

ABSTRACT

Title of dissertation: DATA-DRIVEN OPTIMIZATION AND
STATISTICAL MODELING TO IMPROVE
DECISION MAKING IN LOGISTICS

Debdatta Sinha Roy
Doctor of Philosophy, 2019

Dissertation directed by: Professor Bruce Golden
Department of Decision, Operations
& Information Technologies
Robert H. Smith School of Business

In this dissertation, we develop data-driven optimization and statistical modeling techniques to produce practically applicable and implementable solutions to real-world logistics problems.

First, we address a significant and practical problem encountered by utility companies. These companies collect usage data from meters on a regular basis. Each meter has a signal transmitter that is automatically read by a receiver within a specified distance using radio-frequency identification (RFID) technology. The RFID signals are discontinuous, and each meter differs with respect to the specified distance. These factors could lead to missed reads. We use data analytics, optimization, and Bayesian statistics to address the uncertainty.

Second, we focus on an important problem experienced by delivery and service companies. These companies send out vehicles to deliver customer products and provide services. For the capacitated vehicle routing problem, we show that reducing

route-length variability while generating the routes is an important consideration to minimize the total operating and delivery costs for a company when met with random traffic.

Third, we address a real-time decision-making problem experienced in practice. For example, routing companies participating in competitive bidding might need to respond to a large number of requests regarding route costs in a very short amount of time. Also, during post-disaster aerial surveillance planning or using drones to deliver emergency medical supplies, route-length estimation would quickly need to assess whether the duration to cover a region of interest would exceed the drone battery life. For the close enough traveling salesman problem, we estimate the route length using information about the instances.

Fourth, we address a practical problem encountered by local governments. These organizations carry out road inspections to decide which street segments to repair by recording videos using a camera mounted on a vehicle. The vehicle taking the videos needs to proceed straight or take a left turn to cover an intersection fully. Right turns and U-turns do not capture an intersection fully. We introduce the intersection inspection rural postman problem, a new variant of the rural postman problem involving turns. We develop two integer programming formulations and three heuristics to generate least-cost vehicle routes.

DATA-DRIVEN OPTIMIZATION AND
STATISTICAL MODELING TO IMPROVE
DECISION MAKING IN LOGISTICS

by

Debdatta Sinha Roy

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Bruce Golden, Chair
Professor Edward Wasil
Professor Michael Ball
Professor Tunay Tunca
Professor Paul Schonfeld (*Dean's Representative*)

© Copyright by
Debdatta Sinha Roy
2019

Dedication

to mom and dad...

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Bruce Golden. To say the least, this journey of five years through many successes and failures would not have been possible without his constant motivation and support. I felt a sense of freedom while working under his guidance because he allowed me to freely explore and find solutions to research problems at my own pace. I always tried to make the most of this opportunity to increase my knowledge about the field and to approach a problem from my perspective because I knew that he would always give me a second chance if I failed. I believe this sense of freedom to think brings out the best in a student. He always pushed me for that extra bit of thinking on a research problem that I would not have done on my own. More generally, I am a fan of his sense of humor and his perspective on life. He is a very caring advisor and I have seen him care for the well-being of his students even long after their graduation. It has been an honor to work with him and to become a part of his great academic lineage. I will surely miss the very late night (extremely late for “normal” people) discussions with him over the phone and the Saturday “afternoon” (depends on the perspective) meetings at his home.

I also express my heartfelt gratitude to Professor Edward Wasil for practically being my co-advisor throughout my Ph.D. journey. He spent countless hours to help me improve my academic writing and to help me grow as a scholar. I am an admirer of his straightforward personality and his useful and practical advice on research problems and life. He was extremely patient and helped to channelize my

thoughts. My journey would be a lot different and tougher without his presence and involvement.

I would like to thank Professor Michael Ball and Professor Tunay Tunca for giving me feedback on my research progress from time to time and helping me in my academic journey in various ways. It is also an honor to have them in my dissertation committee. I would also like to thank Professor Michael Fu and Professor Frank Alt for always being there for me and helping me with all kinds of resources and suggestions. Moreover, I am extremely thankful to Professor Paul Schonfeld for being an integral part of my dissertation committee.

I am grateful to all my professors at IISER Mohali and ISI Delhi for grooming me well to take up a challenging PhD journey.

Life in 3330 Van Munching Hall would be a lot different and challenging without Justina Blanco. She is an excellent multi-tasker and keeps the ball rolling in the Smith School Ph.D. office. She forms a personal bond with each student, and I believe that she knows the answers to every question a doctoral student might have.

My Ph.D. experience would not have been as much enjoyable and fulfilling without Janet Cavanagh who has been like a mother to me. She was always there to help me whenever I needed it. She made me feel at home with her care and by inviting me every year to her place with her family during Thanksgiving celebrations.

I am very thankful to my co-authors, Dr. Christof Defryn and Adriano Masone for working with me on some exciting research problems and for making my Ph.D. journey significantly more productive.

I would like to thank Dr. Rui Zhang, Dr. Xingyin Wang, Dr. Oliver Lum, and

all other senior students when I started my Ph.D. journey for their invaluable advice to cope with this challenging journey full of high expectations. I am very grateful to have Dr. Stefan Poikonen and Dr. Cheng Jie as friends for their many hours of discussions on research problems, homework assignments, and ways to understand and get used to the American life.

I believe that after five years of close friendship with Dr. Aishwarya Deep Shukla and Dr. Gokul Iyer, I have a broader perspective about life and people. A special thanks to both of these brilliant people, in their own domains, for passing on to me, their knowledge and experiences. I would always cherish our discussions on science, technology, business, and academic writing.

All work and no play makes one a dull boy. I am very thankful to Anirudh Singh Chauhan, Kunal Dey, Dr. Siddharth Sharma, and Mohit Gupta for their enjoyable company which helped me cool off during times of stress.

A very special thanks to my best friend, Anjali (well, Dr. Gupta now), for always supporting me, guiding me to choose the right path, and helping me reach where I am today. My accomplishments are negligible with her out of the equation.

Last but not least, I am deeply indebted to my parents for all their sacrifices to help me excel. They have always provided unconditional love, trust, encouragement, and support. They have inspired me to keep performing better than yesterday and continuously motivated me to achieve my goal. They have always tried to keep me protected from all sorts of family responsibilities to help me focus on my career. I would not be able to stand here without their strong shoulders and the rock-solid platform they have provided.

Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	vi
List of Tables	ix
List of Figures	xiii
1 Introduction	1
2 Data-Driven Optimization and Statistical Modeling to Improve Meter Reading for Utility Companies	7
2.1 Introduction	7
2.1.1 Background and Literature Review	7
2.1.2 Research Goal and Contributions	12
2.2 Description of the Data Set	14
2.3 Initial Data Analysis	17
2.4 Integer Programming Formulation	21
2.4.1 Stage 1 IP Formulation	22
2.4.2 Stage 2 IP Formulation	24
2.5 Jensen’s Inequality for the Stage 1 IP	26
2.6 Regression	28
2.7 Regression Results	31
2.8 Bayesian Updating	38
2.8.1 Logit Model and Probit Model	41
2.8.2 Metropolis-Hastings Random Walk Algorithm for the Logit Model	42
2.8.3 Gibbs Sampling Algorithm for the Probit Model	43
2.8.4 Hierarchical Probit Model	43
2.8.5 Gibbs Sampling Algorithm for the Hierarchical Probit Model	45
2.9 Bayesian Updating Results	46
2.9.1 Logit Model and Probit Model Results	46
2.9.2 Hierarchical Probit Model Results	50
2.10 Description of the CNG Data Set	52

2.11	Bayesian Updating Results for the CNG Data Set	54
2.11.1	Logit Model and Probit Model Results	54
2.11.2	Hierarchical Probit Model Results	57
2.12	Discussion of Different Formulations and Computational Experiments of the Stage 1 IP	59
2.12.1	Alternative Formulations	59
2.12.1.1	Coefficient Round-Down Inequalities	60
2.12.1.2	Increasing Coefficient Extended Cover Inequalities	60
2.12.1.3	Decreasing Coefficient Extended Cover Inequalities	61
2.12.1.4	Middle Coefficient Extended Cover Inequalities	61
2.12.1.5	Extreme Coefficient Extended Cover Inequalities	62
2.12.1.6	All Inequalities	62
2.12.2	Computational Results	63
2.12.3	Observations from the Computational Experiments	74
2.12.4	Other Insights from the Computational Experiments	75
2.13	Heuristics for the Stage 2 IP	80
2.13.1	Route Generator	80
2.13.2	Route Trimmer	82
2.14	Simulation Experiments	86
2.14.1	Actual Reading Probabilities	87
2.14.2	Simulation Model Overview	88
2.14.3	Generating the Network	90
2.14.4	Simulation Results	91
2.15	Conclusions and Future Directions	99
3	Data-Driven Analysis of the Variability of Routes in the Capacitated Vehicle Routing Problem	101
3.1	Introduction	101
3.2	Capacitated Vehicle Routing Problem Instances	103
3.3	Importance of Standard Deviation of Routes	105
3.4	Effect of Reducing Standard Deviation on Cost	111
3.5	Contribution of Standard Deviation to Cost	120
3.6	Conclusions and Future Directions	122
4	Data-Driven Estimation of the Route Length for the Close-Enough Traveling Salesman Problem	124
4.1	Introduction	124
4.2	The Regression Model	127
4.3	Regression Data and Model Fit Measures	129
4.4	Regression Results	131
4.4.1	Results on all 842 Instances	131
4.4.2	Results on the Second Group of 62 Instances	135
4.4.3	Results on the First Group of 780 Instances	141
4.4.4	Cross-validation for the First Group of 780 Instances	143
4.4.5	Model Selection for the First Group of 780 Instances	152

4.5	Conclusions and Future Directions	162
5	Intersection Inspection Rural Postman Problem on a Mixed Graph	164
5.1	Introduction	164
5.2	Problem Formulations on a Mixed Graph	170
5.2.1	IIRPP Formulation using Node Transformations	172
5.2.2	IIRPP Formulation using Path Transformations	176
5.2.3	Heuristics	180
5.3	Computational Experiments	182
5.3.1	Test Instances	182
5.3.2	Computational Results	184
5.3.3	Summary of Results	192
5.4	Conclusions and Future Directions	195
6	Concluding Remarks	214
	Bibliography	218

List of Tables

2.1	Results based on the analysis from Step 1 to Step 5.	18
2.2	Results based on the analysis from Step 9.	21
2.3	Summary statistics for the dependent and the independent variables.	32
2.4	Correlation between the dependent and the independent variables.	33
2.5	Logistic regression results.	34
2.6	Probit regression results.	36
2.7	Logit model results.	48
2.8	Probit model results.	48
2.9	Hierarchical probit model results for the higher level parameter matrix.	51
2.10	Summary of the CNG data set.	52
2.11	Logit model results for the CNG data set.	54
2.12	Probit model results for the CNG data set.	56
2.13	Hierarchical probit model results for the higher level parameter matrix for the CNG data set.	58
2.14	Results for 20 meters and 10 street segments.	65
2.15	Results for 10 meters and 20 street segments.	66
2.16	Results for 100 meters and 20 street segments.	67
2.17	Results for 20 meters and 100 street segments.	68
2.18	Results for 200 meters and 100 street segments.	69
2.19	Results for 100 meters and 200 street segments.	70
2.20	Results for 1000 meters and 200 street segments.	71
2.21	Results for 200 meters and 1000 street segments.	72
2.22	Results for 2000 meters and 1000 street segments.	73
2.23	Results for 1000 meters and 2000 street segments.	74
2.24	Comparison of the linear Stage 1 IP performance for the three different cost structures.	76
2.25	Comparison of the linear Stage 1 IP objective value for specified likelihood values of 0.95 and 0.75 for all meters.	78
2.26	Local search operators in the variable neighborhood descent metaheuristic.	81
2.27	Algorithm for the remove and repair procedure of the route trimmer.	83

2.28	Average comparison of the total time to read all the meters.	96
3.1	Summary statistics of route times (in hours) for routes generated by two third-party software programs on an actual street network to serve customers.	102
3.2	Eleven CVRP instances (Christofides and Eilon 1969a).	103
3.3	Breakdown of 1000 CVRP solutions for each of the 11 instances in terms of the number of vehicles required to serve the customers.	106
3.4	Parameter values that are used to calculate the total operating and delivery costs for a company to serve its customers (Levy 2018).	107
3.5	Linear regression results for three models.	109
3.6	Average total cost under random traffic conditions for Scenario X.	117
3.7	Average total cost under random traffic conditions for Scenario Y.	118
3.8	Average total cost under random traffic conditions for Scenario Z.	118
3.9	Best average total cost under random traffic conditions across all three scenarios and percent savings compared to Group A.	119
3.10	Number of solutions from the respective buckets for the 20 best total cost solutions under random traffic conditions.	121
4.1	Definitions of the independent variables for the linear regression model.	128
4.2	Regression results for the 842 instances with and without outliers.	132
4.3	Regression results for the second group of 62 instances with and without outliers.	136
4.4	Regression results for the first group of 780 instances with outliers.	140
4.5	Regression results on each training set of 520 instances from the first group of 780 instances.	144
4.6	Best subset models based on R^2 for the first group of 780 instances.	152
4.7	Regression results on the best subset models based on R^2 for the first group of 780 instances.	155
5.1	Comparison of the transformed graphs G^1 and G^2 with the original graph G on 25 nodes.	197
5.2	Comparison of the transformed graphs G^1 and G^2 with the original graph G on 36 nodes.	198
5.3	Comparison of the transformed graphs G^1 and G^2 with the original graph G on 49 nodes.	199
5.4	Comparison of the transformed graphs G^1 and G^2 with the original graph G on 64 nodes.	200
5.5	Comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations on 25 nodes.	201
5.6	Comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations on 36 nodes.	201

5.7	Comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations on 49 nodes.	202
5.8	Comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations on 64 nodes.	202
5.9	Comparison of the running time for the RPP and IIRPP formulations on 25 nodes.	203
5.10	Comparison of the running time for the RPP and IIRPP formulations on 36 nodes.	203
5.11	Comparison of the running time for the RPP and IIRPP formulations on 49 nodes.	204
5.12	Comparison of the running time for the RPP and IIRPP formulations on 64 nodes.	204
5.13	Comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations on 25 nodes.	205
5.14	Comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations on 36 nodes.	205
5.15	Comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations on 49 nodes.	206
5.16	Comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations on 64 nodes.	206
5.17	Comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations on 25 nodes.	207
5.18	Comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations on 36 nodes.	208
5.19	Comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations on 49 nodes.	209
5.20	Comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations on 64 nodes.	210
5.21	Summary of the comparison of the transformed graphs G^1 and G^2 with the original graph G	211
5.22	Summary of the comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations.	212
5.23	Summary of the comparison of the running time for the RPP and IIRPP formulations.	212
5.24	Summary of the comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations.	212

5.25 Summary of the comparison of the RPP based and IIRPP based
heuristics with the IIRPP formulations. 213

List of Figures

2.1	(Color online) A view of the street layer, the service location layer, and the reading events layer. The red lines represent the street segments. The green dots represent the route traversed by the meter reading vehicle. The blue dots and the yellow dots represent meters (customers) in the service location layer that are read and that are missed, respectively, after the vehicle has traversed the route. If multiple meters have the same geographic location, then all those meters are represented by a single dot, although they have distinct account identifiers.	14
2.2	(Color online) A magnified view of the reading events layer.	15
2.3	Minimum time interval between reads.	18
2.4	Maximum read distance.	19
2.5	Histograms of fitted values from the logistic regressions.	35
2.6	Histograms of fitted values from the probit regressions.	37
2.7	Density plots from the MH random walk algorithm for the logit model.	49
2.8	Trace plots from the MH random walk algorithm for the logit model.	49
2.9	Autocorrelation plots from the MH random walk algorithm for the logit model.	50
2.10	Histograms of the means of the lower level parameters from the Gibbs sampling algorithm for the hierarchical probit model.	52
2.11	Density plots from the MH random walk algorithm for the logit model for the CNG data set.	55
2.12	Trace plots from the MH random walk algorithm for the logit model for the CNG data set.	55
2.13	Autocorrelation plots from the MH random walk algorithm for the logit model for the CNG data set.	56
2.14	Histograms of the means of the lower level parameters from the Gibbs sampling algorithm for the hierarchical probit model for the CNG data set.	57

2.15	(Color online) The route generator and the route trimmer applied to a small example. Red lines denote the required street segments. Blue lines denote the deadhead segments. Yellow line denotes the required street segment that is removed. Green line denotes the new street segment added to the route as a replacement for the yellow line. . . .	86
2.16	(Color online) A view of a portion of the actual street network with meter locations in the UTM format serviced by Connecticut Natural Gas in our data set from Hartford, Connecticut. The red dots represent the meters. This network is used for our simulation experiments.	90
2.17	(Color online) Simulation results for the route length.	93
2.18	(Color online) Simulation results for the number of missed meters.	94
3.1	Example to illustrate an iteration of the RTR travel algorithm explaining the three scenarios.	113
3.2	A flowchart showing the relation between the four groups and three scenarios.	115
4.1	An example of a CETSP with 12 customers.	125
4.2	Node locations of instance d493 from the second group of 62 instances.	129
4.3	Studentized residual plot of 842 instances. The lines indicate the Studentized residual values of 2 and -2	133
4.4	Histogram of Studentized residuals of 842 instances.	133
4.5	Normal probability plot of Studentized residuals of 842 instances.	134
4.6	Studentized residual plot for the second group of 62 instances. The lines indicate the Studentized residual values of 2 and -2	137
4.7	Histogram of Studentized residuals for the second group of 62 instances.	137
4.8	Normal probability plot of Studentized residuals for the second group of 62 instances.	138
4.9	Studentized residual plot for the first group of 780 instances. The lines indicate the Studentized residual values of 2 and -2	139
4.10	Histogram of Studentized residuals for the first group of 780 instances.	140
4.11	Normal probability plot of Studentized residuals for the first group of 780 instances.	141
4.12	Studentized residual plot for the first set of 520 training instances. The lines indicate the Studentized residual values of 2 and -2	143
4.13	Histogram of Studentized residuals for the first set of 520 training instances.	145
4.14	Normal probability plot of Studentized residuals for the first set of 520 training instances.	145
4.15	Studentized residual plot for the second set of 520 training instances. The lines indicate the Studentized residual values of 2 and -2	147
4.16	Histogram of Studentized residuals for the second set of 520 training instances.	147

4.17	Normal probability plot of Studentized residuals for the second set of 520 training instances.	148
4.18	Studentized residual plot for the third set of 520 training instances. The lines indicate the Studentized residual values of 2 and -2	149
4.19	Histogram of Studentized residuals for the third set of 520 training instances.	149
4.20	Normal probability plot of Studentized residuals for the third set of 520 training instances.	150
4.21	Plot showing the best subset models for the first group of 780 instances arranged according to adjusted R^2	153
4.22	Plot showing the best subset models for the first group of 780 instances arranged according to Mallows's C_p	153
4.23	Plot showing the best subset models for the first group of 780 instances arranged according to BIC.	154
4.24	Studentized residual plot of 780 instances for the best adjusted R^2 model. The lines indicate the Studentized residual values of 2 and -2	154
4.25	Histogram of Studentized residuals of 780 instances for the best adjusted R^2 model.	156
4.26	Normal probability plot of Studentized residuals of 780 instances for the best adjusted R^2 model.	156
4.27	Studentized residual plot of 780 instances for the best Mallows's C_p model. The lines indicate the Studentized residual values of 2 and -2	157
4.28	Histogram of Studentized residuals of 780 instances for the best Mallows's C_p model.	158
4.29	Normal probability plot of Studentized residuals of 780 instances for the best Mallows's C_p model.	158
4.30	Studentized residual plot of 780 instances for the best BIC model. The lines indicate the Studentized residual values of 2 and -2	160
4.31	Histogram of Studentized residuals of 780 instances for the best BIC model.	160
4.32	Normal probability plot of Studentized residuals of 780 instances for the best BIC model.	161
5.1	(Color online) An intersection with two left turns.	165
5.2	(Color online) An intersection with two right turns.	166
5.3	(Color online) Map of Dupont Circle in Washington, DC.	167
5.4	(Color online) The RPP and the IIRPP solutions on a small grid-like street network.	168
5.5	A mixed graph G (street network) with four nodes. Costs are shown adjacent to the two arcs and one edge.	170
5.6	Transformed graph G^1 from the original graph G shown in Figure 5.5.	173
5.7	Transformed graph G^2 from the original graph G shown in Figure 5.5.	177
5.8	(Color online) RPP-H route on a small grid-like street network.	180
5.9	(Color online) IIRPP-F1-H and IIRPP-F2-H produce the same route on a small grid-like street network.	181

5.10 Example of an 8×8 instance. 183

Chapter 1: Introduction

Business and industry decision making has entered the era of big data. The growing importance of data in decision making has led to better data storage and easy access of data to decision makers. However, they often lack the resources and analytical tools required to understand what type of data they need precisely, how to use the available data, and how to incorporate the vast amount of data from multiple sources including departments inside the organization and clients outside the organization. In this dissertation, we study real-world decision-making problems in logistics. Throughout the progress of this dissertation, we worked in close collaboration with companies to identify the crucial challenges that could be addressed using data analytics, statistics, and optimization. We obtained real-world data, used discussions with companies and their clients, and addressed company requirements. These helped us to formulate our research questions and develop models and solution methods. Since the problems considered in this dissertation are real-world in nature, it is important to model them as data-driven, decision-making problems. We develop data-driven optimization and statistical modeling techniques that can synthesize and analyze different types of data to produce practically applicable and implementable solutions to real-world logistics problems and

to improve decision making for both clients and companies. Even though the collaboration process with different companies for the problems studied in this dissertation was a fulfilling experience, obtaining data and practical insights can be tedious and time-consuming. However, the importance of narrowing the gap between academic research and practical applicability of the solutions is enough of a motivation to strive harder for industry-driven collaborative research.

In Chapter 2, we address a significant and practical problem encountered by utility companies. These companies collect usage data from meters on a regular basis. The usage data are collected automatically using radio-frequency identification (RFID) technology. Each meter transmits signals from an RFID tag that are read by a vehicle-mounted reading device within a specified distance. Currently, utility companies generate meter reading routes by solving the close enough vehicle routing problem (CEVRP) such that all meters are within a specified distance (range of the signal as specified by the manufacturers of the RFID devices) from at least one location on the route. In reality, there is uncertainty while reading meters. The signal transmitted by an RFID tag is discontinuous. The range that each meter can be read is different and stochastic due to weather conditions, surrounding obstacles, interference, and decreasing battery life of the RFID tags. These factors could lead to meters not being read.

Utility companies typically read more than 1.5 million RFID meters on a monthly basis in a mid-sized city in the United States. Around 5-10% of those meters are missed from the planned routes of the meter reading vehicles. Generally, a vehicle is sent at a later time to read the missed meters, resulting in increased costs

for utility companies due to additional operational costs and overtime payments to drivers.

We address the uncertainty issues of the RFID technology using data from a public utility company. We use data analytics, optimization, and Bayesian statistical models to generate routes that are both cost-effective and robust (the number of missed reads is minimized). The stochastic meter reading problem is formulated as a deterministic two-stage integer program (IP). An iterative algorithmic framework addresses the stochasticity. The algorithm starts by learning from the incoming data every time the meter reading vehicle collects readings. A two-stage IP is solved with the updated probability of a meter being read successfully to generate routes that are more robust for addressing the uncertainty. We motivate the use of Bayesian statistical models to update the probabilities and also show that a hierarchical Bayesian statistical model is justified in our problem setup. Simulation experiments using an actual meter reading data set with five million observations are carried out on an actual street network. We show that the hierarchical Bayesian statistical model provides better results than other types of Bayesian statistical models. Utility companies may be able to integrate results from the hierarchical Bayesian statistical model into their route generating software as a decision-support tool to produce routes that are more cost-effective and robust than the routes that they generate currently.

In Chapter 3, we focus on an important problem experienced by delivery and service companies. These companies need to send out multiple vehicles to deliver customer products and provide services to customers in a city every day. Each

vehicle has a capacity constraint and each customer has a demand. A company needs to fulfill all customer demands in a city using a fleet of vehicles. The companies generate vehicle routes using routing software and algorithms provided by third-party vendors. It is essential to maintain a well-defined workload balance among the drivers of a company, i.e., the drivers should have similar route times (inclusive of the service times). However, algorithms found in routing software tend to focus on minimizing the total route time or average route time for the fleet of vehicles serving a city. Large differences in workloads (large route-length variability) are not only unfair to the drivers, but they might also lead to increased operating and delivery costs for these companies.

For the capacitated vehicle routing problem (CVRP), we use regression models to show that reducing route-length variability while generating the routes is an important consideration to minimize the total operating and delivery costs for a company when met with random traffic. We implement fast and easy modifications to some well-known routing algorithms to reduce the standard deviation of the routes, thereby reducing the total operating and delivery costs.

In Chapter 4, we address a real-time decision-making problem experienced in practice that involves the close enough traveling salesman problem (CETSP). In the CETSP, every customer has a service region and is considered visited when the salesman visits any point in the customer's service region. The objective of the CETSP is to visit all customers in the shortest distance traveled. In one application of real-time decision-making, routing companies participating in competitive bidding might need to respond to a large number of requests regarding route costs in a very

short amount of time. In such cases, it may be sufficient to estimate the route lengths using information about the actual instances. In another application, during post-disaster aerial surveillance planning or using drones to deliver emergency medical supplies, route-length estimation would quickly need to assess whether the duration to cover a region of interest would exceed the drone battery life. The traveling salesman problem (TSP) is a special case of the CETSP, so the CETSP is at least as difficult to solve as the TSP.

For the CETSP, we estimate the route length using regression models. Route-length estimation approximates the route length generated by a specific algorithm and does not necessarily approximate the optimal solution. For practical purposes, routing companies need to know the actual costs that would be incurred using a specific algorithm and not the optimal costs even if the optimal costs are lower than the actual costs. The estimation model would be unique to the algorithm that was applied even though the general framework would apply to any routing algorithm.

In Chapter 5, we address a practical problem encountered by local governments. These organizations carry out road inspections to decide which street segments to repair by recording videos using a camera mounted on a vehicle. This process is similar to Google generating street view images. The vehicle taking the videos needs to proceed straight or take a left turn to cover an intersection fully. A right turn does not always capture an intersection fully and a U-turn does not cross an intersection.

We introduce the intersection inspection rural postman problem (IIRPP), a new variant of the rural postman problem (RPP) involving turns. The RPP is an

important arc routing problem. In an RPP, we need to find the shortest way of connecting a given set of required street segments to form a full route. The IIRPP is a hybrid of arc routing and node routing problems. When solving the IIRPP, we have to make sure there is at least one left turn or straight turn at an intersection that is required to be inspected. The RPP is a special case of the IIRPP when there are no intersections to be inspected, so the IIRPP is at least as difficult to solve as the RPP. We develop two integer programming formulations of the IIRPP based on two different graph transformations to generate least-cost vehicle routes. We also develop a heuristic based on the RPP, and two heuristics based on the two IIRPP formulations. We perform computational experiments to compare the performance of the formulations and the heuristics.

In Chapter 6, we present our concluding remarks and briefly summarize our contributions.

Chapter 2: Data-Driven Optimization and Statistical Modeling to Improve Meter Reading for Utility Companies

2.1 Introduction

2.1.1 Background and Literature Review

Utility companies read the electric, gas, and water meters of their residential and commercial customers on a regular basis. Typically, for a residential customer, a meter reader visits a customer and manually reads a meter at the site. Utility companies are interested in generating short and balanced routes where all streets with meters that have to be read are traversed.

In the late 1970s, one of the earliest efforts to solve the meter reading problem was by [Stern and Dror \(1979\)](#). Their algorithm was based on a route-first, cluster-second approach. An Euler cycle covered all required edges (streets) in the network. The cycle was then partitioned into balanced routes.

In the early 1990s, geographic data became available in the form of GBF/DIME (geographic base file/dual independent map encoding) files. This led to the development of optimization algorithms, graphics, and interactive features in meter-reader software systems. However, the incompatibility of street data and geographic data

led to modeling challenges. [Bodin and Levy \(1991\)](#) used an arc-partitioning algorithm to cluster street segments into balanced routes. Their computerized routing system produced much better routes than the routes generated by utility companies.

Utility companies have to read tens of thousands of meters every month. They need to generate highly efficient routes for meter readers. The task of generating efficient routes is complicated due to several factors including balancing workload among routes, meter-reading modes (walking, driving, combination of walking and driving), density of meters in a geographic region, amount of read time, and natural boundaries such as highways, rivers, and lakes. In the early 2000s, geographic information system (GIS) was combined with powerful (near-optimal) routing algorithms to form a highly visual computerized system. [Levy et al. \(2002\)](#) described how a GIS can address the complicating factors. For example, several layers of data such as water, major highways, and railroad tracks can be displayed in a service area by a GIS. The displayed layers can then be used to select a subset of meters to read within the service area for route planning purposes.

In the late 2000s, the use of radio-frequency identification (RFID) technology increased. RFID technology is used extensively in many industries for tracking resources since it holds down cost while increasing accuracy compared to traditional labor-intensive reading methods. During the decade, the accuracy of transmitters and receivers improved and the cost decreased gradually with the advancement of technology, making the RFID technology even more viable and useful. Automatic meter reading (AMR) using RFID technology was first tested in the early 1960s when trials were conducted by AT&T in cooperation with a group of utilities and

Westinghouse but it was not adopted for commercial use by utility companies until the late 2000s. [Eglese et al. \(2014\)](#) gave a brief summary of the meter reading problem from the late 1970s until the late 2000s.

An AMR system has two parts: an RFID tag and a vehicle-mounted reading device. An RFID tag is connected to a physical meter. The tag encodes the identification number of the meter and its current reading into a digital signal. The vehicle-mounted reading device collects the data automatically when it approaches an RFID tag within a specified distance. Utility companies would like to design the routes of the vehicles to cover all customers (meters) in the service area and minimize the total length of the routes or the total cost of the routes. The use of RFID technology in meter reading changes the routing problem from a standard vehicle routing problem (VRP) to a close-enough VRP (CEVRP). Substantial savings over traditional solutions are possible by developing routes that exploit this close-enough feature, i.e., the meter readers have to be within a specified distance from the meters to read them and not manually visit each one.

Most of the research on the close-enough problem is limited to Euclidean distances and uses a node routing formulation. [Dumitrescu and Mitchell \(2003\)](#) studied approximation algorithms for the close-enough traveling salesman problem (CETSP). [Gulczynski et al. \(2006\)](#) and [Dong et al. \(2007\)](#) presented clustering and convex hull heuristics for the CETSP in the context of meter reading. [Mennell \(2009\)](#) and [Behdani and Smith \(2014\)](#) formulated mixed integer programs for the CETSP. [Coutinho et al. \(2016\)](#) proposed an exact algorithm for the CETSP based on a branch-and-bound procedure and second order cone programming. [Groër et al.](#)

(2009) addressed the balanced billing cycle vehicle routing problem (BBCVRP) which occurs when, over time, routes become inefficient and fractured with imbalanced workloads for the meter readers. Their three-stage algorithm for solving the BBCVRP used partitioning heuristics and integer programming to reduce the length of the routes and to balance the workload.

Shuttleworth et al. (2008) were the first to model the CETSP with an arc routing formulation. They developed a two-stage process to solve the CETSP over a street network for a single meter-reader route. In the first stage, two heuristics (weighted bang for buck, distance weighted bang for buck) and two integer programs specify a subset of street segments that have to be traversed by a meter reader. All meters are within distance r from at least one location on at least one of the specified street segments. In the second stage, a travel path (cycle) is generated that traverses the specified street segments. Hà et al. (2014) proposed mathematical formulations and heuristics for the close-enough arc routing problem (CEARP). In the CEARP, traversed street segments only have to be within a specified distance from the points of interest. Ávila et al. (2016) proposed a new mathematical formulation for the CEARP and described its polyhedra. Renaud et al. (2017) considered a version of the CEARP in the context of meter reading in which the probability of reading a meter from a street segment decays exponentially as the distance from the meter to the street segment increases. They proposed an integer programming formulation and presented several heuristics.

There are issues with RFID technology that are not considered in the literature that we need to take into account. The signal transmitted by an RFID tag occurs

at regular time intervals that are not continuous. This is done to extend the battery life of the RFID tags. This leads to the possibility of a missed capture of a signal if the vehicle with the receiver is within the range of the meter only for a short time. Also, the signal range of a meter can vary from the distance specified by the manufacturers of the RFID devices due to weather conditions, surrounding obstacles, signal interference from other meters in the vicinity, and decreasing battery life of the RFID tags.

On average, utility companies read more than 1.5 million RFID meters on a monthly basis. It is observed that around 5-10% of those meters are missed from the planned routes of the meter reading vehicles. Currently, utility companies generate meter reading routes by solving the CEVRP such that all meters are within a specified distance (range of the signal as specified by the manufacturers of the RFID devices) from at least one location on the route. Utility companies always make a special attempt to read the missed meters for commercial and industrial customers because these tend to generate higher revenues. For residential customers, they want to use estimated billing for the missed meters. However, the public utility commission in many areas will not allow estimated billing. For example, in Illinois, utility companies have to perform actual meter reading at least every second billing cycle ([Illinois Administrative Code 2018](#)). Similar examples can be found in Colorado ([Colorado Department of Regulatory Agencies 2018](#)), Michigan ([Michigan Department of Labor and Economic Growth 2018](#)), and Irving, Texas ([Irving, Texas - Code of Ordinances 2018](#)). A vehicle has to be sent at a later time to read the missed meters, and this leads to increased costs due to additional operational costs

and overtime payments to drivers.

2.1.2 Research Goal and Contributions

In the meter reading context, we will address the above-mentioned issues of the RFID technology by generating routes for the CEVRP that are both cost-effective and robust (in the sense that we seek to minimize the number of missed reads). This is done by bringing together data analytics, statistical modeling, and optimization techniques. The idea is to significantly reduce the number of missed meters even though the routes that are generated may be somewhat longer than those currently used by a utility company. For the utility companies, it is much easier and cost-effective if they know ex-ante that they have to traverse a somewhat longer route that leads to fewer missed meters. This substantially reduces the need to dispatch a vehicle to read the missed meters that may be spread throughout the street network. While past research has focused on mathematical formulations and computational experiments on artificially generated networks, we use real networks and actual meter reading data from utility companies to solve a more realistic version of this problem. Real networks that we use are a few orders of magnitude larger than the artificial networks used in the literature. The most important factor is that the way in which street segments and meters are distributed on real networks, is very different from that in artificial networks. [Hà \(2012\)](#) gives a detailed description of how artificial meter reading networks are systematically generated. Therefore, the computational performances of the heuristics discussed in the literature do not have

enough practical relevance.

We summarize the main contributions of this chapter as follows.

1. We formulate the stochastic meter reading problem as a two-stage integer program (IP), where the Stage 1 IP is a linear IP that guarantees a pre-specified likelihood of reading the meters. The Stage 2 IP adds deadhead segments to the solution of the Stage 1 IP to generate the full route. The two-stage IP formulation is deterministic even though the use of the RFID technology makes the meter reading problem inherently stochastic.
2. We develop three Bayesian updating learning models, namely, a logit model, a probit model, and a hierarchical probit model to capture the uncertainty in the data and also to avoid the shortcomings of regression. We show that the hierarchical probit model gives a more accurate estimate of the probability that a meter is read successfully compared to logit and probit models. We perform simulation experiments using an actual street network with meter locations to show that the hierarchical probit model generates robust routes, i.e., the number of missed meters is significantly less compared to the other two Bayesian models, even though the routes may be slightly longer.
3. We present an iterative algorithmic framework. We start by learning from the incoming data every time the meter reading vehicle collects readings. We then re-solve the two-stage IP with the updated probability of a meter being read successfully to generate routes that are more robust for addressing the uncertainty. Utility companies can integrate this algorithm into their route

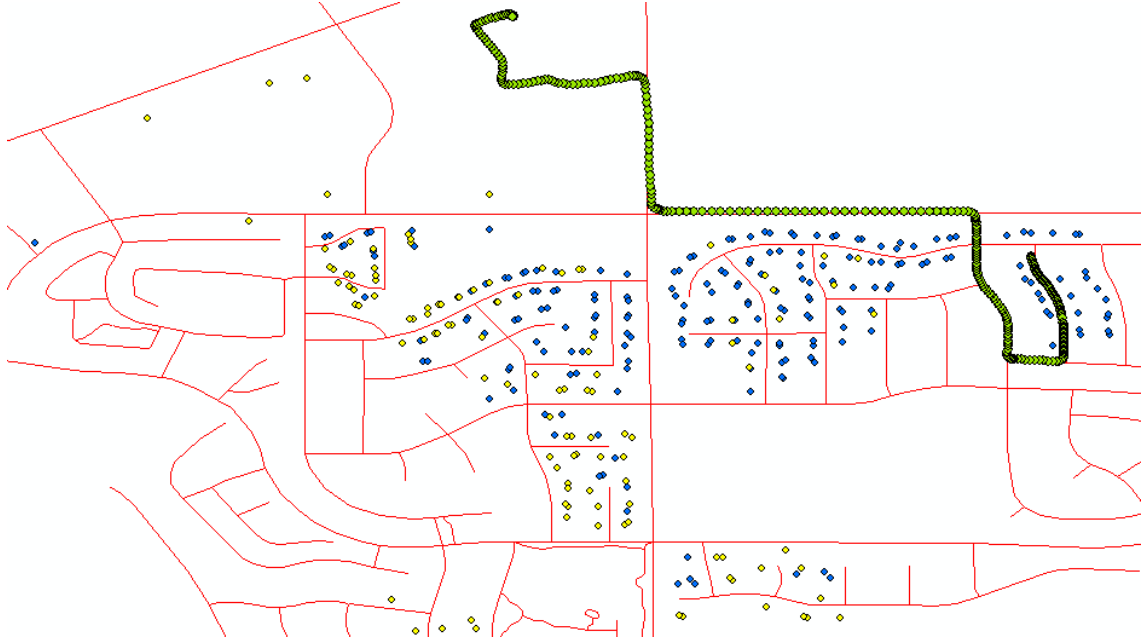


Figure 2.1: (Color online) A view of the street layer, the service location layer, and the reading events layer. The red lines represent the street segments. The green dots represent the route traversed by the meter reading vehicle. The blue dots and the yellow dots represent meters (customers) in the service location layer that are read and that are missed, respectively, after the vehicle has traversed the route. If multiple meters have the same geographic location, then all those meters are represented by a single dot, although they have distinct account identifiers.

generating software as a decision-support tool.

2.2 Description of the Data Set

The data set was gathered during the second half of 2015 by ITRON (a technology and services company) and provided by RouteSmart Technologies. ITRON manufactures radio frequency transmitters and receivers that are used by clients for meter reading. The data are in GIS format and have three layers.

1. Street Level Data. Information about the shape, length, and type of street segments.

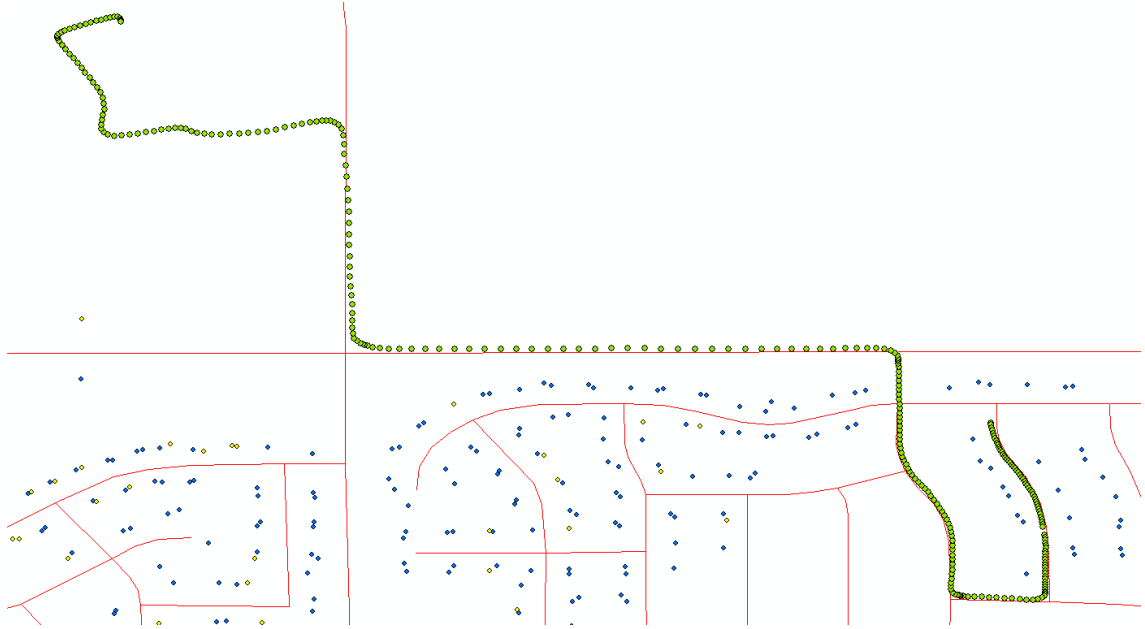


Figure 2.2: (Color online) A magnified view of the reading events layer.

2. Service Location Data. Geographic locations of all meters that are to be read and each indexed by a unique account identifier.
3. Reading Events Data. Records of all read events by the meter reading vehicle in the form of the time of read (with a resolution of one second), the account identifier of the meter that is read, and the geographic location of the vehicle during the read.

The data are represented using ArcGIS ([Steiniger and Hunter 2013](#)). In [Figure 2.1](#), we show how the data appear in GIS format with views of the street layer, the service location layer, and the reading events layer. After the vehicle has traversed a portion of the route marked by the green dots, [Figure 2.1](#) shows the meters in the service location layer that are read (blue dots) and those that are missed (yellow dots). Even though ITRON specifies the range of the RFID signals to be around

500 feet, some missed meters are well within that range, while some meters that are read are well outside of it. The routes generated should address these variabilities.

Figure 2.2 gives a magnified view of the reading events layer. The green dots are farther away from each other on some street segments compared to other street segments. Since the green dots are the vehicle locations every second, we see that the vehicle has traveled at different speeds on different street segments.

From the data, we make the following observations. There are many account identifiers in the reading events file that have no corresponding entry in the service location file, i.e., the RFID readers are picking up signals from nearby RFID transmitters that do not require reading by the utility companies. The read events data have a many-to-one relationship to a service location account identifier, i.e., some of the meters are read more than once by the meter reading device. The vehicle location is tracked every second. However, when the read events for a single meter are recorded, they do not occur every second along a street segment that seems to be within range. Rather, there is generally a regular time gap between occurrences of read events for the same meter. This confirms the fact that the signal transmitted by an RFID tag is at regular time intervals and is not occurring continuously. Some meters that are very close to the vehicle route have been missed, probably due to a discontinuous signal. Missed reads can also be due to the variability of the range of a meter to transmit a signal because of weather conditions, surrounding obstacles, or decreasing battery life of the signal transmitters.

2.3 Initial Data Analysis

Five steps are carried out sequentially to analyze the data.

Step 1. The read events of unwanted account identifiers are separated from the read events of those account identifiers that are in the service location layer.

Step 2. The account identifiers in the service location layer that are read at least once are separated from the account identifiers in the service location layer that are missed.

Step 3. The number of times each of the meters (account identifiers) in the service location layer that were read at least once is calculated.

Step 4. The minimum time interval between any two consecutive reads of each meter in the service location layer that were read more than once is calculated using the time stamp of the read events.

Step 5. The maximum read distance of each of the meters in the service location layer that were read at least once is calculated using the location of the meters and the location of the vehicle during the read events.

In Table [2.1](#), we provide the results based on the analysis from Steps 1 to 5.

In Figure [2.3](#), we use box plots to show the relationship between the minimum time interval between reads and the number of reads. Consider the value of three on the x -axis, i.e., the number of reads is three. We are considering meters in the

Total number of meters in the service location layer	474
Number of meters in the service location layer that are read	209
Total number of read events	28,745
Number of read events from meters in the service location layer	827
Number of street segments traversed in the route	7
Time gap between consecutive signal transmission (sec)	13
Maximum read distance among all meters in the service location layer (feet)	3,510

Table 2.1: Results based on the analysis from Step 1 to Step 5.

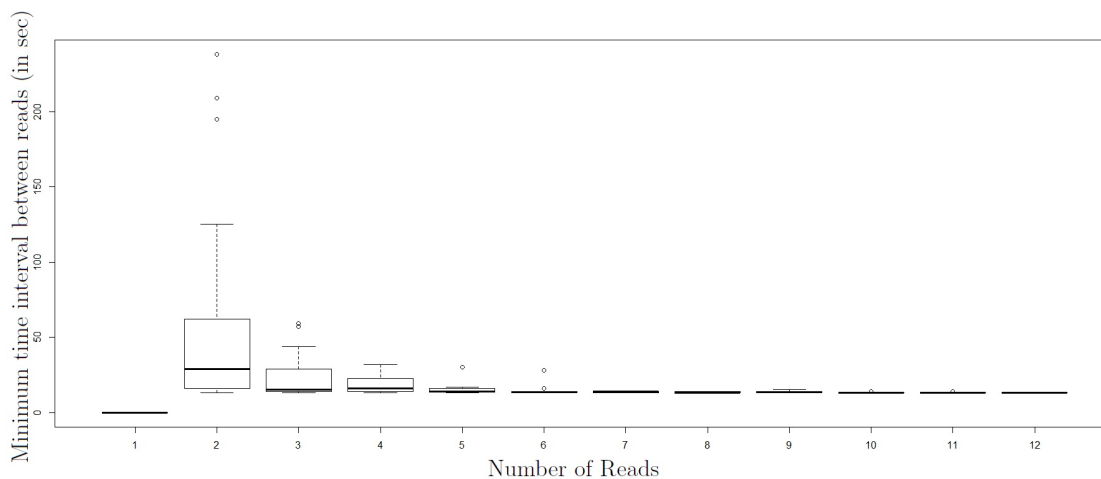


Figure 2.3: Minimum time interval between reads.

service location layer that are read exactly three times from the route traversed by the meter reading vehicle (there are 29 such meters). For these meters, we have a time interval between their first read and second read, and a time interval between their second read and third read. We take the minimum of these two time intervals for each of the 29 meters. The 29 minimum time interval values, which range from 13 seconds to 59 seconds, are shown using a box plot at $x = 3$. We observe that most of the large values of the minimum time interval occur with a fewer number of reads. When the number of reads is less for a meter, the vehicle is probably farther from the meter most of the time on the route. The vehicle probably came close

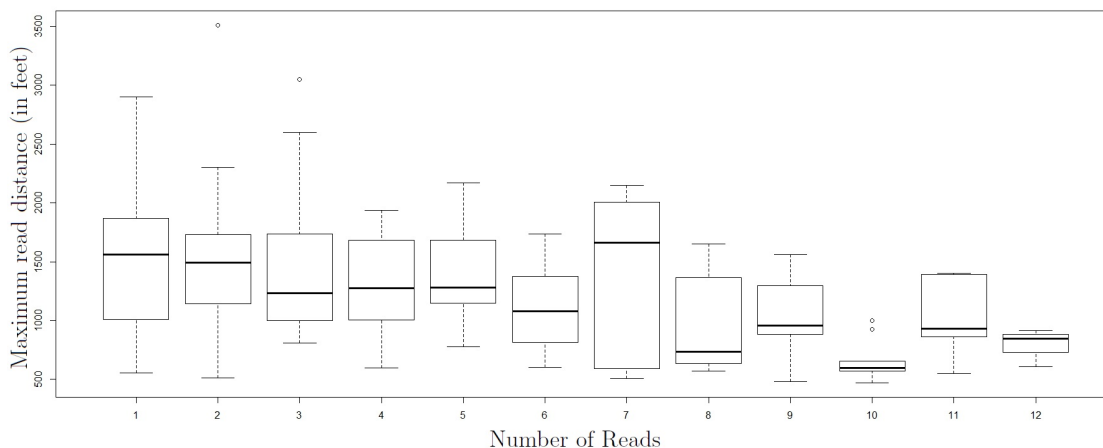


Figure 2.4: Maximum read distance.

to the meter for small portions of the route. When the number of reads is larger for a meter, the vehicle is probably closer to the meter for a longer duration. The vehicle should have read that meter every second. However, this is not the case. Instead, for this data set, the minimum time intervals attain a constant value of 13 seconds. This value is the time gap between consecutive signals sent by the RFID transmitters.

In Figure 2.4, we use box plots to show the relationship between the maximum read distance and the number of reads. Again, consider the value of three on the x -axis. For the 29 meters, we have the read distances for each of their three reads. We take the maximum of these three read distances for each of the meters. These 29 maximum read distance values, which range from 808 feet to 3,052 feet, are shown using a box plot at $x = 3$. From our observations, it seems that the chances of having larger values of the maximum read distance increases for meters with a smaller number of reads. This also confirms our observations from Figure 2.3. When

the number of reads is less for a meter, the vehicle is farther from the meter most of the time on a route; when the number of reads is larger for a meter, the vehicle is closer to the meter for a longer duration.

