand and the inadequate/sub-optimal repositioning. The majority of the fleet is in idling mode and this trend continues to rise till the end of the day. However, single pickup is still significantly lower than carpooling trip and thus, the BM model can still fulfill a lot of demands. In the RL-AET model in Figure 4.17b, it is evident that the controller effectively utilizes the AET fleet with more than 70% of the fleet being in productive status of fulfilling customer trip. We make two interesting observations. First, the fleet productive status shares the same temporal pattern as the demand where there are two peaks in the morning and afternoon respectively. This indicates the RL model is able to predict and strategically prepare for peak hour via repositioning and constantly achieves a satisfactory level of service for every time of the day. Second, the fleet usage for repositioning is actually lower than that of the BM model. The RL is repositioning as few AET as possible (minimizing fleet usage and empty vehicle traveling distance and time) but it has significant impact on the fleet productivity level and quality of service to the customer.



Figure 4.17. Fleet Usage Throughout the Day

4.7. CONCLUSION

This paper proposed a framework for determining the optimal dispatching, repositioning strategy, and recharging for autonomous electric taxi in an environment where human driver taxi also exists, and autonomous taxi is confined within AV-enabled road only. There are three dispatching processes focusing on single pickup, carpooling, and adjacent zone pickup, and these processes are formulated as mixed integer linear programming. The dispatching processes aim to minimize both the pickup time and prioritize demand requests with higher waiting time. The repositioning and recharging process is developed as a reinforcement learning model where the

agent interacted with the simulated environment to define the optimal policy. An asynchronous policy-based solution algorithm is introduced to enhance computational time and solving capability, which makes it suitable with real-world networks. The framework is trained on a simulated taxi demand in the Anaheim network. The trained framework is then applied to a new 50 days simulated taxi demand and compared with the baseline Manual Allocation.

The framework in this research will benefit both ridesharing provider and the community interest since both parties' objectives are considered. The result shows the proposed framework can reduce the wait time and number of cancelled trips by up to 75% and 73% respectively compared to the baseline. The framework aims to preemptively reposition taxis to higher demand zones several timesteps before and thus, it can have more success in single pickup and carpooling. The temporal dimension shows better utilization of time since there is less idling time between fulfilling requests. However, the proposed framework can be improved in a few ways. The first drawback is all of the processes share the same timeline. In the Chicago case study, the frequency of execution is once per 5 minutes. For the repositioning process, this is ideal since it facilitates stability in training and the travel demand pattern does not fluctuate much within a period less than 5 minutes. However, this frequency is not the most efficient for the dispatching processes. Unmatched customer will be transferred to the next matching batch and experiences at least 5 minutes of waiting time, which can be excessive for some. Second, there are still some difficulties in applying RL policy function to reposition vehicles. The repositioning action is from a macro perspective where the controller only directs the total number of vehicles to reposition between zones. There are no details on which vehicle specifically to reposition. In our current algorithm, we choose vehicles in descending order of remaining battery range to reposition. However, compared to other approaches such as greedy

repositioning, our model is not myopic and makes long-term strategic action rather than maximizing current rewards. In addition, the RL policy function and reward are designed to cooperate closely with the dispatching processes to maximize the system long term rewards. These improvements can be investigated further in future research.

APPENDIX 4.A.

Sets

D	Set of Demand		
$G_{TRS} =$	Taxi Demand Shareability Graph		
(V_{TRS}, E_{TRS})			
$\{P_1, P_2, P_3, \dots P_m\}$	Set of Optimal Paths covering the Shareability Graph		
Т	Set of Timestep		
Parameters			

$\begin{bmatrix} pu_d, do_d, t_d^{dem} \end{bmatrix}$	A tuple representing a single demand d which has (1) pickup location, (2) drop-off
$[, c_d, \varphi_d]$	location, (3) pickup time, (4) preference for carpooling, and (5) preference for type
	of service (AET or HV)
$ au_{n_1 o n_2}$	Total path travel time from location n_1 to n_2
$cost_{e(d_1,d_2)}$	Cost of detouring by combining demand d_1 and d_2
$[pu_m, do_m, t_m^{trip}]$	A tuple representing a single trip m which has (1) first pickup location, (2) final
$\varphi_m, N(P_m), p_m, num_m, w_m$	drop-off location, (3) first pickup time, (4) preference for type of service (AET or
L J	HV), (5) Set of demands covered, (6) itinerary, (7) number of demands, and (8)
	total waiting time
$\alpha_1, \alpha_2, \alpha_3$	Weights used in Process 2, 3, and 4
$ ho_m$	Penalty for not completing trip m
rg_v	Range of the vehicle
$cst^{proc}_{v,m}$	Operational Constraints for Process 2, 3, and 4
S_m^{tr}	Status of trip <i>m</i>

$s_t = (\boldsymbol{L}, \boldsymbol{N}^{idling}),$	Environment state at timestep t. It includes 5 matrices representing the unserved
F , H , U)	demand, idling vehicles, Future available vehicle, Historical demand, Undercharge
	vector.
rep_{ij}	Number of vehicles to reposition between zone i and j
ch_j	Number of vehicles to recharge at zone <i>j</i>
$r(s_t, a_t)$	Reward of the agent after taking action a_t at state s_t
$\beta_1,\beta_2,\beta_3,\beta_4$	Weights used in the reward fucntion

Decision Variable

$x_{v,m}$	Decision variables of Process 2, 3, and 4
$a(s_t)$	Action based on environment state s_t

ACKNOWLEDGMENTS

This research has been partially supported by the Center for Transportation Innovations in Education and Research (C-TIER) at the University of Memphis. Any findings and opinions expressed in this paper are those of the authors and do not necessarily reflect the view of C-TIER.