We perform four additional steps of analysis on the data with respect to a meter being read or not being read.

Step 6. The route traversed by a meter reading vehicle is discretized (like the green dots in Figure 2.2) using the distinct geographic coordinates of the vehicle position during the read events.

Step 7. For all the meters in the service location layer, the shortest distance from the route traversed by a vehicle is calculated using the distinct locations of the vehicle. The shortest distance from the route will be used as a proxy for the distance from the meters to the route.

Step 8. Around each of the distinct points in discretized route, a circular disc (radius of 100 feet to 1000 feet with steps of 100) is considered.

Step 9. For each radius, we count the number of meters within at least one of the circular discs and the number read (regardless from where the meters are read). We then calculate the fraction of meters read for each radius.

In Table 2.2, we provide the results based on the analysis from Step 9. The fraction of meters read are calculated both cumulatively and non-cumulatively for each of the 10 different radii, ranging from 100 feet to 1000 feet. The entries in the two columns have the form a/b , where b denotes the number of meters within that

Radius (feet)	Cumulative Success	Non-Cumulative Success
100	14/14 = 1.00	14/14 = 1.00
200	35/35 = 1.00	21/21 = 1.00
300	53/54 = 0.98	18/19 = 0.95
400	64/67 = 0.96	11/13 = 0.85
500	74/78 = 0.95	10/11 = 0.91
600	85/94 = 0.90	11/16 = 0.69
700	97/108 = 0.90	12/14 = 0.86
800	117/131 = 0.89	20/23 = 0.87
900	129/147 = 0.88	12/16 = 0.75
1000	149/171 = 0.87	20/24 = 0.83

Table 2.2: Results based on the analysis from Step 9.

radius for the cumulative case and the number of meters between that radius and the previous lower radius considered for the non-cumulative case, and a denotes the number of meters read out of those b meters. The fractions in the cumulative case show a gradual decrease in success with an increase in the distance of meters from the route. We do not see a specific trend for the non-cumulative case. We note that the smallest value of the fraction occurs for meters that are at a distance of 500 feet to 600 feet from the route. This observation indicates that the shortest distance of meters from routes is not the only key factor for reading a meter successfully. Otherwise, the non-cumulative case would have followed the same trend as the cumulative case.

2.4 Integer Programming Formulation

We formulate the meter reading problem with RFID technology as a two-stage IP. The Stage 1 IP finds the street segments that are to be traversed for reading each meter with a pre-specified chance of being read. The solution of the Stage 1 IP gives

street segments spread across the street network, which does not necessarily form a full route. A mixed rural postman problem finds the shortest way of connecting a given set of required street segments to form a full route on a mixed graph with edges (two-way street segments) and arcs (one-way street segments). The Stage 2 IP solves a mixed rural postman problem that adds deadhead segments (extra street segments not required for reading meters) to the solution of the Stage 1 IP to obtain the full route and it ensures that the depot (denoted by a node on the graph) is a part of the route.

2.4.1 Stage 1 IP Formulation

Consider a street network as a mixed graph denoted by $G = (V, E \cup A)$, where E denotes the set of the edges, A denotes the set of the arcs, and V denotes the set of nodes. Let $c_j \geq 0$ be the cost (length) of street segment j . Let I be the set of the meters. Let p_{ij} be the probability that meter i is read at least once from street segment j . Let $L_i \in [0, 1]$ be the specified likelihood of reading meter i from the full route. We define x_j to be the binary decision variable denoting whether or not street segment j should be traversed. The Stage 1 IP formulation is given on the next page.

The objective function (2.1) minimizes the total cost (length). Constraints (2.2) select the values of the binary decision variables (x_j) so that the probability of reading meter i is at least L_i . Constraints (2.3) define the decision variables. In general, the solution of the Stage 1 IP, i.e., the graph induced by the required

edges and arcs $G_R = (V, E_R \cup A_R)$, where $E_R \subseteq E$ and $A_R \subseteq A$ denote the set of required edges and arcs, respectively, is not connected. The objective value of the Stage 1 IP will be greater for larger values of L_i . The greater the need to read meter i during the next meter reading trip, the larger should be the value of L_i set by the utility company. In cases where the utility company can manage using estimated billing for meter i during the next billing cycle, the value of L_i should be set close to 0. We note that constraints (2.2) can be linearized in the decision variables $\sum_{j \in E \cup A} x_j \times \log(1 - p_{ij}) \leq \log(1 - L_i)$ for all meters i yielding a linear Stage 1 IP.

$$\text{(Stage 1 IP)} \quad \min \quad \sum_{j \in E \cup A} c_j x_j \quad (2.1)$$

$$\text{s.t.} \quad \prod_{j \in E \cup A} (1 - p_{ij})^{x_j} \leq (1 - L_i) \quad \forall i \in I \quad (2.2)$$

$$x_j \in \{0, 1\} \quad \forall j \in E \cup A \quad (2.3)$$

For values of L_i close to 1, constraints (2.2) can be infeasible for some meter i even when the meter reading vehicle traverses all street segments in the network ($x_j = 1$ for all street segments j), i.e., meter i cannot be read automatically with probability of at least L_i . In that case, the driver of the meter reading vehicle will need to park the vehicle on the closest street segment and read meter i manually. This means that meter i is read with probability 1 from the closest street segment, i.e., the Stage 1 IP is solved with $p_{ij} = 1$, where j is the closest street segment to

meter i . This will enforce $x_j = 1$ in the Stage 1 IP solution, and, therefore, street segment j will be in the set of required street segments. Let $M_R \subseteq E_R \cup A_R$ denote the subset of the required street segments that are needed to manually read some of the meters, i.e., $p_{ij} = 1$ for all $j \in M_R$. We consider a constant stoppage time to manually read meter i from street segment j . Accordingly, we add a penalty to the Stage 2 IP objective value as a proxy for the distance that could have been traversed during the stoppage time.

2.4.2 Stage 2 IP Formulation

For $S_1, S_2 \subseteq V$, $(S_1 : S_2)$ denotes the set of edges and arcs with one endpoint in S_1 and the other endpoint in S_2 . $A(S_1 : S_2) = \{(i, j) \in A : i \in S_1, j \in S_2\}$ denotes the set of arcs with one endpoint in S_1 and the other endpoint in S_2 . $E(S_1 : S_2) = \{(i, j) \in E : i \in S_1, j \in S_2\}$ denotes the set of edges with one endpoint in S_1 and the other endpoint in S_2 . For $S \subseteq V$, $\delta^+(S) = A(S : V \setminus S)$, $\delta^-(S) = A(V \setminus S : S)$ and $\delta(S) = E(S : V \setminus S)$, where $E(S)$ and $A(S)$ denote the set of edges and arcs, respectively, with both endpoints in S . $\delta^*(S) = \delta(S) \cup \delta^+(S) \cup \delta^-(S) = (S : V \setminus S)$. If $S = \{v_i\}$, we simply write $\delta(i)$, $\delta^+(i)$, $\delta^-(i)$ or $\delta^*(i)$. The vertex sets of the connected components of G_R are denoted by V_1, \dots, V_p . The depot is denoted by the node $v_0 \in V$. We consider a single meter reading vehicle. We define y_j to be the non-negative integer decision variable denoting the number of times street segment j is traversed in the full route. For $F \subseteq E \cup A$, $Y(F) = \sum_{j \in F} y_j$. The Stage 2 IP formulation is given by the following.

$$\text{(Stage 2 IP)} \quad \min \quad \sum_{j \in E \cup A} c_j y_j \quad (2.4)$$

$$\text{s.t.} \quad Y(\delta^*(0)) \geq 1 \quad (2.5)$$

$$Y(\delta^*(i)) \equiv 0 \pmod{2} \quad \forall i \in V \quad (2.6)$$

$$Y(\delta^+(S)) \geq 1 \quad \forall S = \cup_{k \in Q} V_k, \quad Q \subset \{1, \dots, p\} \quad (2.7)$$

$$Y(\delta^+(S)) - Y(\delta^-(S)) \leq Y(\delta(S)) \quad \forall S \subset V \quad (2.8)$$

$$y_j \geq 1 \text{ and integer} \quad \forall j \in E_R \cup A_R \quad (2.9)$$

$$y_j \geq 0 \text{ and integer} \quad \forall j \in E \cup A \setminus E_R \cup A_R \quad (2.10)$$

The objective function (2.4) minimizes the total cost (length) of the route. Constraint (2.5) ensures that the depot is a part of the route. Constraints (2.6) are the flow conservation constraints, i.e., every node has an even degree in the route. Constraints (2.7) are the disjoint subtour elimination constraints, i.e., the required street segments obtained in the Stage 1 IP are connected in the route. Constraints (2.8) are the balanced-set inequalities, i.e., the difference between the number of arcs in the route entering S and the number of arcs in the route leaving S cannot be more than the number of edges in the route between S and $V \setminus S$. Constraints (2.9) define the decision variables for those street segments j which are required to be traversed by the Stage 1 IP, i.e., $x_j = 1$. Constraints (2.10) define the decision variables for those street segments j which are not required to be traversed by the Stage 1 IP, i.e., $x_j = 0$. The Stage 1 IP solution already meets the specified likelihood L_i of

reading each meter i . The deadhead segments added in the Stage 2 IP increase the likelihood of reading the meters because the meter reading vehicle is also receiving signals while traversing the deadhead segments. The Stage 2 IP formulation without constraint (2.5) is the formulation for the mixed rural postman problem (Corberán et al. 2014).

2.5 Jensen's Inequality for the Stage 1 IP

Jensen's inequality generalizes the statement that the secant line of a convex function lies above the graph of the function. For just two points x_1 and x_2 , the secant line consists of weighted means of the convex function evaluated at the points, $tf(x_1) + (1-t)f(x_2)$, while the graph of the function is the convex function evaluated at the weighted means of the points, $f(tx_1 + (1-t)x_2)$. Thus, Jensen's inequality is

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$$

In the context of probability theory, it is generally stated in the following form: if X is a random variable and φ is a convex function, then

$$\varphi(\mathbb{E}(X)) \leq \mathbb{E}(\varphi(X)).$$

In the context of our meter reading problem, let \hat{p}_{ij} be the random variable denoting the probability that meter i is read from street segment j at least once and $p_{ij} = \mathbb{E}(\hat{p}_{ij})$, where $\mathbb{E}()$ denotes the expected value. From Jensen's inequality it

follows that

$$(1 - p_{ij})^{x_j} \leq \mathbb{E}((1 - \hat{p}_{ij})^{x_j}) \quad (2.11)$$

and the equality holds for x_j values of 0 and 1. In constraint (2.2), $(1 - p_{ij})^{x_j}$ denotes the probability that meter i is missed from street segment j when street segment j is traversed x_j times. If we allow the decision variables x_j to attain integer values greater than 1, i.e., the street segments can be repeated, then the meter reading vehicle may traverse street segment j several times ($x_j > 1$ and integer) to read meters from street segment j with higher probability. In that case, from (2.11) we can observe that the probability that meter i is missed from street segment j when street segment j is traversed x_j times is underestimated and therefore, the probability that meter i is read from street segment j at least once when street segment j is traversed x_j times is overestimated. The equality in (2.11) also holds when it is assumed that the probability that meter i is missed from street segment j is independent across multiple traversals of street segment j , since in that case, $\mathbb{E}((1 - \hat{p}_{ij})^{x_j}) = (\mathbb{E}(1 - \hat{p}_{ij}))^{x_j} = (1 - p_{ij})^{x_j}$. But, if street segment j is traversed multiple times on the same day or even within a span of few days, then the probability that meter i is missed from street segment j should be correlated across those traversals.

Let us consider an example when $x_j = 2$. The worst case scenario would be that if \hat{p}_{ij} values are 0 and 1. So, $p_{ij} = (0 + 1)/2 = 0.5$ and therefore, $(1 - p_{ij})^{x_j} = (1 - 0.5)^2 = 0.25$. On the other hand we have, $\mathbb{E}((1 - \hat{p}_{ij})^{x_j}) = ((1 - 1)^2 + (1 - 0)^2)/2 =$

0.5. So, in this example the probability that meter i is missed from street segment j when street segment j is traversed twice is underestimated by $0.5 - 0.25 = 0.25$, and therefore, the probability that meter i is read from street segment j at least once when street segment j is traversed twice is overestimated by the same amount.

The estimation bias is

$$\begin{aligned}
b(p_{ij}) &= \mathbb{E}((1 - \hat{p}_{ij})^{x_j}) - (1 - p_{ij})^{x_j} \\
&\approx \mathbb{E}\left(\left(\frac{d}{dp_{ij}}(1 - p_{ij})^{x_j}\right)(\hat{p}_{ij} - p_{ij})\right) + \frac{1}{2}\mathbb{E}\left(\left(\frac{d^2}{dp_{ij}^2}(1 - p_{ij})^{x_j}\right)(\hat{p}_{ij} - p_{ij})^2\right) \\
&= 0 + \frac{1}{2}\left(\frac{d^2}{dp_{ij}^2}(1 - p_{ij})^{x_j}\right)(Var(\hat{p}_{ij}))
\end{aligned}$$

Constraint (2.2) can be modified as $\prod_{j \in E \cup A} ((1 - p_{ij})^{x_j} + b(p_{ij})) \leq (1 - L_i)$ to correct for the overestimation in the probability that meter i is read from street segment j at least once when street segment j is traversed x_j times. With this modified constraint the Stage 1 IP cannot be linearized in the decision variables $\sum_{j \in E \cup A} \log((1 - p_{ij})^{x_j} + b(p_{ij})) \leq \log(1 - L_i)$. To compute $b(p_{ij})$ we need to compute $Var(\hat{p}_{ij})$, which is not easy to estimate. So, the most reasonable way around is to solve the linear Stage 1 IP.

2.6 Regression

In order to solve the Stage 1 IP, we need to estimate the values of the p_{ij} 's. We use regression models. The dependent variable in these models is denoted by `Read_OR_Notij` (whether or not meter i was read from street segment j). The

data elements have the form of 1 and 0, where 1 indicates that meter i is read from street segment j and 0 indicates that meter i is not read from street segment j . The predicted values of the dependent variable in the regression model have to be between 0 and 1 which will denote the probability p_{ij} . Based on the type of the data we have and our requirements on the predicted values of the dependent variable, logit and probit models are considered. The independent variables in these models are: $\text{Shortest_Distance}_{ij}$ (shortest distance between meter i and street segment j), No_of_Pulses_j (number of pulses the meter reading vehicle can receive from the meter while traveling on street segment j), and No_of_Customers_i (number of meters within 500 feet from meter i ; 500 feet is the range of the RFID signals as specified by ITRON, so the signals are strong enough to interfere with each other within 500 feet). $\text{Shortest_Distance}_{ij}$ should have a negative coefficient because the larger the shortest distance between meter i and street segment j is, the smaller the value of p_{ij} . No_of_Pulses_j is obtained from the amount of time the meter reading vehicle spent on street segment j divided by the time interval between the RFID signal transmissions. If the meter reading vehicle travels at a higher speed through street segment j , then the time spent by the vehicle on street segment j is smaller and, therefore, the No_of_Pulses_j is lower. No_of_Pulses_j should have a positive coefficient because the greater the number of pulses the meter reading vehicle can receive from the meters while traveling on street segment j , the larger the value of p_{ij} . No_of_Customers_i is a measure of density of meters in a region. It is important because, with a large number of meters in a region, the interference of the RFID signals is greater, so the signals die out quickly. No_of_Customers_i should have a

negative coefficient because the greater the number of meters surrounding meter i , the smaller the value of p_{ij} .

We constructed six logistic regression models and six probit regression models. For the logistic and probit regressions, Model 1 uses three independent variables: $\text{Shortest_Distance}_{ij}$, No_of_Pulses_j , and No_of_Customers_i . Model 2 adds indicator variables for the traversed street segments to Model 1. Model 3 adds indicator variables for the meters to Model 1. Model 4 adds indicator variables for the traversed street segments to Model 3. Model 5 uses $\text{Shortest_Distance}_{ij}$, No_of_Pulses_j , and indicator variables for the meters. Model 6 adds indicator variables for the traversed street segments to Model 5.

In logistic regression and probit regression, the parameters are estimated using the maximum likelihood estimation (MLE) method. We could use McFadden's R^2 ($1 - \text{Residual Deviance}/\text{Null Deviance}$) to assess our models. [Louviere et al. \(2000\)](#) mention that values of McFadden's R^2 between 0.2 and 0.4 are considered to be indicative of extremely good model fits. Simulations by [Domencich and McFadden \(1975\)](#) showed that this range is equivalent to 0.7 to 0.9 for an R^2 value from ordinary least squares (OLS). However, McFadden's R^2 is similar to R^2 from OLS in that its value always increases as new predictors are added to the model. Instead, we use McFadden's Adjusted R^2 ($1 - \text{Akaike Information Criterion}/\text{Null Deviance}$) to assess our models. It is similar to the Adjusted R^2 from OLS in that it penalizes for using additional predictors.

2.7 Regression Results

We use only the traversed street segments to estimate the coefficients of the regression models. We can only determine that a meter was read or not read on street segments traversed by the meter reading vehicle. Thus, the data used for estimating the coefficients has 474 meters and 7 street segments. Therefore, the size of the data set is 3,318 (474×7).

In Table 2.3, we provide summary statistics of the dependent and the independent variables. In Table 2.4, we give the correlation between each pair of variables. From Table 2.3, we see that only 15.2% of the 3,318 values for Read_OR_Not have a value of 1 and the rest have a value of 0. This is due to the fact that many meters are far away from the route, so that many of them are never read from the route. From Table 2.4, we see that the correlation between Read_OR_Not and Shortest_Distance is negative and the largest in magnitude. There is a slight positive correlation between both Read_OR_Not and No_of_Pulses and No_of_Customers. Shortest_Distance and No_of_Customers have a high negative correlation that can make the coefficient of No_of_Customers negative in the regression models.

In Tables 2.5 and 2.6, we present the logistic regression results and the probit regression results, respectively. We give the means of the coefficients of the independent variables and their standard deviations in parenthesis. The results in Tables 2.5 and 2.6 for the six models are similar for both logistic regressions and probit regressions. For each of the six models in both regressions, the coefficient of Shortest_Distance is always significant and negative; the coefficient of No_of_Pulses

Statistic	Read_OR_Not	Shortest_Distance	No_of_Pulses	No_of_Customers
Mean	0.152	2,528.714	5.500	33.181
Standard Deviation	0.359	1,491.172	6.165	15.218
Minimum	0.000	62.804	1.500	0.000
25 th percentile	0.000	1,231.286	1.500	21.000
Median	0.000	2,348.150	2.500	31.000
75 th percentile	0.000	3,695.959	8.500	43.000
Maximum	1.000	5,759.010	19.500	72.000

Table 2.3: Summary statistics for the dependent and the independent variables.

whenever significant is positive; the coefficient of No_of_Customers whenever significant is negative. In Figures 2.5 and 2.6, we give the histograms for the fitted values of the dependent variable from the logistic regression models and the probit regression models, respectively. The histograms for each of the six logistic regression models are similar to the histograms for each of the six respective probit regression

	Read_OR_Not	Shortest_Distance	No_of_Pulses	No_of_Customers
Read_OR_Not	1.000			
Shortest_Distance		-0.508		
No_of_Pulses			0.106	
No_of_Customers				0.143
		1.000	-0.057	-0.484
			1.000	0.000
				1.000

Table 2.4: Correlation between the dependent and the independent variables.

models.

In Tables 2.5 and 2.6 for both regressions, all three independent variables are significant at the 1% level in Model 1. In Model 2, for both regressions, No_of_Pulses is not significant, and the other two independent variables are significant at the 1% level in the presence of street dummies. In Model 3, for both regressions,

	Read_OR_Not					
	(1)	(2)	(3)	(4)	(5)	(6)
Shortest_Distance	-0.003*** (0.0002)	-0.003*** (0.0002)	-0.007*** (0.0005)	-0.007*** (0.0010)	-0.007*** (0.0005)	-0.007*** (0.0010)
No_of_Pulses	0.060*** (0.010)	-0.004 (0.053)	0.091*** (0.016)	0.165** (0.080)	0.091*** (0.016)	0.165** (0.080)
No_of_Customers	-0.017*** (0.005)	-0.021*** (0.005)	0.060 (0.204)	0.057 (0.202)		
Constant	2.852*** (0.258)	3.680*** (0.382)	8.137 (6.035)	7.983 (5.951)	9.577*** (1.669)	9.357*** (1.641)
Street Dummies	No	Yes	No	Yes	No	Yes
Customer Dummies	No	No	Yes	Yes	Yes	Yes
McFadden's Adjusted R ²	0.514	0.525	0.377	0.381	0.377	0.381

*p<0.1; **p<0.05; ***p<0.01

Table 2.5: Logistic regression results.

No_of_Customers is not significant, and the other two independent variables are significant at the 1% level in the presence of customer dummies. In Model 4, for both regressions, No_of_Customers is not significant, Shortest_Distance is significant

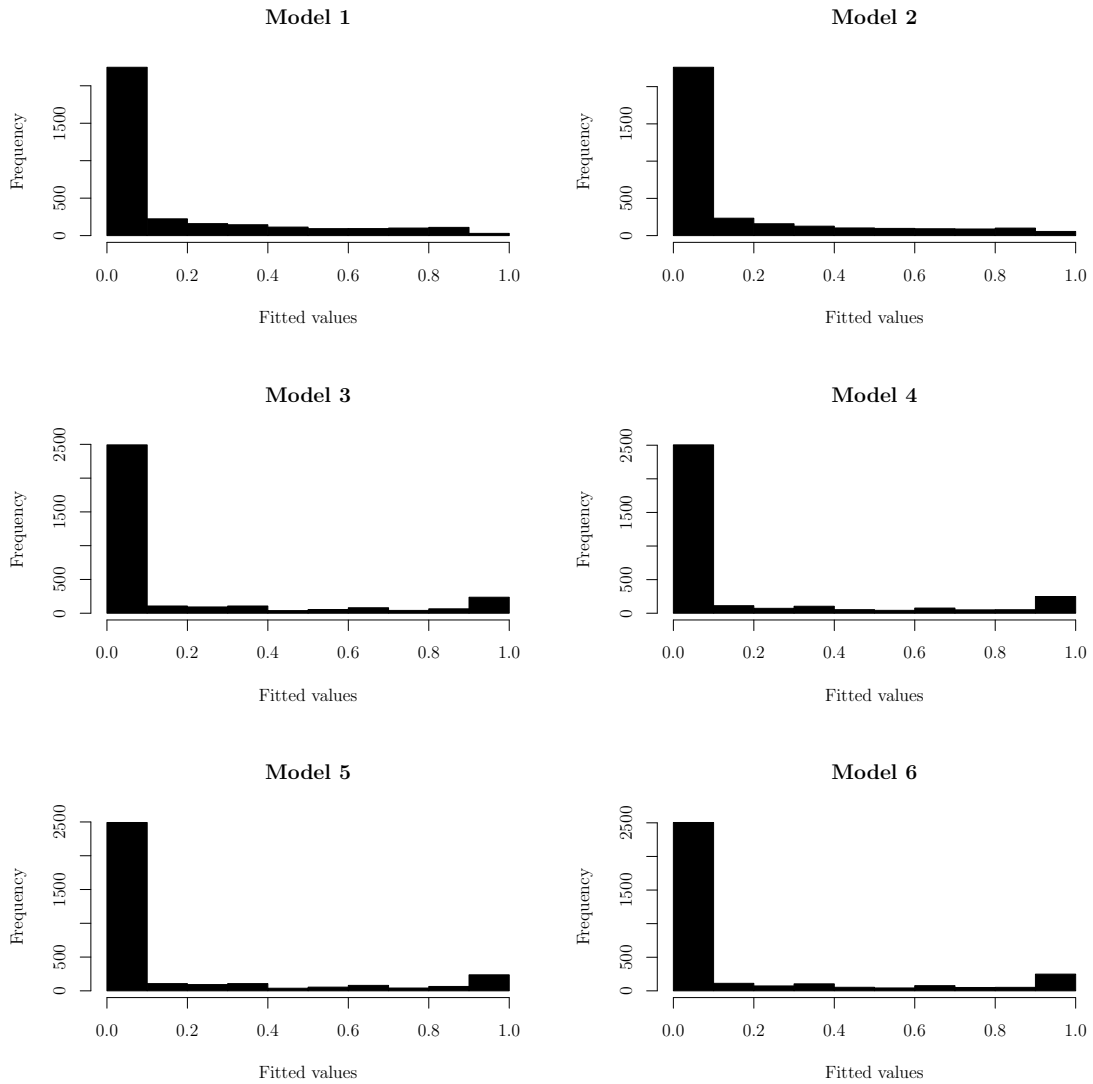


Figure 2.5: Histograms of fitted values from the logistic regressions.

at the 1% level, and No_of_Pulses is significant at the 5% level and the 10% level for logistic regression and probit regression, respectively, in the presence of both street dummies and customer dummies. In two of the first four models, No_of_Customers is not significant, so we construct two models with customer dummies that leave out No_of_Customers. In Model 5, for both regressions, the two independent variables are significant at the 1% level in the presence of customer dummies. In Model 6,

	Read_OR_Not					
	(1)	(2)	(3)	(4)	(5)	(6)
Shortest_Distance	-0.002*** (0.0001)	-0.002*** (0.0001)	-0.004*** (0.0002)	-0.004*** (0.0003)	-0.004*** (0.0002)	-0.004*** (0.0003)
No_of_Pulses	0.032*** (0.005)	-0.009 (0.030)	0.052*** (0.009)	0.084* (0.044)	0.052*** (0.009)	0.084* (0.044)
No_of_Customers	-0.013*** (0.003)	-0.015*** (0.003)	0.032 (0.111)	0.032 (0.110)		
Constant	1.721*** (0.147)	2.180*** (0.213)	4.479 (3.411)	4.378 (3.377)	5.254*** (0.986)	5.138*** (0.977)
Street Dummies	No	Yes	No	Yes	No	Yes
Customer Dummies	No	No	Yes	Yes	Yes	Yes
McFadden's Adjusted R ²	0.512	0.523	0.377	0.381	0.377	0.381

*p<0.1; **p<0.05; ***p<0.01

Table 2.6: Probit regression results.

for both regressions, Shortest_Distance is significant at the 1% level, and No_of_Pulses is significant at the 5% level and the 10% level for logistic regression and probit regression, respectively, in the presence of both street dummies and customer

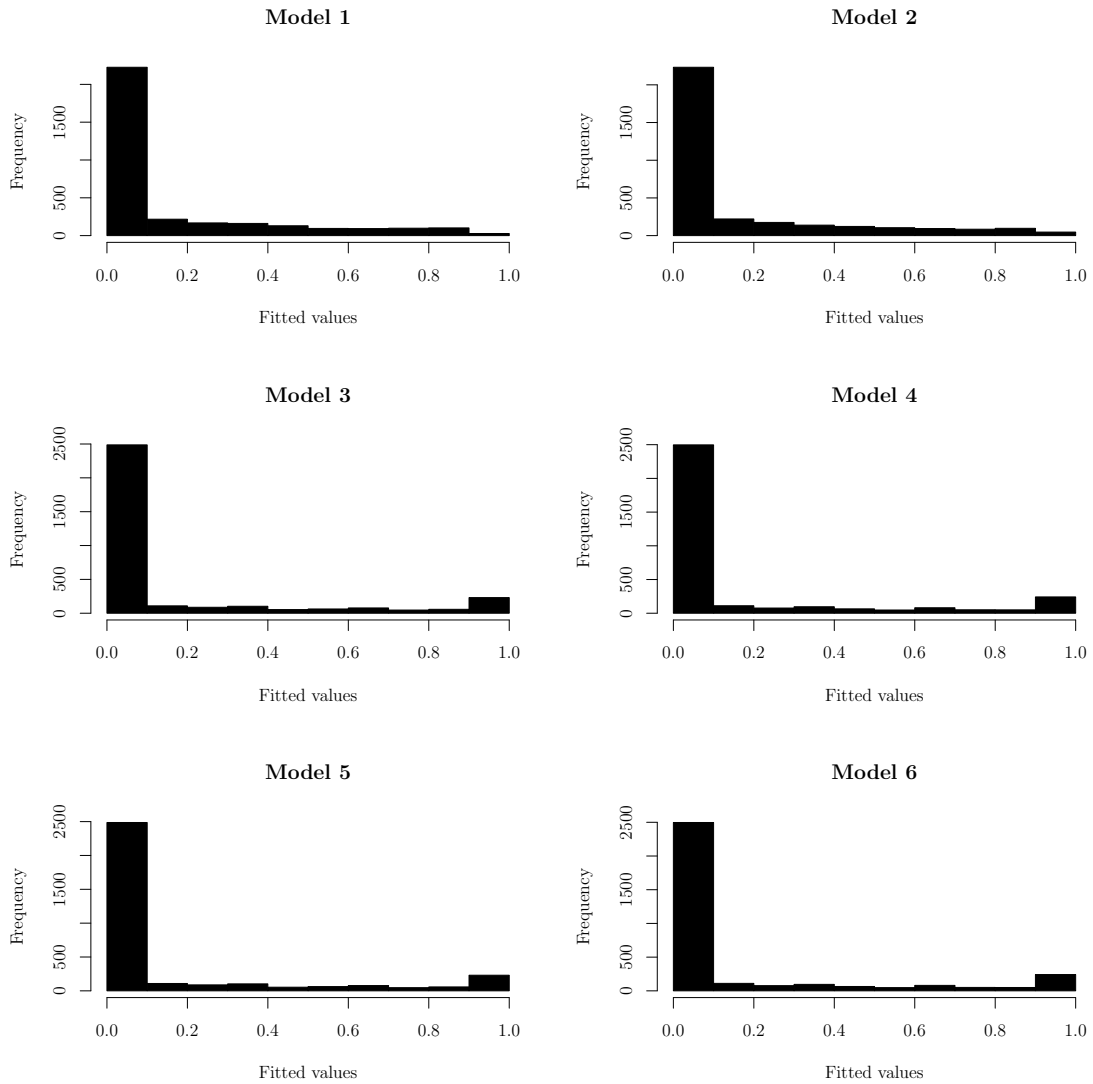


Figure 2.6: Histograms of fitted values from the probit regressions.

dummies. Based on the results in Tables 2.5 and 2.6, Model 1 and Model 2 perform the best for both logistic regression and probit regression, with Model 2 performing slightly better than Model 1. The simplicity of Model 1 without any dummies, however, makes it preferable to Model 2. Model 1 has McFadden’s Adjusted R^2 values of 0.514 and 0.512 for logistic regression and probit regression, respectively. These values indicate very good model fits. We also tried models with higher powers

of independent variables and with interactions between independent variables but none of them performed better than Model 1 for both logistic regression and probit regression.

2.8 Bayesian Updating

Every time the meter reading vehicle collects readings, it adds more data to the previous readings. With more data, we expect that the estimates of the p_{ij} 's will be more accurate. Therefore, the routes generated by the two-stage IP will be higher quality. They will be better at capturing the uncertain signals thereby further reducing the number of missed reads.

There are some serious issues if we use regression to estimate the p_{ij} 's at every time period with the new data. Suppose in time period 1 we observe the first set of i.i.d. data denoted by y_1 . We run the regression on y_1 . In time period 2, we observe a second set of i.i.d. data denoted by y_2 , independent of y_1 . We run the regression on y_1 and y_2 together as a single data set, and so on. We are regressing on the older data sets repeatedly which makes this process of estimation inefficient. Data sets from different time periods are given equal weights in the regression. In practice, utility companies need to use different weights for the data based on seasonality and other factors. For example, during the summer season, meter reading data from the previous summer is more important and accurate compared to the meter reading data from the previous winter. Also, new obstacles may appear between a meter and a street segment, new meters may appear in the vicinity of a meter

causing more interference, and a decrease in the battery level of an RFID tag will reduce the signal range of the meter. All these factors make the most recent meter reading data more accurate. Therefore, utility companies should be able to apply different weights to parts of the data accordingly. If we estimate the p_{ij} 's at every time period with new data, i.e., every time the meter reading vehicle collects data from traversing a route, using concepts from Bayesian statistics, then we can avoid the two drawbacks faced while updating using regression.

Our Bayesian statistical approach is based on updating information using Bayes' Law. Suppose we fully observe the data Y . This is now a fixed and given quantity in the inferential process. Our Bayesian model is a (parametric) statistical model for the observed data, $(Y, f(y|\theta))$, where $f(y|\theta)$ is the probability density function given the parameter θ , and a prior distribution on the parameters, $(\Theta, p(\theta))$, where Θ is the finite dimensional parameter space. Under the assumption that the data are independent and identically distributed (i.i.d.), the likelihood function is $L(\theta|y_1, \dots, y_n) = \prod_{i=1}^n f(y_i|\theta)$ which contains all the information for inference on the unknown parameter θ . Therefore, from Bayes' Law, we have

$$\pi(\theta|y_1, \dots, y_n) = \frac{p(\theta)L(\theta|y_1, \dots, y_n)}{\int_{\Theta} p(\theta)L(\theta|y_1, \dots, y_n)d\theta} \quad (2.12)$$

where $\int_{\Theta} p(\theta)L(\theta|y_1, \dots, y_n)d\theta$ is the marginal likelihood which is independent of θ , and $\pi(\theta|y_1, \dots, y_n)$ is the posterior distribution of θ conditioning on the observed data. Since the denominator in (2.12) does not contribute towards the inference on θ , we can omit it. Therefore, (2.12) can be reduced to $\pi(\theta|y_1, \dots, y_n) \propto$

$p(\theta)L(\theta|y_1, \dots, y_n)$, that is, the posterior distribution of the parameter of interest is proportional to the prior distribution times the likelihood function.

Bayesian updating works as follows:

$$\pi(\theta|y_1) \propto p(\theta)L(\theta|y_1)$$

$$\pi(\theta|y_1, y_2) \propto \pi(\theta|y_1)L(\theta|y_2)$$

\vdots

$$\pi(\theta|y_1, \dots, y_T) \propto \pi(\theta|y_1, \dots, y_{T-1})L(\theta|y_T) = p(\theta)L(\theta|y_1, \dots, y_T).$$

In time period 1, $p(\theta)$ is the prior information on θ , and the posterior distribution is $\pi(\theta|y_1)$. In time period 2, the prior is $\pi(\theta|y_1)$, which is the posterior from time period 1, and so on. This process can be repeated and the model will continue to update the posterior distributions as we collect new data. In time period T , the posterior is $\pi(\theta|y_1, \dots, y_T)$, which does not depend on the sequence in which data arrive. This is exactly the same result that would have obtained if all the i.i.d. data (y_1, \dots, y_T) had been gathered at the same time. Bayesian updating is much faster than regression since analysis is done only on the new incoming data at each time period. Data from different time periods can be weighted differently in Bayesian updating depending on the requirements of the utility companies. The idea is to re-solve the two stage IP at the end of each time period with the new posterior distribution of the probabilities (p_{ij}) that is obtained and, thereby, have an iterative algorithm to generate more robust routes at the end of each time period. The unknown parameters in the Bayesian models are estimated using Markov Chain

Monte Carlo (MCMC) simulations.

2.8.1 Logit Model and Probit Model

Model 1 (three independent variables, no customer dummies, no street dummies) for both logistic regression and probit regression can be represented by

$$g(p_{ij}) = \hat{\beta}_1 + \hat{\beta}_2 \times \text{Shortest_Distance}_{ij} + \hat{\beta}_3 \times \text{No_of_Pulses}_j + \hat{\beta}_4 \times \text{No_of_Customers}_i$$

where g is the link function, $p_{ij} = \mathbb{E}(\text{Read_OR_Not}_{ij})$, $\hat{\beta}_k = \mathbb{E}(\beta_k)$, and $\mathbb{E}()$ denotes the expected value. For the logit model, $g(p_{ij}) = \ln(p_{ij}/1 - p_{ij})$, and for the probit model $g(p_{ij}) = \Phi^{-1}(p_{ij})$, where Φ is the cumulative Normal $(0, 1)$ distribution function. We have to estimate the unknown parameter vector $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)^\top$ which is a 4-dimensional vector of coefficients. The data matrix X is $(N \times 4)$ -dimensional, where N is the size of the data set, and the entries in the first column of X are 1's. X_k denotes row k of X . In time period 1, we do not have any prior information about β . We rely on the information obtained from the data. Therefore, the prior is set to a vague prior, i.e., the prior will have minimal effect on the posterior distribution of time period 1.

Both models have their pros and cons. Error terms in logit models have a logistic distribution, whereas error terms in probit models have a normal distribution. The logistic distribution has heavier tails compared to the normal distribution, so logit models are more robust than probit models. Logit models have a better fit to data that are more spread out in the tails. The normal distribution is the con-

jugate prior for the likelihood function in probit models. Therefore, the unknown parameters in the probit model can be estimated using an exact algorithm. However, neither the normal distribution nor any other distribution from the exponential family is the conjugate prior for the likelihood function in logit models. Therefore, the unknown parameters in the logit model have to be estimated using a non-exact algorithm.

2.8.2 Metropolis-Hastings Random Walk Algorithm for the Logit Model

The Metropolis-Hastings (MH) Random Walk algorithm ([Metropolis et al. 1953](#), [Hastings 1970](#)) is used to estimate the parameters of the logit model and has four steps.

Step 1. Choose a starting value for β .

Step 2. The random walk chain $\beta^{new} = \beta^{old} + \epsilon$ where $\epsilon \sim \text{Multivariate Normal}(0, s^2 H^{-1})$ generates candidate realizations for β , $s = 2.3/\sqrt{\text{dimension}(\beta)}$ ([Marin and Robert 2014](#), [Press 2003](#)), and H is the Hessian of the log likelihood function for the logit model.

Step 3. Accept β^{new} with probability $\alpha = \min\{1, \frac{\pi(\beta^{new}|y,X)}{\pi(\beta^{old}|y,X)}\}$.

Step 4. Repeat Steps 2 and 3.

2.8.3 Gibbs Sampling Algorithm for the Probit Model

The Gibbs sampling algorithm (Albert and Chib 1993) is used to estimate the parameters of the probit model. The setup for this algorithm is as follows. The prior distribution is $p(\beta) = \text{Multivariate Normal}(B_0, V_0)$. We have $y_k^* \sim \text{Normal}(X_k\beta, 1)$ and $y_k = \text{Indicator}(y_k^* > 0)$, where y is the dependent variable and y^* is the latent variable. The distribution of y^* is given by $p(y_k^* | y_k = 0, \beta) = \text{Normal}(X_k\beta, 1) \times \text{Indicator}(y_k^* \leq 0)$ and $p(y_k^* | y_k = 1, \beta) = \text{Normal}(X_k\beta, 1) \times \text{Indicator}(y_k^* > 0)$. Therefore, the posterior distribution is $\pi(\beta | y^*) = \text{Multivariate Normal}((X^\top X + V_0^{-1})^{-1}(X^\top y^* + V_0^{-1}B_0), (X^\top X + V_0^{-1})^{-1})$. The Gibbs sampling algorithm for the probit model has four steps.

Step 1. Choose a starting value for β .

Step 2. Draw $[y^* | y, \beta]$.

Step 3. Draw $[\beta | y^*]$.

Step 4. Repeat Steps 2 and 3.

2.8.4 Hierarchical Probit Model

For our meter reading problem, hierarchical models consider the signal transmission behavior of individual meters and their interactions with the signals from other meters. Bayesian updating for the hierarchical probit model is a more complex method for estimating the p_{ij} 's but the estimates are more accurate compared to Bayesian updating for the logit model and the probit model. The hierarchical

probit model accounts for the uncertain behavior of each meter separately while also accounting for the similarity between meters.

The rationale behind using a hierarchical model for updating the probability estimates is that each meter is inherently different from every other meter. Some meters are surrounded by physical obstacles, some are on elevated ground, and some are old. New meters have better technology. The meters have different stages of battery life. As the battery level of the meters drop below a certain threshold, the signal transmission range decreases. All of these factors affect the signal transmission behavior of the meters.

Let n be the number of meters and m be the number of street segments (counting repetitions) traversed by the meter reading vehicle. Group the observations Read_OR_Not_{ij} into n buckets, where bucket i contains observations on meter i . Each bucket contains m observations with one from each traversed street segment. The probit model for each group i is called the lower level model for meter i and is given by

$$\Phi^{-1}(p_{ij}) = \hat{\beta}_{i,1} + \hat{\beta}_{i,2} \times \text{Shortest_Distance}_{ij} + \hat{\beta}_{i,3} \times \text{No_of_Pulses}_j.$$

where $\hat{\beta}_{i,k} = \mathbb{E}(\beta_{i,k})$ and $\mathbb{E}()$ denotes the expected value. The lower level unknown parameter vector $\beta_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3})^\top$ for each group i are 3-dimensional vector of coefficients and are used as dependent variables in a multivariate linear model called the higher level model. The data matrices X_i for each group i are $(m \times 3)$ -dimensional and the entries in the first column of each matrix are 1's. The multivariate linear

model is $B = Z\Theta + \Delta$, where Δ is the error term. $B = (\beta_1^\top, \dots, \beta_n^\top)^\top$ is an $(n \times 3)$ -dimensional matrix of the dependent variables. $Z = (z_1^\top, \dots, z_n^\top)^\top$ is an $(n \times 2)$ -dimensional matrix of the independent variables, where $z_i = (1, \text{No_of_Customers}_i)^\top$ for each i is a 2-dimensional vector. The higher level unknown parameter matrix $\Theta = \begin{pmatrix} \theta_{1,1} & \theta_{2,1} & \theta_{3,1} \\ \theta_{1,2} & \theta_{2,2} & \theta_{3,2} \end{pmatrix}$ is a (2×3) -dimensional matrix of coefficients from the multivariate linear model. For each i , the multivariate linear model can be written as $\beta_i = \Theta^\top z_i + \delta_i$, where $\delta_i \sim \text{Multivariate Normal}(0, \Lambda)$ is the error term for the group i . We have to estimate the lower level parameters β_i 's for each meter i and the higher level parameters Θ and Λ . In time period 1, we do not have any prior information about the parameters, so we rely on the information obtained from the data. Therefore, the priors are set to vague priors.

2.8.5 Gibbs Sampling Algorithm for the Hierarchical Probit Model

The Gibbs sampling algorithm is used to estimate the parameters of the hierarchical probit model. The setup for this algorithm is as follows. The prior distributions for the lower level parameters are $p(\beta_i) = \text{Multivariate Normal}(\Theta^\top z_i, \Lambda)$ for each i . The prior distributions for the higher level parameters are $p(\text{vec}[\Theta^\top]) = \text{Multivariate Normal}(M_0, V_0)$ and $p(\Lambda) = \text{Inverse-Wishart}(c_0, D_0)$, where vec denotes the operator that transforms a matrix into a vector by concatenating columns. We have $y_i^* \sim \text{Multivariate Normal}(X_i \beta_i, 1)$ and $y_{ij} = \text{Indicator}(y_{ij}^* > 0)$, where $y_i = (y_{i1}, \dots, y_{im})^\top$ is the dependent variable vector and $y_i^* = (y_{i1}^*, \dots, y_{im}^*)^\top$ is the latent variable vector for group i . Therefore, the posterior distributions for the lower

level parameters are $\pi(\beta_i|y_i^*, \Theta, \Lambda) = \text{Multivariate Normal}((X_i^\top X_i + \Lambda^{-1})^{-1}(X_i^\top y_i^* + \Lambda^{-1}\Theta^\top z_i), (X_i^\top X_i + \Lambda^{-1})^{-1})$ for each i , and the posterior distributions for the higher level parameters are $\pi(\text{vec}[\Theta^\top]|\{y_i^*\}, \Lambda, \{\beta_i\}) = \text{Multivariate Normal}((Z^\top Z \otimes \Lambda^{-1} + V_0^{-1})^{-1}((Z^\top \otimes \Lambda^{-1})\text{vec}[B^\top] + V_0^{-1}M_0), (Z^\top Z \otimes \Lambda^{-1} + V_0^{-1})^{-1})$ and $\pi(\Lambda|\{y_i^*\}, \Theta, \{\beta_i\}) = \text{Inverse-Wishart}(c_0 + n, D_0 + (B - Z\Theta)^\top(B - Z\Theta))$. The Gibbs sampling algorithm for the hierarchical probit model has six steps.

Step 1. Choose starting values for β_i 's, Θ , and Λ .

Step 2. Draw $[y_i^*|y_i, \beta_i]$ for each group i .

Step 3. Draw $[\beta_i|y_i^*, \Theta, \Lambda]$ for each group i .

Step 4. Draw $[\Theta|\{y_i^*\}, \Lambda, \{\beta_i\}]$.

Step 5. Draw $[\Lambda|\{y_i^*\}, \Theta, \{\beta_i\}]$.

Step 6. Repeat Steps 2 to 5.

2.9 Bayesian Updating Results

2.9.1 Logit Model and Probit Model Results

The size of the data set used for estimating the parameters of the logit model and the probit model is $N = 474 \times 7 (= 3,318)$. To verify that our choice of the prior distribution on the unknown parameters for both the logit model and the probit model does not have much effect on the posterior distribution, we also estimate the parameters of both the models using regression. If the parameter values

for the regression and the Bayesian estimation match, this indicates that our choice of the prior for the Bayesian estimation fulfills our requirement of a vague prior. In subsequent time periods, when new data are gathered, the parameters can be updated using the Bayesian updating algorithms.

In the MH random walk algorithm for the logit model, the prior for β is set to a multivariate normal distribution with the mean vector as the zero vector, the variances are 10,000, and the covariances are zero. The first 5,000 samples are considered as the burn-in period and the next 10,000 samples are collected for analysis. The starting value for β is set to the maximum likelihood estimator for the likelihood function of the logit model. The acceptance rate for the new values of β generated from the Markov chain is around 33% (the acceptance rate should be between 30-35% for an optimal combination of exploration and exploitation steps).

In the Gibbs sampling algorithm for the probit model, in order to set the prior for β , B_0 is set to the zero vector and V_0 is set to the diagonal matrix with diagonal entries of 10,000. We collected 10,000 samples for analysis. The starting value for β is set to the zero vector.

In Tables 2.7 and 2.8, we give the logit model results and the probit model results, respectively. The mean and standard deviation of the β_i 's from the MH random walk algorithm and logistic regression, and the Gibbs sampling algorithm and probit regression are presented. Since, for each i , β_i values match for both the logistic regression and the MH random walk algorithm, and both the probit regression and the Gibbs sampling algorithm, our choice of the prior in both algorithms serves the purpose of a vague prior. This indicates that we can perform Bayesian

Coefficient	Logistic Regression	MH Random Walk Algorithm
Intercept (β_1)	2.852 (0.258)	2.860 (0.267)
Shortest_Distance (β_2)	-0.003 (0.0002)	-0.003 (0.0001)
No_of_Pulses (β_3)	0.060 (0.010)	0.060 (0.009)
No_of_Customers (β_4)	-0.017 (0.005)	-0.018 (0.005)

Table 2.7: Logit model results.

Coefficient	Probit Regression	Gibbs Sampling Algorithm
Intercept (β_1)	1.721 (0.147)	1.733 (0.166)
Shortest_Distance (β_2)	-0.002 (0.0001)	-0.002 (0.0001)
No_of_Pulses (β_3)	0.032 (0.005)	0.032 (0.005)
No_of_Customers (β_4)	-0.013 (0.003)	-0.013 (0.003)

Table 2.8: Probit model results.

updating for the logit model and the probit model after receiving new data points instead of using logistic regression or probit regression, respectively.

In Figures 2.7, 2.8, and 2.9, we give the density plots, trace plots, and autocorrelation plots, respectively, for the MH random walk algorithm for the logit model. These plots indicate that the Markov chain for this non-exact algorithm has converged without any significant autocorrelation and the resulting posterior distributions for β_i 's are normal distributions.

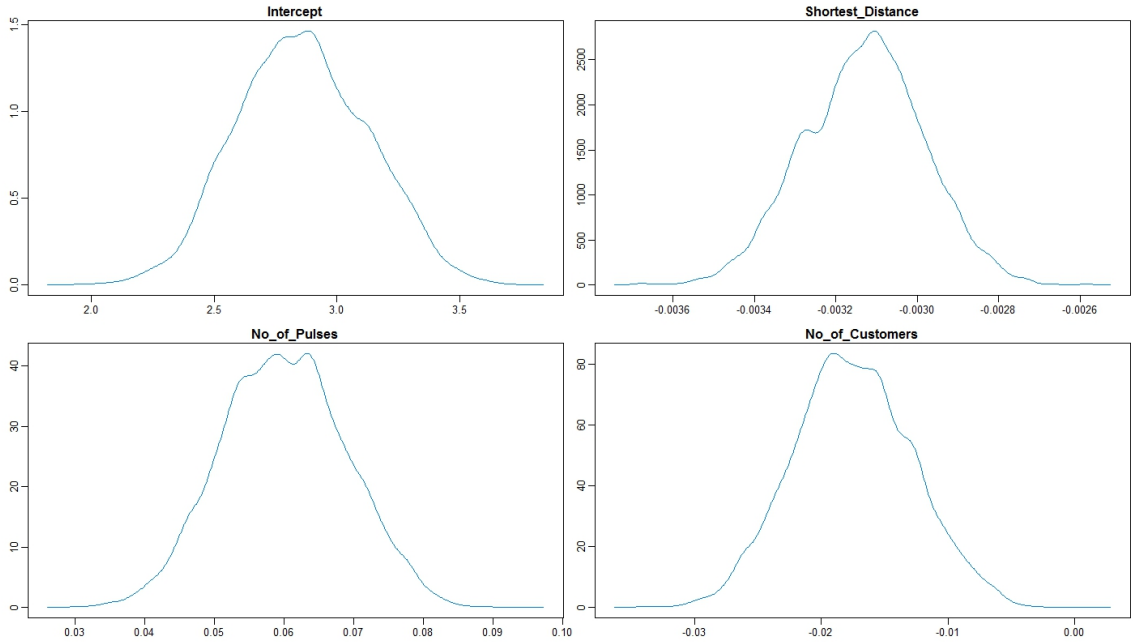


Figure 2.7: Density plots from the MH random walk algorithm for the logit model.

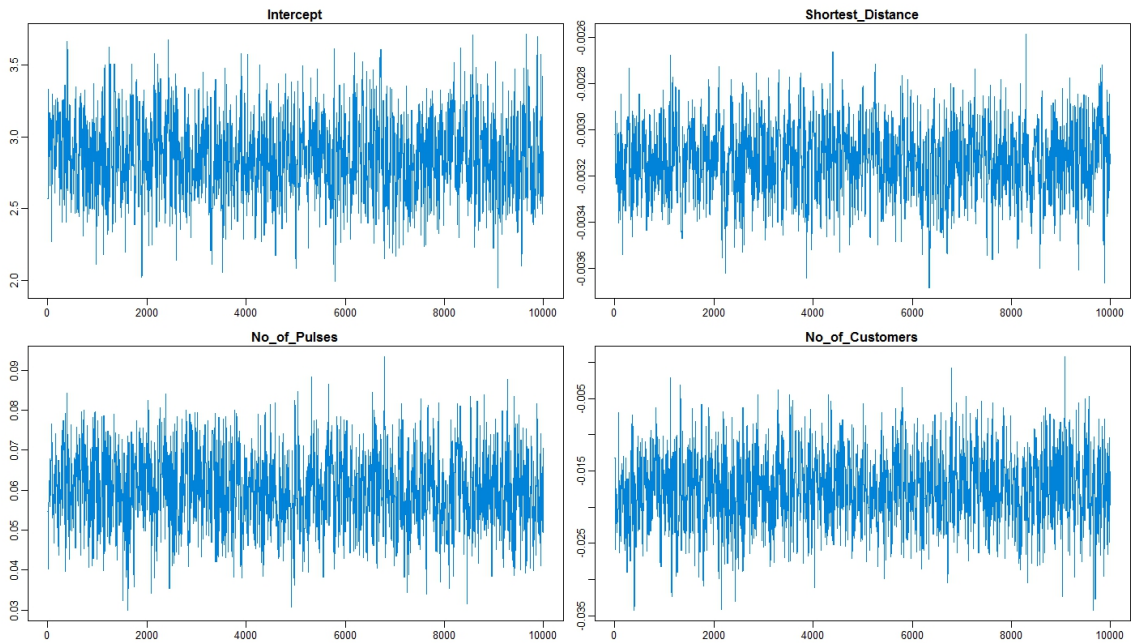


Figure 2.8: Trace plots from the MH random walk algorithm for the logit model.

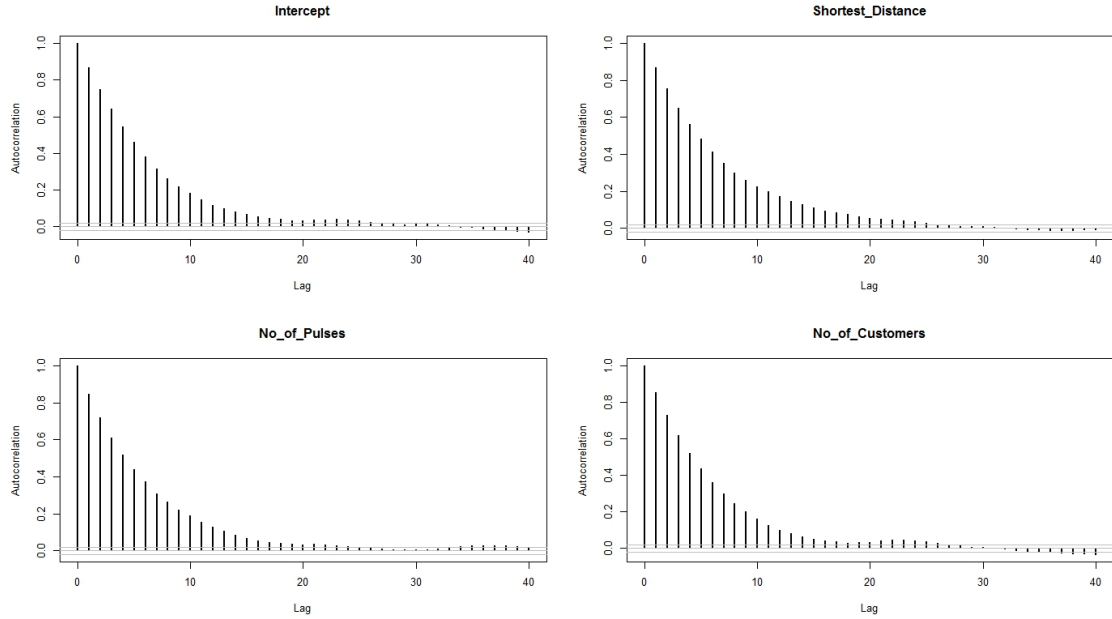


Figure 2.9: Autocorrelation plots from the MH random walk algorithm for the logit model.

2.9.2 Hierarchical Probit Model Results

The size of the data sets used for estimating the lower level parameters in the probit models for each meter i is $m = 7$. The size of the data set used for estimating the higher level parameters in the multivariate linear model is $n = 474$.

In the Gibbs sampling algorithm for the hierarchical probit model, in order to set the priors for β_i 's, Θ , and Λ , M_0 is set to the zero vector and V_0 is set to the diagonal matrix with diagonal entries of 1,000. We set c_0 to seven and D_0 to the diagonal matrix with diagonal entries of three (Rossi et al. 2005, Press 2003). We collected 10,000 samples for analysis. The starting values for the β_i 's and the starting value for Θ are set to the zero vector. The starting value for Λ is sampled from Inverse-Wishart $(c_0 + n, D_0)$.

Coefficient	Gibbs Sampling Algorithm
$\theta_{1,1}$	11.685 (4.008)
$\theta_{2,1}$	-0.010 (0.009)
$\theta_{3,1}$	-0.101 (0.090)
$\theta_{1,2}$	0.013 (0.061)
$\theta_{2,2}$	-0.0002 (0.0002)
$\theta_{3,2}$	0.005 (0.002)

Table 2.9: Hierarchical probit model results for the higher level parameter matrix.

In Table 2.9, we give the hierarchical probit model results for Θ . The mean and standard deviation of the $\theta_{i,j}$'s from the Gibbs sampling algorithm are presented. Using these values, the β_i 's are calculated for each meter i .

In Figure 2.10, we show the histograms of the means of the β_i 's from the Gibbs sampling algorithm for the hierarchical probit model. The three histograms belong to the coefficients of the Intercept ($\hat{\beta}_{i,1}$), the Shortest_Distance ($\hat{\beta}_{i,2}$), and the No_of_Pulses ($\hat{\beta}_{i,3}$), respectively, for the lower level probit model in the hierarchical probit model. The histograms show the variation in the coefficients for different meters that are not captured in the logit model or the probit model. All of the meters have the same equation for estimating the p_{ij} 's for the logit model and the probit model. The hierarchical probit model gives individualized probability predictions p_{ij} for each meter i . Thus, for the hierarchical probit model, each meter has its

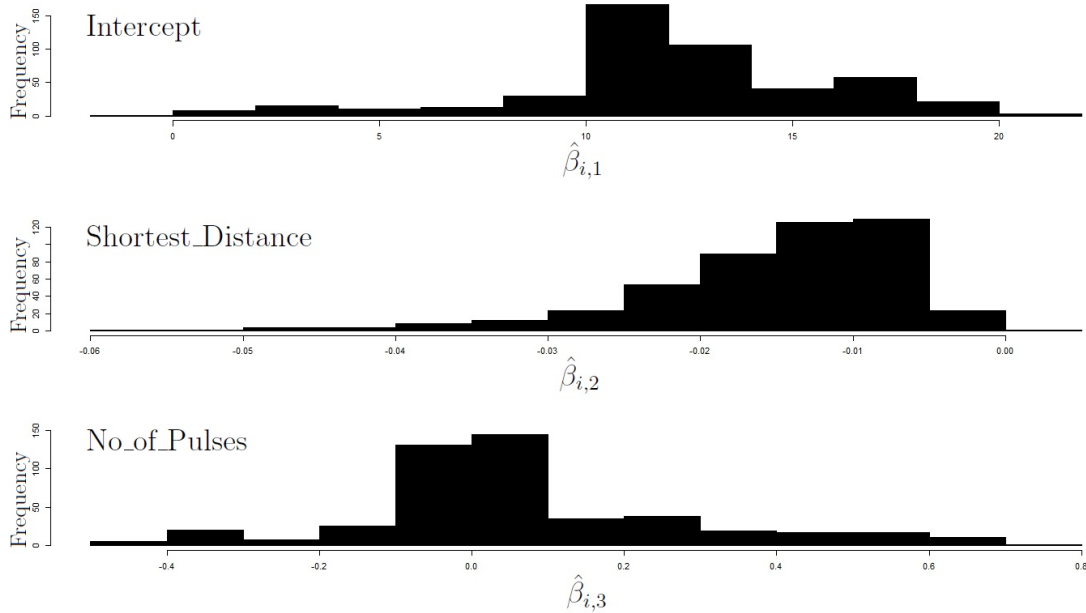


Figure 2.10: Histograms of the means of the lower level parameters from the Gibbs sampling algorithm for the hierarchical probit model.

Total number of meters in the service location layer	6,067
Number of meters in the service location layer that are read	5,720
Total number of read events	337,870
Total number of street segments in the network	1,575
Number of street segments traversed in the route	578
Number of street segments traversed in the route counting repetitions	829
Total number of nodes in the network	1,072
Duration of the route (hours)	6
Time gap between consecutive signal transmission (sec)	3

Table 2.10: Summary of the CNG data set.

unique equation for estimating the p_{ij} 's.

2.10 Description of the CNG Data Set

A larger data set was gathered during the first half of 2016 by ITRON and provided by RouteSmart Technologies. This data set gives meter locations and

reading data serviced by Connecticut Natural Gas (CNG) in Hartford, Connecticut. A summary of the data set is given in Table 2.10.

From the data, we make the following observations. There are many account identifiers in the reading events file that have no corresponding entry in the service location file, i.e., the RFID readers are picking up signals from nearby RFID tags that do not require reading by the utility company. There are a total of 337,870 read events from a route that took six hours and traversed 829 street segments (counting repetitions). Many of those read events are from unwanted RFID tags. The read events data have a many-to-one relationship to a service location account identifier, i.e., some of the meters are read more than once by the meter reading device. Out of the 6,067 meters in our data set, 347 meters are missed. The vehicle location is tracked every second. However, when the read events for a single meter are recorded, they do not occur every second along a street segment that seems to be within range. Rather, there is generally a regular time gap between occurrences of read events for the same meter. This confirms the fact that the signal transmitted by an RFID tag is at regular time intervals and is not occurring continuously. Some meters that are very close to the vehicle route have been missed, probably due to a discontinuous signal. There is a time gap of three seconds between successive signal transmissions from the RFID tags in our data set. Missed reads can also be due to the variability of the range of a meter to transmit a signal.

Coefficient	Logistic Regression	MH Random Walk Algorithm
Intercept (β_1)	-1.242*** (0.010)	-1.242 (0.010)
Shortest_Distance (β_2)	-0.003*** (0.000009)	-0.003 (0.000009)
No_of_Pulses (β_3)	0.019*** (0.00008)	0.019 (0.00008)
No_of_Customers (β_4)	-0.003*** (0.00005)	-0.003 (0.00005)
McFadden's Adjusted R ²	0.223	

***p<0.01

Table 2.11: Logit model results for the CNG data set.

2.11 Bayesian Updating Results for the CNG Data Set

2.11.1 Logit Model and Probit Model Results

The size of the data set used for estimating the parameters of the logit model and the probit model is $N = 6,067 \times 829$ (~ 5 million).

In Table 2.11, we give the logit model results. The mean and standard deviation of the β_i 's from the MH random walk algorithm and logistic regression are presented. Since for each i , β_i values match for both the logistic regression and the MH random walk algorithm, our choice of the prior in the MH random walk algorithm serves the purpose of a vague prior. The McFadden's Adjusted R² value of 0.223 for the logistic regression indicates very good model fit. Also, all coefficients of the regression model are significant at the 1% level, and the signs of the coefficients of the three independent variables are what we expected. This indicates that we

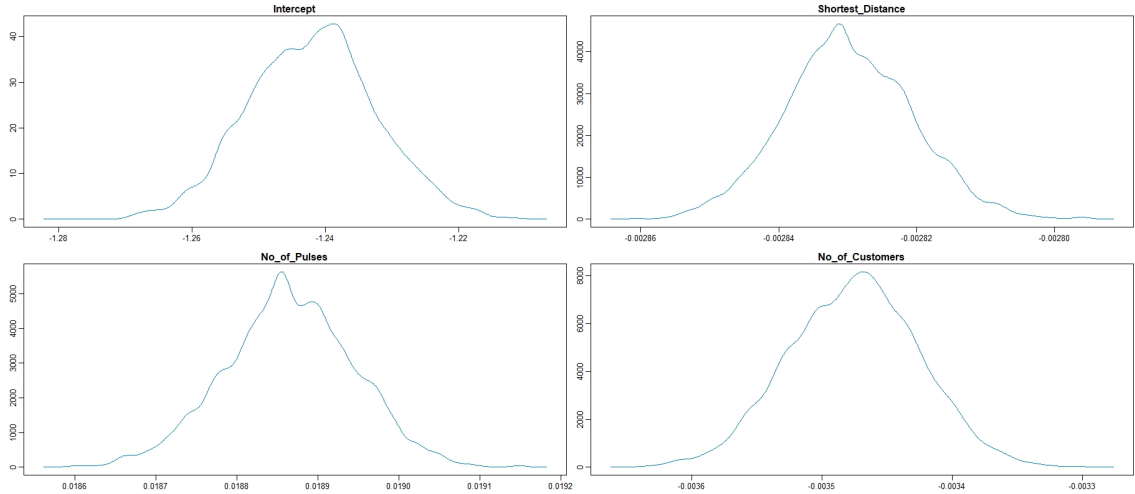


Figure 2.11: Density plots from the MH random walk algorithm for the logit model for the CNG data set.

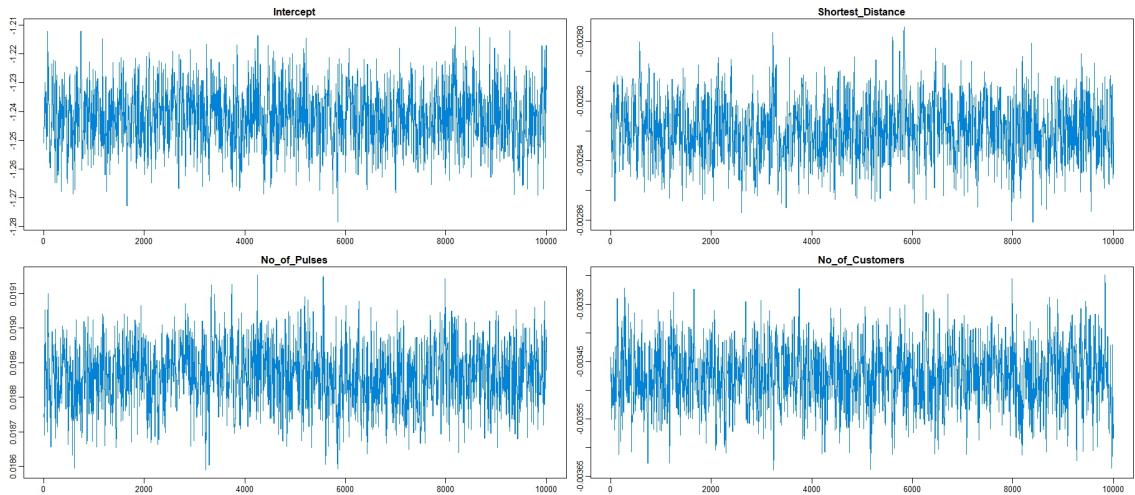


Figure 2.12: Trace plots from the MH random walk algorithm for the logit model for the CNG data set.

can perform Bayesian updating for the logit model after receiving new data points instead of using logistic regression.

In Figures 2.11, 2.12, and 2.13, we give the density plots, trace plots, and autocorrelation plots, respectively, for the MH random walk algorithm for the logit model. These plots indicate that the Markov chain for this non-exact algorithm

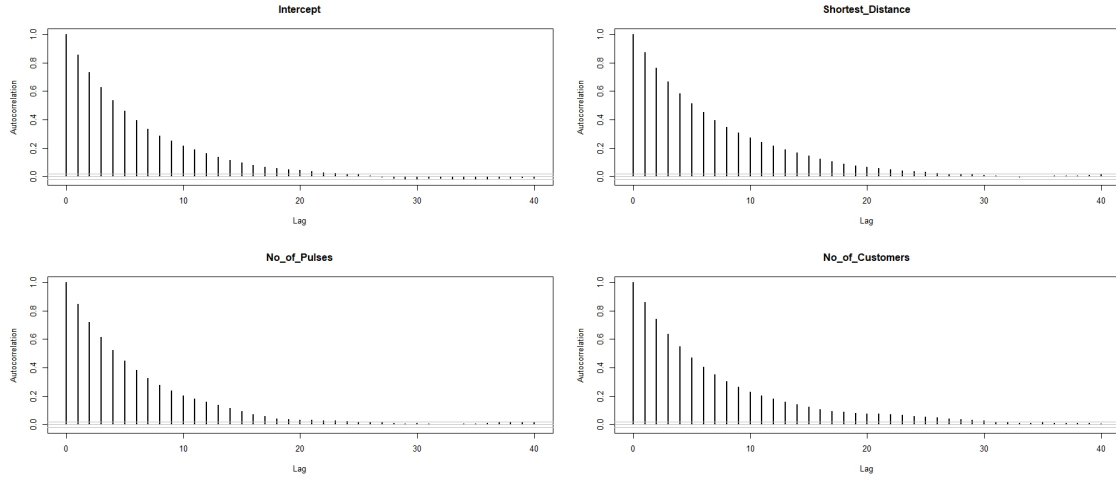


Figure 2.13: Autocorrelation plots from the MH random walk algorithm for the logit model for the CNG data set.

Coefficient	Probit Regression	Gibbs Sampling Algorithm
Intercept (β_1)	-1.000*** (0.004)	-1.024 (0.091)
Shortest_Distance (β_2)	-0.001*** (0.000004)	-0.001 (0.000105)
No_of_Pulses (β_3)	0.007*** (0.00003)	0.007 (0.00069)
No_of_Customers (β_4)	-0.002*** (0.00002)	-0.002 (0.00011)
McFadden's Adjusted R ²	0.200	

***p<0.01

Table 2.12: Probit model results for the CNG data set.

has converged without any significant autocorrelation and the resulting posterior distributions for β_i 's are normal distributions.

In Table 2.12, we give the probit model results. The mean and standard deviation of the β_i 's from the Gibbs sampling algorithm and probit regression are presented. Since for each i , β_i values match for both the probit regression and the

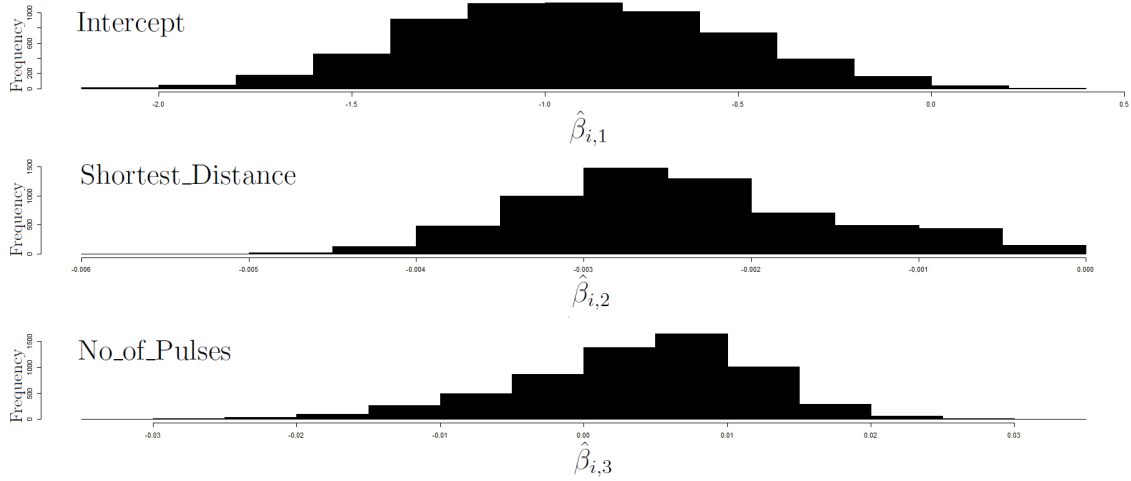


Figure 2.14: Histograms of the means of the lower level parameters from the Gibbs sampling algorithm for the hierarchical probit model for the CNG data set.

Gibbs sampling algorithm, our choice of the prior in the Gibbs sampling algorithm serves the purpose of a vague prior. The McFadden’s Adjusted R^2 value of 0.200 for the probit regression indicates very good model fit. Also, all of the coefficients of the regression model are significant at the 1% level, and the signs of the coefficients of the three independent variables are what we expected. This indicates that we can perform Bayesian updating for the probit model after receiving new data points instead of using probit regression.

2.11.2 Hierarchical Probit Model Results

The size of the data sets used for estimating the lower level parameters in the probit models for each meter i is $m = 829$. The size of the data set used for estimating the higher level parameters in the multivariate linear model is $n = 6,067$.

In Table 2.13, we give the hierarchical probit model results for Θ . The mean

Coefficient	Gibbs Sampling Algorithm
$\theta_{1,1}$	-0.890 (0.241)
$\theta_{2,1}$	-0.002 (0.0009)
$\theta_{3,1}$	0.004 (0.002)
$\theta_{1,2}$	-0.0002 (0.0001)
$\theta_{2,2}$	-0.000003 (0.000004)
$\theta_{3,2}$	0.0000006 (0.000005)

Table 2.13: Hierarchical probit model results for the higher level parameter matrix for the CNG data set.

and standard deviation of the $\theta_{i,j}$'s from the Gibbs sampling algorithm are presented. In Figure 2.14, we show the histograms of the means of the β_i 's from the Gibbs sampling algorithm for the hierarchical probit model. The three histograms belong to the coefficients of the Intercept ($\hat{\beta}_{i,1}$), the Shortest_Distance ($\hat{\beta}_{i,2}$), and the No_of_Pulses ($\hat{\beta}_{i,3}$), respectively, for the lower level probit model in the hierarchical probit model. The histograms show the variation in the coefficients for different meters that are not captured in the logit model or the probit model. The hierarchical probit model gives individualized probability predictions p_{ij} for each meter i . Thus, for the hierarchical probit model, each meter has its unique equation for estimating the p_{ij} 's.

2.12 Discussion of Different Formulations and Computational Experiments of the Stage 1 IP

We discuss different formulations of the Stage 1 IP. The goal is to develop a formulation that is able to obtain the optimal value within a reasonable amount of time for a large data (several thousands of meters and few thousands of street segments).

2.12.1 Alternative Formulations

We have already discussed that the Stage 1 IP can be linearized in the decision variables. The formulation of the linear Stage 1 IP (denoted by L) is given by the following.

$$(L) \quad \min \quad \sum_{j \in E \cup A} c_j x_j \quad (2.13)$$

$$\text{s.t.} \quad \sum_{j \in E \cup A} a_{ij} x_j \geq b_i \quad \forall i \in I \quad (2.14)$$

$$x_j \in \{0, 1\} \quad \forall j \in E \cup A \quad (2.15)$$

where $a_{ij} = -\log(1 - p_{ij}) \geq 0$ for all i, j and $b_i = -\log(1 - L_i) \geq 0$ for all i .

Under an affine transformation of the decision variables in L, the linear Stage 1 IP can be expressed using 0-1 knapsack constraints. Let P denote the transformed linear Stage 1 IP and the formulation is given by the following.

$$(P) \quad \min \quad \sum_{j \in E \cup A} -c_j \bar{x}_j + \sum_{j \in E \cup A} c_j \quad (2.16)$$

$$\text{s.t.} \quad \sum_{j \in E \cup A} a_{ij} \bar{x}_j \leq \bar{b}_i \quad \forall i \in I \quad (2.17)$$

$$\bar{x}_j \in \{0, 1\} \quad \forall j \in E \cup A \quad (2.18)$$

where $\bar{x}_j = 1 - x_j$ for all j and $\bar{b}_i = \sum_{j \in E \cup A} a_{ij} - b_i$ for all i . We, assume that the a_{ij} 's and \bar{b}_i 's are positive.

Different types of valid inequalities are generated from the 0-1 knapsack constraint in P and can be added to the formulation P to create stronger formulations. We discuss these in the next five subsections.

2.12.1.1 Coefficient Round-Down Inequalities

The a_{ij} 's and \bar{b}_i 's are rounded down to generate valid inequalities. For each constraint $\sum_{j \in E \cup A} a_{ij} \bar{x}_j \leq \bar{b}_i$, a new constraint $\sum_{j \in E \cup A} \lfloor a_{ij} \rfloor \bar{x}_j \leq \lfloor \bar{b}_i \rfloor$ is generated. We add these coefficient round-down constraints to P and denote the formulation by PR.

2.12.1.2 Increasing Coefficient Extended Cover Inequalities

Without loss of generality, we assume that, for each $i \in I$, $a_{i1} \leq \dots \leq a_{i|E \cup A|}$. For each constraint $\sum_{j \in E \cup A} a_{ij} \bar{x}_j \leq \bar{b}_i$, a new constraint $\sum_{j=1}^k \bar{x}_j \leq k-1$ is generated where k is such that $\sum_{j=1}^k a_{ij} > \bar{b}_i$ but $\sum_{j=1}^{k-1} a_{ij} \leq \bar{b}_i$. These are then extended to

form extended cover inequalities. We add these increasing coefficient extended cover inequalities to P and denote the formulation by PDI.

To illustrate, consider the 0-1 knapsack constraint given by $1x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 + 7x_7 + 8x_8 + 9x_9 + 10x_{10} \leq 20$. The generated valid inequality is $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 5$ which is then extended to $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \leq 5$.

2.12.1.3 Decreasing Coefficient Extended Cover Inequalities

Without loss of generality, we assume that, for each $i \in I$, $a_{i|E \cup A|} \leq \dots \leq a_{i1}$. For each constraint $\sum_{j \in E \cup A} a_{ij} \bar{x}_j \leq \bar{b}_i$, a new constraint $\sum_{j=1}^k \bar{x}_j \leq k-1$ is generated where k is such that $\sum_{j=1}^k a_{ij} > \bar{b}_i$ but $\sum_{j=1}^{k-1} a_{ij} \leq \bar{b}_i$. These are extended cover inequalities. We add these decreasing coefficient extended cover inequalities to P and denote the formulation by PCI.

To illustrate, consider the 0-1 knapsack constraint given by $10x_1 + 9x_2 + 8x_3 + 7x_4 + 6x_5 + 5x_6 + 4x_7 + 3x_8 + 2x_9 + 1x_{10} \leq 20$. The generated valid inequality is $x_1 + x_2 + x_3 \leq 2$.

2.12.1.4 Middle Coefficient Extended Cover Inequalities

Without loss of generality, we assume that, for each $i \in I$, $a_{i(|E \cup A|-1)} \leq a_{i(|E \cup A|-3)} \leq \dots \leq a_{i1} \leq a_{i2} \leq a_{i4} \leq \dots \leq a_{i|E \cup A|}$, when $|E \cup A|$ is even, and $a_{i(|E \cup A|-1)} \leq a_{i(|E \cup A|-3)} \leq \dots \leq a_{i2} \leq a_{i1} \leq a_{i3} \leq \dots \leq a_{i|E \cup A|}$, when $|E \cup A|$ is odd. For each constraint $\sum_{j \in E \cup A} a_{ij} \bar{x}_j \leq \bar{b}_i$, a new constraint $\sum_{j=1}^k \bar{x}_j \leq k-1$ is gener-

ated where k is such that $\sum_{j=1}^k a_{ij} > \bar{b}_i$ but $\sum_{j=1}^{k-1} a_{ij} \leq \bar{b}_i$. These are then extended to form extended cover inequalities. We add these middle coefficient extended cover inequalities to P and denote the formulation by PMI.

To illustrate, consider the 0-1 knapsack constraint given by $5x_1 + 6x_2 + 4x_3 + 7x_4 + 3x_5 + 8x_6 + 2x_7 + 9x_8 + 1x_9 + 10x_{10} \leq 20$. The generated valid inequality is $x_1 + x_2 + x_3 + x_4 \leq 3$ which is then extended to $x_1 + x_2 + x_3 + x_4 + x_6 + x_8 + x_{10} \leq 3$.

2.12.1.5 Extreme Coefficient Extended Cover Inequalities

Without loss of generality, we assume that, for each $i \in I$, $a_{i2} \leq a_{i4} \leq \dots \leq a_{i|E \cup A|} \leq a_{i(|E \cup A|-1)} \leq a_{i(|E \cup A|-3)} \leq \dots \leq a_{i1}$, when $|E \cup A|$ is even, and $a_{i2} \leq a_{i4} \leq \dots \leq a_{i(|E \cup A|-1)} \leq a_{i|E \cup A|} \leq a_{i(|E \cup A|-2)} \leq \dots \leq a_{i1}$, when $|E \cup A|$ is odd. For each constraint $\sum_{j \in E \cup A} a_{ij} \bar{x}_j \leq \bar{b}_i$, a new constraint $\sum_{j=1}^k \bar{x}_j \leq k-1$ is generated where k is such that $\sum_{j=1}^k a_{ij} > \bar{b}_i$ but $\sum_{j=1}^{k-1} a_{ij} \leq \bar{b}_i$. These are extended cover inequalities. We add these extreme coefficient extended cover inequalities to P and denote the formulation by PEI.

To illustrate, consider the 0-1 knapsack constraint given by $10x_1 + 1x_2 + 9x_3 + 2x_4 + 8x_5 + 3x_6 + 7x_7 + 4x_8 + 6x_9 + 5x_{10} \leq 20$. The generated valid inequality is $x_1 + x_2 + x_3 + x_4 \leq 3$.

2.12.1.6 All Inequalities

In the formulation P, we include increasing coefficient extended cover inequalities, decreasing coefficient extended cover inequalities, middle coefficient extended

cover inequalities, extreme coefficient extended cover inequalities, and coefficient round-down inequalities and denote this formulation by PI. For each $i \in I$, five new constraints are added to P to form PI.

2.12.2 Computational Results

We perform computational experiments that are designed to examine the performance on eight models (L, P, PR, PDI, PCI, PMI, PEI, PI) and how performance varies with respect to the size of the data set. Understanding the performance of the various formulations of the Stage 1 IP on a large data set is important because our CNG data set has 6,067 meters and 1,575 street segments.

We test the eight models for different values of the number of meters ($|I|$) and the number of street segments ($|E \cup A|$), where the range of both values is 10 to 2000. The probabilities (p_{ij}) are generated randomly from a Uniform (0, 1) distribution. The costs (c_j) of the street segments are generated from three different distributions. In the first case, costs are generated randomly from a Uniform (0, 1) distribution, and then the values are multiplied by 100. In the second case, unit costs are considered for all street segments. In the third case, costs are generated randomly from a truncated Normal (0, 1) distribution with the left tail truncated at 0 and the right tail truncated at 1, and then the values are multiplied by 100. We use R software version 3.3.1 to generate the data for the computational experiments and Gurobi version 7.0 to solve the models. We use an i7 CPU with 32 GB RAM and a one-hour time limit. If an optimal value (denoted by v) is not found within

the time limit, the best feasible solution value (denoted by V) and the best available bound (denoted by B) are reported. We ensure that the generated data are feasible for all the eight models and that all entries in the constraint matrix are non-zero (the p_{ij} 's in the real world will be non-zero). The constraint matrices have dimension $|I| \times |E \cup A|$ for models L and P, $2 \times |I| \times |E \cup A|$ for models PR, PDI, PCI, PMI and PEI, and $6 \times |I| \times |E \cup A|$ for model PI.

The comparison of the performance for the eight models using different values of $|I|$ and $|E \cup A|$ based on the three cost structures are given in Tables 2.14 to 2.23. For all these analyses, the values of the specified likelihood L_i is taken to be 0.95 for all i . In Tables 2.14 to 2.23, the branch column gives the number of branches or nodes created and the time column gives the running time in seconds. The v(LP) column gives the LP relaxation values.

Table 2.14 gives the results for 20 meters and 10 street segments. All eight models are solved to optimality for all three cost structures at the root node within one-hundredth of a second. LP relaxation values indicate that formulations for PCI and PI are stronger than the other six models for the uniform cost structure and the normal cost structure. Formulations for PDI, PMI, and PI are stronger than the other five models for the unit cost structure.

Table 2.15 gives the results for 10 meters and 20 street segments. All eight models are solved to optimality for all three cost structures close to one-hundredth of a second. For the unit cost structure, all eight models are solved at the root node. LP relaxation values indicate that formulations for PCI and PI are stronger than the other six models for all three cost structures.

Model	Random Uniform Cost			Unit Cost			Random Normal Cost		
	Branch	Time	v(LP)	Branch	Time	v(LP)	Branch	Time	v(LP)
	v = 210.3068			v = 5			v = 185.8380		
L	0	0.000	170.9240	0	0.016	3.8705	0	0.016	154.1182
P	0	0.001	170.9240	0	0.019	3.8705	0	0.000	154.1182
PR	0	0.016	170.9240	0	0.016	3.8705	0	0.016	154.1182
PDI	0	0.008	170.9240	0	0.009	4.0000	0	0.008	154.1182
PCI	0	0.004	175.3579	0	0.016	3.8800	0	0.000	156.4817
PMI	0	0.017	170.9240	0	0.000	4.0000	0	0.000	154.1182
PEI	0	0.006	170.9240	0	0.016	3.8705	0	0.016	154.1182
PI	0	0.016	175.3579	0	0.016	4.0000	0	0.000	156.4817

Table 2.14: Results for 20 meters and 10 street segments.

Table 2.16 gives the results for 100 meters and 20 street segments. All eight models are solved to optimality for all three cost structures in nearly one-tenth of a second. For the unit cost structure, all eight models required a larger number of nodes than the number of nodes in the other two cost structures. LP relaxation values indicate that formulations for PCI and PI are stronger than the other six

Model	Random Uniform Cost			Unit Cost			Random Normal Cost		
	Branch	Time	v(LP)	Branch	Time	v(LP)	Branch	Time	v(LP)
	v = 90.9012			v = 4			v = 78.6606		
L	9	0.047	65.3144	0	0.016	2.9883	8	0.031	56.4202
P	8	0.010	65.3144	0	0.000	2.9883	8	0.016	56.4202
PR	8	0.005	65.3144	0	0.000	2.9883	8	0.016	56.4202
PDI	8	0.010	65.3144	0	0.005	2.9883	8	0.031	56.4202
PCI	7	0.022	65.7364	0	0.016	3.0370	7	0.019	56.8099
PMI	8	0.016	65.3144	0	0.000	2.9883	8	0.031	56.4202
PEI	8	0.008	65.3144	0	0.000	2.9883	8	0.031	56.4202
PI	6	0.016	65.7364	0	0.000	3.0370	6	0.031	56.8099

Table 2.15: Results for 10 meters and 20 street segments.

models for the uniform cost structure and the normal cost structure.

Table 2.17 gives the results for 20 meters and 100 street segments. All eight models are solved to optimality for all three cost structures within one-hundredth of a second. For the unit cost structure, all eight models required a considerably larger more number of nodes than the number of nodes in the other two cost structures.

Model	Random Uniform Cost			Unit Cost			Random Normal Cost		
	Branch	Time	v(LP)	Branch	Time	v(LP)	Branch	Time	v(LP)
L	19	0.190	142.3955	80	0.105	4.0408	18	0.111	125.7916
P	11	0.134	142.3955	57	0.098	4.0408	9	0.117	125.7916
PR	11	0.134	142.3955	57	0.077	4.0408	9	0.116	125.7916
PDI	11	0.151	142.3955	57	0.081	4.0408	9	0.112	125.7916
PCI	11	0.289	142.9973	80	0.097	4.0408	12	0.050	126.2463
PMI	14	0.117	142.3955	57	0.068	4.0408	16	0.157	125.7916
PEI	13	0.195	142.3955	81	0.085	4.0408	10	0.050	125.7916
PI	13	0.379	142.9973	80	0.113	4.0408	12	0.062	126.2463

Table 2.16: Results for 100 meters and 20 street segments.

LP relaxation values indicate that all eight formulations are equivalent for all three cost structures.

Table 2.18 gives the results for 200 meters and 100 street segments. All eight models are solved to optimality for the uniform cost structure and the normal cost structure within one second, and for the unit cost structure within 125 seconds. For

Model	Random Uniform Cost			Unit Cost			Random Normal Cost		
	Branch	Time	v(LP)	Branch	Time	v(LP)	Branch	Time	v(LP)
	$v = 42.0435$			$v = 4$			$v = 36.0238$		
L	17	0.031	30.8630	244	0.081	2.2912	21	0.043	26.4746
P	16	0.047	30.8630	212	0.063	2.2912	16	0.031	26.4746
PR	16	0.031	30.8630	212	0.078	2.2912	16	0.031	26.4746
PDI	16	0.043	30.8630	212	0.067	2.2912	16	0.023	26.4746
PCI	18	0.050	30.8630	235	0.093	2.2912	18	0.038	26.4746
PMI	16	0.050	30.8630	212	0.085	2.2912	16	0.031	26.4746
PEI	16	0.028	30.8630	231	0.083	2.2912	16	0.047	26.4746
PI	18	0.047	30.8630	190	0.109	2.2912	18	0.047	26.4746

Table 2.17: Results for 20 meters and 100 street segments.

the unit cost structure, all eight models required a larger number of nodes than the number of nodes in the other two cost structures. LP relaxation values indicate that all eight formulations are equivalent for all three cost structures.

Table 2.19 gives the results for 100 meters and 200 street segments. All eight models are solved to optimality for the uniform cost structure and the normal cost

Model	Random Uniform Cost			Unit Cost			Random Normal Cost		
	Branch	Time	v(LP)	Branch	Time	v(LP)	Branch	Time	v(LP)
			$v = 31.1048$			$v = 5$			$v = 26.6314$
L	71	0.172	21.4948	28393	56.458	2.7859	71	0.158	18.4024
P	54	0.160	21.4948	28031	109.678	2.7859	70	0.181	18.4024
PR	54	0.173	21.4948	28031	109.976	2.7859	70	0.184	18.4024
PDI	54	0.211	21.4948	9234	7.190	2.7859	70	0.234	18.4024
PCI	62	0.382	21.4948	27823	159.750	2.7859	65	0.414	18.4024
PMI	54	0.219	21.4948	9234	7.364	2.7859	70	0.211	18.4024
PEI	54	0.275	21.4948	8598	11.496	2.7859	57	0.266	18.4024
PI	62	0.524	21.4948	8876	11.941	2.7859	65	0.525	18.4024

Table 2.19: Results for 100 meters and 200 street segments.

models are solved to optimality for the uniform cost structure and the normal cost structure within six seconds. For the unit cost structure, none of the models are solved to optimality within the one-hour time limit. LP relaxation values indicate that all eight formulations are equivalent for all three cost structures.

Table 2.21 gives the analysis results for 200 meters and 1000 street segments.

Random Uniform Cost		Unit Cost		Random Normal Cost		
$v = 50.0808$		$V = 8, B = 5$		$v = 42.8871$		
Model	Branch	Time	v(LP)	Branch	Time	v(LP)
L	509	2.359	36.4209	573516	3600.025	3.2982
P	416	2.451	36.4209	525333	3600.025	3.2982
PR	416	2.538	36.4209	532719	3600.029	3.2982
PDI	416	2.809	36.4209	540225	3600.039	3.2982
PCI	429	5.370	36.4209	343069	3600.051	3.2982
PMI	416	2.945	36.4209	532268	3600.043	3.2982
PEI	416	5.386	36.4209	327526	3600.058	3.2982
PI	429	5.961	36.4209	307213	3600.118	3.2982

Table 2.20: Results for 1000 meters and 200 street segments.

All eight models are solved to optimality for the uniform cost structure and the normal cost structure within five seconds. For the unit cost structure, none of the models are solved to optimality within the one-hour time limit. LP relaxation values indicate that all eight formulations are equivalent for all three cost structures.

Table 2.22 gives the results for 2000 meters and 1000 street segments. All eight

Model	Random Uniform Cost				Unit Cost				Random Normal Cost					
	Branch	Time	v(LP)	Branch	Branch	Time	v(LP)	Branch	Branch	Time	v(LP)	Branch	Time	v(LP)
					V = 6, B = 4				v = 4.3551					
					v = 5.0900									
L	78	0.688	3.7080	1012646	3600.026	2.7328	2.7328	78	0.762	3.1727				
P	75	0.696	3.7080	732113	3600.029	2.7328	2.7328	75	0.723	3.1727				
PR	75	0.720	3.7080	721351	3600.025	2.7328	2.7328	75	0.708	3.1727				
PDI	75	0.898	3.7080	680384	3600.037	2.7328	2.7328	75	0.873	3.1727				
PCI	75	2.055	3.7080	1389916	3600.044	2.7328	2.7328	75	2.194	3.1727				
PMI	75	0.865	3.7080	715694	3600.034	2.7328	2.7328	75	0.905	3.1727				
PEI	75	1.247	3.7080	938173	3600.043	2.7328	2.7328	75	1.271	3.1727				
PI	75	4.104	3.7080	910063	3600.073	2.7328	2.7328	75	3.903	3.1727				

Table 2.21: Results for 200 meters and 1000 street segments.

models are solved to optimality for the uniform cost structure and the normal cost structure within 105 seconds. For the unit cost structure, none of the models are solved to optimality within the one-hour time limit. LP relaxation values indicate that all eight formulations are equivalent for all three cost structures.

Table 2.23 gives the results for 1000 meters and 2000 street segments. All

Random Uniform Cost		Unit Cost		Random Normal Cost					
v = 7.4823		V = 9, B = 4		v = 6.4021					
Model	Branch	Time	v(LP)	Branch	Time	v(LP)	Branch	Time	v(LP)
L	306	10.327	5.4391	4066	3600.063	3.0643	306	10.564	4.6540
P	292	11.130	5.4391	4506	3600.139	3.0643	292	11.404	4.6540
PR	292	11.613	5.4391	4505	3600.604	3.0643	292	11.720	4.6540
PDI	292	16.794	5.4391	4384	3600.106	3.0643	292	15.500	4.6540
PCI	455	41.847	5.4391	4156	3601.407	3.0643	450	39.208	4.6540
PMI	292	16.700	5.4391	4349	3600.131	3.0643	292	15.330	4.6540
PEI	292	42.715	5.4391	4255	3600.326	3.0643	292	40.500	4.6540
PI	404	102.281	5.4391	4195	3601.171	3.0643	403	104.360	4.6540

Table 2.22: Results for 2000 meters and 1000 street segments.

eight models are solved to optimality for the uniform cost structure and the normal cost structure within 66 seconds. For the unit cost structure, none of the models are solved to optimality within the one-hour time limit. LP relaxation values indicate that all eight formulations are equivalent for all three cost structures.

Model	Random Uniform Cost				Unit Cost				Random Normal Cost				
	Branch	Time	v(LP)	Branch	Branch	Time	v(LP)	Branch	Branch	Time	v(LP)	Branch	Time
	$v = 1.9747$				$V = 8, B = 4$				$v = 1.6896$				
L	51	6.102	1.4088	7026	3600.249	2.9371	51	6.201	1.2054				
P	33	6.229	1.4088	9345	3605.910	2.9371	33	6.630	1.2054				
PR	33	6.441	1.4088	9345	3605.748	2.9371	33	6.805	1.2054				
PDI	33	8.728	1.4088	9511	3604.998	2.9371	33	8.950	1.2054				
PCI	47	26.887	1.4088	9341	3601.234	2.9371	47	27.510	1.2054				
PMI	33	8.734	1.4088	9510	3606.107	2.9371	33	9.178	1.2054				
PEI	33	11.500	1.4088	4731	3600.849	2.9371	33	11.580	1.2054				
PI	47	60.986	1.4088	4431	3600.431	2.9371	47	65.302	1.2054				

Table 2.23: Results for 1000 meters and 2000 street segments.

2.12.3 Observations from the Computational Experiments

All eight formulations performed similarly for the three respective cost structures in terms of the number of nodes created and the LP relaxation value. The formulations L and P had the smallest running times. Neither the affine transfor-

mation of the decision variables in L into 0-1 knapsack constraints in P, nor the addition of coefficient round-down inequalities and various other extended cover inequalities to P made the formulation stronger. Most likely, this is due to the fact that Gurobi already takes into account similar transformations and valid inequalities during the pre-processing stages while solving the models. The addition of the valid inequalities to P increased the running times. Based on these observations, we should use formulation L (linear Stage 1 IP) to solve the Stage 1 IP when using large data sets.

The uniform cost structure and the normal cost structure performed similarly and were easy to solve in terms of the running times and the number of nodes created. For each pair of $|I|$ and $|E \cup A|$, the number of branches created and the running times are significantly larger for the unit cost structure compared to the other two cost structures. With larger values of $|I|$ and $|E \cup A|$, none of the eight models are solved to optimality within the one-hour time limit. This is probably due to the inherent symmetry in the Stage 1 IP for the unit cost structure, since each street segment has the same weight in the objective function.

2.12.4 Other Insights from the Computational Experiments

In Table 2.24, we summarize the performance for formulation L (linear Stage 1 IP), based on the results from Tables 2.14 to 2.23, for the three cost structures with respect to the number of branches created, the running times (in seconds), and the optimality gap ($100(\frac{V-B}{B})$). The value of the specified likelihood L_i is set at 0.95 for

	$ I $	$ E \cup A $	Uniform Cost				Unit Cost				Normal Cost			
			Branch	Time	Gap	Branch	Time	Gap	Branch	Time	Gap	Branch	Time	Gap
	20	10	0	0.000	Optimal	0	0.016	Optimal	0	0.016	Optimal	0	0.016	Optimal
	10	20	9	0.047	Optimal	0	0.016	Optimal	8	0.031	Optimal	8	0.031	Optimal
	100	20	19	0.190	Optimal	80	0.105	Optimal	18	0.111	Optimal	18	0.111	Optimal
	20	100	17	0.031	Optimal	244	0.081	Optimal	21	0.043	Optimal	21	0.043	Optimal
	200	100	127	0.224	Optimal	70776	68.857	Optimal	117	0.237	Optimal	117	0.237	Optimal
	100	200	71	0.172	Optimal	28393	56.458	Optimal	71	0.158	Optimal	71	0.158	Optimal
	1000	200	509	2.359	Optimal	573516	3600.025	60%	657	2.269	Optimal	657	2.269	Optimal
	200	1000	78	0.688	Optimal	1012646	3600.026	50%	78	0.762	Optimal	78	0.762	Optimal
	2000	1000	306	10.327	Optimal	4066	3600.063	125%	306	10.564	Optimal	306	10.564	Optimal
	1000	2000	51	6.102	Optimal	7026	3600.249	100%	51	6.201	Optimal	51	6.201	Optimal

Table 2.24: Comparison of the linear Stage 1 IP performance for the three different cost structures.

all i . For each pair of $|I|$ and $|E \cup A|$, the linear Stage 1 IP had similar performance for both the uniform cost structure and the normal cost structure in terms of the

number of branches created, the running times, and the optimality gap. All the models are solved to optimality with the running times ranging from one-hundredth of a second for $|I| \times |E \cup A|$ value of 200 to 10 seconds for $|I| \times |E \cup A|$ value of 2,000,000. The number of branches created ranges from less than 10 to around 500. For each pair of $|I|$ and $|E \cup A|$, the number of branches created and the running times are significantly larger for the unit cost structure compared to the other two cost structures. This is probably due to the inherent symmetry in the linear Stage 1 IP for the unit cost structure, since each street segment has the same weight in the objective function. The models are not solved to optimality within the one-hour time limit for $|I| \times |E \cup A|$ values greater than 20,000. The optimality gap is more than 50% and more than 100% for $|I| \times |E \cup A|$ value of 200,000 and 2,000,000, respectively.