REFERENCES

- Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand highcapacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. U.S.A. 114, 462–467. https://doi.org/10.1073/pnas.1611675114
- Chen, C., Yao, F., Mo, D., Zhu, J., Chen, X. (Michael), 2021. Spatial-temporal pricing for ridesourcing platform with reinforcement learning. Transportation Research Part C: Emerging Technologies 130, 103272. https://doi.org/10.1016/j.trc.2021.103272
- Chen, L., Mislove, A., Wilson, C., 2015. Peeking Beneath the Hood of Uber, in: Proceedings of the 2015 Internet Measurement Conference, IMC '15. Association for Computing Machinery, New York, NY, USA, pp. 495–508. https://doi.org/10.1145/2815675.2815681
- Chen, T.D., Kockelman, K.M., Hanna, J.P., 2016. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. Transportation Research Part A: Policy and Practice 94, 243–254. https://doi.org/10.1016/j.tra.2016.08.020
- Chen, X. (Michael), Chen, X., Zheng, H., Xiao, F., 2021. Efficient dispatching for on-demand ride services: Systematic optimization via Monte-Carlo tree search. Transportation Research Part C: Emerging Technologies 127, 103156. https://doi.org/10.1016/j.trc.2021.103156
- Fagnant, D.J., Kockelman, K.M., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transportation Research Part C: Emerging Technologies 40, 1–13. https://doi.org/10.1016/j.trc.2013.12.001

- Fuller, M., 2016. Wireless charging in California: Range, recharge, and vehicle electrification. Transportation Research Part C: Emerging Technologies 67, 343–356. https://doi.org/10.1016/j.trc.2016.02.013
- Harold Kuhn, 1955. The Hungarian method for the assignment problem. Naval research logistics quarterly 2(1-2), 83–97.
- Jiao, Y., Tang, X., Qin, Z. (Tony), Li, S., Zhang, F., Zhu, H., Ye, J., 2021. Real-world ridehailing vehicle repositioning using deep reinforcement learning. Transportation Research Part C: Emerging Technologies 130, 103289. https://doi.org/10.1016/j.trc.2021.103289
- Liang, E., Wen, K., Lam, W.H.K., Sumalee, A., Zhong, R., 2021. An Integrated Reinforcement Learning and Centralized Programming Approach for Online Taxi Dispatching. IEEE Transactions on Neural Networks and Learning Systems 1–1. https://doi.org/10.1109/TNNLS.2021.3060187
- Lin, K., Zhao, R., Xu, Z., Zhou, J., 2018. Efficient Large-Scale Fleet Management via Multi-Agent Deep Reinforcement Learning, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Presented at the KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, London United Kingdom, pp. 1774–1783. https://doi.org/10.1145/3219819.3219993
- Ma, T.-Y., Rasulkhani, S., Chow, J.Y.J., Klein, S., 2019. A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. Transportation Research Part E: Logistics and Transportation Review 128, 417–442. https://doi.org/10.1016/j.tre.2019.07.002