In Table 2.25, for each of the three cost structures we show the comparison of the objective value for the linear Stage 1 IP for L_i values of 0.95 and 0.75 for all i . If the linear Stage 1 IP is solved to optimality, the objective value is smaller for L_i values of 0.75 compared to 0.95 for each pair of $|I|$ and $|E \cup A|$ and for each of the three cost structures. For the unit cost structure and for $|I| \times |E \cup A|$ values greater than 20,000, the interval $[B, V]$ is tighter and shifted towards zero for L_i values of 0.75 compared to 0.95. This demonstrates the fact that the smaller the specified likelihood values for reading meters, the smaller is the Stage 1 IP objective value.

The Stage 1 IP selects the street segments with the lowest total cost (length) that guarantee the specified likelihood of reading the meters. The Stage 2 IP adds deadhead segments with the lowest total cost (length) to complete the full route,

$ I $	$ E \cup A $	Uniform Cost		Unit Cost		Normal Cost	
		0.95	0.75	0.95	0.75	0.95	0.75
20	10	210.3068	90.1101	5	3	185.8380	77.9464
10	20	90.9012	44.0025	4	2	78.6606	37.7965
100	20	185.6485	101.2878	6	4	161.9885	88.9582
20	100	42.0435	23.7557	4	3	36.0238	20.3450
200	100	60.6152	27.9134	6	4	51.9349	23.9070
100	200	31.1048	16.9587	5	3	26.6314	14.5172
1000	200	50.0808	23.3395	B = 5, V = 8	B = 3, V = 5	42.8871	19.9795
200	1000	5.0900	2.7574	B = 4, V = 6	B = 3, V = 4	4.3551	2.3593
2000	1000	7.4823	3.8587	B = 4, V = 9	B = 2, V = 6	6.4021	3.3016
1000	2000	1.9747	0.9722	B = 4, V = 8	B = 2, V = 5	1.6896	0.8319

Table 2.25: Comparison of the linear Stage 1 IP objective value for specified likelihood values of 0.95 and 0.75 for all meters.

starting and ending at the depot. It might be the case that the street segments selected by the Stage 1 IP are not the best selections for the full route. The street segments selected by the Stage 1 IP might be farther away from each other. This may lead to a larger total length for the full route. To help prevent this, an alternate

objective function for the Stage 1 IP may be minimizing the total number of street segments (unit cost structure), i.e., $c_j = 1$ for all j . For the unit cost structure, our computational experiments showed that the linear Stage 1 IP with a one-hour running time had an optimality gap of more than 100% for the $|I| \times |E \cup A|$ value of 2,000,000. Since the $|I| \times |E \cup A|$ value for our data set ($6,067 \times 1,575$) is an order of magnitude larger, it is not possible to arrive at Stage 1 IP solutions with smaller optimality gaps in a reasonable amount of time. So, rather than using the unit cost structure as the only objective function in the Stage 1 IP, it might be useful to explore the option of having the unit cost structure added to the Stage 1 IP as the second objective function. We compare the routes generated using the single-objective Stage 1 IP (discussed in Section 2.4) and the bi-objective Stage 1 IP in the simulation experiments conducted in Section 2.14 with our data set. The bi-objective Stage 1 IP is given on the next page.

The objective function (2.19) is the same as the objective function (2.1) in the single-objective Stage 1 IP. The objective function (2.20) minimizes the total number of street segments selected. Constraints (2.21) and (2.22) are the same as constraints (2.2) and (2.3), respectively, in the single-objective Stage 1 IP. A lexicographic approach (establishing a pre-defined ordering between the competing objective functions) is used to solve the bi-objective Stage 1 IP. First, the single-objective Stage 1 IP is solved with the objective function (2.19). Then, the value of the objective function (2.20) is improved without allowing the value of the objective function (2.19) to increase.

$$\text{(Bi-objective Stage 1 IP)} \quad \min \quad \sum_{j \in E \cup A} c_j x_j \quad (2.19)$$

$$\min \quad \sum_{j \in E \cup A} x_j \quad (2.20)$$

$$\text{s.t.} \quad \prod_{j \in E \cup A} (1 - p_{ij})^{x_j} \leq (1 - L_i) \quad \forall i \in I \quad (2.21)$$

$$x_j \in \{0, 1\} \quad \forall j \in E \cup A \quad (2.22)$$

2.13 Heuristics for the Stage 2 IP

[Frederickson et al. \(1978\)](#) showed that the mixed rural postman problem is NP-complete. A mixed rural postman problem with a particular node (depot) that is a part of the route is also NP-complete. This is the problem we solve in the Stage 2 IP. Therefore, for any realistic large data sets, similar to the size of our CNG data set (6,067 meters, 1,575 street segments, and 1,072 nodes), any exact method of solving the Stage 2 IP potentially has long running times with solutions that have large optimality gaps. We discuss a fast metaheuristic to generate near-optimal solutions in a short amount of time.

2.13.1 Route Generator

Given a set of required street segments, the goal is to find the shortest route for the meter reading vehicle that starts and ends at the depot and traverses all required street segments. By assuming that the vehicle always takes the shortest

Reverse	Change the direction in which a required edge is traversed by the vehicle. Required arcs remain unaffected.
Relocate	Remove a required street segment from the current solution and insert it at a different location in the route.
2-opt	Reverse the order in which the vehicle visits the street segments in a subsequence of required street segments.

Table 2.26: Local search operators in the variable neighborhood descent metaheuristic.

path between any two required street segments, the aim of the route generator is reduced to finding an optimal permutation of all required street segments and, in the case of edges, the direction in which they should be traversed. This shortest path between each pair of nodes is computed using Dijkstra’s algorithm ([Dijkstra 1959](#)).

The route generator has two phases, the constructive phase and the improvement phase. During the constructive phase, the route starts from the depot node and, based on the nearest-neighbor principle, consecutively visits the closest required street segments. Since the edges can be traversed in both directions, the node (among the two nodes representing an edge) closest to the preceding node in the route is visited first. After a complete initial route is constructed, the improvement phase improves the solution using three different local search operators embedded in a variable neighborhood descent metaheuristic. The three local search operators are briefly described in [Table 2.26](#). The variable neighborhood descent metaheuristic framework is considered very effective for solving routing problems ([Defryn and Sörensen 2017](#), [Hansen et al. 2017](#), [Wassan et al. 2017](#)).

2.13.2 Route Trimmer

The full route obtained from the route generator contains deadhead street segments in addition to the required street segments. This leads to the fact that the probabilities of successfully reading the meters are more than the specified likelihoods (L_i), which is already guaranteed by the required street segments. Although this will further reduce the chance of missing meters, it comes at a cost of increased route length. To account for this, the route trimmer aims at decreasing the total route length while still assuring feasibility, i.e., the probability of successfully reading each meter i from the full route is at least the specified likelihood L_i .

The route trimmer makes use of a remove and repair procedure to decrease the number of required street segments and, therefore, the total route length. In Table 2.27, we give the algorithm for the remove and repair procedure of the route trimmer. Lines 1-4 are the initialization steps. *HaveToBeRequired* is the subset of the required street segments that should be a part of the route. *CandidateList* is the subset of the required street segments that could potentially be removed from the route or replaced with other street segments. *RequiredSegments* is the set of required street segments that will be used by the route generator to generate the full route. *BestRoute* is the route that is feasible and shortest in length. Line 5 indicates that the algorithm will loop until the *CandidateList* is empty.

Lines 6-13 represent the remove procedure. Street segment s_{mc} in the *CandidateList* with the highest marginal cost in the *BestRoute* is removed. *CandidateList* and *RequiredSegments* are updated and route $R_{s_{mc}}$ is generated using the route gen-

```

1:  HaveToBeRequired  $\leftarrow M_R$ ;
2:  CandidateList  $\leftarrow E_R \cup A_R \setminus M_R$ ;
3:  RequiredSegments  $\leftarrow E_R \cup A_R$ ;
4:  BestRoute  $\leftarrow$  Full route obtained using route generator;
5:  while CandidateList  $\neq \emptyset$  do
6:      Find  $s_{mc} \in$  CandidateList which has the highest marginal cost
       in BestRoute;
7:      CandidateList  $\leftarrow$  CandidateList  $\setminus s_{mc}$ ;
8:      RequiredSegments  $\leftarrow$  HaveToBeRequired  $\cup$  CandidateList;
9:      Generate full route  $R_{s_{mc}}$  using route generator;
10:     if  $R_{s_{mc}}$  meets the specified likelihood  $L_i$  of reading each meter
         $i$  then
11:         CountInfeasible  $\leftarrow 0$ ;
12:         if Length of  $R_{s_{mc}}$   $<$  Length of BestRoute then
13:             BestRoute  $\leftarrow R_{s_{mc}}$ ;
14:         else
15:             Success  $\leftarrow$  FALSE;
16:             PossibleSegments  $\leftarrow \emptyset$ ;
17:             for all  $s_n \notin$  RequiredSegments  $\cup s_{mc}$  do
18:                 if constraint (2.2) is feasible for all the meters with
                     $x_j = 1$  and  $j \in$  RequiredSegments  $\cup s_n$  then
19:                     PossibleSegments  $\leftarrow$  PossibleSegments  $\cup s_n$ ;
20:                 for all  $s_p \in$  PossibleSegments do
21:                     RequiredSegments  $\leftarrow$  RequiredSegments  $\cup s_p$ ;
22:                     Generate full route  $R_{s_{mc}}^{s_p}$  using route generator;
23:                     RequiredSegments  $\leftarrow$  RequiredSegments  $\setminus s_p$ ;
24:                     if Length of  $R_{s_{mc}}^{s_p}$   $<$  Length of BestRoute then
25:                         BestRoute  $\leftarrow R_{s_{mc}}^{s_p}$ ;
26:                         Success  $\leftarrow$  TRUE;
27:                 if Success = FALSE then
28:                     HaveToBeRequired  $\leftarrow$  HaveToBeRequired  $\cup s_{mc}$ ;
29:                     CountInfeasible  $\leftarrow$  CountInfeasible + 1;
30:                     if CountInfeasible  $>$  10 then
31:                         break;

```

Table 2.27: Algorithm for the remove and repair procedure of the route trimmer.

erator. If $R_{s_{mc}}$ is feasible, *CountInfeasible*, an infeasibility counter for a route, is set to zero, and if the length of $R_{s_{mc}}$ is smaller than the length of the *BestRoute*, $R_{s_{mc}}$ becomes the new *BestRoute*.

Lines 14-30 represent the repair procedure. If $R_{s_{mc}}$ is infeasible, the repair

procedure tries to add another street segment to remove the infeasibility. *Success*, a binary variable representing the success of the repair procedure, is initialized to FALSE. *PossibleSegments* is the subset of street segments outside of *RequiredSegments* and s_{mc} , each of which, if added to *RequiredSegments*, has the potential of generating a shorter route compared to the current *BestRoute*. If any street segment s_n , outside of the *RequiredSegments* and s_{mc} , satisfies constraint (2.2) of the Stage 1 IP along with the *RequiredSegments*, then s_n is added to the *PossibleSegments*. For each street segment s_p in *PossibleSegments*, a route $R_{s_{mc}}^{s_p}$ is generated using the route generator with the required street segments being the *RequiredSegments* and s_p . The routes $R_{s_{mc}}^{s_p}$, for each s_p , and the *BestRoute* are compared and the shortest route is set as the new *BestRoute*. If the repair procedure improves the *BestRoute*, *Success* is updated to TRUE. If *Success* is FALSE after the repair procedure, i.e., both the remove and the repair procedures involving street segment s_{mc} is unsuccessful, then s_{mc} is added to the *HaveToBeRequired* and the *CountInfeasible* is increased by one. The loop repeats by choosing a new s_{mc} from the current *CandidateList*, given that it is non-empty, based on the current *BestRoute* unless *CountInfeasible* is greater than the stopping criterion, which is set to 10.

When the remove and repair procedure terminates, a second procedure of the route trimmer looks for remaining redundancies in the list of required street segments for the current *BestRoute*. When a required street segment lies on the shortest path between its predecessor and successor required street segments, that particular required street segment would be visited by the vehicle as a deadhead segment. Therefore, it can be removed from the list of the required street segments without

affecting feasibility or the total route length. This simplifies the representation of the vehicle route and speeds up the simulation experiments conducted in Section 2.14.

The route generator is used extensively in the route trimming procedure. This strengthens our motivation for choosing a fast metaheuristic for the route generator. To further speed up the route generator and to avoid solving the same instance multiple times during our simulation experiments, a pool of solutions is maintained. Before using the route generator, this pool is checked for instances that have been solved already.

In Figure 2.15, we show how the route generator and the route trimmer work for a small example. Figure 2.15a shows the street network. The route of the vehicle starts from and ends at the depot denoted by the black square. The remaining nodes in the network are denoted by black dots. The black lines are the edges in the network (assume that there are no arcs). Figure 2.15b shows the Stage 1 IP solution. The red lines are the required street segments that the meter reading vehicle needs to traverse to guarantee the specified service levels (L_i). Figure 2.15c shows the route produced by the route generator. The blue lines are the deadhead segments added by the route generator to connect the required street segments in the shortest possible manner. Figure 2.15d shows the route produced by the route trimmer. The route trimmer finds the required street segment that has the largest marginal cost in the current route (Figure 2.15c), denoted by the yellow line, and replaces it with another street segment, denoted by the green line, such the new route (Figure 2.15d) has a smaller length and the specified service levels (L_i) are

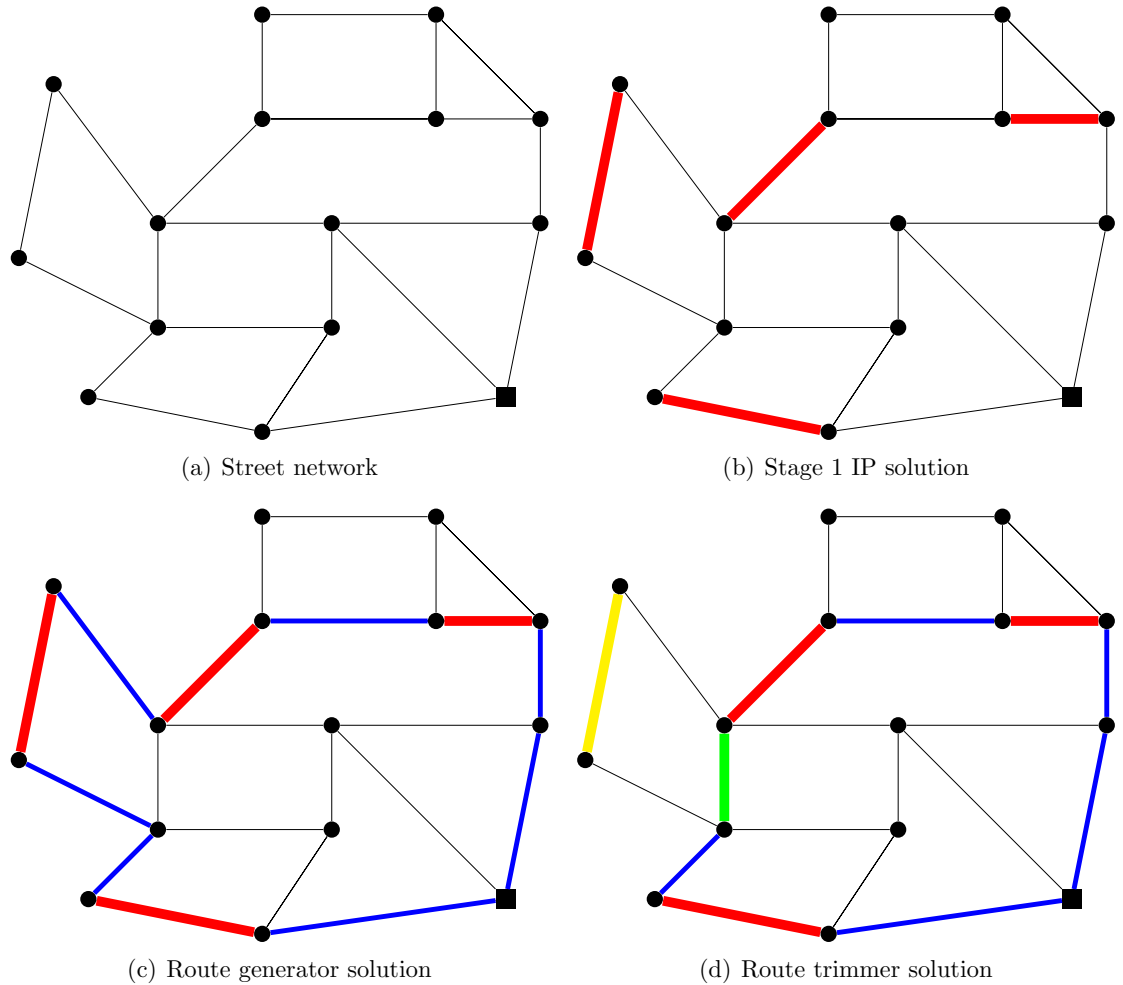


Figure 2.15: (Color online) The route generator and the route trimmer applied to a small example. Red lines denote the required street segments. Blue lines denote the deadhead segments. Yellow line denotes the required street segment that is removed. Green line denotes the new street segment added to the route as a replacement for the yellow line.

still satisfied.

2.14 Simulation Experiments

On a regular basis, utility companies need to decide on the routes for their meter reading vehicles. The quality of a route is determined by its length and robustness. Shorter routes are more cost effective as fuel and labor costs are lower.

The smaller the number of missed meters the more robust the route is for reading the uncertain RFID signals. To compare and quantify the performance of the three Bayesian updating models on route quality, simulation experiments are conducted using the CNG data set with 6,067 meters, 1,575 street segments, and 1,072 nodes. As the meter reading vehicle makes more trips and collects more data, the parameters of the Bayesian updating models should get closer to the actual parameter values. The actual parameter values depend on the street network and the distribution of the meters in the street network. This will be demonstrated by the fact that the vehicle routes will be adjusted over time to reduce the number of missed meters and still being cost-effective.

2.14.1 Actual Reading Probabilities

To calculate the probabilities p_{ij} 's for the three Bayesian updating models, we use the parameter values estimated in Section 2.11. The p_{ij} 's obtained are considered to be the actual probabilities, denoted by \bar{p}_{ij} 's, with which meter i is read from street segment j at least once. This assumption is reasonable since the size of the data set on which the model parameters are estimated is very large ($N \sim 5$ million for the logit model and the probit model, and $m = 829$ and $n = 6,067$ for the hierarchical probit model). To determine whether or not meter i is successfully read from street segment j , we consider a binomial random variable $Y_{ij} \sim \text{Binomial}(M, \bar{p}_{ij})$, where M is the number of times street segment j is traversed in the route. Meter i is considered to be missed from the route of the vehicle if $Y_{ij} = 0$ for all street

segments j in the route, i.e., meter i has not been read from any of the street segments traversed by the vehicle.

For the logit model, $\ln(\bar{p}_{ij}/1 - \bar{p}_{ij}) = -1.242 - 0.003 \times \text{Shortest_Distance}_{ij} + 0.019 \times \text{No_of_Pulses}_j - 0.003 \times \text{No_of_Customers}_i$, where $\text{Shortest_Distance}_{ij}$ is in meters.

For the probit model, $\Phi^{-1}(\bar{p}_{ij}) = -1.024 - 0.001 \times \text{Shortest_Distance}_{ij} + 0.007 \times \text{No_of_Pulses}_j - 0.002 \times \text{No_of_Customers}_i$, where $\text{Shortest_Distance}_{ij}$ is in meters.

For the hierarchical probit model, $\mathbb{E}(\beta_i) = \mathbb{E}(\Theta^T)z_i$ for each i . Therefore,

$$\begin{pmatrix} \hat{\beta}_{i,1} \\ \hat{\beta}_{i,2} \\ \hat{\beta}_{i,3} \end{pmatrix} = \begin{pmatrix} -0.890 & -0.0002 \\ -0.002 & -0.000003 \\ 0.004 & 0.0000006 \end{pmatrix} \begin{pmatrix} 1 \\ \text{No_of_Customers}_i \end{pmatrix}.$$

So, $\Phi^{-1}(\bar{p}_{ij}) = (-0.890 - 0.0002 \times \text{No_of_Customers}_i) + (-0.002 - 0.000003 \times \text{No_of_Customers}_i) \times \text{Shortest_Distance}_{ij} + (0.004 + 0.0000006 \times \text{No_of_Customers}_i) \times \text{No_of_Pulses}_j$, where $\text{Shortest_Distance}_{ij}$ is in meters.

2.14.2 Simulation Model Overview

A simulation starts from an initial vehicle route. This is iteration zero or the initialization step. To construct the initial route, p_{ij} is set to 1 if meter i is within 500 feet from street segment j and 0 otherwise. First, the linear Stage 1 IP is solved to obtain the required street segments. The specified likelihood (L_i) values do not affect the Stage 1 IP solution in iteration zero because of the particular choice of

the p_{ij} 's. Then the full route is produced using the route generator and the route trimmer. This initial route is a deterministic CEVRP solution which is currently used by utility companies. Therefore, we use this initial route as a benchmark for our experiments. In the first iteration, the Y_{ij} 's are generated based on the initial route. Since we do not have any information about the p_{ij} 's during the first iteration (this is analogous to time period 1 as described in Section 2.8), the prior distributions for the parameters of the Bayesian updating models are set to the vague priors. Using the priors and the meter reading data (Y_{ij} as the dependent variable and Shortest-Distance $_{ij}$, No_of_Pulses $_j$, and No_of_Customers $_i$ as the independent variables), we obtain the posterior distributions of the parameters, and thereby the updated p_{ij} 's. A new route is produced based on the updated p_{ij} 's. For all subsequent iterations, the Y_{ij} 's are generated based on the current route; the posterior distributions from the previous iteration are used as the prior distributions.

An iteration in the simulation experiment represents the generation of a new route for the next meter reading day after updating the probabilities p_{ij} 's using the previous meter reading data. Therefore, our simulation model can be used by a utility company as a decision-support tool to generate robust and cost effective routes. The only difference would be that the utility companies have access to the actual dependent variable values (Read_OR_Not $_{ij}$) after the vehicle has traversed the route, whereas, for the simulations we generate the Y_{ij} 's based on the \bar{p}_{ij} 's and the route to identify the missed meters.



Figure 2.16: (Color online) A view of a portion of the actual street network with meter locations in the UTM format serviced by Connecticut Natural Gas in our data set from Hartford, Connecticut. The red dots represent the meters. This network is used for our simulation experiments.

2.14.3 Generating the Network

Our data set is from Hartford, Connecticut. The OAR Bench software ([Lum et al. 2018](#)) is used to extract the metadata of the street network (information about nodes, edges, and arcs). The coordinate system used by OAR Bench is the World Geodetic System (WGS) 84. The meter locations in the ArcGIS data are also in WGS 84 format. WGS 84 is the reference coordinate system used by the Global

Positioning System (GPS). The Universal Transverse Mercator (UTM) conformal projection (map projections that preserve angles locally) uses a 2-dimensional Cartesian coordinate system to give locations on the surface of the Earth by dividing the Earth into sixty zones. The metadata of the street network and the meter locations are converted from the WGS 84 format to the UTM format using the UTM zone number of Hartford, Connecticut. The street network and the meter locations in the UTM format are considered to be on a flat surface. Therefore, Euclidean geometry can be used to find spatial distances between any two points in the network. Figure 2.16 gives a view of the actual street network with meter locations in the UTM format.

2.14.4 Simulation Results

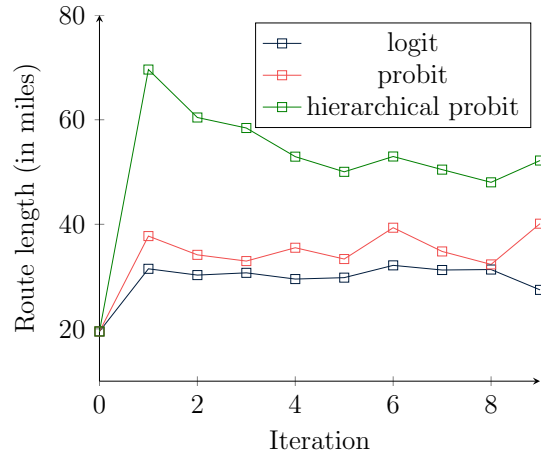
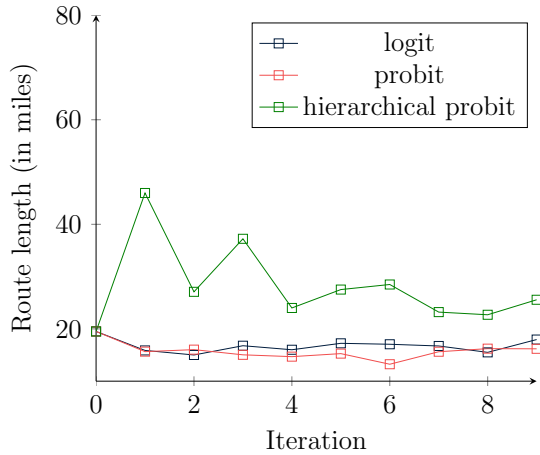
Simulations are conducted for each of the three Bayesian updating models, for both the single-objective and the bi-objective linear Stage 1 IP, and for L_i values of 0.95 and 0.75 for all i . In total, we perform 12 ($3 \times 2 \times 2$) simulation experiments. Each simulation is run for nine iterations. The initial route in iteration zero depends on the version of the Stage 1 IP used for that particular simulation experiment. A simulation using the single-objective linear Stage 1 IP has a different initial route compared to a simulation using the bi-objective linear Stage 1 IP.

The meter reading vehicles are driven at five miles per hour in residential neighborhoods. Drivers are encouraged to drive at a slow speed to increase the chances of reading the uncertain RFID signals. We assume five minutes to manually

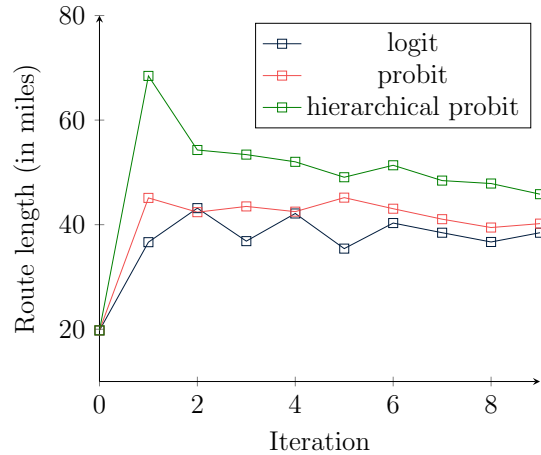
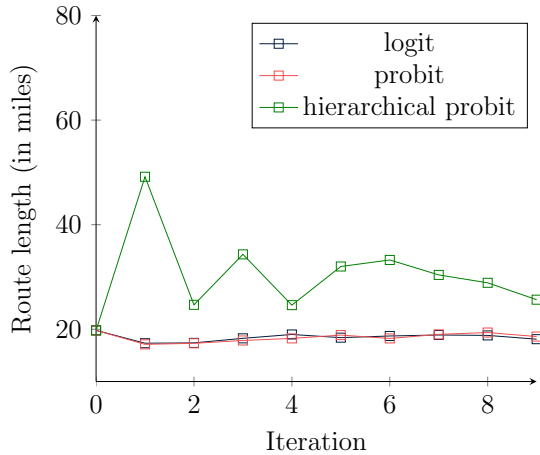
read a meter (these are the meters that are infeasible with respect to constraints (2.2) in the Stage 1 IP) after parking the vehicle on the closest street segment. The route lengths are in miles. For each meter that is supposed to be manually read, we add 0.42 miles to the full route length as a proxy for the distance that could have been traversed in five minutes at five miles per hour. We use an i7 CPU with 32 GB RAM for the simulations. We use R software version 3.3.1 to run the Bayesian updating models and Gurobi version 7.5 to solve the linear Stage 1 IP. Each time the linear Stage 1 IP is solved or the route generator is used, a time limit of 10 minutes and two minutes, respectively, is imposed.

In Figures 2.17 and 2.18, we show the results of the simulation experiments for the route length and the number of missed meters, respectively. Both figures show the results of the three Bayesian updating models in four different scenarios, namely, single-objective linear Stage 1 IP and likelihood values of 0.75 (scenario a), single-objective linear Stage 1 IP and likelihood values of 0.95 (scenario b), bi-objective linear Stage 1 IP and likelihood values of 0.75 (scenario c), and bi-objective linear Stage 1 IP and likelihood values of 0.95 (scenario d). Figure 2.17 shows the results from iteration 0 (initial route) through iteration 9, whereas, Figure 2.18 shows the results from iteration 1 through iteration 9. The missed meters (Figure 2.18) in iteration k are based on the route (Figure 2.17) in iteration $k - 1$ and also leads to the formation of the new route in iteration k .

In each of the four scenarios depicted in Figures 2.17 and 2.18, the logit model and the probit model do not show any significant differences. The initial routes are approximately 20 miles and there are 279 missed meters in scenarios a and b and



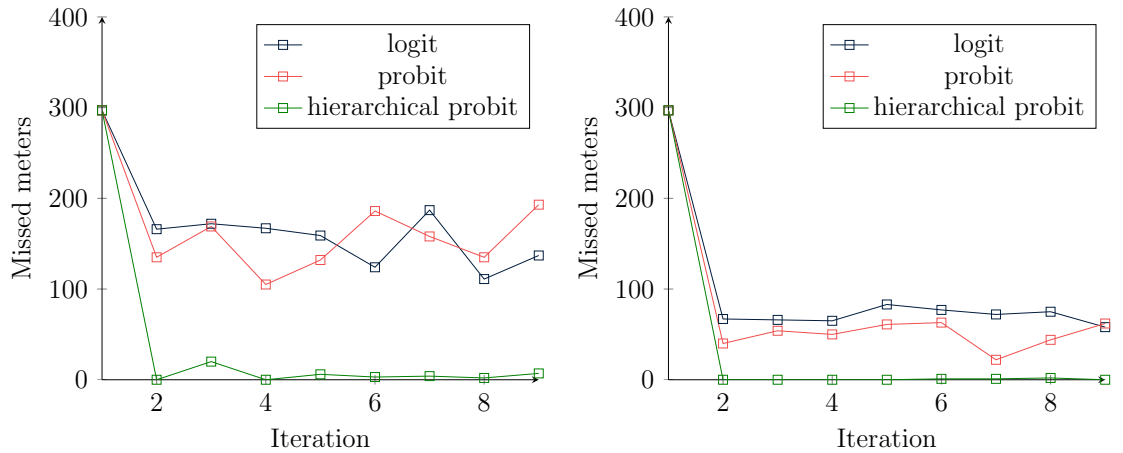
(a) Single-objective Stage 1 and likelihood values of 0.75 (b) Single-objective Stage 1 and likelihood values of 0.95



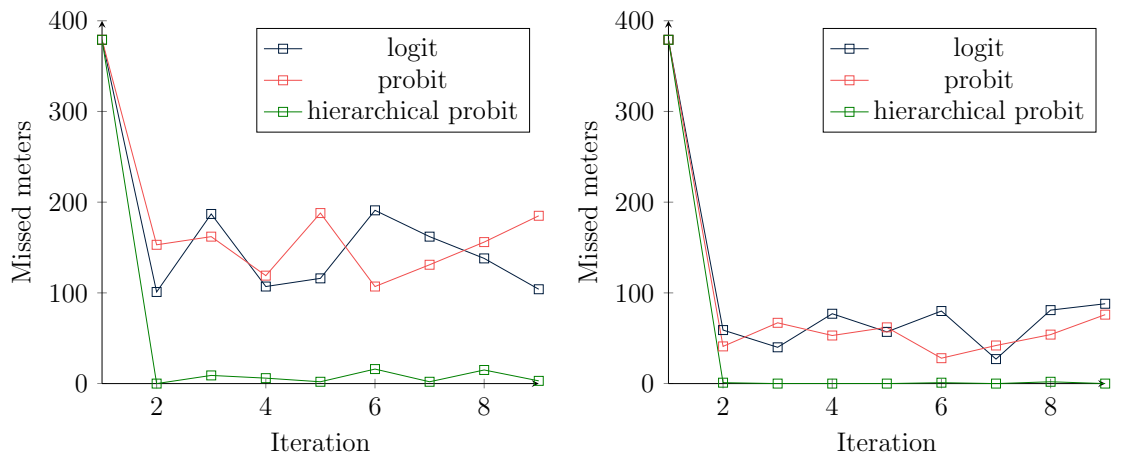
(c) Bi-objective Stage 1 and likelihood values of 0.75 (d) Bi-objective Stage 1 and likelihood values of 0.95

Figure 2.17: (Color online) Simulation results for the route length.

379 missed meters in scenarios c and d. The logit model and the probit model, on average, generated routes 17 miles long and missed 148 meters in scenarios a and c, and routes 37 miles long and missed 60 meters in scenarios b and d. An increase in the likelihood values from 0.75 to 0.95 increased the route length from 17 miles to 37 miles and reduced the number of missed meters from 148 to 60. However, the choice of the Stage 1 IP did not have any substantial impact on the results. For the routes to be operationally impactful to the extent that there is no need to send



(a) Single-objective Stage 1 and likelihood values of 0.75 (b) Single-objective Stage 1 and likelihood values of 0.95



(c) Bi-objective Stage 1 and likelihood values of 0.75 (d) Bi-objective Stage 1 and likelihood values of 0.95

Figure 2.18: (Color online) Simulation results for the number of missed meters.

another vehicle at a later time to read the missed meters, the number of missed meters needs to be reduced further.

The hierarchical probit model, in each of the four scenarios shown in Figure 2.17, shows a large increase in route lengths compared to the initial routes of 20 miles. However, the route lengths gradually decreased to around 28 miles in scenarios a and c, and around 52 miles in scenarios b and d. The hierarchical probit model captures the uncertain behavior of each meter uniquely, and therefore it explores the street network to learn the reading potential of the different street segments. As

a result, the hierarchical probit model, as shown in Figure 2.18, on average, missed six meters in scenarios a and c and one meter in scenarios b and d. The choice of the Stage 1 IP did not have any significant impact on the results. The probability estimates of the hierarchical probit model get very close to the actual probabilities within a few iterations. Therefore, even with likelihood values of 0.75, the routes generated are very high quality (only six meters are missed from these routes that are just 8 miles longer than the initial routes). These routes should be of high operational and practical relevance for a utility company. Likelihood values greater than 0.75 make the routes substantially longer, thereby, increasing the cost.

In Bayesian updating, the posterior distribution of the model parameters is a compromise between prior information and the information provided by the new data. This helps in the inference on the parameters. If the new data are smaller in size (this leads to model parameters with distributions of high variance), we want to rely more on prior knowledge. Conversely, if the data are plentiful and contain high-quality information, then we should not care much about the form of the prior information. The Bayesian updating process automatically considers this trade-off. For the hierarchical probit model, the meter reading data obtained from a route are divided into small segments to learn about the behavior of each meter separately. Therefore, the size of the data is effectively a few orders of magnitude smaller for the hierarchical probit model compared to the logit model and the probit model. This leads to more dependence on the priors for the posterior distributions of the hierarchical probit model. When we estimated the parameters of the Bayesian updating models in Section 2.11 using our real meter reading data, the logit model

Model	d_1	h	d_2	Total time
Initial (benchmark) route	20	329	25.58	33.12
Logit or Probit for likelihood values of 0.75	17	148	24.62	17.37
Logit or Probit for likelihood values of 0.95	37	60	18.37	13.62
Hierarchical Probit for likelihood values of 0.75	28	6	8.35	6.65
Hierarchical Probit for likelihood values of 0.95	52	1	6.99	10.95

Table 2.28: Average comparison of the total time to read all the meters.

and the probit model used $6,067 \times 829$ (~ 5 million) data points, whereas, the hierarchical probit model used 829 data points for each of the lower level probit models and 6,067 data points for the higher level multivariate linear model. The simulation experiments were started using vague priors because we did not have any prior information on the parameters to accurately capture the signal transmission behavior of the meters in the street network. Therefore, to lower the variance of the posterior distributions, the hierarchical probit model produced longer routes to gather more meter reading data and, thereby, improved the quality of the probability estimates. Finally, the logit and the probit model need to estimate only four parameters. However, the hierarchical probit model needs to estimate three lower level parameters for each of the 6,067 meters and seven higher level parameters. Therefore, the hierarchical probit model is able to capture the heterogeneous behavior of the meters and build routes accordingly, unlike the logit and the probit model which capture the average effect of all the meters.

For a utility company, the total cost of reading meters in each time period (iteration) is divided into two phases. The first phase is the cost of the CEVRP routes to read the meters automatically. The second phase is the cost of reading the meters that are missed from the first phase route. Typically, a public utility commission

does not allow billing cycles to shift more than two or three days. Therefore, it is necessary to send out another vehicle to manually read the missed meters within a few days after the completion of the first phase. We compare the quality of the initial (benchmark) route and the routes generated by the three Bayesian updating models taking into account the cost from both phases. Let d_1 denote the length (in miles) of the route in the first phase. A meter reading vehicle is driven at five miles per hour, so the first phase route will take $d_1/5$ hours to complete. Let h denote the number of meters missed from the first phase route. During the second phase, the missed meters are read manually to ensure success with probability one (a vehicle is driven at a speed of around 15 miles per hour and stops at each missed meter to read it). The second phase is a standard VRP route. A utility company knows which meters are missed from the first phase route. It will have the exact standard VRP route length to read the missed meters. In our simulations, we discuss the results averaged over the nine iterations and need to estimate the route length for the second phase. [Kwon et al. \(1995\)](#) showed that $(0.8326 - 0.0011(h + 1) + 1.1147G/(h + 1))\sqrt{(h + 1)D}$, where D is the area of the rectangular network and G is the ratio of length and breadth of the network such that $G \geq 1$, gives a reasonable estimate of the standard VRP route length as a function of the number of customers (h) on the route. For our data set, $G = 1.5$ and $D = 8.8$ square miles. Let d_2 denote the estimate of the length (in miles) of the route in the second phase; the route will take $d_2/15$ hours to complete. If we assume five minutes to manually read a meter, it will take $h/12$ hours to read h missed meters. The total time (in hours) required to read all meters is the sum of the first phase time ($d_1/5$) and the second phase time ($d_2/15 + h/12$).

In Table 2.28, we show the average comparison of the total time for the benchmark route and the routes generated by the three Bayesian updating models. All three models took much less time than the benchmark route. For the logit and the probit models, the total time was a few hours less for likelihood values of 0.95 compared to likelihood values of 0.75. This is because the time required to read the missed meters in the second phase was smaller even though the first phase routes were longer for likelihood values of 0.95. The hierarchical probit model does considerably better than the logit or the probit models. For likelihood values of 0.75, the hierarchical probit model takes only 6.5 hours to read all the meters. The hierarchical probit model for likelihood values of 0.95 has a longer first phase route without significantly reducing the second phase time compared to the likelihood values of 0.75.

We point out that there is an inherent trade-off between the first phase time and the second phase time. Longer first phase routes would reduce the number of missed meters. Therefore, longer route times in the first phase would generally lead to shorter route times in the second phase and vice-versa. Beyond a certain level, lengthening the first phase route would lead to a diminishing return on the total time to read all meters. We should be able to find the optimal value for the likelihood levels (unique for each statistical model and street network) of reading the meters with respect to the total time. Likelihood levels greater than or lesser than the optimal value would lead to an increase in the total time.

2.15 Conclusions and Future Directions

We developed an iterative methodology to read uncertain RFID signals from utility meters using vehicles at some distance. Every time we get access to new meter reading data, we learn about the probabilities p_{ij} 's. The two-stage IP, representing the meter reading problem, is then re-solved to generate routes that are robust for addressing the uncertainty. Even though the routes generated by our procedure are a few miles longer than the benchmark route, the routes are cost-effective when compared to the costs incurred by sending a vehicle at a later time to read the missed meters. The two-stage IP formulation is deterministic even though the meter reading problem has an inherent stochastic set up. The Stage 1 IP, which gives us the required street segments to reach a specified service level for each meter, is linear in the decision variables. Computational experiments showed that the linear Stage 1 IP can be optimally solved within a few seconds for large data sets. Since the Stage 2 IP is NP-complete, we developed a fast metaheuristic that generated and further improved the full route, and still maintained the specified service levels for each meter. We developed three Bayesian updating models to learn from the new incoming data in an efficient way and avoid the drawbacks faced in regression. We cross-checked our choice for the priors in the Bayesian models by comparing the parameter estimates of the logit model and the probit model with their regression counterparts. We showed that the hierarchical probit Bayesian updating model produces more accurate probability estimates for each meter compared to the other two models. We conducted simulation experiments to compare the route qualities

for the three Bayesian models and the benchmark routes used by utility companies. In our simulations, we used an actual street network and meter locations, which is different from the artificial networks used in the literature. Our simulation results showed that the routes generated by the hierarchical probit model with likelihood values of 0.75 are operationally useful because almost all meters are read with only a few miles of extra travel compared to the initial route and the total time in the two phases to read all the meters is 6.5 hours which fits into a typical 8 hour workday schedule for the drivers.

Typically, the drivers of service vehicles are more comfortable traveling through the same neighborhoods on their routes. For a utility company, this leads to meter reading routes that look similar in every period. Because of this, we do not have any information about the actual meter reading potential from most of the street segments in the network. The iterative framework proposed and discussed in this paper to generate robust routes will be more effective when we have actual meter reading data from a larger set of street segments. In future work, using Bayesian decision theory and route optimization, we would like to be able to identify new street segments to traverse at each time period so that the information gain is maximized without having to travel many extra miles.

Chapter 3: Data-Driven Analysis of the Variability of Routes in the Capacitated Vehicle Routing Problem

3.1 Introduction

Everyday, delivery and service companies dispatch many vehicles to deliver customer products and provide services to customers in a city. These companies generate vehicle routes using software and algorithms provided by third-party vendors. These algorithms focus on minimizing the total route time or average route time for the fleet of vehicles serving a city.

Typically, these companies need to maintain a well-defined workload balance amongst the company drivers, i.e., drivers should have similar route times (inclusive of customer service times). Large differences in workloads are not only unfair to the drivers, but they might also lead to increased operating and delivery costs for these companies. Other factors such as mileage, number of vehicles used, driver wages, and traffic conditions during the actual service hours also contribute to the costs that are often not considered when generating routes using these algorithms. It is important to consider the difference between planned costs and operational costs. Third-party routing software and algorithms consider the costs that are generally taken into

Statistic	Program Alpha	Program Beta
Number of routes generated	25	23
Route time minimum	5.17	5.24
Route time 1 st quartile	5.30	5.65
Route time median	5.34	5.94
Route time 3 rd quartile	5.39	6.12
Route time maximum	5.46	6.43
Route time inter-quartile range	0.09	0.47
Route time range	0.29	1.19
Route time standard deviation	0.07	0.33
Route time mean	5.33	5.85
Route time total	133.27	135.33

Table 3.1: Summary statistics of route times (in hours) for routes generated by two third-party software programs on an actual street network to serve customers.

account while building routes. These costs form the planned costs. However, the operational costs might have extra components in addition to the planned costs depending on the specific delivery and service company. Special attention should be given to the additional elements that form the operational costs while generating the routes.

In Table 3.1, we show the summary statistics of route times (in hours) for routes generated by two third-party software programs on an actual street network to serve customers. The company that markets Program Alpha was bidding for a client that already uses Program Beta to generate its routes. The solution generated by Program Alpha requires 25 vehicles to serve the city in a total of 133.27 hours. The solution generated by Program Beta requires two fewer vehicles with two more hours than the solution produced by Program Alpha. The route times generated by Program Beta have greater variability (greater workload imbalance) compared to the route times generated by Program Alpha. Drivers might be more satisfied

Instance name	Number of customers	Vehicle capacity	Instance new name
E-n22-k4	21	6000	E-n22-c6000
E-n23-k3	22	4500	E-n23-c4500
E-n30-k3	29	4500	E-n30-c4500
E-n33-k4	32	8000	E-n33-c8000
E-n51-k5	50	160	E-n51-c160
E-n76-k7	75	220	E-n76-c220
E-n76-k8	75	180	E-n76-c180
E-n76-k10	75	140	E-n76-c140
E-n76-k14	75	100	E-n76-c100
E-n101-k8	100	200	E-n101-c200
E-n101-k14	100	112	E-n101-c112

Table 3.2: Eleven CVRP instances ([Christofides and Eilon 1969a](#)).

with the routes produced by Program Alpha because they will be getting similar wages due to the similar number of hours they are driving and serving customers on a route. In contrast, it might be cheaper to operate two fewer vehicles with just two extra hours for the company when using the routes produced by Program Beta. For this example, the summary statistics of the route times are not enough to determine which of the two solutions is better in terms of total operating and delivery costs. It is important for the client to examine the effect of the workload imbalance created by Program Beta. The planned costs to generate the route using Program Beta do not include the cost of workload imbalance. However, in practice, the workload imbalance might be an integral part of the operational costs.

3.2 Capacitated Vehicle Routing Problem Instances

We focus on the Capacitated Vehicle Routing Problem (CVRP) to understand the importance of route balance in calculating the total operating and delivery costs

for delivery companies. A CVRP models a real-life scenario, where a vehicle has a capacity constraint and each customer has a demand. A company needs to fulfill all customer demands in a city using a fleet of vehicles. The number of vehicles required to serve the city depends on the total demand, the capacity of each vehicle, and the algorithm (see Table 3.1). In this chapter, we use the 11 CVRP instances given by [Christofides and Eilon \(1969a\)](#) which have the node coordinates to perform our analysis. In Table 3.2, we give the name of each instance, the number of customers, the capacity of each vehicle, and the new name of each instance that we will use. For example, in E-n22-k4, there are 22 nodes with 21 customers and 1 depot (denoted by n22) and 4 vehicles (denoted by k4). The capacity of each vehicle is 6000. The four instances starting with E-n76 have the same locations for the customers and the depot and the same customer demands. They differ with respect to the number of vehicles and the capacity of each vehicle. This also holds for the two instances starting with E-n101. As the vehicle capacity decreases for these six instances, we expect to use more vehicles to serve the customers.

The optimal or best-known solution to each of the 11 instances given in the [Capacitated Vehicle Routing Problem Library \(2014\)](#) uses exactly the number of vehicles given in the instance name shown in the left-most column in Table 3.2. In our work, we will let our algorithm determine how many vehicles to use. We will use the new names (that reflect only vehicle capacity) shown in the right-most column of Table 3.2. For example, E-n22-k4 becomes E-n22-c6000.

3.3 Importance of Standard Deviation of Routes

We want to understand and compare the effects of the standard deviation of the route lengths on the total route time, the total route time under random traffic conditions, and the total operating and delivery costs for a company to serve its customers.

We use a weighted savings modification of the Clarke and Wright (C&W) algorithm (Clarke and Wright 1964) to generate routes for an instance. In the initialization step of the standard C&W algorithm, each customer is served by its own vehicle from the depot. If i and j are two customers that are served separately from the depot (denoted by 0), then the savings obtained from merging the two routes is $s_{ij} = c_{i0} + c_{0j} - c_{ij}$, where c_{i0} and c_{0j} are the costs of serving i from the depot and j from the depot, respectively, and c_{ij} is the cost of going from i to j . At each iteration of the standard C&W algorithm, routes are merged based on the largest savings. At each iteration of our weighted savings modification algorithm, we determine the three largest savings s_1 , s_2 , and s_3 . Then we randomly choose to merge the routes with probability $p_k = s_k / (s_1 + s_2 + s_3)$ where $k \in \{1, 2, 3\}$. For each instance, we generated 1000 solutions using our weighted savings modification algorithm with the objective of minimizing the total route length. In Table 3.3, we show the breakdown of the 1000 solutions for each instance in terms of the number of vehicles required to serve the customers. To illustrate, in 1000 solutions generated by the weighted savings modification algorithm for the instance E-n22-c6000, 824 solutions required four vehicles and 176 solutions required five vehicles.

Instance	Number of vehicles															
	3	4	5	6	7	8	9	10	11	14	15	16				
E-n22-c6000		824	176													
E-n23-c4500	1000															
E-n30-c4500		1000														
E-n33-c8000		997	3													
E-n51-c160			500	500												
E-n76-c220					1000											
E-n76-c180						967	33									
E-n76-c140								135	865							
E-n76-c100										15	979	6				
E-n101-c200									1000							
E-n101-c112										999	1					

Table 3.3: Breakdown of 1000 CVRP solutions for each of the 11 instances in terms of the number of vehicles required to serve the customers.

The four instances starting with E-n76 and the two instances starting with E-n101 increasingly require more vehicles to serve the customers as the capacity of the vehicles decreases.

We build three linear regression models that can be represented by $y_{ij} = \hat{\beta}_1 + \hat{\beta}_2 \times \text{No.of.Routes}_{ij} + \hat{\beta}_3 \times \text{No.of.Customers}_i + \hat{\beta}_4 \times \text{Vehicle.Capacity}_i + \hat{\beta}_5 \times$

Parameter	Value
Fixed cost for using a vehicle	\$50 per day
Mileage	10 miles per gallon
Gas price	\$2.5 per gallon
Speed of vehicle with no traffic	25 miles per hour
Regular work hours for a driver	6 hours per day
Total wage for regular work hours	\$18 per hour for all 6 hours ($18 \times 6 =$ \$108)
Per hour wage for overtime work hours	\$27 per hour for each overtime hour
Service time for each customer	10 minutes

Table 3.4: Parameter values that are used to calculate the total operating and delivery costs for a company to serve its customers (Levy 2018).

Standard_Deviation $_{ij}$, where $y_{ij} = \mathbb{E}(Y_{ij})$, $\hat{\beta}_k = \mathbb{E}(\beta_k)$, $k \in \{1, \dots, 5\}$, $i \in \{1, \dots, 11\}$ denotes the instance number, $j \in \{1, \dots, 1000\}$ denotes the solution number of an instance, and $\mathbb{E}()$ denotes the expected value. No_of_Routes $_{ij}$ is specific for each solution of an instance as shown in Table 3.3. No_of_Customers $_i$ and Vehicle_Capacity $_i$ is the same for any solution given an instance as shown in Table 3.2. Standard_Deviation $_{ij}$ is the route length standard deviation calculated for each solution, and so it is specific for each solution of an instance. The dependent variable Y_{ij} is total route time for Model 1, total route time under random traffic conditions for Model 2, and total cost under random traffic conditions for Model 3. All three dependent variables are specific for each solution of an instance. In Table 3.4, we give the values of the parameters (Levy 2018) that are used to calculate the dependent variable values. For each solution of an instance, we have the route length for each vehicle, and we divide that by the speed of the vehicle with no traffic to get the route time for each vehicle. We sum the route times for all vehicles for each solution to get the total route times for Model 1. We randomly increase the route time for each vehi-

cle by $t\%$, where $t \sim \text{Uniform}(0, 10)$, to capture random traffic conditions during actual service by the vehicles. This assumes independence of traffic patterns across streets. There has been some work in the literature that looks at correlated traffic patterns between streets (Laporte et al. 1992, Kenyon and Morton 2003, Rostami et al. 2017) while generating the routes. However, the correlation of traffic patterns will not have an effect on our understanding of the impact of route length standard deviation on the total cost. We sum the increased route times for all vehicles for each solution to get the total route times under random traffic conditions for Model 2. The total costs under random traffic conditions for Model 3 has three components: driving cost, fixed cost, total wages. Driving cost and fixed cost are invariant under random traffic conditions. For each solution of an instance, total route length \times (gas price/mileage) gives us the driving cost; number of vehicles \times fixed cost for using a vehicle gives us the fixed cost. We add the increased route time under random traffic conditions and the customer service time (number of customers served on the route \times service time for each customer) for each vehicle to get the total working hours under random traffic conditions for each driver. If the total working hours are less than the regular work hours, then the driver is paid the total wage for regular work hours. If the total working hours are greater than the regular work hours, an additional per hour overtime wage is paid to the driver for each overtime hour. We sum the wages for all drivers for each solution to get the total wages. The dependent variables of Model 1 and Model 2 are in hours; the dependent variable of Model 3 is in dollars.

In Table 3.5, we present the linear regression results for all three models. The

Coefficient	Model 1	Model 2	Model 3
Intercept (β_1)	-5.266*** (0.167)	-5.530*** (0.176)	-257.014*** (5.568)
No_of_Routes (β_2)	1.688*** (0.011)	1.774*** (0.012)	137.939*** (0.382)
No_of_Customers (β_3)	0.272*** (0.002)	0.286*** (0.002)	9.111*** (0.068)
Vehicle_Capacity (β_4)	0.002*** (0.00002)	0.002*** (0.00002)	0.062*** (0.0006)
Standard_Deviation (β_5)	0.076*** (0.001)	0.080*** (0.001)	3.285*** (0.034)
Adjusted R ²	0.90	0.90	0.97

***p<0.001

Table 3.5: Linear regression results for three models.

size of the data set is 11,000 (11×1000) for each model. We give the means of the coefficients of the independent variables and their standard errors in parenthesis. All five coefficients for all three models are significant at the 0.1% level. The adjusted R² values indicate very good model fits. The coefficient of No_of_Routes is very large for Model 3 compared to Models 1 and 2. An additional route leads to an increased fixed cost (for using an additional vehicle) and increased total wages (at least for the regular work hours of an additional driver) in the calculation of the total operating and delivery costs. However, it reduces the total overtime work hours, thereby, reducing the cost for the total overtime wages. Comparing the coefficient of Standard_Deviation across the three models, we see that the effect of the route length standard deviation increases for the total route time under random traffic conditions compared to the total route time with no traffic. It is largest for

the total operating and delivery costs under random traffic conditions. Since the driving cost and fixed cost are invariant under random traffic conditions, the route length standard deviation affects the total wages because drivers having to work more than the regular working hours will have a large impact on the cost. When routing algorithms are used to generate routes, it is important to consider the traffic conditions during the time of actual service and, thereby, also consider the effect on the total operating and delivery costs.

The regression results show that route balance is not just of secondary importance due to perhaps union regulations. Route balance directly affects total operating and delivery costs. Under random traffic conditions, route times increase from the solution that was originally produced by the routing algorithm. When starting with routes that are already imbalanced, longer routes will take even more time to complete in the presence of heavy traffic. Imbalanced routes directly and significantly affect the total wages component in the total cost calculation because longer routes lead to a larger chance of a driver working more than the regular work hours. Thus, when routes are imbalanced, a delivery company may pay more overtime wages to its drivers. Our regression analysis shows that when we consider a real-world CVRP (in terms of complexity and random traffic conditions), as opposed to benchmark problems found in the literature (e.g., a simple extension of the TSP), route balance is an important determinant of low-cost solutions.

We can interpret the regression analysis in a practical setting as follows. Suppose we seek to solve a real-world CVRP (using data like those presented in Table 3.4) and obtain a high-quality solution in terms of total cost. We can apply a sim-

ple algorithm, such as the weighted savings modification of the C&W algorithm, repeatedly to obtain many solutions. We can then select a solution with a small total route length and a small route length standard deviation to achieve low operating and delivery costs.

3.4 Effect of Reducing Standard Deviation on Cost

After we select a high-quality solution among the many solutions generated by the C&W savings algorithm, we might still be able to reduce the total operating and delivery costs under random traffic conditions by decreasing the standard deviation of the routes. We implement fast and easy modifications to the solutions of the C&W savings algorithm in order to achieve this reduction.

We consider random traffic versions of the instances given in Table 3.2. For each instance, we divide the distance between each pair of customers (including the depot) by the speed of the vehicle with no traffic. This gives us the travel time with no traffic between each pair of customers. We generate 1000 different pseudo versions of each instance by randomly increasing the travel time between customer pairs by $t\%$, where $t \sim \text{Uniform}(0, 10)$. This captures random traffic conditions during actual service by the vehicles. For each pseudo version (random traffic induced instance), we use the C&W savings algorithm to generate the solution (set of routes) with the objective of minimizing the total increased route times including service times. For each instance, we have 1000 C&W solutions under random traffic conditions (Group A). Group A is our benchmark because it is obtained by minimizing the total route

times including service times which is typical of any route generating algorithm.

We apply three different modifications of the record-to-record (RTR) travel algorithm (Golden et al. 1998) to each solution. Route time now includes the time to service customers. A record is the total route time of the C&W solution. A deviation is $r\%$ of the record, where $r \in \{0, \dots, 10\}$. The RTR travel algorithm will accept a new solution (set of routes) if the total route time is less than the sum of record and deviation. The objective of the RTR travel algorithm is to minimize the standard deviation of the route time. In the first modification (Scenario X) of the RTR travel algorithm, we sort the routes of a solution from the longest to the shortest in terms of the route time. We try to move a customer from a longer route to a shorter route so that the new total route time is less than the sum of record and deviation, and the route time standard deviation decreases. The sequence of trials for moving a customer from a long route starts with the longest route followed by the second longest route and so on. The sequence of trials for moving a customer to a short route starts with the shortest route followed by the second shortest route and so on. After every customer move, we update the sequence of routes. In the second modification (Scenario Y) of the RTR travel algorithm, we use a total route time constraint in addition to using the sorted sequence as described in Scenario X. The total route time is segmented in the following way: the segment value for anything less than six hours is considered as six hours, for anything between six and seven hours is considered as seven hours, and so on. This segmentation is in line with the wage structure for the drivers described in Table 3.4. A customer move is allowed under Scenario Y using the sorted sequence of routes if the new total route

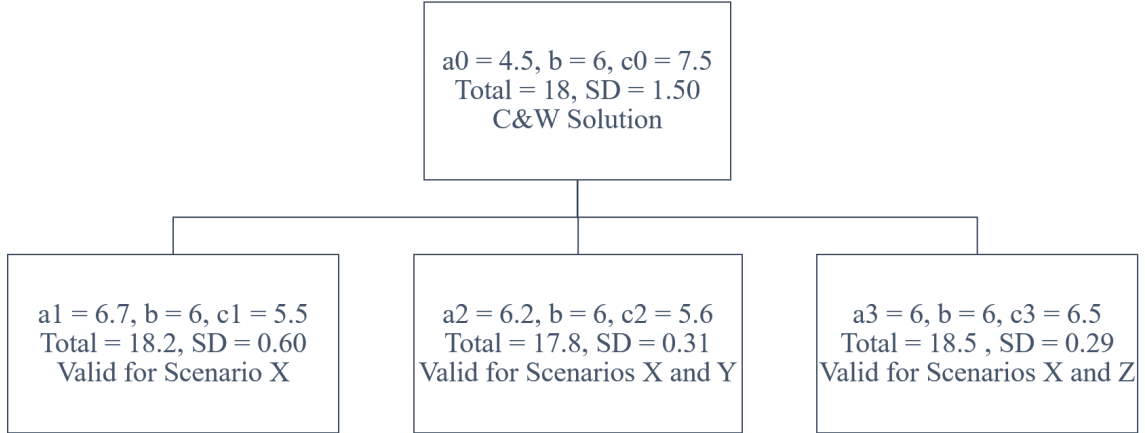


Figure 3.1: Example to illustrate an iteration of the RTR travel algorithm explaining the three scenarios.

time is less than the sum of record and deviation and less than the segment value for the current total route time, and the route time standard deviation decreases. After every customer move, we update the sequence of routes and the total route time segment value. In the third modification (Scenario Z) of the RTR travel algorithm, we use a route time constraint for each route in addition to using the sorted sequence as described in Scenario X. The route times are segmented in the same way as in Scenario Y. A customer move is allowed under Scenario Z using the sorted sequence of routes if the new total route time is less than the sum of record and deviation, the new route times for each route are less than the respective segment values for the current route times, and the route time standard deviation decreases. After every customer move, we update the sequence of routes and the route time segment values for each route.

In Figure 3.1, we show an example to illustrate an iteration of the RTR travel algorithm explaining the three scenarios. Suppose the C&W savings algorithm generates three routes a_0 , b , and c_0 for a CVRP instance with the route times as 4.5

hours, 6 hours, and 7.5 hours, respectively. The total route time is 18 hours, and the route time standard deviation is 1.5 hours. Let us look at an iteration of the RTR travel algorithm with the deviation parameter r as 3%. Therefore, the sum of record and deviation for the routes a0, b, and c0 is 18.54 ($18 + 0.03 \times 18$) hours. The segment value for the total route time is 18 hours. The segment values for routes a0, b, and c0 are 6 hours, 6 hours, and 8 hours, respectively. Suppose a customer move from c0 to a0 changes the two routes to a1 and c1 with the route times as 6.7 hours and 5.5 hours, respectively. The new total route time is 18.2 hours, and the new route time standard deviation is 0.60 hours. This is a valid customer move under Scenario X because the new total route time is less than the sum of record and deviation, and the route time standard deviation decreases. However, this is not a valid customer move for Scenarios Y and Z. The new total route time is more than the total route time segment value violating Scenario X, and the route time for a1 is more than the segment value for a0 violating Scenario Z. Suppose a customer move from c0 to a0 changes the two routes to a2 and c2 with the route times as 6.2 hours and 5.6 hours, respectively. The new total route time is 17.8 hours, and the new route time standard deviation is 0.31 hours. This is a valid customer move under Scenarios X and Y because the new total route time is less than the sum of record and deviation and less than the total route time segment value, and the route time standard deviation decreases. However, this is not a valid customer move for Scenario Z. The route time for a2 is more than the segment value for a0 violating Scenario Z. Suppose a customer move from c0 to a0 changes the two routes to a3 and c3 with the route times as 6 hours and 6.5 hours, respectively. The new total route

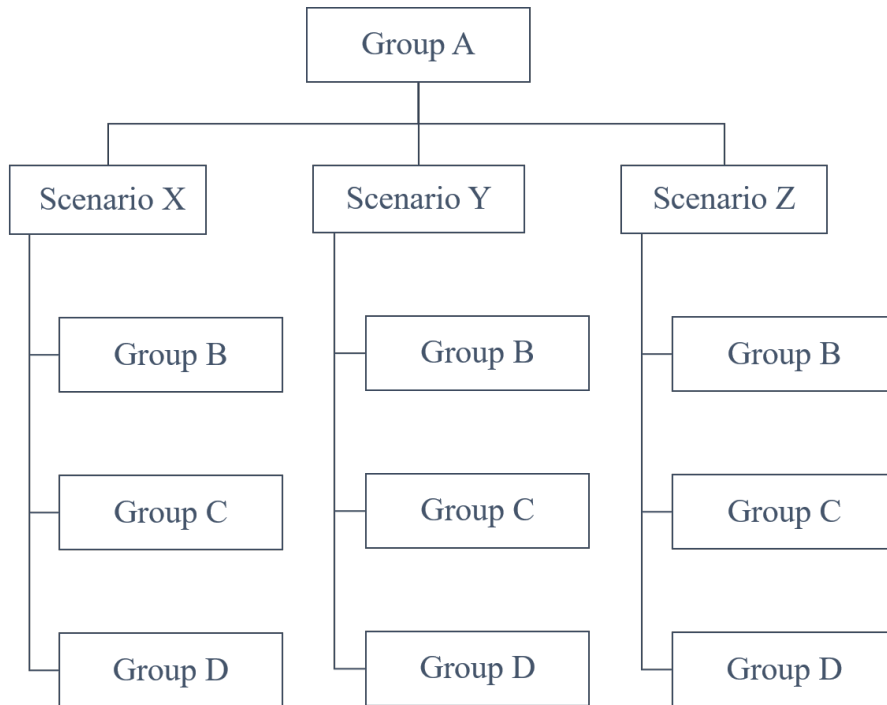


Figure 3.2: A flowchart showing the relation between the four groups and three scenarios.

time is 18.5 hours, and the new route time standard deviation is 0.29 hours. This is a valid customer move under Scenarios X and Z because the new total route time is less than the sum of record and deviation, the route times for a3 and c3 are less than the segment values for a0 and c0, respectively, and the route time standard deviation decreases. However, this is not a valid customer move for Scenario Y. The new total route time is more than the total route time segment value violating Scenario Y.

We apply all three modified versions of the RTR travel algorithm to each of the 1000 C&W solutions of every instance under random traffic conditions (Group A). For each modified version of the RTR travel algorithm, each C&W solution produces 11 different solution sequences, one for each value (0 through 10) of the deviation

parameter r . The RTR travel algorithm stops when it is not possible to find a solution that decreases the route time standard deviation and satisfies all criteria of the particular modification (scenario) of the RTR travel algorithm. Out of the 11 sequences starting from any C&W solution under any particular scenario, we retain three solutions. First, we retain the best intermediate solution (any solution other than the initial C&W solution and the final solution) in terms of the shortest total route time. For each instance and each scenario, we have 1000 best intermediate total route time solutions (Group B). Second, we retain the best final solution in terms of the shortest total route time. For each instance and each scenario, we have 1000 best final total route time solutions (Group C). Third, we retain the best final solution in terms of the smallest route time standard deviation. For each instance and each scenario, we have 1000 best final route time standard deviation solutions (Group D). We do not retain the best intermediate solution in terms of the smallest route time standard deviation because the standard deviation will always be the smallest in the final solution of each sequence. The reason is that the objective of the RTR travel algorithm is to minimize the route time standard deviation. In Figure 3.2, we show the relation between the four groups and three scenarios using a flowchart. The C&W solutions on the random traffic induced instances form Group A. Each of the three Scenarios X, Y, and Z of the RTR travel algorithm are applied to each solution from Group A to obtain three solutions, one each in Groups B, C, and D. For each solution, we then calculate the total operating and delivery cost under random traffic conditions using the parameters given in Table 3.4. To calculate the total cost for each solution, we need the route times of each driver

Instance	Group A	Group B	Group C	Group D
E-n22-c6000	740.98	737.46	737.19	747.78
E-n23-c4500	1051.56	1000.87	1004.88	1043.74
E-n30-c4500	1007.10	977.38	970.48	983.18
E-n33-c8000	1395.63	1388.76	1392.75	1521.67
E-n51-c160	1194.81	1170.38	1166.95	1175.44
E-n76-c220	1545.11	1538.11	1531.57	1554.61
E-n76-c180	1630.70	1626.05	1623.57	1678.71
E-n76-c140	2021.21	2005.65	1997.69	1975.17
E-n76-c100	2653.86	2649.10	2646.53	2671.40
E-n101-c200	1863.45	1850.06	1836.63	1935.02
E-n101-c112	2594.27	2590.75	2587.77	2553.66

Bold indicates best solution

Table 3.6: Average total cost under random traffic conditions for Scenario X.

and the total route length. We already have the route times for each driver from the C&W savings algorithm for Group A, and from modifications of the RTR travel algorithm for Groups B, C, and D. The total route length is computed from the original instances without the traffic modifications based on the solution. Finally, for each instance, each scenario, and each group, we find the average of the total cost under random traffic conditions of the 1000 solutions. The total cost value is given in dollars.

In Tables 3.6, 3.7, and 3.8, we present the average total cost under random traffic conditions of the 1000 solutions for each instance and each group for Scenarios X, Y, and Z, respectively. A scenario represents the modified version of the RTR travel algorithm applied to the solution of the C&W algorithm to reduce the route time standard deviation. Group A corresponds to the C&W solutions; the average total cost values for Group A are the same in all three scenarios. For all three scenarios, the attempt to reduce the route time standard deviation pays off because

Instance	Group A	Group B	Group C	Group D
E-n22-c6000	740.98	737.46	737.22	746.78
E-n23-c4500	1051.56	997.57	989.91	992.19
E-n30-c4500	1007.10	977.88	967.33	972.92
E-n33-c8000	1395.63	1386.97	1384.72	1402.00
E-n51-c160	1194.81	1171.39	1167.62	1133.95
E-n76-c220	1545.11	1535.47	1523.73	1520.45
E-n76-c180	1630.70	1626.09	1623.24	1629.33
E-n76-c140	2021.21	2007.44	1998.10	1973.95
E-n76-c100	2653.86	2649.38	2647.04	2671.25
E-n101-c200	1863.45	1848.39	1828.70	1859.06
E-n101-c112	2594.27	2590.82	2587.91	2553.27

Bold indicates best solution

Table 3.7: Average total cost under random traffic conditions for Scenario Y.

Instance	Group A	Group B	Group C	Group D
E-n22-c6000	740.98	737.45	737.19	747.09
E-n23-c4500	1051.56	993.15	988.66	1015.55
E-n30-c4500	1007.10	976.30	963.77	972.64
E-n33-c8000	1395.63	1386.97	1388.63	1471.47
E-n51-c160	1194.81	1167.26	1164.33	1161.32
E-n76-c220	1545.11	1535.97	1525.45	1543.83
E-n76-c180	1630.70	1625.86	1623.17	1672.28
E-n76-c140	2021.21	2005.45	1997.26	1974.16
E-n76-c100	2653.86	2649.10	2646.53	2671.29
E-n101-c200	1863.45	1847.69	1829.93	1905.80
E-n101-c112	2594.27	2590.70	2587.58	2554.12

Bold indicates best solution

Table 3.8: Average total cost under random traffic conditions for Scenario Z.

none of the average total cost values from Group A is the best for any instance.

After we have obtained an initial solution by reducing the total route time using a fast and easy to implement algorithm such as the C&W savings algorithm, there is still room to improve a solution in terms of the total cost by reducing the route time standard deviation.

In Table 3.9, we present the best average total cost under random traffic

Instance	Group A	Best	Scenario	Percent Savings
E-n22-c6000	740.98	737.19	X and Z	0.51
E-n23-c4500	1051.56	988.66	Z	5.98
E-n30-c4500	1007.10	963.77	Z	4.30
E-n33-c8000	1395.63	1384.72	Y	0.78
E-n51-c160	1194.81	1133.95	Y	5.09
E-n76-c220	1545.11	1520.45	Y	1.60
E-n76-c180	1630.70	1623.17	Z	0.46
E-n76-c140	2021.21	1973.95	Y	2.34
E-n76-c100	2653.86	2646.53	X and Z	0.28
E-n101-c200	1863.45	1828.70	Y	1.86
E-n101-c112	2594.27	2553.27	Y	1.58

Table 3.9: Best average total cost under random traffic conditions across all three scenarios and percent savings compared to Group A.

conditions for each instance by comparing the best values across all three scenarios.

We also present the percent savings of the best average total cost compared to the average total cost of Group A, which is our benchmark. In six of eleven instances, Scenario Y provided the best solution. In the remaining five instances, Scenario Z provided the best solution. Scenario X provided the best solution in two of the five instances. For the two instances where Scenarios X and Z gave the same best solution, the additional constraints on the route times of each vehicle in the modified RTR travel algorithm did not have any effect. The percent savings is calculated as $100 \times (\text{C\&W Cost} - \text{Best Cost}) / \text{C\&W Cost}$. The maximum percent savings is 5.98% for instance E-n23-c4500, and the minimum percent savings is 0.28% for instance E-n76-c100. The average percent savings for the 11 instances is 2.25%.

3.5 Contribution of Standard Deviation to Cost

We have shown that total route length and route length standard deviation are both important in reducing the total operating and delivery costs under random traffic conditions. We now want to understand the separate contributions of these two factors. We want to determine whether a higher fraction of solutions with the best total cost are the ones with lower total route length or lower route length standard deviation.

We use the weighted savings modification of the C&W algorithm with the top three savings as described in Section 3.3 to generate 1000 solutions for each instance with the objective of minimizing the total length. The breakdown of the 1000 solutions for each instance in terms of the number of vehicles required to serve the customers is given in Table 3.3. We choose the 100 best solutions with the lowest total route length (Bucket RL) and 100 best solutions with the lowest route length standard deviation (Bucket SD). In total, we have chosen at most 200 solutions out of 1000 because some solutions will appear in both Bucket RL and Bucket SD. For each of the 200 solutions, we divide the route length for each vehicle by the speed of the vehicle with no traffic to get the route time for each vehicle. We randomly increase the route time for each vehicle by $t\%$, where $t \sim \text{Uniform}(0, 10)$, to capture random traffic conditions during actual service by a vehicle. We repeat the randomization 1000 times for each of the 200 solutions. For each of the 1000 randomized versions, we calculate the total operating and delivery costs under random traffic conditions using the parameters given in Table 3.4. Driving cost and fixed cost are invariant

Instance	Bucket RL	Bucket SD	Both Buckets
E-n22-c6000	20	12	12
E-n23-c4500	18	20	18
E-n30-c4500	19	18	17
E-n33-c8000	20	2	2
E-n51-c160	19	20	19
E-n76-c220	13	18	11
E-n76-c180	18	14	12
E-n76-c140	10	20	10
E-n76-c100	10	13	3
E-n101-c200	11	12	3
E-n101-c112	3	18	1

Table 3.10: Number of solutions from the respective buckets for the 20 best total cost solutions under random traffic conditions.

for the randomized versions of each solution. The total wages will be specific to each randomized version. For each of the 200 solutions, we find the average total cost among the 1000 randomized versions. We then choose the 20 best solutions with the lowest total cost under random traffic conditions among the 200 solutions and determine whether these are from Bucket RL or Bucket SD.

In Table 3.10, we give the number of solutions for each instance that are from Bucket RL, Bucket SD, or both buckets among the 20 best solutions with the lowest total cost under random traffic conditions. In four of eleven instances, there are more solutions from Bucket RL than from Bucket SD. In the remaining seven instances, there are more solutions from Bucket SD than from Bucket RL. For the seven instances, greater number of the best solutions in terms of the total cost under random traffic conditions are also the best in terms of route length standard deviation. For nine of eleven instances, the number of solutions in both buckets is close to the maximum possible, which is the minimum number of solutions among

the two buckets. However, for instances E-n76-100 and E-n101-c200, the number of solutions in both buckets is only three compared to the maximum possible number of 10 and 11, respectively. Important observations can be made from Table 3.10 and the last column of Table 3.9. The percent savings in total cost from the RTR travel algorithm tend to be larger for those instances where we have a greater number of solutions from Bucket SD and the number of solutions in both buckets is close to the maximum possible. This observation makes sense because the RTR travel algorithm is able to substantially reduce the total cost under random traffic conditions for those instances where the total cost is driven by the standard deviation of the routes with smaller total route lengths.

3.6 Conclusions and Future Directions

It is not always possible to accurately judge the quality of a route with respect to total cost by only assessing the total route length or the total route time. We used standard CVRP instances from the literature to show that, under random traffic conditions, the standard deviation of routes has an impact on total route time and on total operating and delivery costs. For delivery companies, it is important to understand the objective function that third-party routing software programs are using. We implemented fast and easy modifications to the solutions of the C&W savings algorithm. We used modified versions of the RTR travel algorithm to reduce the standard deviation of routes. We showed that the improved solutions in terms of standard deviation have also improved in terms of total cost. Finally, we showed

that the RTR travel algorithm could make a substantial impact if the total cost of a solution is driven by the standard deviation.

There are big gaps in the literature in assessing the impact of the standard deviation of routes on overall route quality. There could be improvements in heuristics that further improve the total cost savings by adjusting the standard deviation of the routes in a smarter way. We need to understand the properties of those instances that have the potential of producing solutions with lower total cost and smaller standard deviation. It might be useful for the delivery companies if we could quantify such instances using some easy to understand metrics. These metrics would potentially indicate when it would be worthwhile to try to reduce the standard deviation in order to produce substantial cost reductions.

Chapter 4: Data-Driven Estimation of the Route Length for the Close-Enough Traveling Salesman Problem

4.1 Introduction

The Close-Enough Traveling Salesman Problem (CETSP) is a variant of the Traveling Salesman Problem (TSP). Both are defined on a Euclidean plane. The TSP requires a salesman to visit customers at their exact locations starting and ending at the depot. In the CETSP, every customer has a service region and is considered visited when the salesman visits any point in the customer's service region. A service region is assumed to be a circular disk centered at the customer location with a specified radius. Similar to the TSP, the objective of the CETSP is to visit all customers in the shortest distance traveled starting and ending at the depot. The TSP is a special case of the CETSP when all customer radii are zero, making the CETSP at least as difficult to solve as the TSP. In order to solve an instance of the CETSP, it is not enough to determine the sequence in which the customers are visited. We must also determine the locations at which these customers are visited within their respective service regions. In Figure 4.1, we show an example of a CETSP with 12 customers denoted by C_1, \dots, C_{12} and a depot denoted by C_0 .

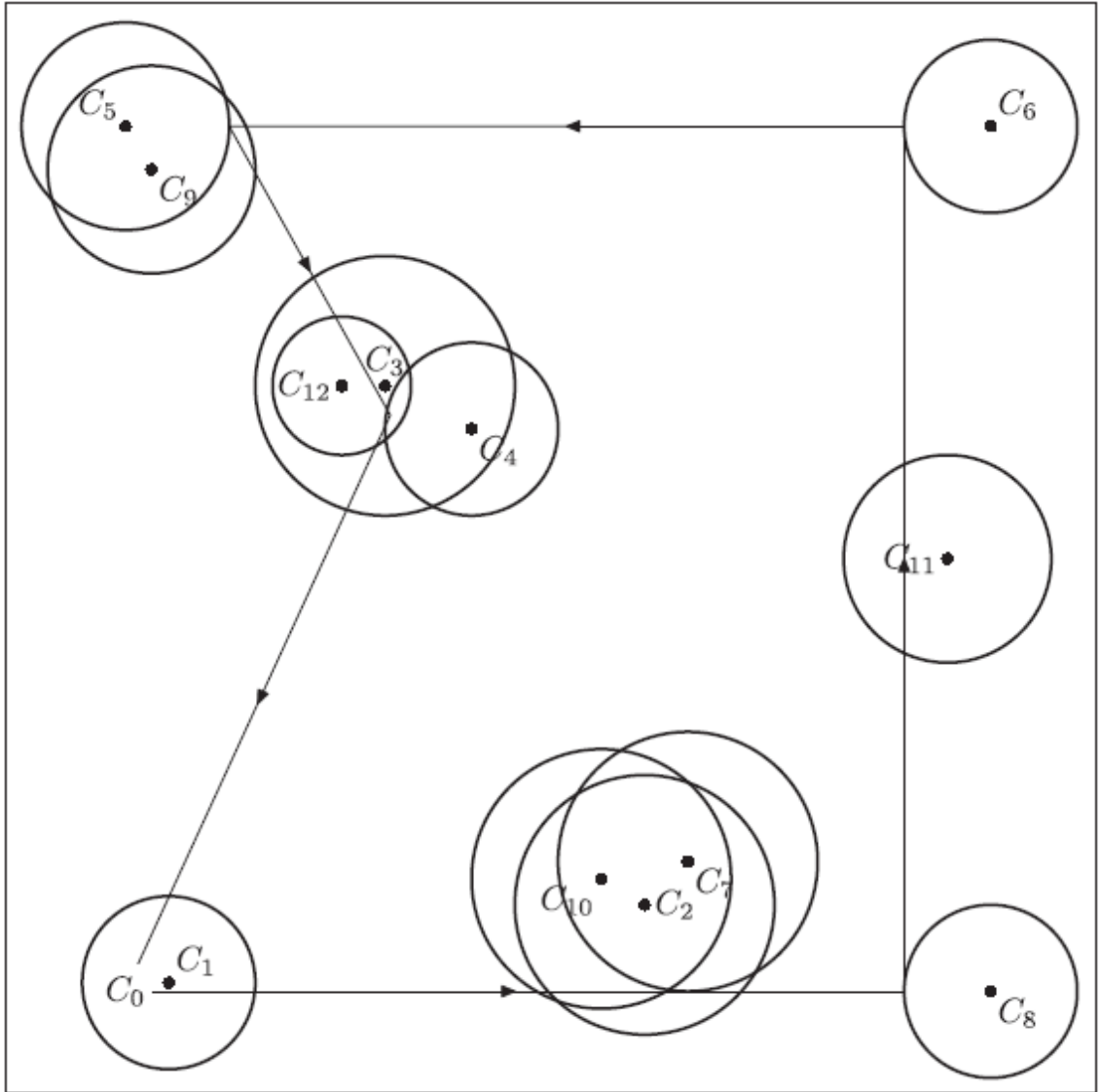


Figure 4.1: An example of a CETSP with 12 customers.

The service region is given by a circle centered at the customer's location. A feasible CETSP tour is shown by the solid lines with arrows. The tour passes through at least one point in the service region of each customer. In practice, applications such as meter reading by utility companies using radio frequency identification (RFID) technology and surveillance by a pilot in an airplane or an unmanned drone can be modeled as a CETSP. In both of these applications, it is sufficient to get close

enough to the target and not exactly visit the target.

Over the years, many exact and heuristic algorithms have been developed to solve the CETSP. [Gulczynski et al. \(2006\)](#) and [Dong et al. \(2007\)](#) developed several heuristics that first selected a set of supernodes such that each customer service region contains at least one supernode. Then a TSP tour was generated through these supernodes. [Mennell \(2009\)](#), [Mennell et al. \(2011\)](#), and [Wang et al. \(2019\)](#) developed several heuristics based on Steiner zones. [Silberholz and Golden \(2007\)](#), [Yuan et al. \(2007\)](#), and [Yang et al. \(2018\)](#) developed heuristics based on genetic algorithms. [Carrabs et al. \(2017\)](#) provided tight lower and upper bounds. [Behdani and Smith \(2014\)](#) and [Coutinho et al. \(2016\)](#) developed exact approaches using a discretization scheme and a branch-and-bound algorithm, respectively.

In some applications, it may not be necessary to find the routes of the CETSP using exact or heuristic algorithms. For example, routing companies participating in competitive bidding might need to respond to a large number of requests regarding route costs in a very short amount of time. In such cases, it is sufficient to estimate the route lengths using information about the actual instances. Also, during post-disaster aerial surveillance planning or using drones to deliver emergency medical supplies, route length estimation would quickly need to assess whether the duration to cover a region of interest would exceed the drone battery life. Route-length estimation for an instance would approximate the route length generated by a particular heuristic and not necessarily approximate the optimal or the best-known solution. The variables in the estimation model would capture the features of the instance that would be exploited by a specific heuristic. For practical purposes,

routing companies need to know the actual costs that would be incurred using a specific algorithm and not the optimal costs even if the optimal costs are lower than the actual costs. The estimation model would be unique to the algorithm that was applied even though the general framework would apply to any routing algorithm.

Estimating TSP route lengths has been studied in the operations research literature since the late 1950s. [Beardwood et al. \(1959\)](#) were among the first to estimate route lengths using analytically derived formulas. These formulas were improved by [Christofides and Eilon \(1969b\)](#) using empirically estimated parameters. [Golden and Alt \(1979\)](#) provided interval estimates of the optimal solution. [Chien \(1992\)](#), [Kwon et al. \(1995\)](#), [Hindle and Worthington \(2004\)](#), and [Cavdar and Sokol \(2015\)](#) used parameters for the shape of the area covering the customers and the depot, the distance between customers, and the coordinates of the customers. [Nicola et al. \(2019\)](#) provided a detailed regression-based estimation method. However, estimating the route length for a CETSP has not been addressed in the literature. We will use a regression model to estimate the CETSP route lengths.

4.2 The Regression Model

The Steiner zone variable neighborhood search (SZVNS) heuristic given in [Wang et al. \(2019\)](#) finds high-quality solutions to instances of the CETSP. There are three steps in the SZVNS heuristic. First, Steiner zones of degree three and less that are not dominated by other Steiner zones of degree three and less are detected. Second, a set covering problem is solved to choose a subset of Steiner zones such that

Independent Variable	Definition
n	Number of nodes
A	Area of the smallest rectangle covering all nodes
MinP	Minimum distance across all pairs of nodes
MaxP	Maximum distance across all pairs of nodes
VarP	Variance of distances across all pairs of nodes
SumMinP	Sum of distances to the nearest neighbor of each node
SumMaxP	Sum of distances to the farthest neighbor of each node
MinM	Minimum distance to the average node
MaxM	Maximum distance to the average node
SumM	Sum of distances to the average node
VarM	Variance of distances to the average node
VarX×VarY	Product of variances of the nodes across two axes
AvgR	Average radius of the customer service regions
VarR	Variance of the radii of the customer service regions
SZ	Number of Steiner zones of degree three and less that are not dominated by other Steiner zones of degree three and less

Table 4.1: Definitions of the independent variables for the linear regression model.

each customer service region has at least one selected Steiner zone. Third, different search operators are incorporated into a variable neighborhood search framework to improve solutions.

We build a linear regression model to estimate the route length generated by the SZVNS heuristic for the CETSP. The model can be represented by $y_i = \hat{\beta}_1 + \hat{\beta}_2 \times n_i + \hat{\beta}_3 \times A_i + \hat{\beta}_4 \times \text{MinP}_i + \hat{\beta}_5 \times \text{MaxP}_i + \hat{\beta}_6 \times \text{VarP}_i + \hat{\beta}_7 \times \text{SumMinP}_i + \hat{\beta}_8 \times \text{SumMaxP}_i + \hat{\beta}_9 \times \text{MinM}_i + \hat{\beta}_{10} \times \text{MaxM}_i + \hat{\beta}_{11} \times \text{SumM}_i + \hat{\beta}_{12} \times \text{VarM}_i + \hat{\beta}_{13} \times (\text{VarX} \times \text{VarY})_i + \hat{\beta}_{14} \times \text{AvgR}_i + \hat{\beta}_{15} \times \text{VarR}_i + \hat{\beta}_{16} \times \text{SZ}_i$, where $y_i = \mathbb{E}(Y_i)$, $\hat{\beta}_k = \mathbb{E}(\beta_k)$, $k \in \{1, \dots, 16\}$, i denotes a CETSP instance, and $\mathbb{E}()$ denotes the expected value. The dependent variable Y_i is the route length generated by the SZVNS heuristic. In Table 4.1, we give the definitions of the independent variables for the linear regression model. Nodes represent customers and the depot. n and

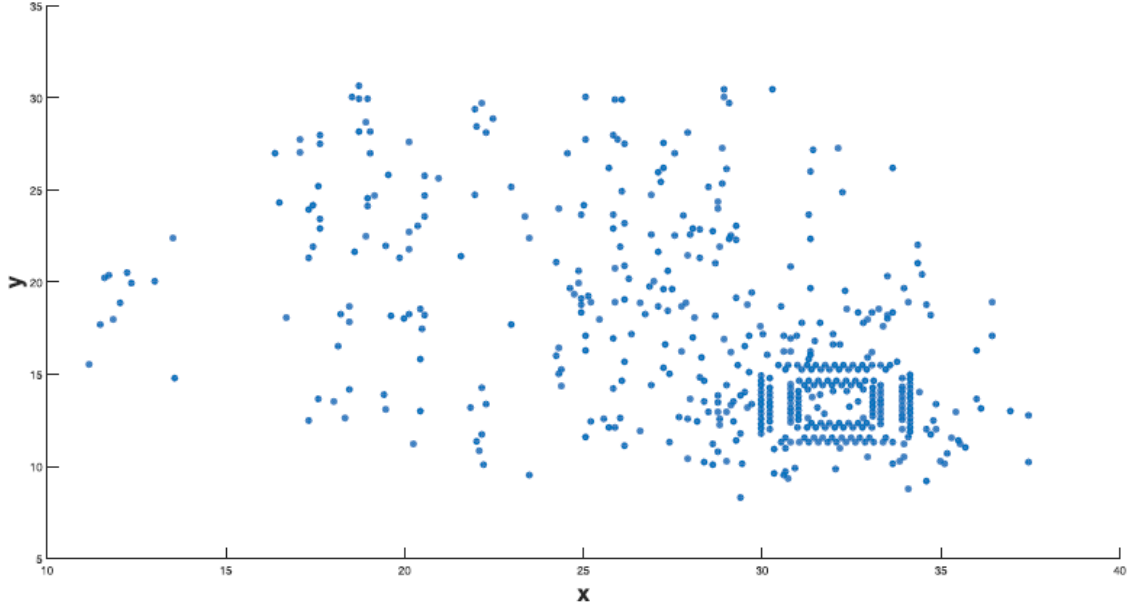


Figure 4.2: Node locations of instance d493 from the second group of 62 instances.

A capture the size of the instance. MinP , MaxP , VarP , SumMinP , and SumMaxP capture the distances between nodes. MinM , MaxM , SumM , and VarM capture the distances to the average node. $\text{VarX} \times \text{VarY}$ captures the spread of the instance across the two axes. AvgR and VarR capture the mean and variability of the radii of the customer service regions. The service region radius of a depot is always zero. SZ captures the feature of the instance that is exploited by the SZVNS heuristic.

4.3 Regression Data and Model Fit Measures

We use the 842 CETSP benchmark instances and their route lengths generated using the SZVNS heuristic given in Wang et al. (2019). The instances can be divided into two groups. The first group has 780 instances with the node locations generated randomly. All customers in an instance have the same radius for the

service regions. The second group has 62 instances with the node locations generated in various structured ways. Some instances have different radii for the customer service regions. In Figure 4.2, we show the node locations of instance d493 from the second group. This instance has nodes forming concentric rectangles and nodes distributed randomly.

We use mean percentage error (MPE) and mean absolute percentage error (MAPE) to assess the quality of the approximation of the CETSP route lengths from the regression model (y_i) with respect to the route lengths from the SZVNS heuristic (Y_i). MPE and MAPE are defined as $100 \times (\sum_{i=1}^N (Y_i - y_i) / Y_i) / N$ and $100 \times (\sum_{i=1}^N |Y_i - y_i| / Y_i) / N$, respectively, where N denotes the number of instances. A value of MPE close to zero indicates that there is almost an equal distribution of instances with route lengths being overestimated ($Y_i < y_i$) and underestimated ($Y_i > y_i$). The value of MAPE is always positive, and a low value indicates that the route length estimates are close to the SZVNS route lengths for most of the instances.

We use R^2 , adjusted R^2 , Studentized residuals, outliers, normal probability plots, Shapiro-Wilk hypothesis test for normality, Mallows's C_p , and Bayesian information criterion (BIC) to assess the quality of the model fits. R^2 increases as extra variables are added to the model. Adjusted R^2 can decrease if the penalty for adding an extra variable to the model is more than the improvement to the model. Residuals ($Y_i - y_i$) have a mean of zero. Studentized residuals are scaled residuals with unit variance. The Studentized residual plot should show a horizontal band of points around the horizontal axis at zero to denote the absence of any heteroscedas-

ticity in the model. The plot also shows whether the linear form of the model is adequate to capture all underlying patterns. Any data point (instance) with a Studentized residual value of greater than 2 or less than -2 can be considered as an outlier and may be removed from the model for a better fit. The normal probability plot matches the quantiles of the Studentized residuals with the quantiles of the standard normal distribution. The Shapiro-Wilk hypothesis test indicates whether (null hypothesis) or not (alternative hypothesis) the Studentized residuals follows a standard normal distribution. Mallows's C_p and BIC are used in the context of model selection where the goal is to find the best model involving a subset of the independent variables. Mallows's C_p addresses the issue of model overfitting by penalizing for adding extra variables. Mallows's C_p value should be greater but close to the number of independent variables in the model (p) to indicate the absence of overfitting. Mallows's C_p is equivalent to Akaike information criterion (AIC) for linear regression. Both AIC and BIC penalize a model for having more independent variables. However, BIC uses a larger penalty as the number of instances increases. The lower the value of BIC, the better is the model fit.

4.4 Regression Results

4.4.1 Results on all 842 Instances

In Table 4.2, we present the regression results for all 842 instances with and without outliers. The regression model with outliers (842 instances) has an adjusted R^2 value of 0.978 which might indicate a very good model fit. VarR is insignificant

Coefficient	With outliers	Without outliers
Intercept (β_1)	-49.226***	-18.970***
n (β_2)	0.197***	0.059***
A (β_3)	-0.006*	0.006***
MinP (β_4)	-7.580***	-0.053
MaxP (β_5)	7.942***	4.696***
VarP (β_6)	-0.362***	-0.263***
SumMinP (β_7)	0.190***	0.190***
SumMaxP (β_8)	-0.005*	0.006***
MinM (β_9)	4.742***	0.343
MaxM (β_{10})	-3.259*	-1.353***
SumM (β_{11})	0.014*	-0.012***
VarM (β_{12})	0.357***	0.389***
VarX \times VarY (β_{13})	0.00002***	0.000003***
AvgR (β_{14})	-17.039***	-12.876***
VarR (β_{15})	0.056	0.156***
SZ (β_{16})	-0.009***	-0.005***
Number of instances	842	811
Adjusted R ²	0.978	0.999
MPE	1.530%	0.327%
MAPE	21.141%	9.821%

*p<0.1; *p<0.05; **p<0.01; ***p<0.001

Table 4.2: Regression results for the 842 instances with and without outliers.

at the 10% level. A, SumMaxP, MaxM, and SumM are significant at the 5% level. All other variables are significant at the 0.1% level. In Figure 4.3, we give the Studentized residual plot of 842 instances. The lines show the Studentized residuals at values of 2 and -2. The Studentized residual plot show some linear trend which may be due to outliers. There are 31 instances from the second group of 62 instances with Studentized residual values greater than 2 or less than -2. The Studentized residuals have values from 18 to -21. In Figure 4.4, we give the histogram of Studentized residuals of 842 instances. The histogram shows that there is almost an equal distribution of instances with positive and negative residuals. This is also

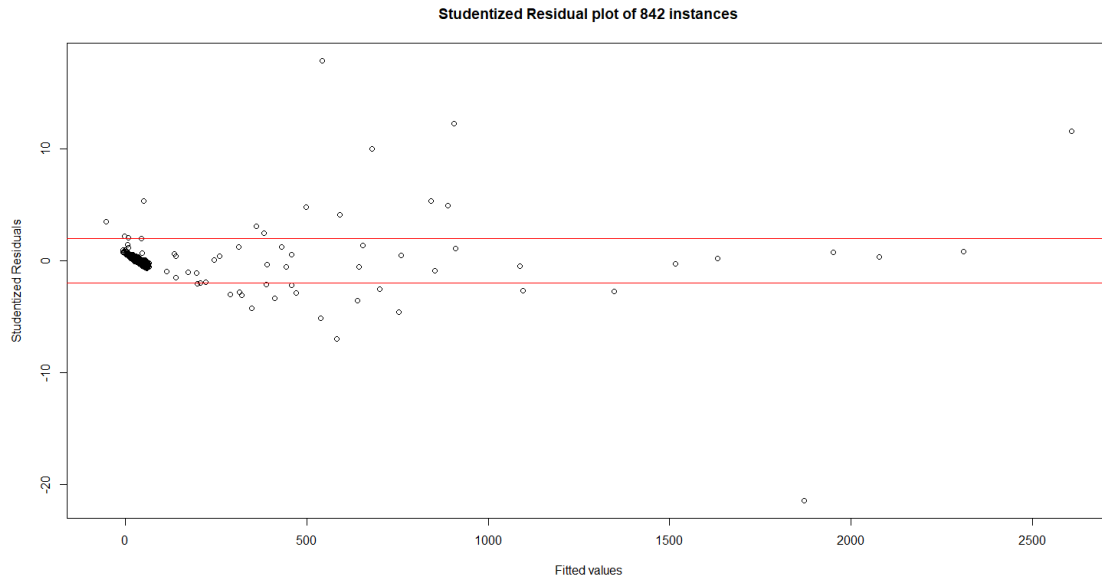


Figure 4.3: Studentized residual plot of 842 instances. The lines indicate the Studentized residual values of 2 and -2 .

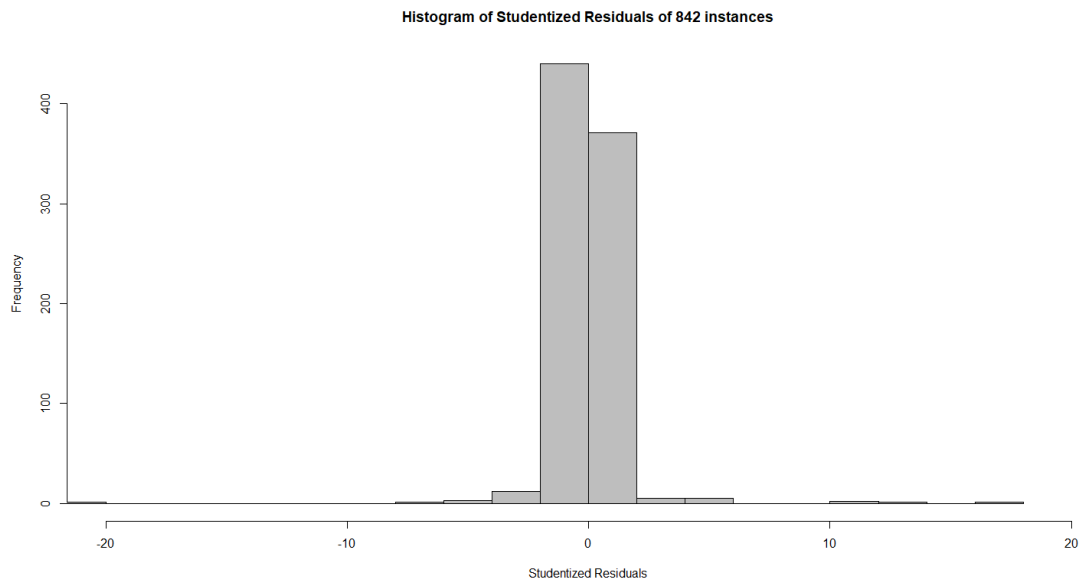


Figure 4.4: Histogram of Studentized residuals of 842 instances.

indicated by the MPE value of 1.530% which is close to zero. In Figure 4.5, we give the normal probability plot of Studentized residuals of 842 instances. Both the histogram and the normal probability plot show that the Studentized residuals do

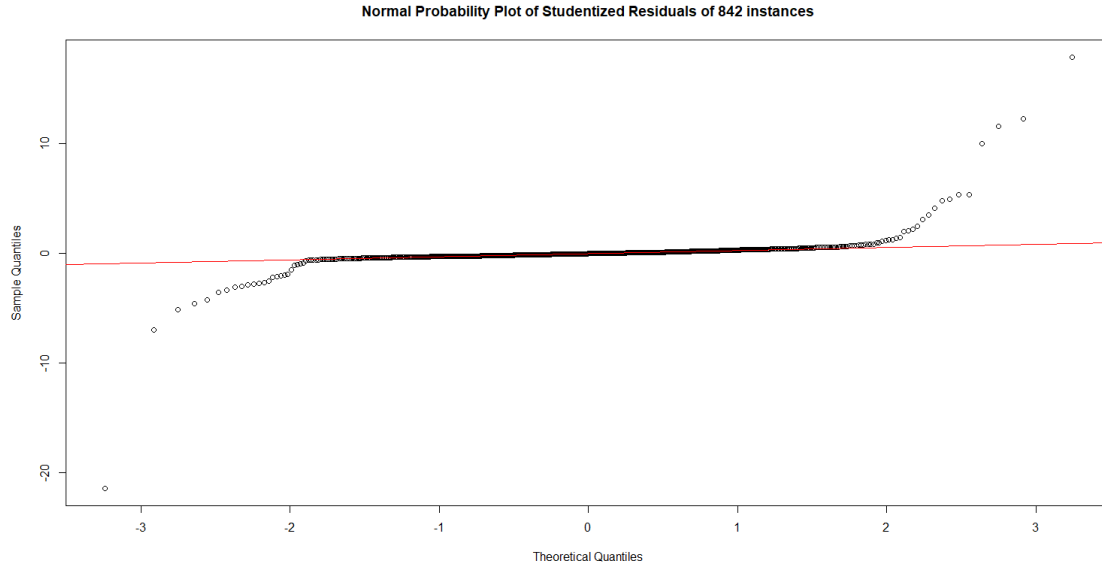


Figure 4.5: Normal probability plot of Studentized residuals of 842 instances.

not follow a standard normal distribution. The non-normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which rejects the null hypothesis at the 0.1% significance level. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 21.141% from the SZVNS route lengths. Even though the model has a very high adjusted R^2 value, MAPE and the result of the Shapiro-Wilk hypothesis test indicate that the model does not perform well, specifically for the second group of 62 instances with 31 outliers.