- Ma, Z., Koutsopoulos, H.N., 2022. Near-on-demand mobility. The benefits of user flexibility for ride-pooling services. Transportation Research Part C: Emerging Technologies 135, 103530. https://doi.org/10.1016/j.trc.2021.103530
- Mao, C., Liu, Y., Shen, Z.-J. (Max), 2020. Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. Transportation Research Part C: Emerging Technologies 115, 102626. https://doi.org/10.1016/j.trc.2020.102626
- Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T.P., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous Methods for Deep Reinforcement Learning. arXiv:1602.01783 [cs].
- Shi, J., Gao, Y., Wang, W., Yu, N., Ioannou, P.A., 2020. Operating Electric Vehicle Fleet for Ride-Hailing Services With Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems 21, 4822–4834. https://doi.org/10.1109/TITS.2019.2947408
- Shou, Z., Di, X., Ye, J., Zhu, H., Zhang, H., Hampshire, R., 2020. Optimal passenger-seeking policies on E-hailing platforms using Markov decision process and imitation learning. Transportation Research Part C: Emerging Technologies 111, 91–113. https://doi.org/10.1016/j.trc.2019.12.005
- Simonetto, A., Monteil, J., Gambella, C., 2019. Real-time city-scale ridesharing via linear assignment problems. Transportation Research Part C: Emerging Technologies 101, 208– 232. https://doi.org/10.1016/j.trc.2019.01.019
- Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: an introduction, Second edition. ed, Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts.
- Tang, X., Li, M., Lin, X., He, F., 2020. Online operations of automated electric taxi fleets: An advisor-student reinforcement learning framework. Transportation Research Part C: Emerging Technologies 121, 102844. https://doi.org/10.1016/j.trc.2020.102844
- Turan, B., Pedarsani, R., Alizadeh, M., 2020. Dynamic pricing and fleet management for electric autonomous mobility on demand systems. Transportation Research Part C: Emerging Technologies 121, 102829. https://doi.org/10.1016/j.trc.2020.102829
- Vazifeh, M.M., Santi, P., Resta, G., Strogatz, S.H., Ratti, C., 2018. Addressing the minimum fleet problem in on-demand urban mobility. Nature 557, 534–538. https://doi.org/10.1038/s41586-018-0095-1
- Yang, H., Qin, X., Ke, J., Ye, J., 2020. Optimizing matching time interval and matching radius in on-demand ride-sourcing markets. Transportation Research Part B: Methodological 131, 84–105. https://doi.org/10.1016/j.trb.2019.11.005
- Zhan, X., Qian, X., Ukkusuri, S.V., 2016. A Graph-Based Approach to Measuring the Efficiency of an Urban Taxi Service System. IEEE Trans. Intell. Transport. Syst. 17, 2479–2489. https://doi.org/10.1109/TITS.2016.2521862
- Zhou, M., Jin, J., Zhang, W., Qin, Z., Jiao, Y., Wang, C., Wu, G., Yu, Y., Ye, J., 2019. Multi-Agent Reinforcement Learning for Order-dispatching via Order-Vehicle Distribution Matching. Proceedings of the 28th ACM International Conference on Information and Knowledge Management 2645–2653. https://doi.org/10.1145/3357384.3357799

5. CONCLUSION

"A man is like a novel: until the very last page you don't know how it will end. Otherwise, it wouldn't be worth reading."

Yevgeny Zamyatin

In this dissertation, the author has successfully developed three data-driven methodologies to solve real-world transportation problems as first introduced in Chapter 1 of Introduction. The first research presents a modeling framework for optimally positioning dynamic wireless charging (DWC) infrastructure in a transportation network for Battery Electric Vehicles (BEVs). The paper uses a bi-level modeling framework to minimize both the Total System Travel Time (TSTT) and the Total System Net Energy Consumption (TSNEC). The results showed that the suggested DWC infrastructure investment is different for TSTT and TSNEC minimization, and the optimal DWC plan of the TSTT model can lower the total system travel time by 0.0055%, while the TSNEC model can lower the total system net energy consumption by 28%. The paper provides valuable insights for planners and policymakers in making informed decisions, and the model is applied to a real-world Montgomery County network from Maryland, USA, which requires a 100 million dollar expense in DWC to sufficiently recharge all BEVs within the network. Future research includes analysis of DWC network in a mixed environment of conventional vehicles and BEVs, estimation of power availability from neighborhood electric grids, and induced demand because of DWC implementation.

The second paper proposes a sequential three-step framework to solve the problem of Network Wide Dynamic Link Travel Speed using only the Taxi Trip Dataset. The framework includes a novel deep learning model that consists of two main components which are Traffic Graph Convolution (TGCN) for capturing spatial relation and TGCNIstm for temporal relation. The result suggests that the framework can estimate dynamic link level travel speed in dense urban areas with high accuracy, and the model is capable of obtaining network-wide travel speed for larger networks with up to 9,500 links. However, the three-step framework has several drawbacks such as low computational efficiency in the first step, the Yen's Algorithm generates an alternative path set that lacks diversity, and the model cannot accurately evaluate step 1 and 2. Future research can fully address these drawbacks by making taxi trips GPS traces available and incorporating the effect of signals in path travel time estimation. The paper provides valuable insights for urban planners in the ITS domain, especially those in developing economies without state-of-the-art infrastructure already in place.

Finally, the third research presents a framework for determining the optimal dispatching, repositioning, and recharging strategy for a mixed autonomous electric taxi fleet using reinforcement learning. The dispatching processes aim to minimize pickup time and prioritize demand requests with higher waiting time. The repositioning and recharging process is developed as a reinforcement learning model, which is trained on a simulated taxi demand in the Anaheim network. The trained framework is then applied to a new 50 days simulated taxi demand taxi demand and compared with the baseline Manual Allocation. The results show that the proposed framework can reduce the wait time and number of cancelled trips by up to 75% and 73% respectively compared to the baseline. The framework aims to preemptively reposition taxis to higher demand zones several timesteps before and thus, it can have more success in single

pickup and carpooling. However, there are still some difficulties in applying RL policy function to reposition vehicles. Further research can investigate improvements such as the frequency of execution and the vehicle-specific repositioning action. Overall, the framework in this research will benefit both ridesharing provider and the community interest since both parties' objectives are considered, and it can lead to more efficient and effective use of the autonomous electric taxi fleet.