We now examine the results of the model without the outliers (Table 4.2). The regression model without outliers (811 instances) has an adjusted R^2 value of 0.999 which might indicate a very good model fit. MinP and MinM are not significant at the 10% level. All other variables are significant at the 0.1% level. The MPE value of 0.327% is close to zero which indicates that there is almost an equal distribution

of instances with positive and negative residuals. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 9.821% from the SZVNS route lengths. Even though the average route-length prediction error dropped from 21.141% to 9.821% after removing the 31 outlier instances, the prediction error of about 10% might still be above an acceptable limit for a routing manager. Therefore, it is worthwhile to examine the performance of the regression model separately on the first group of 780 instances and on the second group of 62 instances.

Nicola et al. (2019) provided detailed regression-based estimation models for the TSP and similar routing problems. The estimation models use around 400 instances. The quality of the estimation models was assessed using MPE and MAPE, in addition to adjusted R^2 . For many of the regression models they studied, the adjusted R^2 values were in the range 0.97 to 0.99 with MAPE values greater than 10%. These observations indicated that instances with different geometric properties can have large errors in a practical sense if their route lengths are estimated using the same model even if the model has a very high adjusted R^2 . However, splitting up the instances according to their geometric properties and building specific estimation models significantly lowered the MAPE values.

4.4.2 Results on the Second Group of 62 Instances

In Table 4.3, we present the regression results for the second group of 62 instances with and without outliers. The regression model with outliers (62 instances)

Coefficient	With outliers	Without outliers
Intercept (β_1)	49.407	60.461
n (β_2)	0.099	0.029
A (β_3)	0.003	0.0008
MinP (β_4)	-19.059	-20.600*
MaxP (β_5)	8.129	9.245**
VarP (β_6)	-0.112	-0.099
SumMinP (β_7)	0.184*	0.166**
SumMaxP (β_8)	0.008	0.010
MinM (β_9)	26.047	12.840
MaxM (β_{10})	-7.918	-9.189
SumM (β_{11})	-0.020	-0.024
VarM (β_{12})	0.112	0.019
VarX \times VarY (β_{13})	0.000005	0.000007
AvgR (β_{14})	-17.349***	-19.728***
VarR (β_{15})	-0.038	0.202
SZ (β_{16})	-0.010***	-0.008***
Number of instances	62	57
Adjusted R ²	0.954	0.978
MPE	-1.237%	1.184%
MAPE	26.546%	25.638%

*p<0.1; **p<0.05; ***p<0.01; ****p<0.001

Table 4.3: Regression results for the second group of 62 instances with and without outliers.

has an adjusted R² value of 0.954 which might indicate a very good model fit. AvgR and SZ are significant at the 0.1% level. SumMinP is significant at the 5% level. MaxP and MinM are significant at the 10% level. All other variables are not significant at the 10% level. In Figure 4.6, we give the Studentized residual plot for the second group of 62 instances. The lines show the Studentized residuals at values of 2 and -2. The Studentized residual plot shows some linear trend which may be due to outliers. There are five instances with Studentized residual values greater than 2 or less than -2. The Studentized residuals have values from 5 to -6. In Figure 4.7, we give the histogram of Studentized residuals for the second group of 62 in-

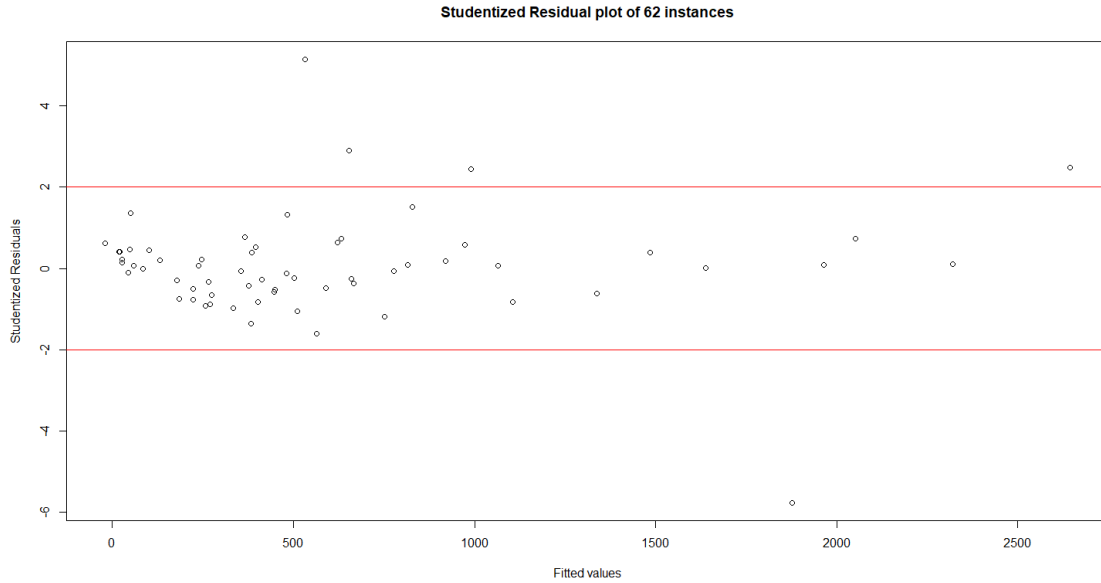


Figure 4.6: Studentized residual plot for the second group of 62 instances. The lines indicate the Studentized residual values of 2 and -2 .

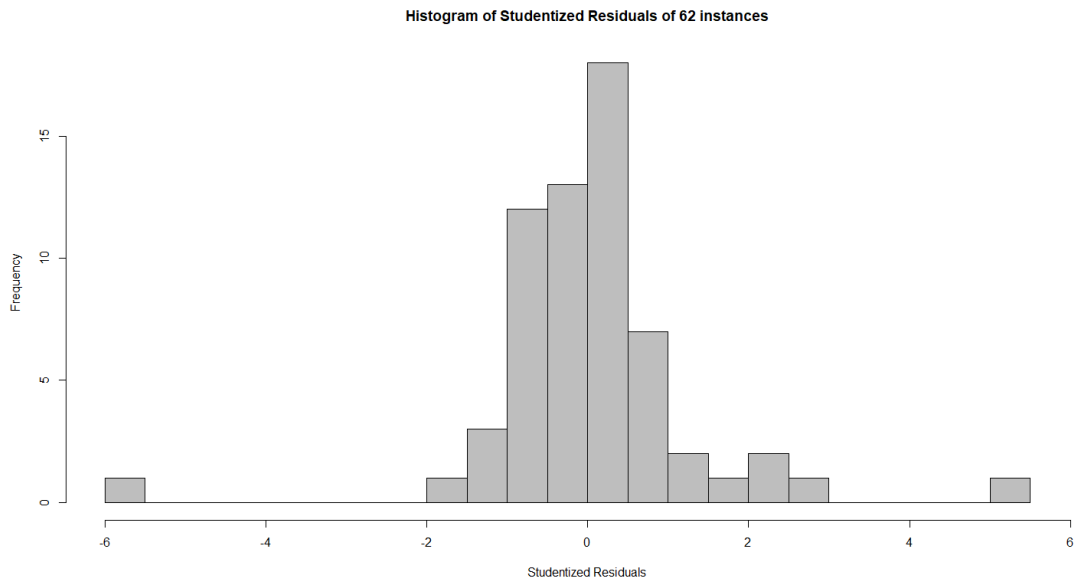


Figure 4.7: Histogram of Studentized residuals for the second group of 62 instances.

stances. The histogram shows that there is almost an equal distribution of instances with positive and negative residuals. This is also indicated by the MPE value of

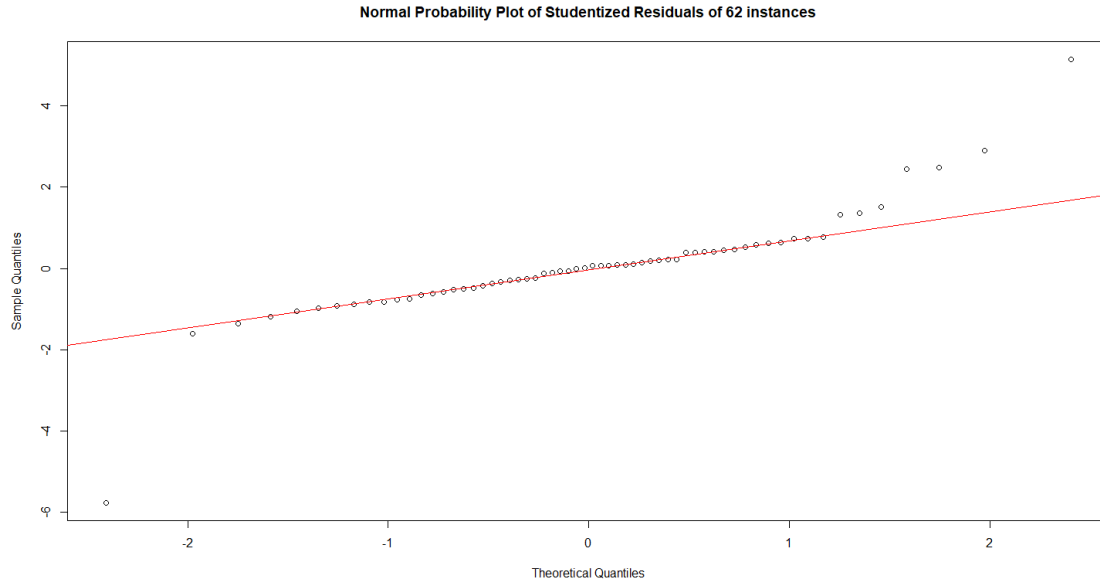


Figure 4.8: Normal probability plot of Studentized residuals for the second group of 62 instances.

-1.237% which is close to zero. In Figure 4.8, we give the normal probability plot of Studentized residuals for the second group of 62 instances. Both the histogram and the normal probability plot show that the Studentized residuals do not follow a standard normal distribution. The non-normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which rejects the null hypothesis at the 0.1% significance level. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 26.546% from the SZVNS route lengths. Even though the model has a very high adjusted R^2 value, MAPE and the result of the Shapiro-Wilk hypothesis test indicate that the model does not perform well.

We now examine the results of the model without the outliers (Table 4.3). The regression model without outliers (57 instances) has an adjusted R^2 value of

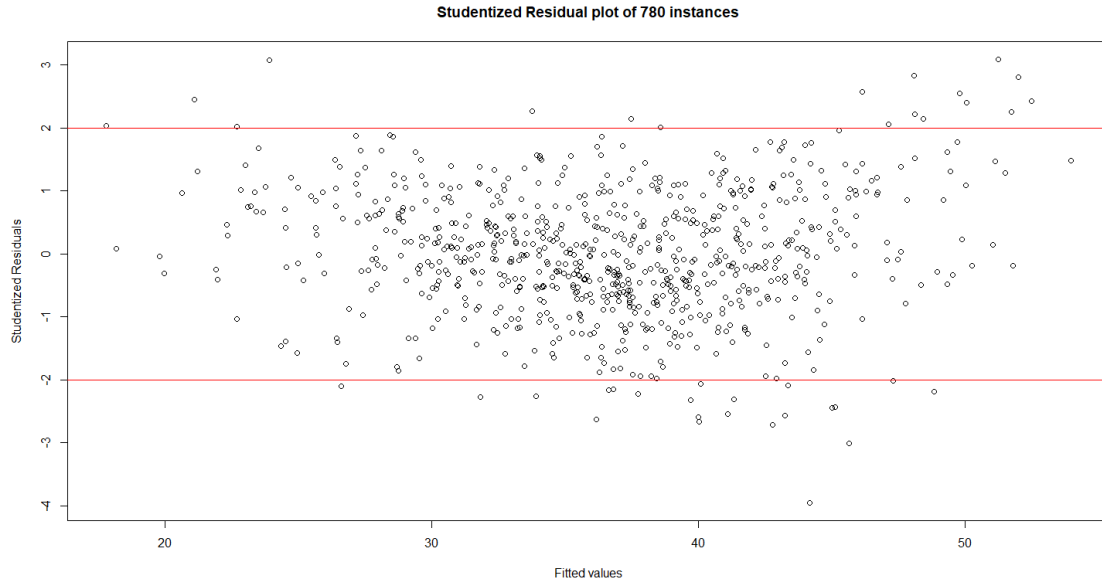


Figure 4.9: Studentized residual plot for the first group of 780 instances. The lines indicate the Studentized residual values of 2 and -2 .

0.978 which might indicate a very good model fit. AvgR and SZ are significant at the 0.1% level. MaxP and SumMinP are significant at the 1% level. MinP is significant at the 5% level. All other variables are not significant at the 10% level. The MPE value of 1.184% is close to zero which indicates that there is almost an equal distribution of instances with positive and negative residuals. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 25.638% from the SZVNS route lengths. The average route-length prediction error did not change much after removing the five outlier instances. The prediction error of nearly 26% is very high for practical purposes.

Coefficient	With outliers
Intercept (β_1)	15.231***
n (β_2)	0.225
A (β_3)	0.048***
MinP (β_4)	-0.654***
MaxP (β_5)	0.224*
VarP (β_6)	0.106
SumMinP (β_7)	0.361***
SumMaxP (β_8)	0.036*
MinM (β_9)	0.459**
MaxM (β_{10})	-0.418*
SumM (β_{11})	-0.067*
VarM (β_{12})	0.683***
VarX \times VarY (β_{13})	0.014***
AvgR (β_{14})	-9.092***
SZ (β_{16})	-0.026
Adjusted R ²	0.921
MPE	-0.192%
MAPE	3.984%

*p<0.1; **p<0.05; ***p<0.01; ****p<0.001

Table 4.4: Regression results for the first group of 780 instances with outliers.

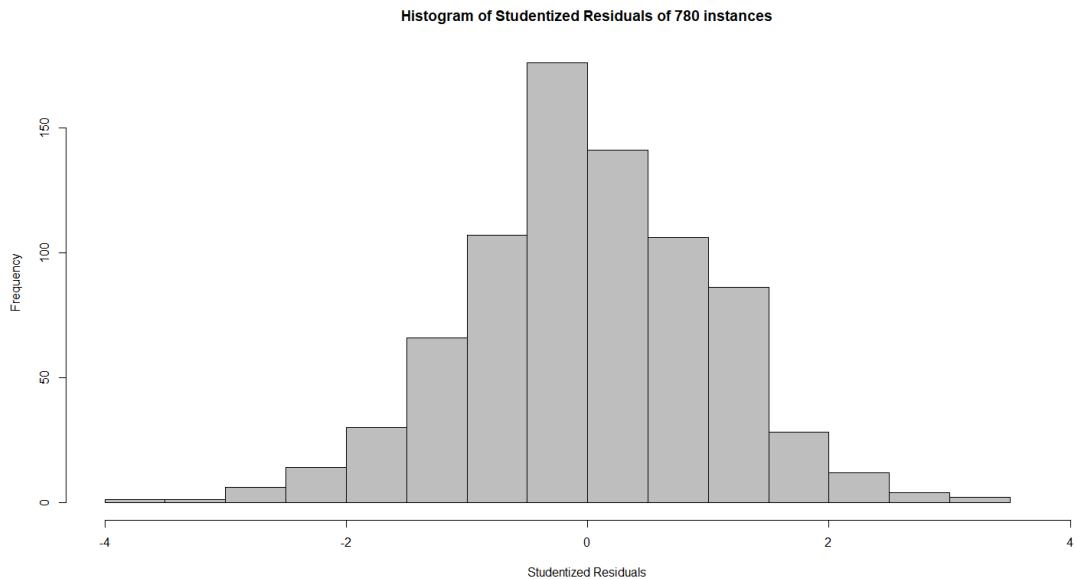


Figure 4.10: Histogram of Studentized residuals for the first group of 780 instances.

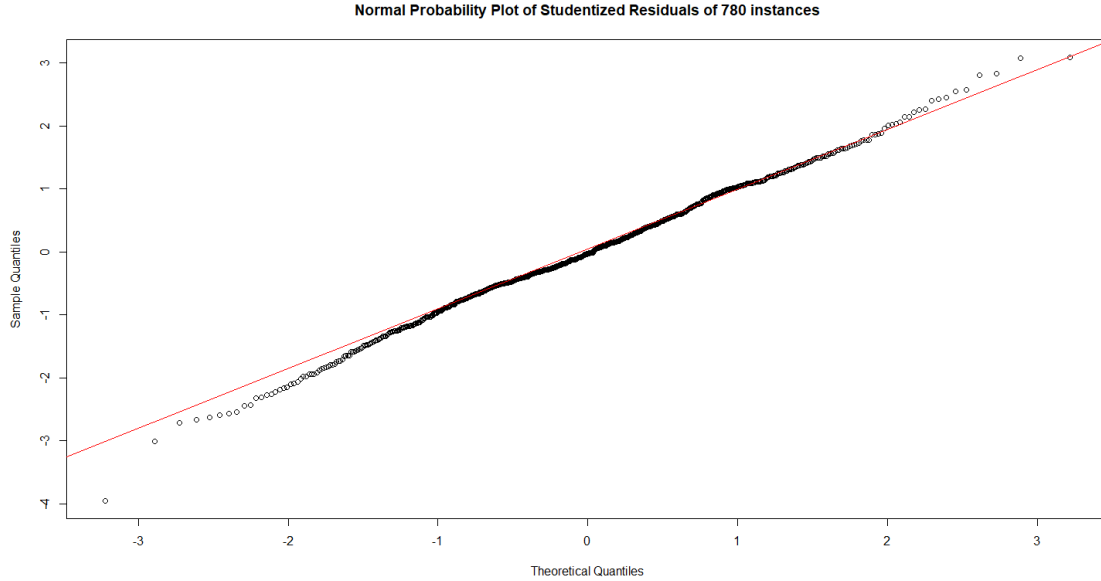


Figure 4.11: Normal probability plot of Studentized residuals for the first group of 780 instances.

4.4.3 Results on the First Group of 780 Instances

In Table 4.4, we present the regression results for the first group of 780 instances. The regression model has an adjusted R^2 value of 0.921 which indicates a very good model fit. VarR is not present in the model because the variance of the radii for the customer service regions is zero for all instances in this group. n and SZ are not significant at the 10% level. VarP is significant at the 10% level. MaxP, SumMaxP, MaxM, and SumM are significant at the 5% level. MinM is significant at the 1% level. All other variables are significant at the 0.1% level. In Figure 4.9, we give the Studentized residual plot for the first group of 780 instances. The lines show the Studentized residuals at values of 2 and -2 . The Studentized residual plot shows a horizontal band of points. There are 40 instances with Studentized resid-

ual values greater than 2 or less than -2 . The Studentized residuals have values from 3 to -4 . Even though there are outliers, the magnitude of the deviations of the outliers is not very high. In Figure 4.10, we give the histogram of Studentized residuals for the first group of 780 instances. The histogram shows that there is almost an equal distribution of instances with positive and negative residuals. This is also indicated by the MPE value of -0.192% which is close to zero. In Figure 4.11, we give the normal probability plot of Studentized residuals for the first group of 780 instances. Both the histogram and the normal probability plot show that the Studentized residuals follow a standard normal distribution. The normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which does not reject the null hypothesis at the 10% significance level. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 3.984% from the SZVNS route lengths. The adjusted R^2 value, MAPE, and the result of the Shapiro-Wilk hypothesis test indicate that the model performs well in predicting the SZVNS route lengths. Therefore, it would be useful to generalize the use of a linear regression model in predicting the SZVNS route lengths using the variables given in Table 4.1 for the CETSP. We need to validate the regression model on out-of-sample data for the generalization to work well on instances similar to this group.

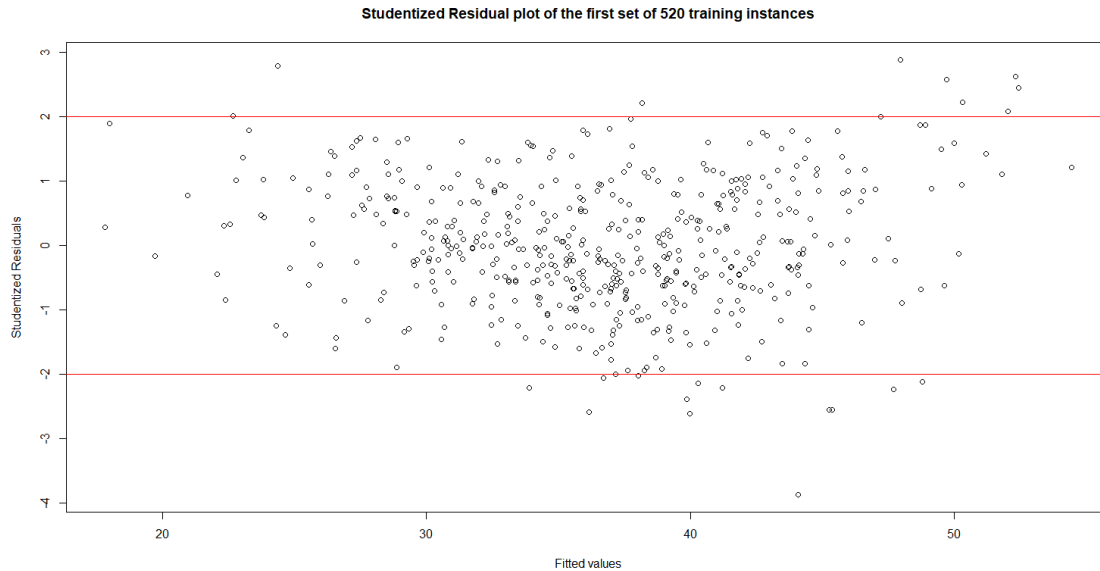


Figure 4.12: Studentized residual plot for the first set of 520 training instances. The lines indicate the Studentized residual values of 2 and -2 .

4.4.4 Cross-validation for the First Group of 780 Instances

We perform 3-fold cross-validation to test the performance of the linear regression model on out-of-sample data. We randomly partition the first group of 780 instances into three equal groups of size 260 each. First, we train the model on the second and the third groups and test the model on the first group. Second, we train the model on the first and the third groups and test the model on the second group. Third, we train the model on the first and the second groups and test the model on the third group. We look at the performance of the regression model in each of the three scenarios with the training data sets of size 520, and the testing data set of size 260.

In Table 4.5, we present the regression results for three training sets. The regression model on the first training set has an adjusted R^2 value of 0.919 which

Coefficient	Training set 1	Training set 2	Training set 3
Intercept (β_1)	16.436***	11.987***	17.691***
n (β_2)	0.009	0.628***	-0.041
A (β_3)	0.061***	0.035***	0.043***
MinP (β_4)	-0.635**	-0.608**	-0.972***
MaxP (β_5)	0.152	0.178	0.301*
VarP (β_6)	0.131*	0.198**	-0.039
SumMinP (β_7)	0.379***	0.400***	0.331***
SumMaxP (β_8)	0.061**	-0.008	0.050**
MinM (β_9)	0.543**	0.463*	0.410*
MaxM (β_{10})	-0.533*	-0.185	-0.505*
SumM (β_{11})	-0.094*	-0.035	-0.047
VarM (β_{12})	0.591***	0.820***	0.739***
VarX \times VarY (β_{13})	0.011***	0.016***	0.015***
AvgR (β_{14})	-9.145***	-9.101***	-9.014***
SZ (β_{16})	-0.025	-0.034	-0.025
Adjusted R ²	0.919	0.927	0.920
MPE (in-sample)	-0.193%	-0.179%	-0.190%
MAPE (in-sample)	4.075%	3.817%	3.948%
MPE (out-of-sample)	-0.712%	0.315%	-0.246%
MAPE (out-of-sample)	3.945%	4.388%	4.183%

·p<0.1; *p<0.05; **p<0.01; ***p<0.001

Table 4.5: Regression results on each training set of 520 instances from the first group of 780 instances.

indicates a very good model fit. VarR is not present in the model. n, MaxP, and SZ are not significant at the 10% level. VarP, MaxM, and SumM are significant at the 5% level. MinP, SumMaxP, and MinM are significant at the 1% level. All other variables are significant at the 0.1% level. In Figure 4.12, we give the Studentized residual plot for the first set of 520 training instances. The lines show the Studentized residuals at values of 2 and -2. The Studentized residual plot shows a horizontal band of points. There are 23 instances with Studentized residual values greater than 2 or less than -2. The Studentized residuals have values from 3 to -4. Even though there are outliers, the magnitude of the deviations of the outliers is not

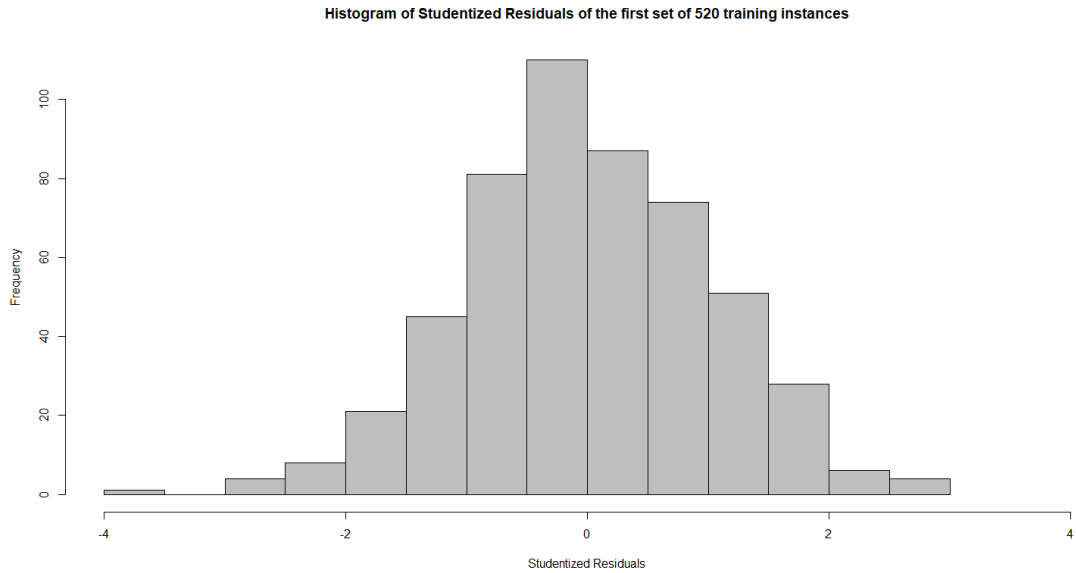


Figure 4.13: Histogram of Studentized residuals for the first set of 520 training instances.

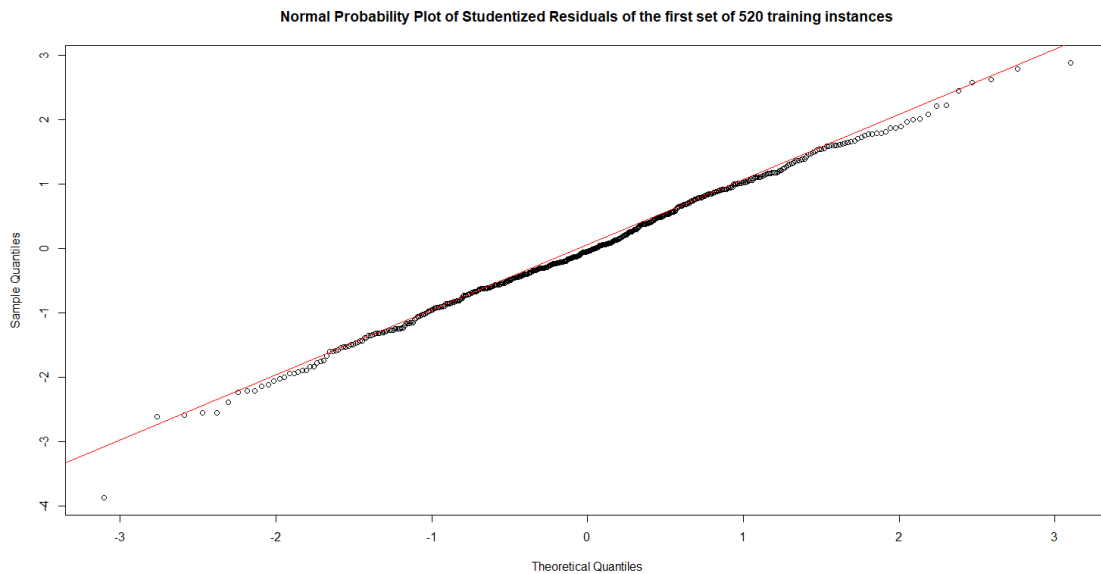


Figure 4.14: Normal probability plot of Studentized residuals for the first set of 520 training instances.

very high. In Figure 4.13, we give the histogram of Studentized residuals for the first set of 520 training instances. The histogram shows that there is almost an

equal distribution of instances with positive and negative residuals. This is also indicated by the MPE value of -0.193% which is close to zero. In Figure 4.14, we give the normal probability plot of Studentized residuals for the first set of 520 training instances. Both the histogram and the normal probability plot show that the Studentized residuals follow a standard normal distribution. The normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which does not reject the null hypothesis at the 10% significance level. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 4.075% from the SZVNS route lengths. The adjusted R^2 value, MAPE, and the result of the Shapiro-Wilk hypothesis test indicate that the model performs well in predicting the SZVNS route lengths. The out-of-sample MPE and MAPE values are -0.712% and 3.945% , respectively, which indicate that the regression model performs well on new instances from this group.

The regression model on the second training set has an adjusted R^2 value of 0.927 which indicates a very good model fit. VarR is not present in the model. MaxP, SumMaxP, MaxM, SumM, and SZ are not significant at the 10% level. MinM is significant at the 5% level. MinP and VarP are significant at the 1% level. All other variables are significant at the 0.1% level. In Figure 4.15, we give the Studentized residual plot for the second set of 520 training instances. The lines show the Studentized residuals at values of 2 and -2 . The Studentized residual plot shows a horizontal band of points. There are 24 instances with Studentized residual values greater than 2 or less than -2 . The Studentized residuals have values from 4 to -3 . Even though there are outliers, the magnitude of the deviations of the outliers

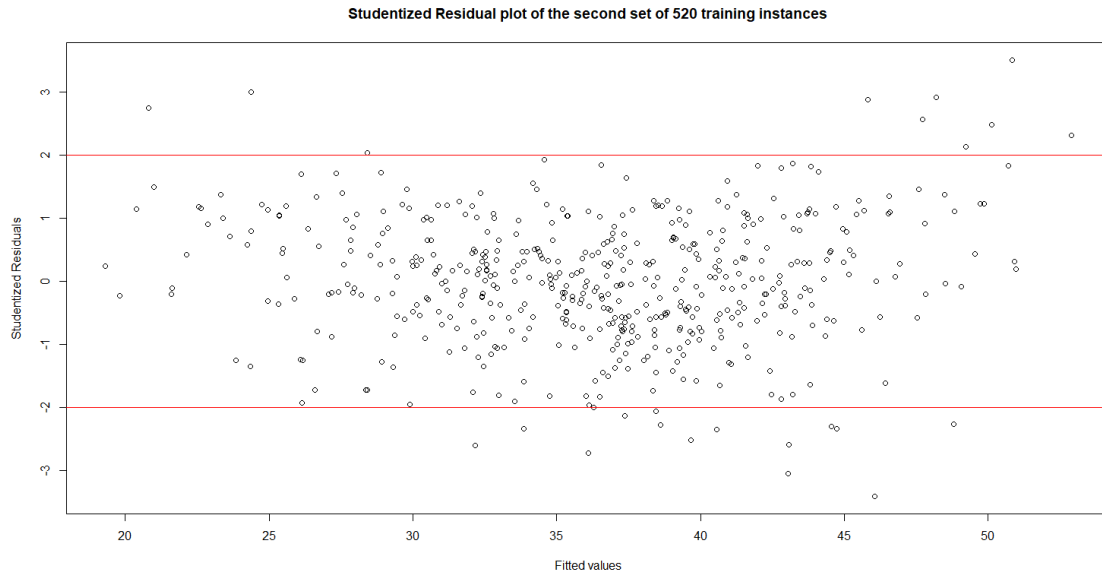


Figure 4.15: Studentized residual plot for the second set of 520 training instances. The lines indicate the Studentized residual values of 2 and -2 .

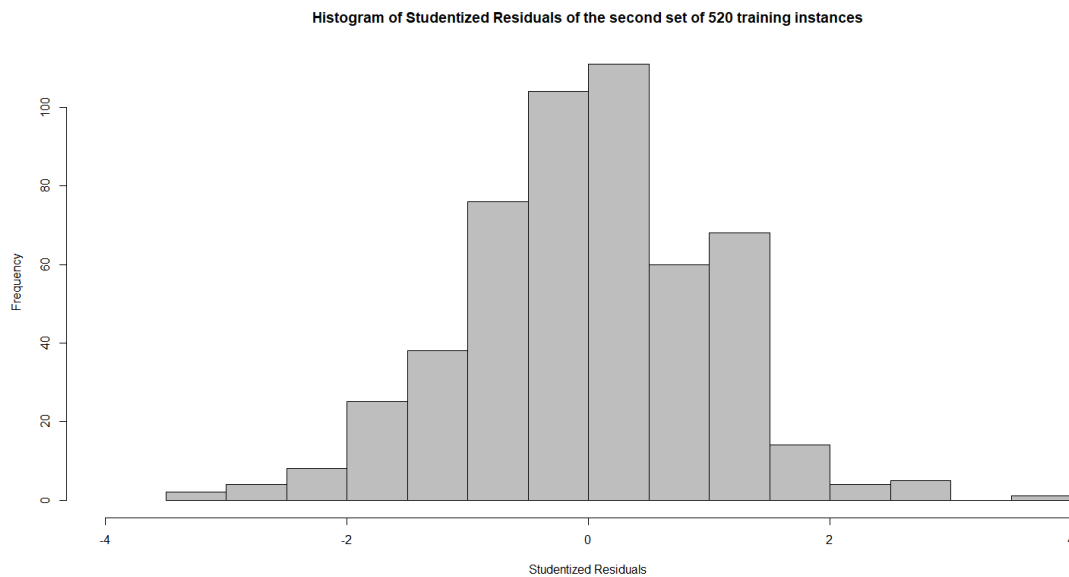


Figure 4.16: Histogram of Studentized residuals for the second set of 520 training instances.

is not very high. In Figure 4.16, we give the histogram of Studentized residuals for the second set of 520 training instances. The histogram shows that there is almost

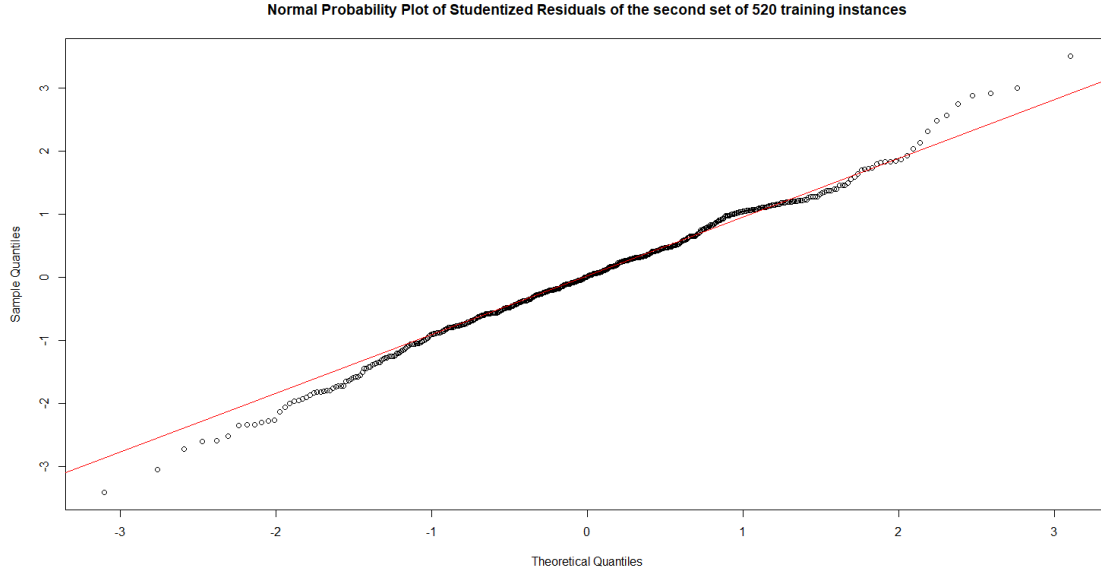


Figure 4.17: Normal probability plot of Studentized residuals for the second set of 520 training instances.

an equal distribution of instances with positive and negative residuals. This is also indicated by the MPE value of -0.179% which is close to zero. In Figure 4.17, we give the normal probability plot of Studentized residuals for the second set of 520 training instances. Both the histogram and the normal probability plot show that the Studentized residuals follow a standard normal distribution. The normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which does not reject the null hypothesis at the 1% significance level. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 3.817% from the SZVNS route lengths. The adjusted R^2 value, MAPE, and the result of the Shapiro-Wilk hypothesis test indicate that the model performs well in predicting the SZVNS route lengths. The out-of-sample MPE and MAPE values are 0.315% and 4.388%, respectively, which indicate that the regression model

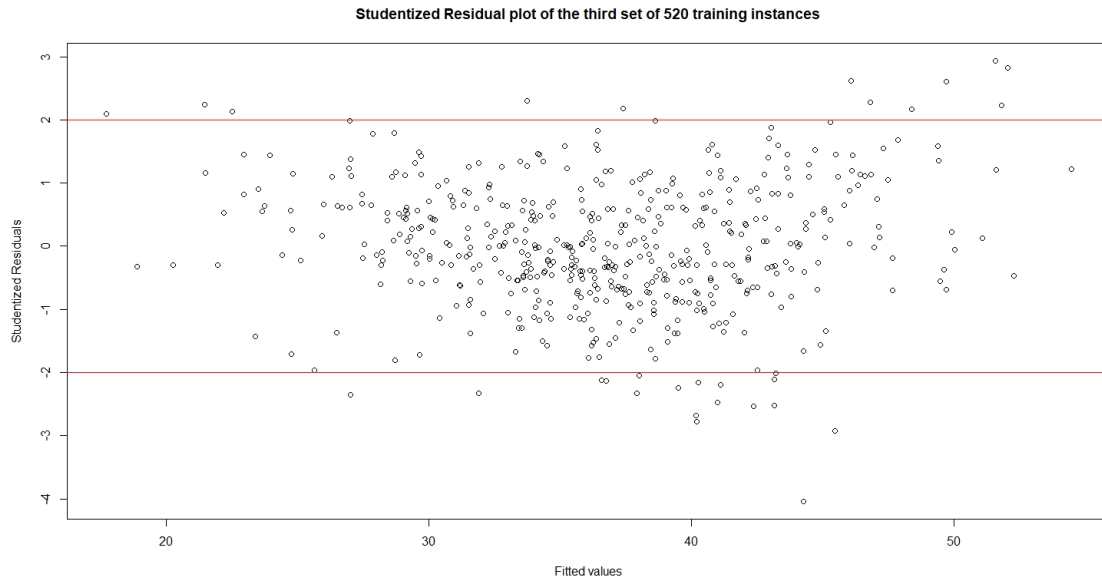


Figure 4.18: Studentized residual plot for the third set of 520 training instances. The lines indicate the Studentized residual values of 2 and -2 .

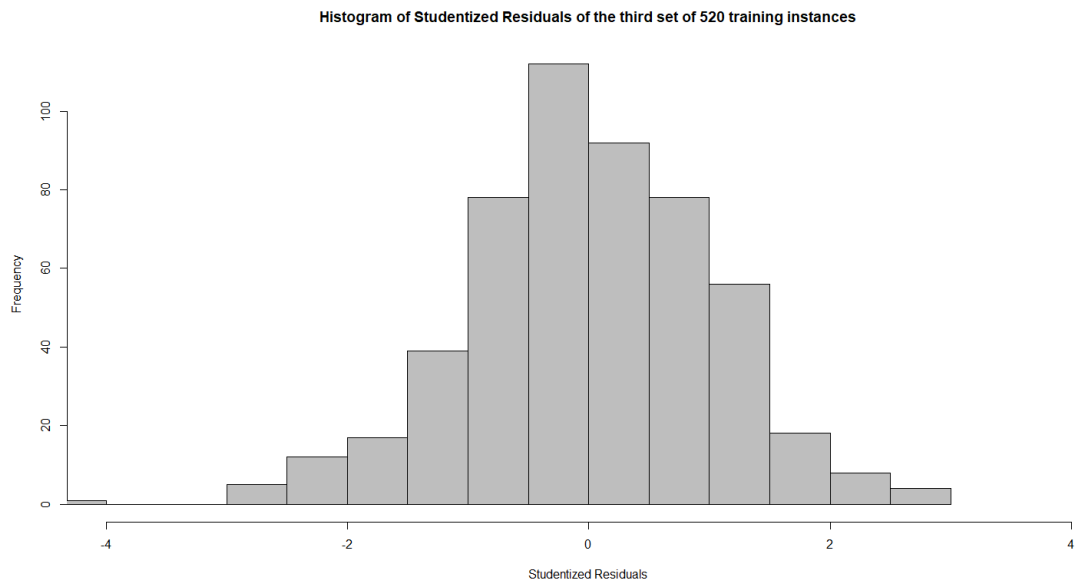


Figure 4.19: Histogram of Studentized residuals for the third set of 520 training instances.

performs well on new instances from this group.

The regression model on the third training set has an adjusted R^2 value of

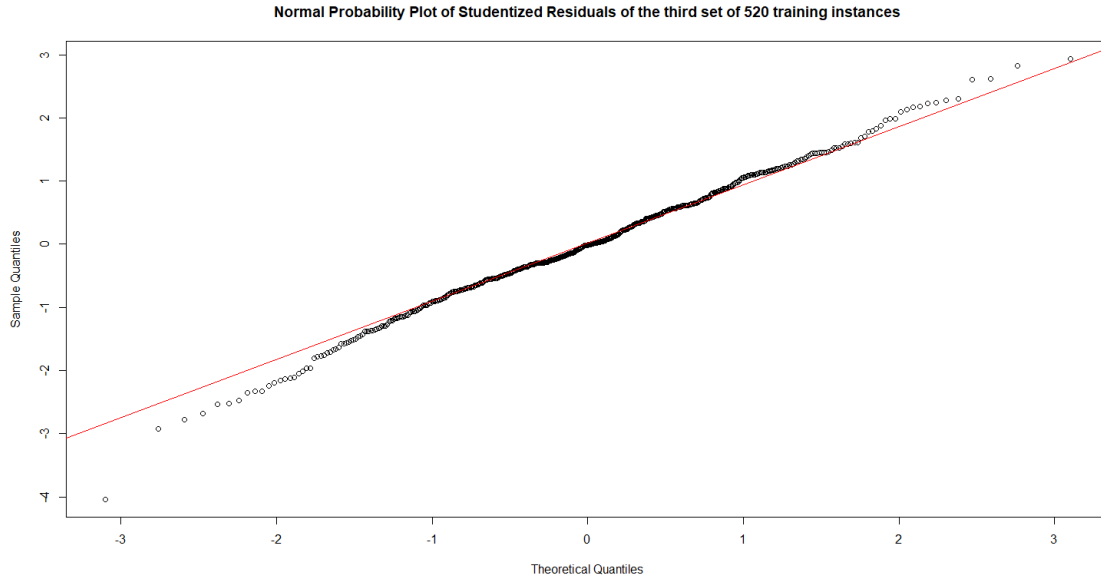


Figure 4.20: Normal probability plot of Studentized residuals for the third set of 520 training instances.

0.920 which indicates a very good model fit. VarR is not present in the model. n, VarP, SumM, and SZ are not significant at the 10% level. MaxP, MinM, and MaxM are significant at the 5% level. SumMaxP is significant at the 1% level. All other variables are significant at the 0.1% level. In Figure 4.18, we give the Studentized residual plot for the third set of 520 training instances. The lines show the Studentized residuals at values of 2 and -2 . The Studentized residual plot shows a horizontal band of points. There are 30 instances with Studentized residual values greater than 2 or less than -2 . The Studentized residuals have values from 3 to -4 . Even though there are outliers, the magnitude of the deviations of the outliers is not very high. In Figure 4.19, we give the histogram of Studentized residuals for the third set of 520 training instances. The histogram shows that there is almost an equal distribution of instances with positive and negative residuals. This is also

indicated by the MPE value of -0.190% which is close to zero. In Figure 4.20, we give the normal probability plot of Studentized residuals for the third set of 520 training instances. Both the histogram and the normal probability plot show that the Studentized residuals follow a standard normal distribution. The normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which does not reject the null hypothesis at the 10% significance level. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 3.948% from the SZVNS route lengths. The adjusted R^2 value, MAPE, and the result of the Shapiro-Wilk hypothesis test indicate that the model performs well in predicting the SZVNS route lengths. The out-of-sample MPE and MAPE values are -0.246% and 4.183%, respectively, which indicate that the regression model performs well on new instances from this group.

The average MPE and MAPE values on the three training data sets (in-sample) of size 520 are -0.188% and 3.947%, respectively. The average MPE and MAPE values on the three testing data sets (out-of-sample) of size 260 are -0.196% and 4.172%, respectively. The average MAPE value is very close for in-sample data and out-of-sample data. The performance of the regression model on out-of-sample data indicates that the linear model can be applied to predict SZVNS route lengths on CETSP instances with properties similar to the first group of 780 instances. Given the robustness of the linear regression model, it would be useful to find the best subset model without compromising the performance.

Variable	Number of variables													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
n									*				*	*
A			*	*	*	*	*	*	*	*	*	*	*	*
MinP								*	*	*	*	*	*	*
MaxP									*	*	*	*	*	*
VarP												*	*	*
SumMinP			*	*	*	*	*	*	*	*	*	*	*	*
SumMaxP	*	*		*		*	*	*		*	*	*	*	*
MinM							*	*	*	*	*	*	*	*
MaxM										*	*	*	*	*
SumM					*						*	*	*	*
VarM					*	*	*	*	*	*	*	*	*	*
VarX×VarY						*	*	*	*	*	*	*	*	*
AvgR		*	*	*	*	*	*	*	*	*	*	*	*	*
SZ														*

Table 4.6: Best subset models based on R^2 for the first group of 780 instances.

4.4.5 Model Selection for the First Group of 780 Instances

In Table 4.6, we present the best subset models based on R^2 for the first group of 780 instances. VarR is not present in any of the models. We select the best 1-variable model, best 2-variable model, and so on based on the highest R^2 values. As we increase the number of variables from k to $k + 1$, the best $(k + 1)$ -variable model might drop some variables that were in the best k -variable model. Out of the 14 best subset models shown, we select three models based on adjusted R^2 , Mallows's C_p , and BIC. In Figures 4.21, 4.22, and 4.23, we show the 14 best subset models for the first group of 780 instances according to adjusted R^2 , Mallows's C_p , and BIC, respectively. The 14 models are arranged in decreasing order of performance according to the respective criterion with the first row indicating the best model and the last row indicating the worst model. We select the best models (first row)

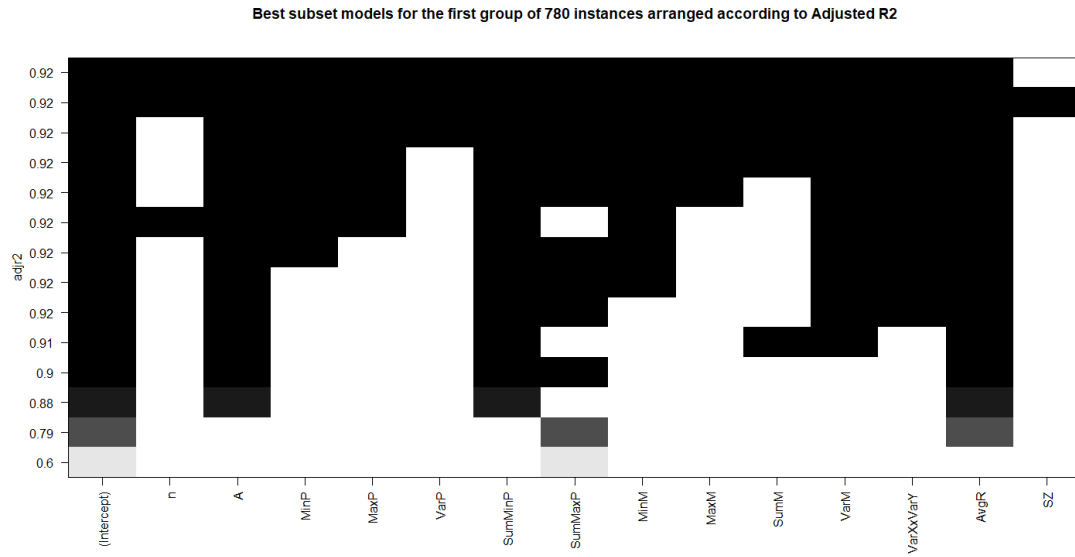


Figure 4.21: Plot showing the best subset models for the first group of 780 instances arranged according to adjusted R^2 .

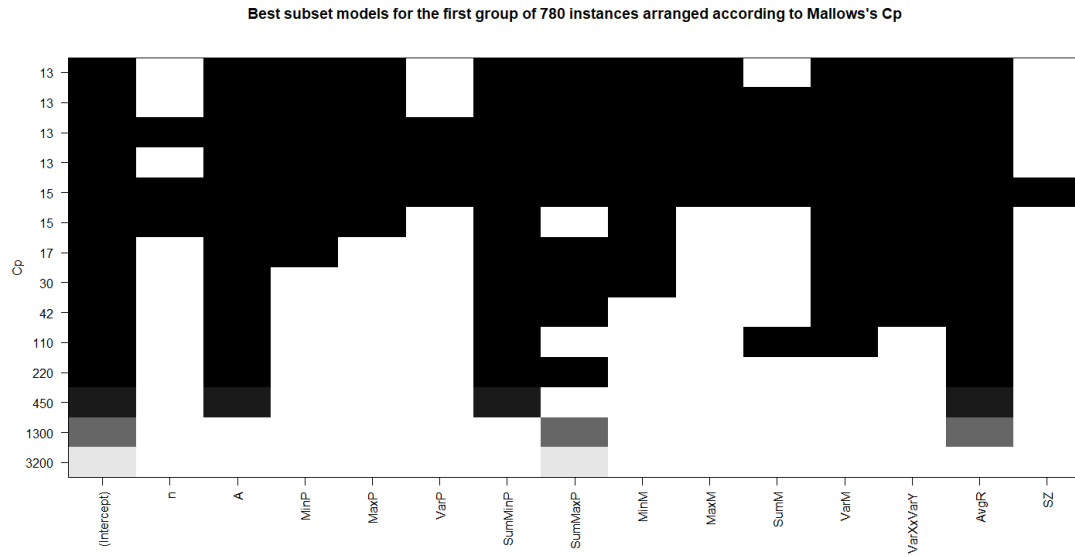


Figure 4.22: Plot showing the best subset models for the first group of 780 instances arranged according to Mallows's C_p .

according to each of the three criteria for further analysis of model performance.

In Table 4.7, we present the regression results for three best subset models

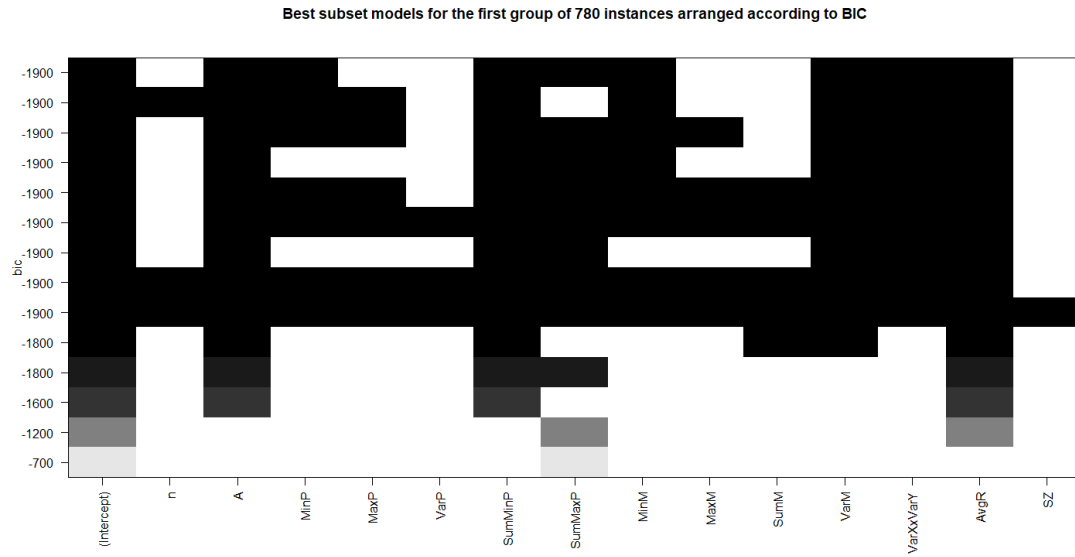


Figure 4.23: Plot showing the best subset models for the first group of 780 instances arranged according to BIC.

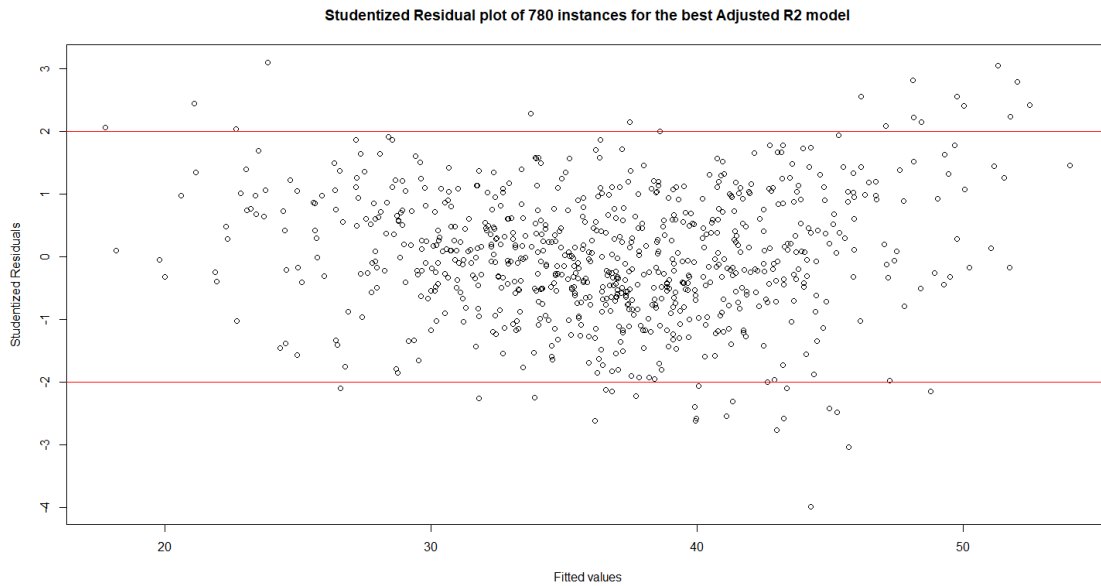


Figure 4.24: Studentized residual plot of 780 instances for the best adjusted R^2 model. The lines indicate the Studentized residual values of 2 and -2 .

based on adjusted R^2 , Mallows's C_p , and BIC. The regression model for the best subset based on adjusted R^2 has 13 variables and an adjusted R^2 value of 0.921

Coefficient	Best adjusted R ²	Best Mallows's C _p	Best BIC
Intercept (β_1)	15.320***	15.485***	16.334***
n (β_2)	0.188		
A (β_3)	0.048***	0.046***	0.044***
MinP (β_4)	-0.668***	-0.734***	-0.674***
MaxP (β_5)	0.224*	0.230*	
VarP (β_6)	0.101·		
SumMinP (β_7)	0.359***	0.362***	0.362***
SumMaxP (β_8)	0.036*	0.025***	0.027***
MinM (β_9)	0.455**	0.508***	0.498***
MaxM (β_{10})	-0.410*	-0.273*	
SumM (β_{11})	-0.064*		
VarM (β_{12})	0.685***	0.805***	0.797***
VarX×VarY (β_{13})	0.014***	0.011***	0.012***
AvgR (β_{14})	-9.059***	-9.059***	-9.059***
SZ (β_{16})			
Number of variables	13	10	8
Adjusted R ²	0.921	0.921	0.920
MPE	-0.192%	-0.194%	-0.202%
MAPE	3.983%	3.995%	4.008%

·p<0.1; *p<0.05; **p<0.01; ***p<0.001

Table 4.7: Regression results on the best subset models based on R² for the first group of 780 instances.

which indicates a very good model fit. SZ is not in the model. n is not significant at the 10% level. VarP is significant at the 10% level. MaxP, SumMaxP, MaxM, and SumM are significant at the 5% level. MinM is significant at the 1% level. All other variables are significant at the 0.1% level. In Figure 4.24, we give the Studentized residual plot of 780 instances for the best adjusted R² model. The lines show the Studentized residuals at values of 2 and -2. The Studentized residual plot shows a horizontal band of points. There are 38 instances with Studentized residual values greater than 2 or less than -2. The Studentized residuals have values from 3 to -4. Even though there are outliers, the magnitude of the deviations of the outliers

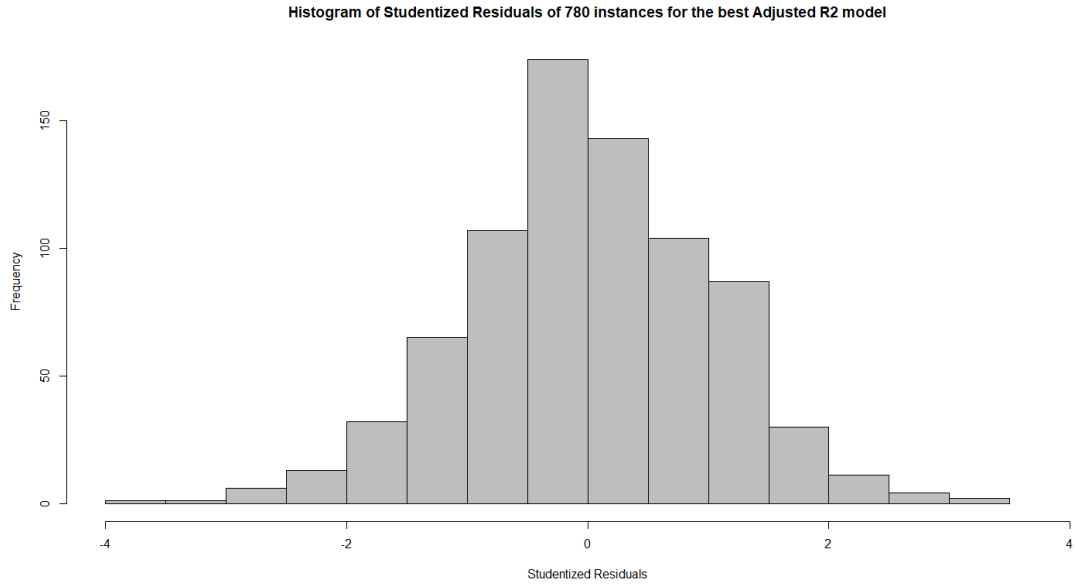


Figure 4.25: Histogram of Studentized residuals of 780 instances for the best adjusted R^2 model.

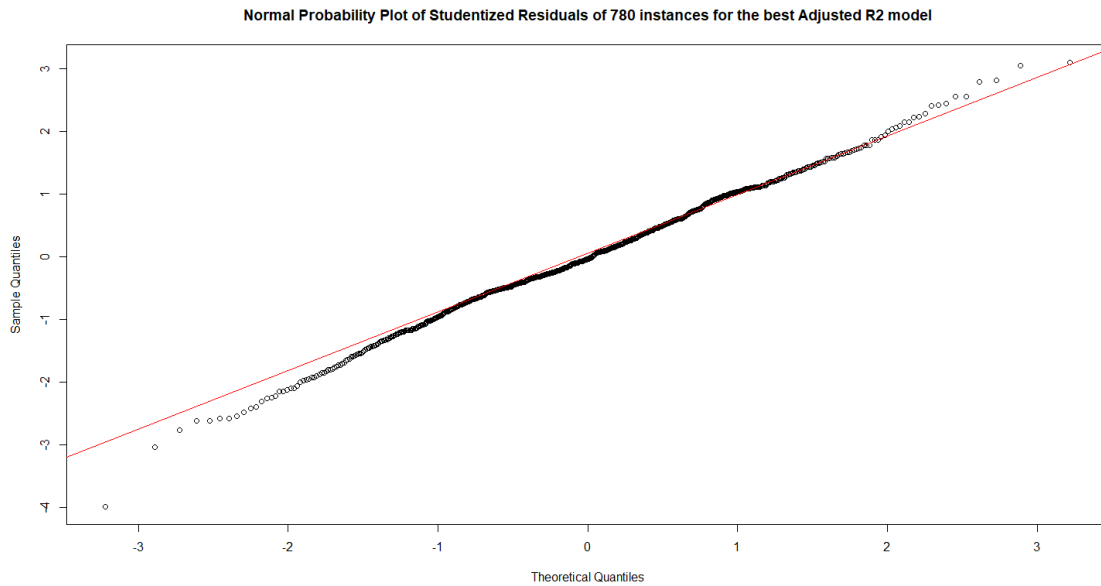


Figure 4.26: Normal probability plot of Studentized residuals of 780 instances for the best adjusted R^2 model.

is not very high. In Figure 4.25, we give the histogram of Studentized residuals of 780 instances for the best adjusted R^2 model. The histogram shows that there is

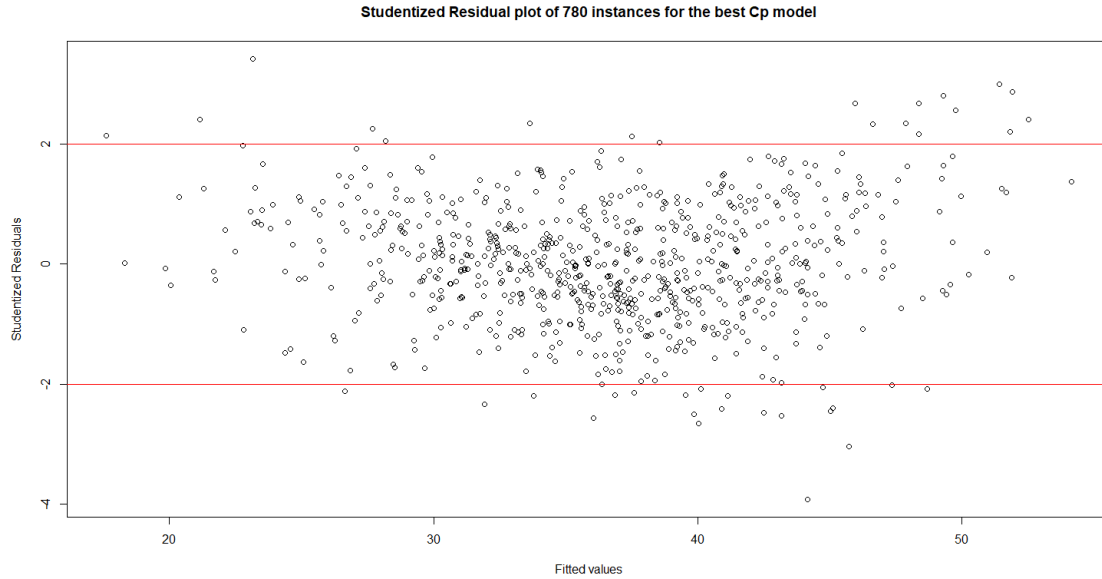


Figure 4.27: Studentized residual plot of 780 instances for the best Mallows's C_p model. The lines indicate the Studentized residual values of 2 and -2 .

almost an equal distribution of instances with positive and negative residuals. This is also indicated by the MPE value of -0.192% which is close to zero. In Figure 4.26, we give the normal probability plot of Studentized residuals of 780 instances for the best adjusted R^2 model. Both the histogram and the normal probability plot show that the Studentized residuals follow a standard normal distribution. The normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which does not reject the null hypothesis at the 10% significance level. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 3.983% from the SZVNS route lengths. The adjusted R^2 value, MAPE, and the result of the Shapiro-Wilk hypothesis test indicate that the model performs well in predicting the SZVNS route lengths.

The regression model for the best subset based on Mallows's C_p has 10 variables

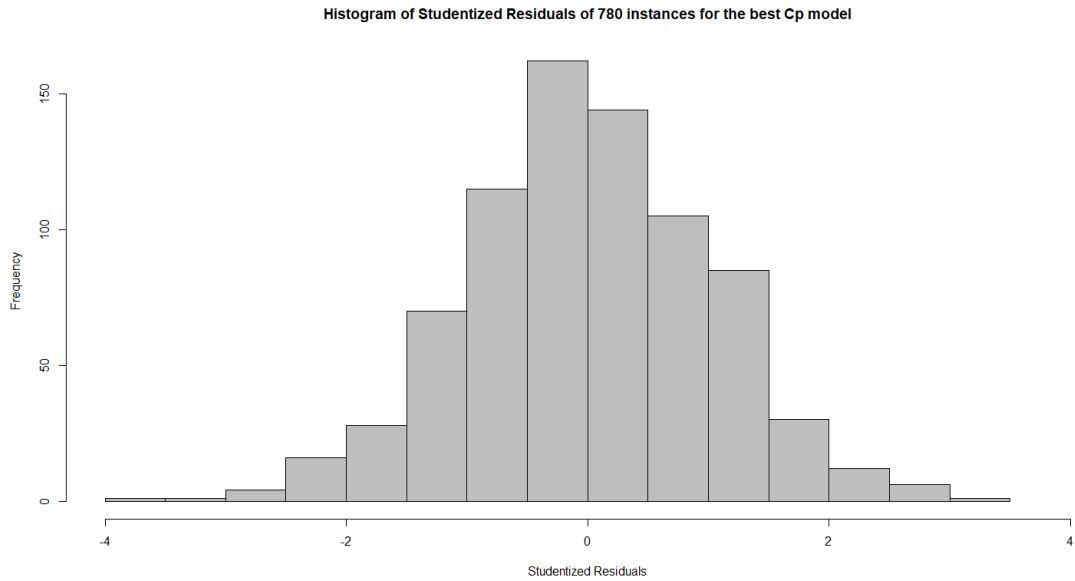


Figure 4.28: Histogram of Studentized residuals of 780 instances for the best Mallows's C_p model.

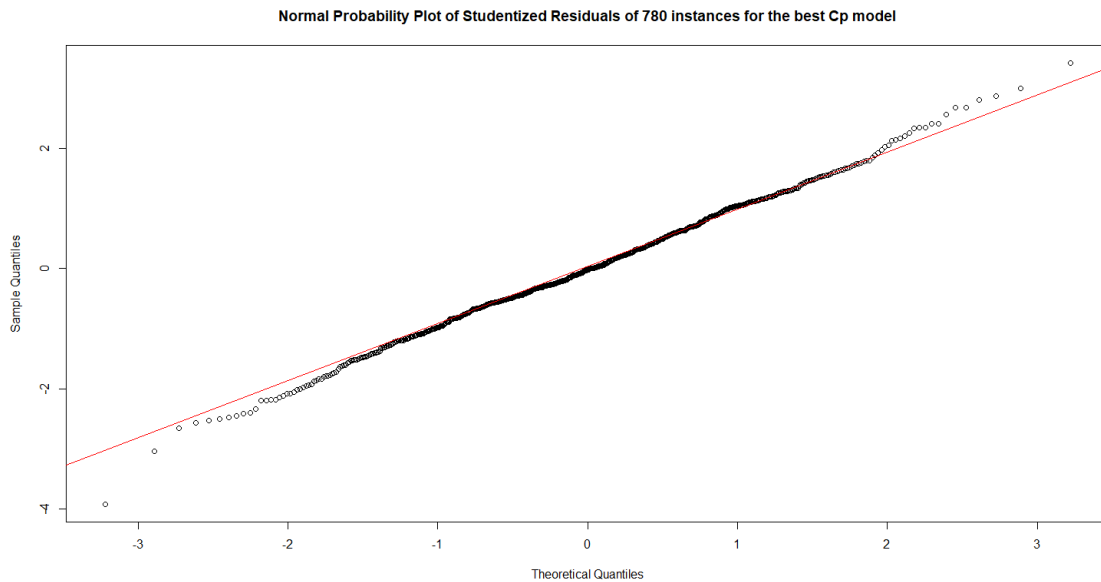


Figure 4.29: Normal probability plot of Studentized residuals of 780 instances for the best Mallows's C_p model.

and an adjusted R^2 value of 0.921 which indicates a very good model fit. n , VarP , SumM , and SZ are not in the model. MaxP and MaxM are significant at the 5%

level. All other variables are significant at the 0.1% level. In Figure 4.27, we give the Studentized residual plot of 780 instances for the best Mallows's C_p model. The lines show the Studentized residuals at values of 2 and -2 . The Studentized residual plot shows a horizontal band of points. There are 41 instances with Studentized residual values greater than 2 or less than -2 . The Studentized residuals have values from 3 to -4 . Even though there are outliers, the magnitude of the deviations of the outliers is not very high. In Figure 4.28, we give the histogram of Studentized residuals of 780 instances for the best Mallows's C_p model. The histogram shows that there is almost an equal distribution of instances with positive and negative residuals. This is also indicated by the MPE value of -0.194% which is close to zero. In Figure 4.29, we give the normal probability plot of Studentized residuals of 780 instances for the best Mallows's C_p model. Both the histogram and the normal probability plot show that the Studentized residuals follow a standard normal distribution. The normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which does not reject the null hypothesis at the 10% significance level. The MAPE value indicates that the route-length estimates from the regression model differ by an average of 3.995% from the SZVNS route lengths. The adjusted R^2 value, MAPE, and the result of the Shapiro-Wilk hypothesis test indicate that the model performs well in predicting the SZVNS route lengths.

The regression model for the best subset based on BIC has eight variables and an adjusted R^2 value of 0.920 which indicates a very good model fit. n , MaxP, VarP, MaxM, SumM, and SZ are not in the model. All other variables are significant at the 0.1% level. In Figure 4.30, we give the Studentized residual plot of 780 instances

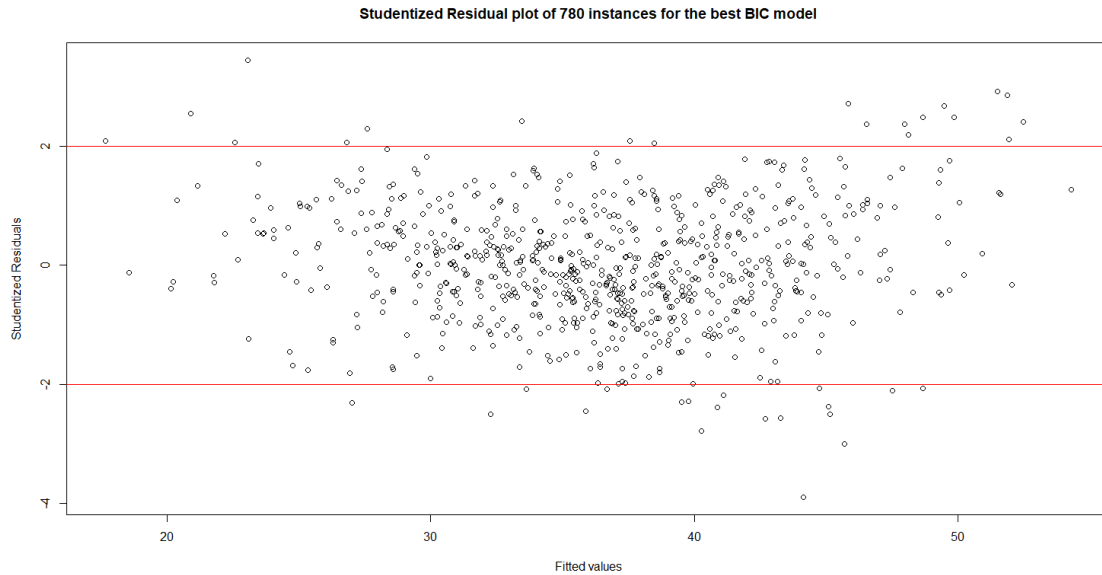


Figure 4.30: Studentized residual plot of 780 instances for the best BIC model. The lines indicate the Studentized residual values of 2 and -2 .

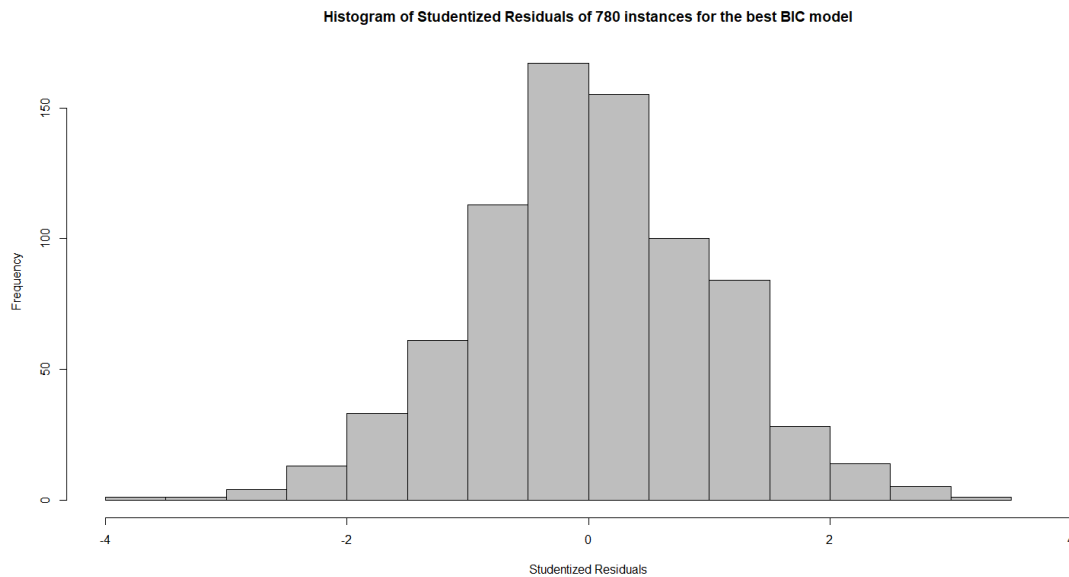


Figure 4.31: Histogram of Studentized residuals of 780 instances for the best BIC model.

for the best BIC model. The lines show the Studentized residuals at values of 2 and -2 . The Studentized residual plot shows a horizontal band of points. There are

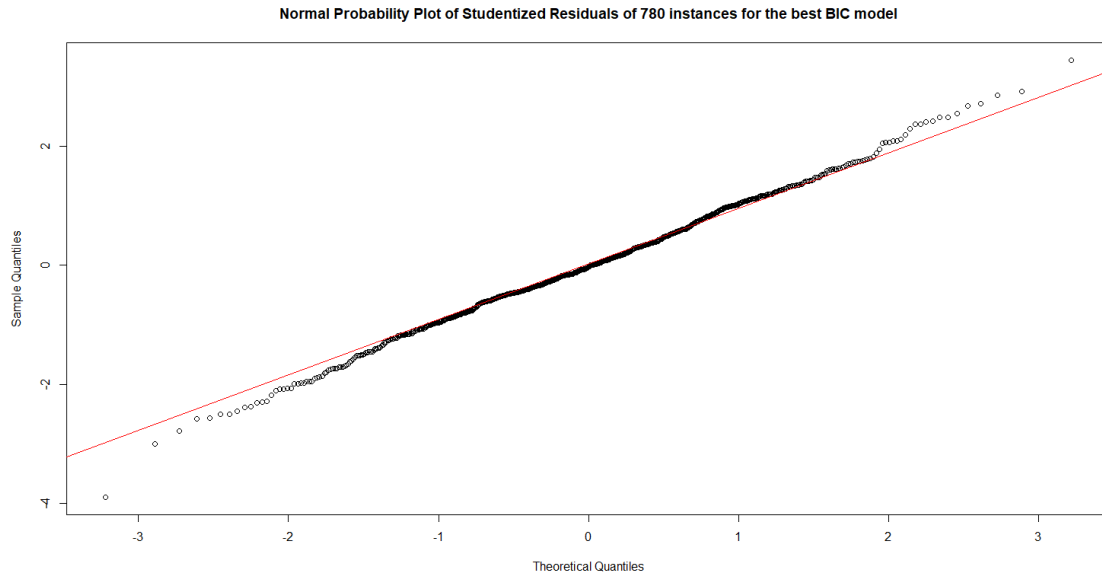


Figure 4.32: Normal probability plot of Studentized residuals of 780 instances for the best BIC model.

39 instances with Studentized residual values greater than 2 or less than -2 . The Studentized residuals have values from 3 to -4 . Even though there are outliers, the magnitude of the deviations of the outliers is not very high. In Figure 4.31, we give the histogram of Studentized residuals of 780 instances for the best BIC model. The histogram shows that there is almost an equal distribution of instances with positive and negative residuals. This is also indicated by the MPE value of -0.202% which is close to zero. In Figure 4.32, we give the normal probability plot of Studentized residuals of 780 instances for the best BIC model. Both the histogram and the normal probability plot show that the Studentized residuals follow a standard normal distribution. The normality of the Studentized residuals is indicated by the Shapiro-Wilk hypothesis test which does not reject the null hypothesis at the 10% significance level. The MAPE value indicates that the route-length estimates from the regression

model differ by an average of 4.008% from the SZVNS route lengths. The adjusted R^2 value, MAPE, and the result of the Shapiro-Wilk hypothesis test indicate that the model performs well in predicting the SZVNS route lengths.

All three best subset models based on adjusted R^2 , Mallows's C_p , and BIC performed similarly as indicated by their adjusted R^2 value, MAPE, and the result of the Shapiro-Wilk hypothesis test. The MAPE value was 3.983% for the best subset model on 13 variables according to adjusted R^2 , 3.995% for the best subset model on 10 variables according to Mallows's C_p , and 4.008% for the best subset model on eight variables according to BIC. The linear model with eight variables predicts the SZVNS route lengths for CETSP instances with properties similar to the first group of 780 instances almost as well as any other linear model with more variables. All eight variables in the model are highly significant at the 0.1% level which indicates that these variables are essential in predicting the SZVNS route lengths.

4.5 Conclusions and Future Directions

We showed that it is possible to have a fast and accurate method of predicting CETSP route lengths using a linear regression model without generating the actual routes. The exact model would be able to predict the route lengths generated by using a specific routing algorithm. However, the overall regression framework can be adapted to any algorithm or heuristic. We showed that the SZVNS route lengths for CETSP could be predicted with an average error of about 4% using

a linear regression model. Therefore, we recommend to use the linear regression model with A, MinP, SumMinP, SumMaxP, MinM, VarM, VarX×VarY, and AvgR as the independent variables for predicting the SZVNS routes lengths on CETSP instances with random node locations, and all customers having the same radius for the service regions. However, the linear regression model did not perform well for instances with node locations generated in different structured ways. There could be improvements in predicting the route lengths for CETSP instances with node locations not generated randomly that are also fast to compute. We need to represent these instances in the regression models using more appropriate variables.

Chapter 5: Intersection Inspection Rural Postman Problem on a Mixed Graph

5.1 Introduction

The Rural Postman Problem (RPP) is an important arc routing problem. In an RPP, we need to find the shortest way of connecting a given set of required street segments to form a full route. It has many real-world applications including street sweeping, meter reading, postal delivery, and snow plowing. [Frederickson et al. \(1978\)](#) showed that the RPP on a mixed graph is NP-complete. [Corberán et al. \(2014\)](#) described problem formulations for directed, mixed, and windy graphs. There has also been interest in the RPP with turn penalties which was introduced by [Benavent and Soler \(1999\)](#). The idea is that the quality of a tour is determined not only by the length of the tour but also by the types of turns that are made at street intersections. For example, most truck drivers prefer to travel straight ahead for as long as possible. Turning left or even turning right can be dangerous and time consuming. U-turns are impossible for long trucks. In snow plowing in the U.S., left turns and U-turns are often discouraged because they take more time and push snow into an intersection. So, along with solving an RPP, there is a cost associated

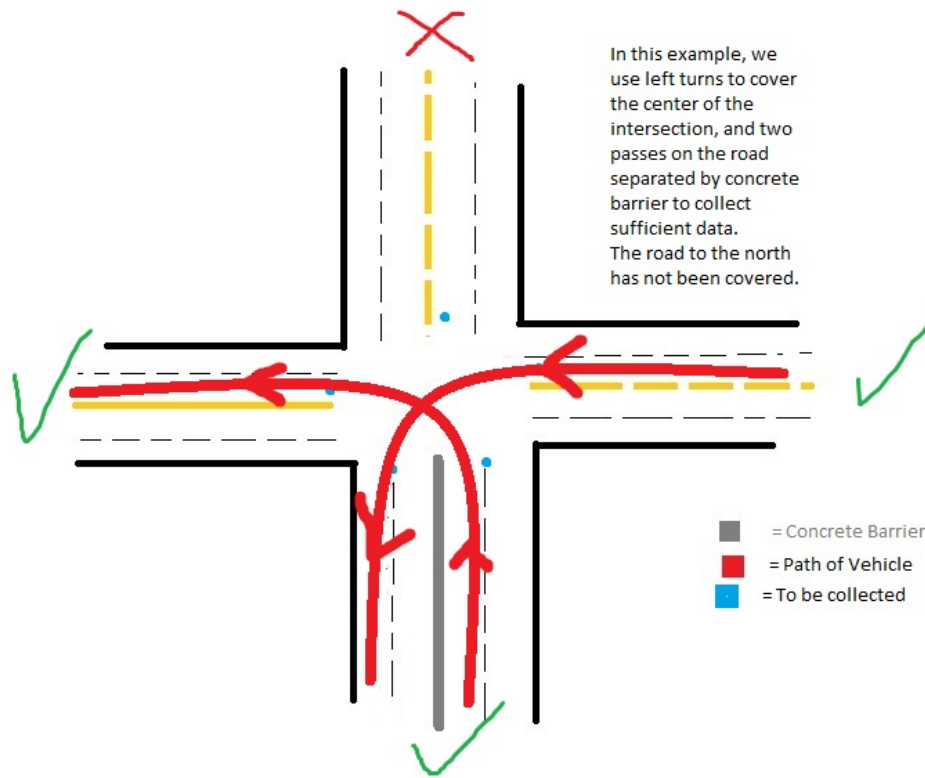


Figure 5.1: (Color online) An intersection with two left turns.

with each turn that needs to be taken into account. [Clossey et al. \(2001\)](#) used a two-stage approach to deal with turn penalties. In the first stage, the problem was solved as an RPP to obtain a Eulerian graph. In the second stage, an end-pairing algorithm was used to generate a Eulerian tour taking into consideration the turn penalties. [Cerrone et al. \(2019\)](#) further improved this two-stage approach using heuristics.

We introduce another important variant of the RPP involving turns. City governments and highway authorities carry out road inspections to decide which street segments to repair by taking videos using a camera mounted on a vehicle. This process is similar to Google generating street view images. The vehicle taking

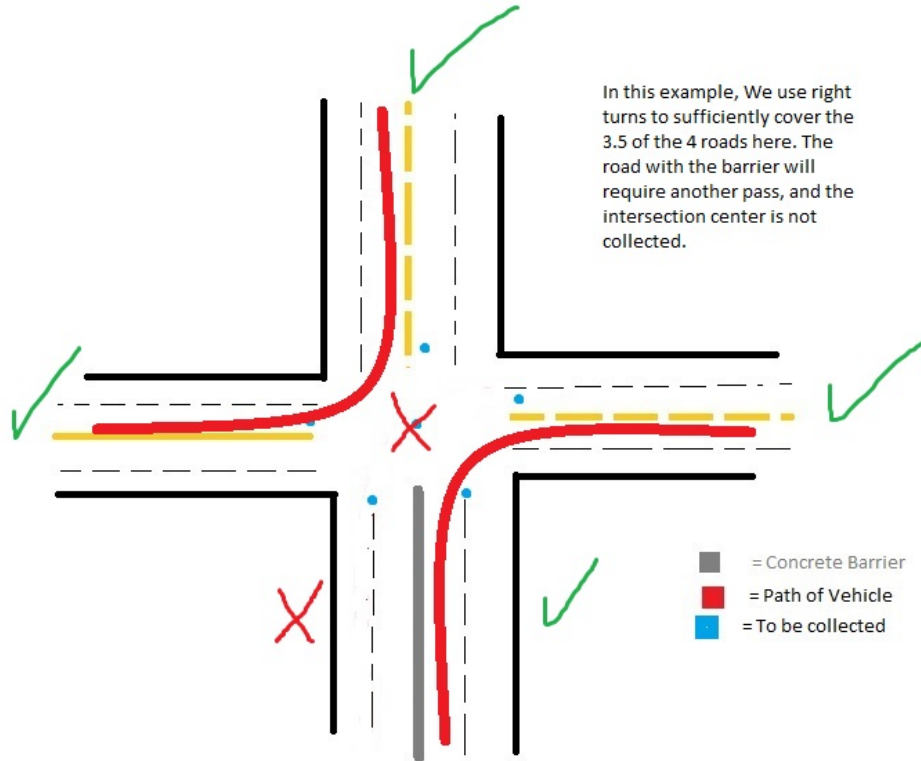


Figure 5.2: (Color online) An intersection with two right turns.

the videos needs to proceed straight or take a left turn to cover an intersection fully. A right turn does not always capture an intersection fully and a U-turn does not cross an intersection. In Figure 5.1, we show an intersection with two left turns. A blue dot on a street segment or an intersection represents that a proper pass by the vehicle is required through that region to cover them. One left turn is going from east to south, covering the street segment to the east and the left side of the street segment to the south. The other left turn is going from south to west, covering the street segment to the west and the right side of the street segment to the south. Only one pass in either direction is required to cover a street segment unless there is a concrete barrier through the middle of the street segment that blocks the view

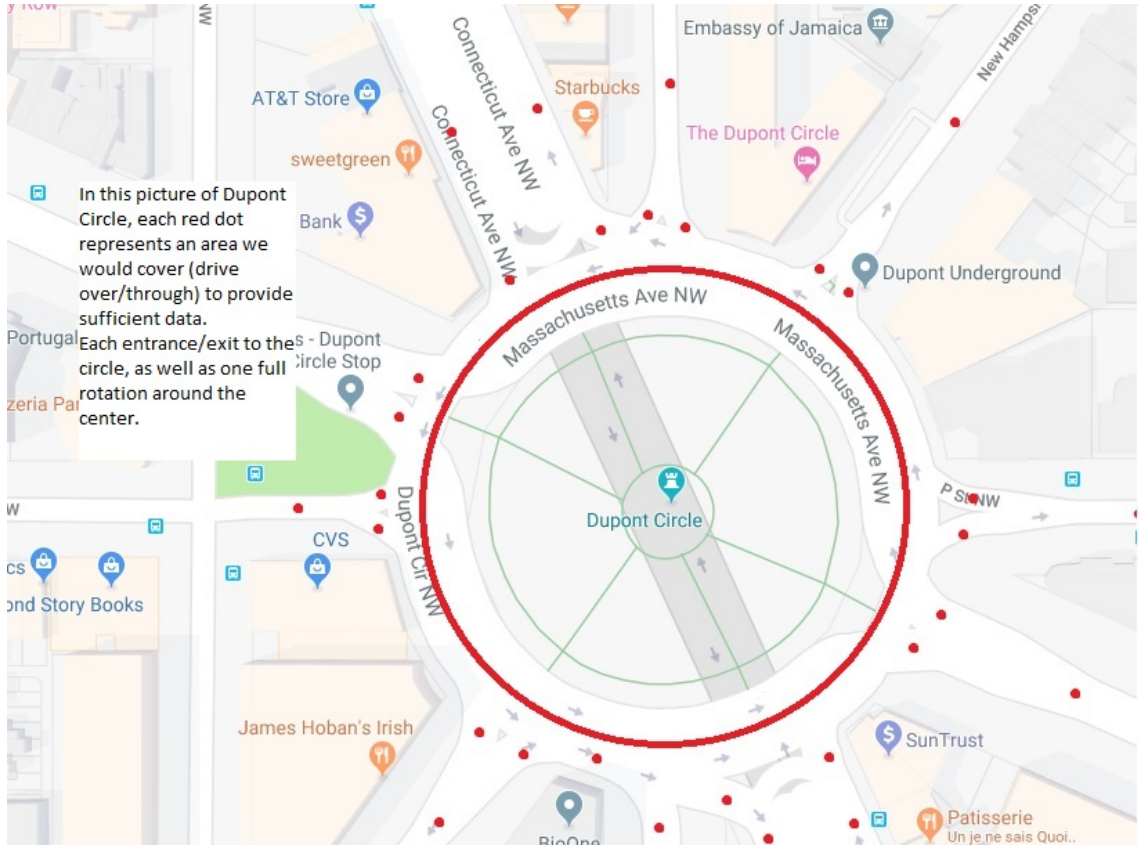
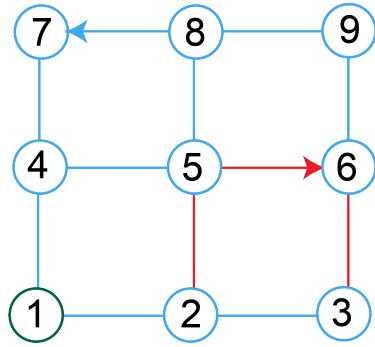
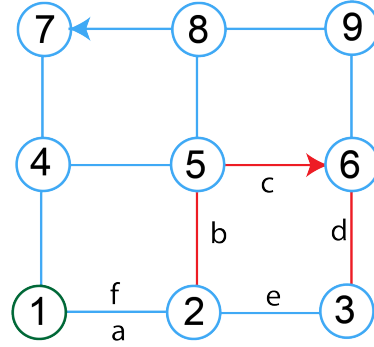


Figure 5.3: (Color online) Map of Dupont Circle in Washington, DC.

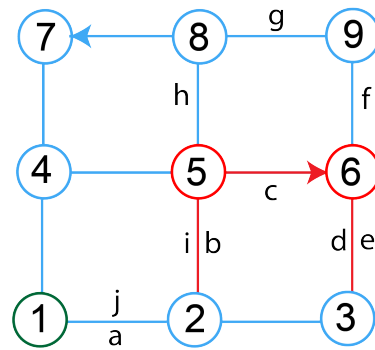
of the camera mounted on the vehicle. Each of these two left turns covers the intersection. However, the street segment to the north is not covered. In Figure 5.2, we show an intersection with two right turns. One right turn is going from south to east, covering the street segment to the east and the right side of the street segment to the south. The other right turn is going from north to west, covering the street segment to the north and the street segment to the west. These two right turns do not cover the intersection and the left side of the street segment to the south. In Figure 5.3, we show the map of Dupont Circle in Washington, DC. Even though there are multiple lanes in the circle, just one pass is required through any lane to cover the circle as denoted by the red line. The red dots denote the various street



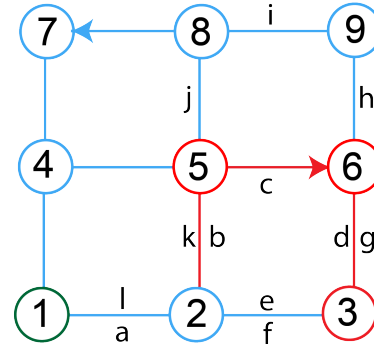
(a) Street network with required edges and arcs



(b) RPP solution of length 6



(c) IIRPP solution of length 10 with 5 and 6 as required vertices



(d) IIRPP solution of length 12 with 3, 5, and 6 as required vertices

Figure 5.4: (Color online) The RPP and the IIRPP solutions on a small grid-like street network.

segments connected to the circle that need to be traversed since they are not covered during the pass through the circle. It is not required to cover the circle in one pass, rather segments of the circle can be covered by different sections of the route.

The Intersection Inspection Rural Postman Problem (IIRPP) is a hybrid of arc routing and node routing problems. We address the problem on a mixed graph, the most general case. In addition to solving an RPP, we have to make sure there is at least one left turn or straight turn at an intersection that is required to be

inspected. We consider the RPP as a special case of the IIRPP when there are no intersections to be inspected. Therefore, the IIRPP is at least as difficult to solve as the RPP. Unlike the RPP with turn penalties, there is no route cost associated with the turns in the IIRPP. In Figure 5.4, we show the RPP and the IIRPP solutions on a small grid-like street network. Figure 5.4a shows the street network. The blue lines are the edges (two-way street segments), and the blue arrows are the arcs (one-way street segments). The red lines are the required edges, and the red arrows are the required arcs. The green node is the starting and ending point of the route. The red nodes are the intersections to be inspected. The length of each edge or arc is 1 unit. Figure 5.4b shows the optimal RPP solution ($a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f$) with a route length of 6 units. Figure 5.4c shows the optimal IIRPP solution ($a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow h \rightarrow i \rightarrow j$) with a route length of 10 units. Figure 5.4d shows the optimal IIRPP solution ($a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow h \rightarrow i \rightarrow j \rightarrow k \rightarrow l$) with a route length of 12 units. A route that is feasible for the RPP might not be feasible for the IIRPP. A feasible IIRPP route is always feasible for the RPP. Therefore, the optimal IIRPP objective value cannot be less than the optimal RPP objective value. In Figure 5.4, we demonstrate that the optimal IIRPP objective value, for a street network with a given set of required edges, required arcs, and intersections to be inspected, is a non-decreasing function as we add more intersections for inspection.

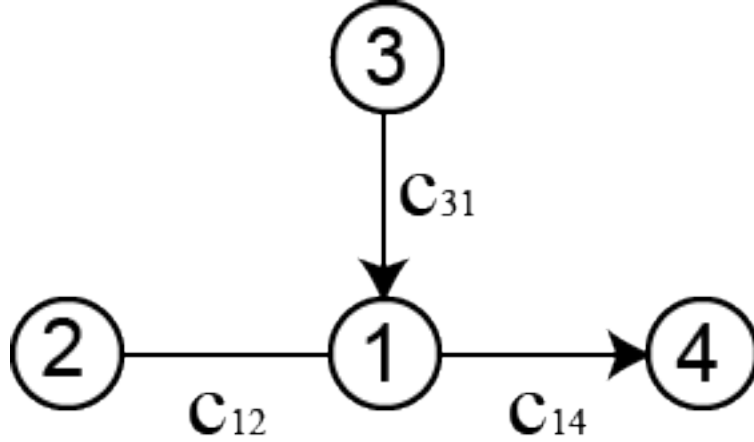


Figure 5.5: A mixed graph G (street network) with four nodes. Costs are shown adjacent to the two arcs and one edge.

5.2 Problem Formulations on a Mixed Graph

Consider a street network as a mixed graph denoted by $G = (V, E \cup A)$, where E denotes the set of the edges (two-way street segments), A denotes the set of the arcs (one-way street segments), and V denotes the set of nodes (intersections). Let $c_{ij} \geq 0$ be the traversal cost (length) of street segment (i, j) , where $i \in V$ and $j \in V$. It is possible to travel from i to j and also from j to i on an edge $(i, j) \in E$. However, it is only possible to travel from i to j and not from j to i on an arc $(i, j) \in A$. In Figure 5.5, we show a mixed graph G (street network) with four nodes, two arcs, one edge, and traversal costs.

Let $E_R \subseteq E$ and $A_R \subseteq A$ be the set of required edges and arcs, respectively, that need to be video recorded. $V_R = \{i \in V \mid \exists j \in V, (i, j) \in A_R \text{ or } (j, i) \in A_R \text{ or } (i, j) \in E_R \text{ or } (j, i) \in E_R\} \subseteq V$ is the set of nodes defining the required edges and arcs. $G_R = (V_R, E_R \cup A_R)$ is the graph induced by the required edges and arcs. The vertex sets of the connected components of G_R are denoted by V_1, \dots, V_p .

Let $\delta^+(i) = \{j \in V | (i, j) \in A\}$ be the set of nodes connected to node i by an outgoing arc from i . Let $\delta^-(i) = \{j \in V | (j, i) \in A\}$ be the set of nodes connected to node i by an incoming arc to i . Let $\delta(i) = \{j \in V | (i, j) \text{ or } (j, i) \in E\}$ be the set of nodes connected to node i by an edge. $V_I \subseteq V$ is the set of nodes that need to be recorded. For practicality, we assume that $V_I \subset V_R$. Let $o \in V_R \setminus V_I$ denote the starting and ending node of the route. At node i , $RT(i) = \{(k, i, j) | k \in \delta^-(i) \cup \delta(i), j \in \delta^+(i) \cup \delta(i), (k, i)(i, j) \text{ is a right turn}\}$, $LT(i) = \{(k, i, j) | k \in \delta^-(i) \cup \delta(i), j \in \delta^+(i) \cup \delta(i), (k, i)(i, j) \text{ is a left turn}\}$, and $ST(i) = \{(k, i, j) | k \in \delta^-(i) \cup \delta(i), j \in \delta^+(i) \cup \delta(i), (k, i)(i, j) \text{ is a straight turn}\}$ denotes the right turns, left turns, and straight turns, respectively, using 3-tuples. We define the non-negative integer decision variable x_{ij} to be the number of times street segment (i, j) is traversed in the route. The RPP formulation on the graph G is given on the next page.

The objective function (5.1) minimizes the total cost (length) of the route. Constraint (5.2) ensures that the depot is a part of the route. Constraints (5.3) ensure that the required edges are a part of the route. Constraints (5.4) are the flow conservation constraints. Constraints (5.5) are the disjoint subtour elimination constraints. Constraints (5.6) define the decision variables for the required arcs. Constraints (5.7) define the decision variables for the non-required arcs. Constraints (5.8) define the decision variables for the edges.

$$(RPP) \quad \min \quad \sum_{(i,j) \in E \cup A} c_{ij} x_{ij} + \sum_{(i,j) \in E} c_{ij} x_{ji} \quad (5.1)$$

$$\text{s.t.} \quad \sum_{j \in \delta^+(o) \cup \delta(o)} x_{oj} \geq 1 \quad (5.2)$$

$$x_{ij} + x_{ji} \geq 1 \quad \forall (i, j) \in E_R \quad (5.3)$$

$$\sum_{j \in \delta^+(i) \cup \delta(i)} x_{ij} - \sum_{j \in \delta^-(i) \cup \delta(i)} x_{ji} = 0 \quad \forall i \in V \quad (5.4)$$

$$\sum_{i \in S} \left(\sum_{j \in (\delta^+(i) \cup \delta(i)) \setminus S} x_{ij} \right) \geq 1 \quad \forall S = \cup_{l \in Q} V_l, \quad Q \subset \{1, \dots, p\} \quad (5.5)$$

$$x_{ij} \geq 1 \text{ and integer} \quad \forall (i, j) \in A_R \quad (5.6)$$

$$x_{ij} \geq 0 \text{ and integer} \quad \forall (i, j) \in A \setminus A_R \quad (5.7)$$

$$x_{ij}, x_{ji} \geq 0 \text{ and integer} \quad \forall (i, j) \in E \quad (5.8)$$

5.2.1 IIRPP Formulation using Node Transformations

Let $G^1 = (V^1, A^1)$ denote the transformed graph, where $V^1 = V_E^1 \cup V_A^1$ is the set of nodes and $A^1 = A_E^1 \cup A_A^1 \cup A_T^1$ is the set of arcs. $V_A^1 = \{n_{i,j,i}, n_{i,j,j} | (i, j) \in A\}$ and $V_E^1 = \{n_{i,j,i}, n_{j,i,i}, n_{i,j,j}, n_{j,i,j} | (i, j) \in E\}$ are the set of nodes in G^1 corresponding to each arc $(i, j) \in A$ and each edge $(i, j) \in E$, respectively, in the original graph G . $A_E^1 = \{(n_{i,j,i}, n_{i,j,j}), (n_{j,i,j}, n_{j,i,i}), (n_{j,i,i}, n_{i,j,i}), (n_{i,j,j}, n_{j,i,j}) | (i, j) \in E\}$ and $A_A^1 = \{(n_{i,j,i}, n_{i,j,j}) | (i, j) \in A\}$ are the set of arcs in G^1 corresponding to each edge $(i, j) \in E$ and each arc $(i, j) \in A$, respectively, in the original graph G . The cost of the arc $(n_{i,j,i}, n_{i,j,j}) \in A_A^1$ is equal to the cost of the arc $(i, j) \in A$. The cost of the arcs

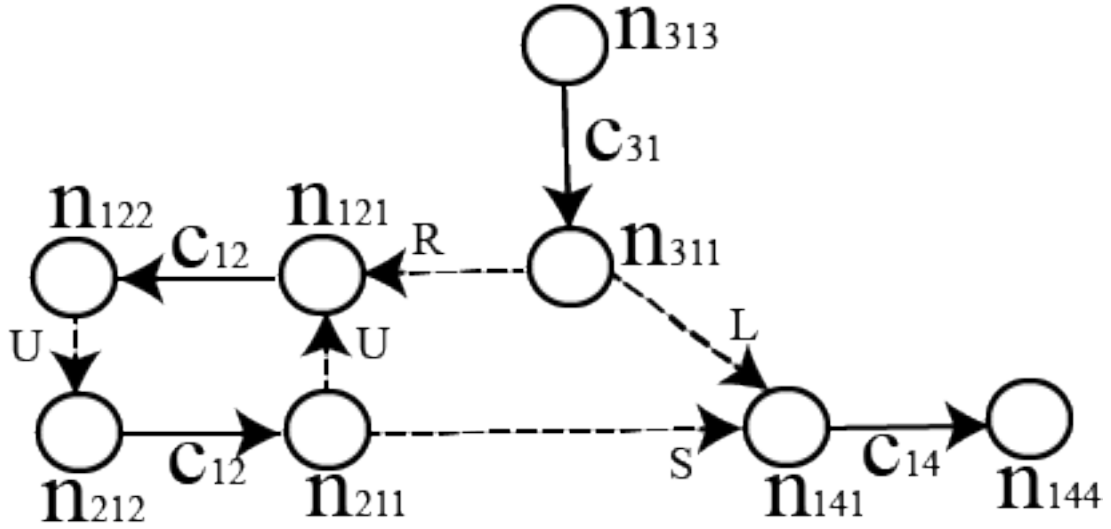


Figure 5.6: Transformed graph G^1 from the original graph G shown in Figure 5.5.

$(n_{i,j,i}, n_{i,j,j}) \in A_E^1$ and $(n_{j,i,j}, n_{j,i,i}) \in A_E^1$ are equal to the cost of the edge $(i, j) \in E$. Arcs $(n_{j,i,i}, n_{i,j,i}) \in A_E^1$ and $(n_{i,j,j}, n_{j,i,j}) \in A_E^1$ represent the U-turns in G^1 and have a cost of zero. $A_T^1 = \{(n_{k,i,i}, n_{i,j,i}) | i \in V, (k, i, j) \in RT(i) \cup LT(i) \cup ST(i)\}$ is the set of arcs in G^1 representing the right turns, left turns, and straight turns in the original graph G , each with a cost of zero.

In Figure 5.6, we show the transformed graph G^1 from the original graph G shown in Figure 5.5. Node 1 in G is split into nodes n_{121} , n_{211} , n_{141} , and n_{311} in G^1 . Node 2 in G is split into nodes n_{122} and n_{212} in G^1 . Nodes 3 and 4 in G are represented as nodes n_{313} and n_{144} , respectively, in G^1 . Arcs (1,4) and (3,1) in G are represented as arcs (n_{141}, n_{144}) with a cost of c_{14} and (n_{313}, n_{311}) with a cost of c_{31} , respectively, in G^1 . Edge (1,2) in G is represented as arcs (n_{121}, n_{122}) and (n_{212}, n_{211}) each with a cost of c_{12} and U-turn arcs (n_{211}, n_{121}) and (n_{122}, n_{212}) each with a cost of zero in G^1 . The right turn (3,1)(1,2), left turn (3,1)(1,4), and straight

turn (2,1)(1,4) in G are represented as arcs (n_{311}, n_{121}) , (n_{311}, n_{141}) , and (n_{211}, n_{141}) , respectively, in G^1 , each with a cost of zero.

$$(IIRPP-F1) \quad \min \quad \sum_{(i,j) \in E \cup A} c_{ij} x_{n_{i,j}, i n_{i,j,j}} + \sum_{(i,j) \in E} c_{ij} x_{n_{j,i,j} n_{j,i,i}} \quad (5.9)$$

$$\text{s.t.} \quad \sum_{j \in \delta^+(o) \cup \delta(o)} x_{n_{o,j}, o n_{o,j,j}} \geq 1 \quad (5.10)$$

$$x_{n_{i,j}, i n_{i,j,j}} + x_{n_{j,i,j} n_{j,i,i}} \geq 1 \quad \forall (i,j) \in E_R \quad (5.11)$$

$$x_{n_{i,j}, i n_{i,j,j}} - \sum_{k \in \delta^-(i) \cup \delta(i)} x_{n_{k,i,i} n_{i,j,i}} = 0 \quad \forall n_{i,j,i} \in V_A^1 \quad (5.12)$$

$$\sum_{k \in \delta^+(j) \cup \delta(j)} x_{n_{i,j,j} n_{j,k,j}} - x_{n_{i,j}, i n_{i,j,j}} = 0 \quad \forall n_{i,j,j} \in V_A^1 \quad (5.13)$$

$$x_{n_{i,j}, i n_{i,j,j}} - x_{n_{j,i,i} n_{i,j,i}} - \sum_{k \in \delta^-(i) \cup \delta(i) \setminus \{j\}} x_{n_{k,i,i} n_{i,j,i}} = 0 \quad \forall n_{i,j,i} \in V_E^1 \quad (5.14)$$

$$\sum_{k \in \delta^+(j) \cup \delta(j) \setminus \{i\}} x_{n_{i,j,j} n_{j,k,j}} + x_{n_{i,j,j} n_{j,i,j}} - x_{n_{i,j}, i n_{i,j,j}} = 0 \quad \forall n_{i,j,j} \in V_E^1 \quad (5.15)$$

$$\sum_{(k,i,j) \in LT(i) \cup ST(i)} x_{n_{k,i,i} n_{i,j,i}} \geq 1 \quad \forall i \in V_I \quad (5.16)$$

$$\sum_{n_{k,i,i} \in S^1} \left(\sum_{(n_{k,i,i}, n_{i,j,i}) \in A_T^1 : n_{i,j,i} \notin S^1} x_{n_{k,i,i} n_{i,j,i}} \right) \geq 1 \quad \forall S^1 = \cup_{l \in Q} V_l^1, \quad Q \subset \{1, \dots, p\} \quad (5.17)$$

$$x_{n_{i,j}, i n_{i,j,j}} \geq 1 \text{ and integer} \quad \forall (i,j) \in A_R \quad (5.18)$$

$$x_{n_{i,j}, i n_{i,j,j}} \geq 0 \text{ and integer} \quad \forall (i,j) \in A \setminus A_R \quad (5.19)$$

$$x_{n_{i,j}, i n_{i,j,j}}, x_{n_{j,i,j} n_{j,i,i}}, x_{n_{j,i,i} n_{i,j,i}}, x_{n_{i,j,j} n_{j,i,i}} \geq 0 \text{ and integer} \quad \forall (i,j) \in E \quad (5.20)$$

$$x_{n_{k,i,i} n_{i,j,i}} \geq 0 \text{ and integer} \quad \forall (n_{k,i,i}, n_{i,j,i}) \in A_T^1 \quad (5.21)$$

Let $V_R^1 = V_{E_R}^1 \cup V_{A_R}^1 \subseteq V^1$. $V_{E_R}^1 = \{n_{i,j,i}, n_{j,i,i}, n_{i,j,j}, n_{j,i,j} | (i, j) \in E_R\} \subseteq V_E^1$ is the set of nodes in G^1 corresponding to the required edges in G . $V_{A_R}^1 = \{n_{i,j,i}, n_{i,j,j} | (i, j) \in A_R\} \subseteq V_A^1$ is the set of nodes in G^1 corresponding to the required arcs in G . Let $A_R^1 = A_{E_R}^1 \cup A_{A_R}^1 \cup A_{T_R}^1 \subseteq A^1$. $A_{A_R}^1 = \{(n_{i,j,i}, n_{i,j,j}) | (i, j) \in A_R\} \subseteq A_A^1$ is the set of arcs in G^1 corresponding to the required arcs in G . $A_{E_R}^1 = \{(n_{i,j,i}, n_{i,j,j}), (n_{j,i,j}, n_{j,i,i}), (n_{j,i,i}, n_{i,j,i}), (n_{i,j,j}, n_{j,i,j}) | (i, j) \in E_R\} \subseteq A_E^1$ is the set of arcs in G^1 corresponding to the required edges in G . $A_{T_R}^1 = \{(n_{k,i,i}, n_{i,j,i}) | i \in V, (k, i, j) \in RT(i) \cup LT(i) \cup ST(i), n_{k,i,i} \in V_R^1, n_{i,j,i} \in V_R^1\} \subseteq A_T^1$ is the set of arcs in G^1 corresponding to the turns between required arcs or edges in G . Let $G_R^1 = (V_R^1, A_R^1)$ be the graph induced from G^1 . The vertex sets of the connected components of G_R^1 are denoted by V_1^1, \dots, V_p^1 . We define the non-negative integer decision variable $x_{n_{a,b,c}, n_{d,e,f}}$ to be the number of times arc $(n_{a,b,c}, n_{d,e,f})$ is traversed in the route on G^1 . The IIRPP formulation on the transformed graph G^1 (IIRPP-F1) based on the original graph G is given on the previous page.

The objective function (5.9) minimizes the total cost (length) of the route. Constraint (5.10) ensures that the depot in G is a part of the route. Constraints (5.11) ensure that the required edges in G are a part of the route. Constraints (5.12), (5.13), (5.14), and (5.15) are the flow conservation constraints for the nodes in G^1 . Constraints (5.16) ensure that the nodes in G that need to be video recorded are covered by at least one left turn or straight turn. Constraints (5.17) are the disjoint subtour elimination constraints. Constraints (5.18) define the decision variables corresponding to the required arcs in G . Constraints (5.19) define the decision variables corresponding to the non-required arcs in G . Constraints (5.20) define the decision

variables corresponding to the edges in G . Constraints (5.21) define the decision variables for the arcs in G^1 representing the right turns, left turns, and straight turns in G . This formulation does not limit an intersection being covered with only left turns and straight turns. If the field of view of the camera improves in the future enabling two right turns to cover an intersection (the two right turns as shown in Figure 5.2 might be able to cover the intersection with an improved camera), we would modify constraints (5.16) as $\sum_{(k,i,j) \in LT(i) \cup ST(i)} x_{n_{k,i}, i, n_{i,j}, i} + \frac{1}{2} \sum_{(k,i,j) \in RT(i)} x_{n_{k,i}, i, n_{i,j}, i} \geq 1$ for all $i \in V_I$.

5.2.2 IIRPP Formulation using Path Transformations

Let $G^2 = (V^2, A^2)$ denote the transformed graph, where $V^2 = V \cup V_E^2 \cup V_A^2$ is the set of nodes and $A^2 = A_E^2 \cup A_A^2$ is the set of arcs. $V_A^2 = \{n_{i,j} | (i,j) \in A\}$ and $V_E^2 = \{n_{i,j}, n_{j,i} | (i,j) \in E\}$ are the set of nodes in G^2 corresponding to each arc $(i,j) \in A$ and each edge $(i,j) \in E$, respectively, in the original graph G . $A_A^2 = \{(i, n_{i,j}), (n_{i,j}, j) | (i,j) \in A\}$ and $A_E^2 = \{(i, n_{i,j}), (n_{i,j}, j), (j, n_{j,i}), (n_{j,i}, i) | (i,j) \in E\}$ are the set of arcs in G^2 corresponding to each arc $(i,j) \in A$ and each edge $(i,j) \in E$, respectively, in the original graph G . The cost of the arcs $(i, n_{i,j}) \in A_A^2$ and $(n_{i,j}, j) \in A_A^2$ are half the cost of the arc $(i,j) \in A$. The cost of the arcs $(i, n_{i,j}) \in A_E^2$, $(n_{i,j}, j) \in A_E^2$, $(j, n_{j,i}) \in A_E^2$, and $(n_{j,i}, i) \in A_E^2$ are half the cost of the edge $(i,j) \in E$.

In Figure 5.7, we show the transformed graph G^2 from the original graph G shown in Figure 5.5. All four nodes in G are also in G^2 . Nodes n_{12} , n_{21} , n_{14} , and n_{31} are added in G^2 . Arc $(1,4)$ in G is replaced by arcs $(1, n_{14})$ and $(n_{14}, 4)$ each

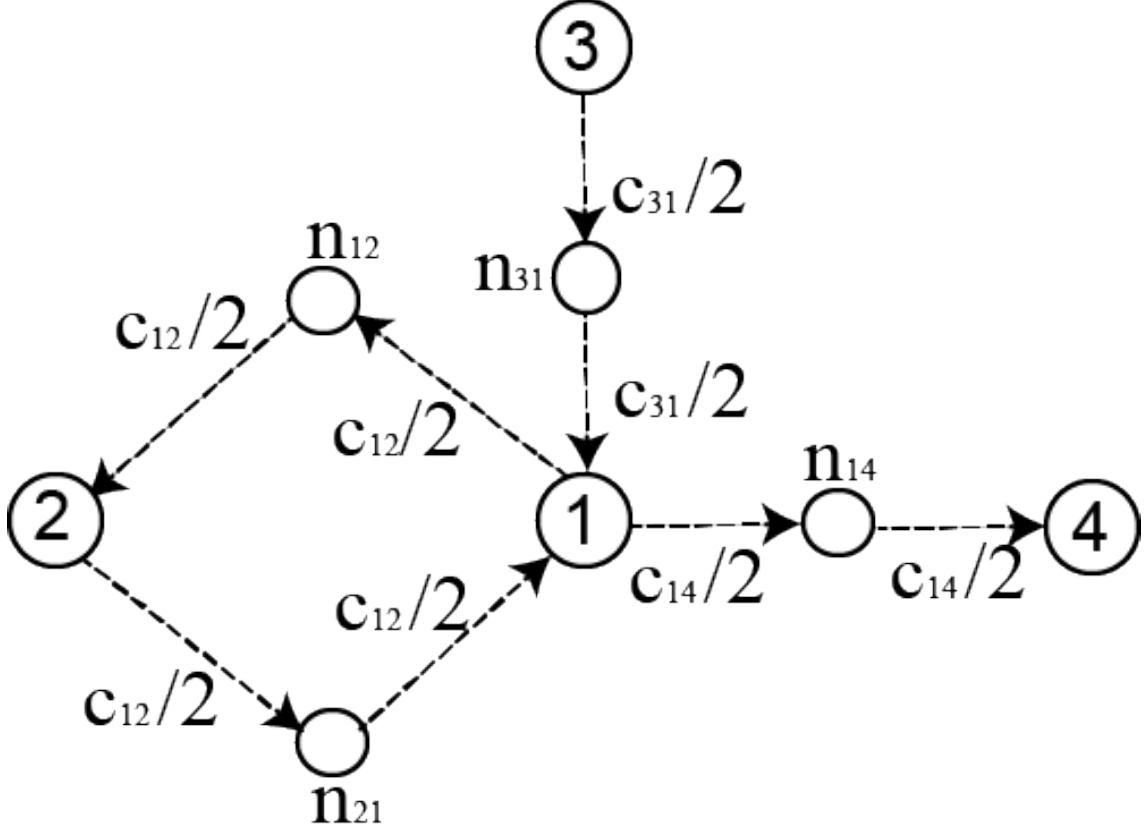


Figure 5.7: Transformed graph G^2 from the original graph G shown in Figure 5.5.

with a cost of $c_{14}/2$ in G^2 . Arc $(3,1)$ in G is replaced by arcs $(3, n_{31})$ and $(n_{31}, 1)$ each with a cost of $c_{31}/2$ in G^2 . Edge $(1,2)$ in G is replaced by arcs $(1, n_{12})$, $(n_{12}, 2)$, $(2, n_{21})$, and $(n_{21}, 1)$ each with a cost of $c_{12}/2$ in G^2 .

Let $V_R^2 = V_R \cup V_{E_R}^2 \cup V_{A_R}^2 \subseteq V^2$. $V_{E_R}^2 = \{n_{i,j}, n_{j,i} | (i,j) \in E_R\} \subseteq V_E^2$ is the set of nodes in G^2 corresponding to the required edges in G . $V_{A_R}^2 = \{n_{i,j} | (i,j) \in A_R\} \subseteq V_A^2$ is the set of nodes in G^2 corresponding to the required arcs in G . Let $A_R^2 = A_{E_R}^2 \cup A_{A_R}^2 \subseteq A^2$. $A_{A_R}^2 = \{(i, n_{i,j}), (n_{i,j}, j) | (i,j) \in A_R\} \subseteq A_A^2$ is the set of arcs in G^2 corresponding to the required arcs in G . $A_{E_R}^2 = \{(i, n_{i,j}), (n_{i,j}, j), (j, n_{j,i}), (n_{j,i}, i) | (i,j) \in E_R\} \subseteq A_E^2$ is the set of arcs in G^2 corresponding to the required edges in G . Let $G_R^2 = (V_R^2, A_R^2)$ be the graph induced from G^2 . The vertex sets of the connected

components of G_R^2 are denoted by V_1^2, \dots, V_p^2 . We define the binary decision variable $z_{n_{a,b}bn_{b,c}}$ denoting the first time arcs $(n_{a,b}, b)$ and $(b, n_{b,c})$ are traversed consecutively in the route on G^2 , i.e., the first time the turn $(a, b)(b, c)$ is made in G . We define the non-negative integer decision variables $y_{n_{a,b}b}$ and $y_{bn_{b,c}}$ to be the number of times arcs $(n_{a,b}, b)$ and $(b, n_{b,c})$ are traversed non-consecutively or consecutively after the first time, respectively, in the route on G^2 . The IIRPP formulation on the transformed graph G^2 (IIRPP-F2) based on the original graph G is given on the next page.

The objective function (5.22) minimizes the total cost (length) of the route. Constraint (5.23) ensures that the depot in G is a part of the route. Constraints (5.24) ensure that the required arcs in G are a part of the route. Constraints (5.25) ensure that the required edges in G are a part of the route. Constraints (5.26) and (5.27) are the flow conservation constraints for the nodes in G^2 . Constraints (5.28) ensure that the nodes in G that need to be video recorded are covered by at least one left turn or straight turn. Constraints (5.29) are the disjoint subtour elimination constraints. Constraints (5.30) define the decision variables corresponding to the arcs in G . Constraints (5.31) define the decision variables corresponding to the edges in G . Constraints (5.32) define the decision variables corresponding to the right turns, left turns, and straight turns in G . This formulation does not limit an intersection being covered with only left turns and straight turns. If the field of view of the camera improves in the future enabling two right turns to cover an intersection, we would modify constraints (5.28) as $\sum_{(k,i,j) \in LT(i) \cup ST(i)} z_{n_{k,i}in_{i,j}} + \frac{1}{2} \sum_{(k,i,j) \in RT(i)} z_{n_{k,i}in_{i,j}} \geq 1$ for all $i \in V_I$.

$$\begin{aligned}
(\text{IIRPP-F2}) \quad \min \quad & \frac{1}{2} \sum_{(i,j) \in E} c_{ij} \left(y_{jn_{j,i}} + \sum_{k \in \delta^-(j) \cup \delta(j) \setminus \{i\}} z_{n_{k,j}jn_{j,i}} + y_{n_{j,i}i} + \sum_{k \in \delta^+(i) \cup \delta(i) \setminus \{j\}} z_{n_{j,i}in_{i,k}} \right) \\
& + \frac{1}{2} \sum_{(i,j) \in E \cup A} c_{ij} \left(y_{in_{i,j}} + \sum_{k \in \delta^-(i) \cup \delta(i) \setminus \{j\}} z_{n_{k,i}in_{i,j}} + y_{n_{i,j}j} + \sum_{k \in \delta^+(j) \cup \delta(j) \setminus \{i\}} z_{n_{i,j}jn_{j,k}} \right)
\end{aligned} \tag{5.22}$$

$$\text{s.t.} \quad \sum_{j \in \delta^+(o) \cup \delta(o)} y_{on_{o,j}} \geq 1 \tag{5.23}$$

$$y_{in_{i,j}} + \sum_{k \in \delta^-(i) \cup \delta(i)} z_{n_{k,i}in_{i,j}} + y_{n_{i,j}j} + \sum_{k \in \delta^+(j) \cup \delta(j)} z_{n_{i,j}jn_{j,k}} \geq 2 \quad \forall (i,j) \in A_R \tag{5.24}$$

$$\begin{aligned}
& y_{in_{i,j}} + \sum_{k \in \delta^-(i) \cup \delta(i) \setminus \{j\}} z_{n_{k,i}in_{i,j}} + y_{n_{i,j}j} + \sum_{k \in \delta^+(j) \cup \delta(j) \setminus \{i\}} z_{n_{i,j}jn_{j,k}} + \\
& y_{jn_{j,i}} + \sum_{k \in \delta^-(j) \cup \delta(j) \setminus \{i\}} z_{n_{k,j}jn_{j,i}} + y_{n_{j,i}i} + \sum_{k \in \delta^+(i) \cup \delta(i) \setminus \{j\}} z_{n_{j,i}in_{i,k}} \geq 2 \quad \forall (i,j) \in E_R
\end{aligned} \tag{5.25}$$

$$y_{n_{i,j}j} + \sum_{k \in \delta^+(j) \cup \delta(j) \setminus \{i\}} z_{n_{i,j}jn_{j,k}} - y_{in_{i,j}} - \sum_{k \in \delta^-(i) \cup \delta(i) \setminus \{j\}} z_{n_{k,i}in_{i,j}} = 0 \quad \forall n_{i,j} \in V_E^2 \cup V_A^2 \tag{5.26}$$

$$\sum_{j \in \delta^+(i) \cup \delta(i)} y_{in_{i,j}} - \sum_{j \in \delta^-(i) \cup \delta(i)} y_{n_{j,i}i} = 0 \quad \forall i \in V \tag{5.27}$$

$$\sum_{(k,i,j) \in LT(i) \cup ST(i)} z_{n_{k,i}in_{i,j}} \geq 1 \quad \forall i \in V_I \tag{5.28}$$

$$\sum_{i \in S^2} \left(\sum_{j \in (\delta^+(i) \cup \delta(i)) \setminus S^2} \left(y_{in_{i,j}} + \sum_{k \in (\delta^-(i) \cup \delta(i)) \cap S^2} z_{n_{k,i}in_{i,j}} \right) \right) \geq 1 \quad \forall S^2 = \cup_{l \in Q} V_l^2, \quad Q \subset \{1, \dots, p\} \tag{5.29}$$

$$y_{in_{i,j}}, y_{n_{i,j}j} \geq 0 \text{ and integer} \quad \forall (i,j) \in A \tag{5.30}$$

$$y_{in_{i,j}}, y_{n_{i,j}j}, y_{jn_{j,i}}, y_{n_{j,i}i} \geq 0 \text{ and integer} \quad \forall (i,j) \in E \tag{5.31}$$

$$z_{n_{k,i}in_{i,j}} \in \{0, 1\} \quad \forall i \in V, (k,i,j) \in RT(i) \cup LT(i) \cup ST(i) \tag{5.32}$$

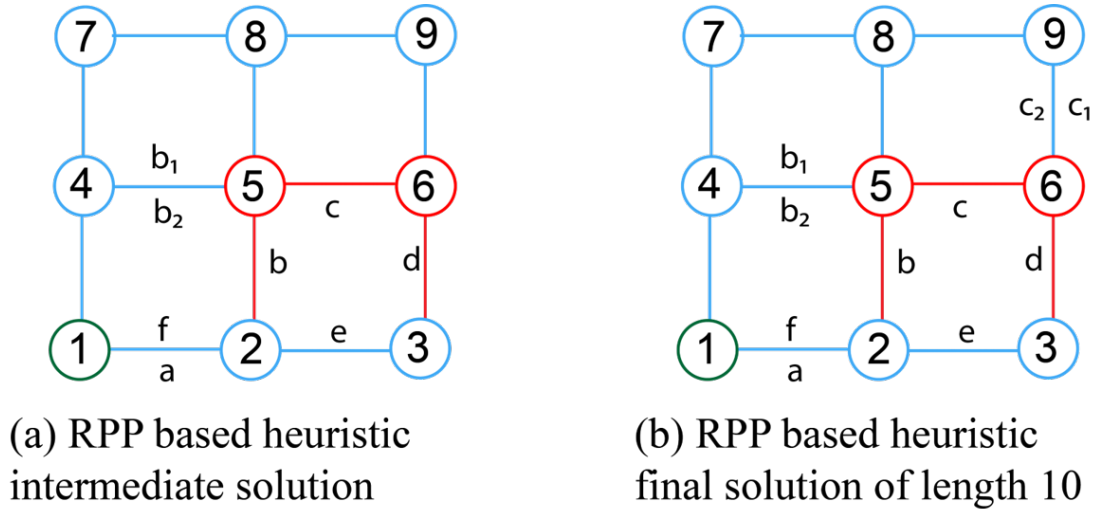


Figure 5.8: (Color online) RPP-H route on a small grid-like street network.

5.2.3 Heuristics

We develop three heuristics for the IIRPP. The first heuristic (RPP-H) starts by solving the RPP optimally on G . For each $i \in V_I$ that is not covered by at least one left turn or straight turn, the RPP route is locally modified to add the cheapest possible left turn or straight turn at i without affecting other parts of the route. Therefore, the RPP objective value is a lower bound for the RPP-H objective value. The second heuristic (IIRPP-F1-H) starts by solving the IIRPP-F1 optimally on G^1 without the disjoint subtour elimination constraints 5.17. The third heuristic (IIRPP-F2-H) starts by solving the IIRPP-F2 optimally on G^2 without the disjoint subtour elimination constraints 5.29. The disjoint subtours obtained are connected by solving a generalized traveling salesman problem (Kara et al. 2012) optimally on G^1 and G^2 to obtain the IIRPP-F1-H and IIRPP-F2-H solutions, respectively.

In Figure 5.8, we show the RPP-H solution on a small grid-like street network.

disjoint subtour elimination constraints. The intermediate solution has two disjoint subtours $(a \rightarrow b)$ and $(c \rightarrow d \rightarrow e \rightarrow f)$ with a total route length of six units. In this example, the first subtour $(a \rightarrow b)$ connects the green node and the second subtour $(c \rightarrow d \rightarrow e \rightarrow f)$ covers all the red edges and red nodes. Figure 5.9b shows the final IIRPP-F1-H and IIRPP-F2-H solution obtained by optimally solving a generalized traveling salesman problem connecting the two disjoint subtours. The final IIRPP-F1-H and IIRPP-F2-H solution $(a \rightarrow b \rightarrow b_1 \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow f_1)$ has a route length of eight units. The optimal IIRPP solution $(1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 2 \rightarrow 1)$ on this grid has a route length of six units.

5.3 Computational Experiments

5.3.1 Test Instances

We randomly generate test instances. Each graph (street network) represents a strongly connected grid-like structure. Each node is located within 10% of its exact position on the grid in order to produce street segments with slightly different lengths. The cost assigned to each street segment is proportional to the Euclidean distance between the nodes defining the street segment, where the average cost for a street segment is equal to 100. We have four scenarios for the number of nodes $|V| \in \{5 \times 5, 6 \times 6, 7 \times 7, 8 \times 8\}$. We first fix the position of the nodes. We then randomly assign a percentage p_a of street segments to be arcs and a percentage $1 - p_a$ of street segments to be edges, with two scenarios for $p_a \in \{5\%, 15\%\}$. We randomly remove a percentage p_d of arcs and the same percentage p_d of edges, with two

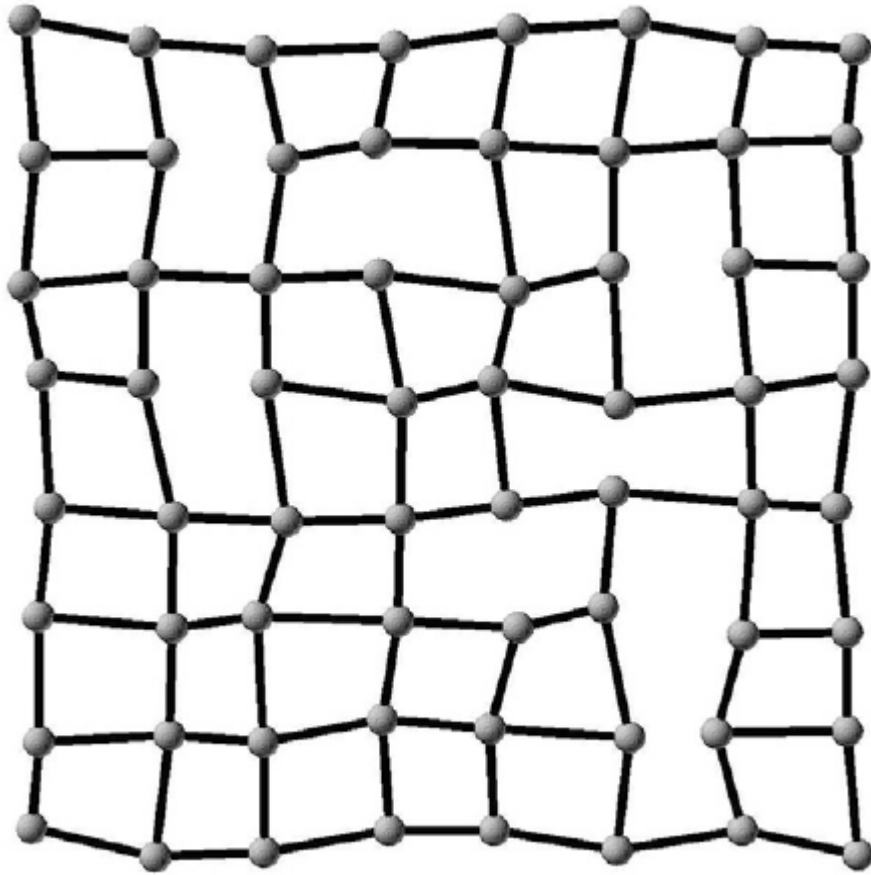


Figure 5.10: Example of an 8×8 instance.

scenarios for $p_d \in \{5\%, 15\%\}$. We use only strongly connected graphs. We randomly assign a percentage p_r of the remaining arcs as required and the same percentage p_r of the remaining edges as required, with two scenarios for $p_r \in \{20\%, 40\%\}$. Finally, we assign a percentage p_i of nodes defining the required edges and arcs to be video recorded, with two scenarios for $p_i \in \{20\%, 40\%\}$. We only use a graph that has at least one left turn or straight turn possible for each node that needs to be recorded. Therefore, for each of the four grids, we have 16 ($2 \times 2 \times 2 \times 2$) scenarios. We generate 10 graphs for each scenario, giving a total of 640 ($4 \times 16 \times 10$) test instances. In Figure 5.10, we show an example of an 8×8 instance.

5.3.2 Computational Results

In our computational experiments, we use Gurobi version 8.1, an i7 CPU with 32GB RAM, and a one-hour time limit. Practically, we do not need to fully transform the graph G to solve IIRPP-F1 and IIRPP-F2. For G^1 , we only need to transform the nodes in V_I , and the edges and arcs connected to those nodes. The remaining nodes, edges, and arcs need not be transformed because that would unnecessarily increase the problem complexity. For G^2 , we only need to transform the edges and arcs defined by the nodes in V_I on both ends. The remaining edges and arcs need not be transformed because that would unnecessarily increase the problem complexity. Therefore, the transformed graphs G^1 and G^2 could be mixed graphs. The RPP is solved optimally for all 640 instances within the time limit. All three heuristics RPP-H, IIRPP-F1-H, and IIRPP-F2-H are completed on all 640 instances within the time limit. For IIRPP-F1 and IIRPP-F2, we obtain the best feasible solution and the best lower bound for instances that are not solved optimally within the time limit. However, for IIRPP-F1 on eight instances and for IIRPP-F2 on one instance, a feasible solution was not found within the time limit.

In Tables 5.1 to 5.4, we compare the transformed graphs G^1 and G^2 to the original graph G . The first seven columns give the instance parameters and the number of nodes, edges, and arcs in G . The graph transformations depend on E_R , A_R , and V_I . The next five columns give the number of nodes, edges, and arcs in G^1 averaged over 10 instances, the number of instances out of 10 that are optimally solved (NOS) by IIRPP-F1, and the number of instances out of 10 that are not

optimally solved but for which a feasible solution is obtained (NFS) by IIRPP-F1. The last five columns give the number of nodes, edges, and arcs in G^2 averaged over 10 instances, the number of instances out of 10 that are optimally solved (NOS) by IIRPP-F2, and the number of instances out of 10 that are not optimally solved but for which a feasible solution is obtained (NFS) by IIRPP-F2.

In Table 5.1, the 160 instances on the grid with 25 nodes have an average of 33.00 edges and 4.00 arcs. G^1 has an average of 45.33 nodes, 22.54 edges, and 63.01 arcs. G^2 has an average of 27.37 nodes, 31.87 edges, and 8.63 arcs. Both IIRPP-F1 and IIRPP-F2 produced optimal solutions on all 160 instances. In Table 5.2, the 160 instances on the grid with 36 nodes have an average of 49.25 edges and 5.75 arcs. G^1 has an average of 67.63 nodes, 33.33 edges, and 97.89 arcs. G^2 has an average of 40.26 nodes, 47.21 edges, and 14.09 arcs. IIRPP-F1 produced optimal solutions on 158 instances and feasible solutions on the remaining two instances. IIRPP-F2 produced optimal solutions on all 160 instances. In Table 5.3, the 160 instances on the grid with 49 nodes have an average of 68.75 edges and 7.75 arcs. G^1 has an average of 94.85 nodes, 31.17 edges, and 142.33 arcs. G^2 has an average of 55.21 nodes, 65.83 edges, and 19.80 arcs. IIRPP-F1 produced optimal solutions on 150 instances and feasible solutions on nine instances. IIRPP-F1 did not produce a feasible solution on the remaining one instance. IIRPP-F2 produced optimal solutions on 159 instances and feasible solution on the remaining one instance. In Table 5.4, the 160 instances on the grid with 64 nodes have an average of 91.75 edges and 10 arcs. G^1 has an average of 126.88 nodes, 35.21 edges, and 194.76 arcs. G^2 has an average of 72.64 nodes, 87.64 edges, and 26.86 arcs. IIRPP-F1 produced

optimal solutions on 112 instances and feasible solutions on 41 instances. IIRPP-F1 did not generate a feasible solution to the remaining seven instances. IIRPP-F2 produced optimal solutions on 142 instances and feasible solutions on 17 instances. IIRPP-F2 did not produce a feasible solution on the remaining one instance. The only instance on 64 nodes that did not have a feasible solution by IIRPP-F2 was also among the seven instances that did not have a feasible solution by IIRPP-F1. With increasing grid size, the size of the transformed graph increases and it is more difficult to generate the optimal solution within the time limit. However, the number of nodes and number of arcs are always greater for G^1 compared to G^2 which makes IIRPP-F1 harder to solve.

In Tables 5.5 to 5.8, we show the percentage optimality gap between the best feasible solution (V) and the best lower bound (B) for the IIRPP formulations. The percentage gap is $100 \times \frac{V-B}{V}$. In calculating the average percentage optimality gap, we use only those instances that have feasible solutions for both IIRPP-F1 and IIRPP-F2. The first four columns give the instance parameters. The next two columns give the average percentage gap for the IIRPP-F1 for the two p_i scenarios. The last two columns give the average percentage gap for the IIRPP-F2 for the two p_i scenarios.

In Table 5.5, all instances are optimal for both IIRPP-F1 and IIRPP-F2. In Table 5.6, all instances with $p_i = 0.2$ are optimal and the average percentage gap is 0.26% for instances with $p_i = 0.4$ for IIRPP-F1. All instances are optimal for IIRPP-F2. In Table 5.7, the average percentage gap is 0.25% for instances with $p_i = 0.2$ and 0.27% for instances with $p_i = 0.4$ for IIRPP-F1. All instances

are optimal for IIRPP-F2. In Table 5.8, the average percentage gap is 2.45% for instances with $p_i = 0.2$ and 1.72% for instances with $p_i = 0.4$ for IIRPP-F1. The average percentage gap is 0.50% for instances with $p_i = 0.2$ and 0.78% for instances with $p_i = 0.4$ for IIRPP-F2. The larger percentage optimality gaps for IIRPP-F1 compared to IIRPP-F2 indicate that IIRPP-F1 is harder to solve.

In Tables 5.9 to 5.12, we show the running times for the RPP and IIRPP formulations. We use all instances in calculating the average RPP running times. In calculating the average IIRPP running times, we use only those instances that have feasible solutions for both IIRPP-F1 and IIRPP-F2. The first four columns give the instance parameters. The fifth column gives the average running time for the RPP, which is equivalent to the IIRPP with $p_i = 0$. The next two columns give the average running time for the IIRPP-F1 for the two p_i scenarios. The last two columns give the average running time for the IIRPP-F2 for the two p_i scenarios.

In Table 5.9, the average running time is 0.02 seconds for RPP. The average running time is 0.10 seconds for instances with $p_i = 0.2$ and 0.28 seconds for instances with $p_i = 0.4$ for IIRPP-F1. The average running time is 0.06 seconds for instances with $p_i = 0.2$ and 0.07 seconds for instances with $p_i = 0.4$ for IIRPP-F2. In Table 5.10, the average running time is 0.04 seconds for RPP. The average running time is 1.37 seconds for instances with $p_i = 0.2$ and 107.28 seconds for instances with $p_i = 0.4$ for IIRPP-F1. The average running time is 0.36 seconds for instances with $p_i = 0.2$ and 2.75 seconds for instances with $p_i = 0.4$ for IIRPP-F2. In Table 5.11, the average running time is 0.05 seconds for RPP. The average running time is 261.90 seconds for instances with $p_i = 0.2$ and 286.15 seconds for instances with

$p_i = 0.4$ for IIRPP-F1. The average running time is 21.17 seconds for instances with $p_i = 0.2$ and 56.82 seconds for instances with $p_i = 0.4$ for IIRPP-F2. In Table 5.12, the average running time is 0.09 seconds for RPP. The average running time is 1076.64 seconds for instances with $p_i = 0.2$ and 1338.47 seconds for instances with $p_i = 0.4$ for IIRPP-F1. The average running time is 320.72 seconds for instances with $p_i = 0.2$ and 747.15 seconds for instances with $p_i = 0.4$ for IIRPP-F2. The larger running times for IIRPP-F1 compared to IIRPP-F2 indicate that IIRPP-F1 is harder to solve.

In Tables 5.13 to 5.16, we show the percentage gap between the RPP optimal solution (RPP OS) and the best feasible solutions from the IIRPP (IIRPP FS) formulations. The percentage gap is $100 \times \frac{\text{IIRPP FS} - \text{RPP OS}}{\text{RPP OS}}$. We use all instances in calculating the average RPP optimal solutions. In calculating the average percentage gap, we use only those instances that have feasible solutions for both IIRPP-F1 and IIRPP-F2. The first four columns give the instance parameters. The fifth column gives the average optimal solution for the RPP, which is equivalent to the IIRPP with $p_i = 0$. The next two columns give the average percentage gap for the IIRPP-F1 for the two p_i scenarios. The last two columns give the average percentage gap for the IIRPP-F2 for the two p_i scenarios.

In Table 5.13, the average RPP optimal solution is 2088.01. The average percentage gap is 3.86% for instances with $p_i = 0.2$ and 7.41% for instances with $p_i = 0.4$ for both IIRPP-F1 and IIRPP-F2. In Table 5.14, the average RPP optimal solution is 3110.55. The average percentage gap is 3.73% for instances with $p_i = 0.2$ and 8.56% for instances with $p_i = 0.4$ for IIRPP-F1. The average percentage gap is

3.73% for instances with $p_i = 0.2$ and 8.45% for instances with $p_i = 0.4$ for IIRPP-F2. In Table 5.15, the average RPP optimal solution is 4245.20. The average percentage gap is 3.64% for instances with $p_i = 0.2$ and 7.76% for instances with $p_i = 0.4$ for IIRPP-F1. The average percentage gap is 3.62% for instances with $p_i = 0.2$ and 7.73% for instances with $p_i = 0.4$ for IIRPP-F2. In Table 5.16, the average RPP optimal solution is 5695.42. The average percentage gap is 6.04% for instances with $p_i = 0.2$ and 8.92% for instances with $p_i = 0.4$ for IIRPP-F1. The average percentage gap is 3.75% for instances with $p_i = 0.2$ and 8.51% for instances with $p_i = 0.4$ for IIRPP-F2. The larger percentage gaps for IIRPP-F1 compared to IIRPP-F2 indicate that IIRPP-F1 is harder to solve.

In Tables 5.17 to 5.20, we compare the RPP based and IIRPP based heuristics with the IIRPP formulations. We show the percentage gap between the heuristic solutions (H FS) and the IIRPP best feasible solution (IIRPP BFS). For each instance, the IIRPP best feasible solution is the lower of the two best feasible solutions from IIRPP-F1 and IIRPP-F2. The percentage gap is $100 \times \frac{\text{H FS} - \text{IIRPP BFS}}{\text{IIRPP BFS}}$. We compare the running times of the heuristic solutions with the best running time of the IIRPP best feasible solution. If the best feasible solutions from IIRPP-F1 and IIRPP-F2 are equal for an instance, we select the lower of the two running times. In calculating the average heuristic solutions and the average running times, we use only those instances that have at least a feasible solution from IIRPP-F1 or IIRPP-F2. The first five columns give the instance parameters. The sixth column gives the average IIRPP best feasible solution and the seventh column gives the average best running time. The next three columns give the average percentage gap and the

average running time for RPP-H, and the number of instances out of 10 (NBS) for which the RPP-H solution is equal to or lower than the corresponding IIRPP best feasible solution. The next three columns give the average percentage gap and the average running time for IIRPP-F1-H, and the number of instances out of 10 (NBS) for which the IIRPP-F1-H solution is equal to or lower than the corresponding IIRPP best feasible solution. The last three columns give the average percentage gap and the average running time for IIRPP-F2-H, and the number of instances out of 10 (NBS) for which the IIRPP-F2-H solution is equal to or lower than the corresponding IIRPP best feasible solution.

In Table 5.17, the average IIRPP best feasible solution is 2205.38 and the average best running time is 0.06 seconds. The average percentage gap is 8.30% and the average running time is 0.03 seconds for RPP-H. The RPP-H solutions are at least as good as the IIRPP best feasible solutions on an average of 2.94 out of 10 instances. The average percentage gap is 16.82% and the average running time is 0.13 seconds for IIRPP-F1-H. The IIRPP-F1-H solutions are at least as good as the IIRPP best feasible solutions on an average of 2.00 out of 10 instances. The average percentage gap is 16.42% and the average running time is 0.07 seconds for IIRPP-F2-H. The IIRPP-F2-H solutions are at least as good as the IIRPP best feasible solutions on an average of 1.81 out of 10 instances. In Table 5.18, the average IIRPP best feasible solution is 3299.45 and the average best running time is 1.56 seconds. The average percentage gap is 9.81% and the average running time is 0.06 seconds for RPP-H. The RPP-H solutions are at least as good as the IIRPP best feasible solutions on an average of 1.44 out of 10 instances. The average percentage

gap is 20.37% and the average running time is 0.40 seconds for IIRPP-F1-H. The IIRPP-F1-H solutions are at least as good as the IIRPP best feasible solutions on an average of 0.50 out of 10 instances. The average percentage gap is 20.38% and the average running time is 0.14 seconds for IIRPP-F2-H. The IIRPP-F2-H solutions are at least as good as the IIRPP best feasible solutions on an average of 0.50 out of 10 instances. In Table 5.19, the average IIRPP best feasible solution is 4501.58 and the average best running time is 76.43 seconds. The average percentage gap is 11.14% and the average running time is 0.10 seconds for RPP-H. The RPP-H solutions are at least as good as the IIRPP best feasible solutions on an average of 0.75 out of 10 instances. The average percentage gap is 22.21% and the average running time is 1.10 seconds for IIRPP-F1-H. The IIRPP-F1-H solutions are at least as good as the IIRPP best feasible solutions on an average of 0.31 out of 10 instances. The average percentage gap is 21.43% and the average running time is 0.31 seconds for IIRPP-F2-H. The IIRPP-F2-H solutions are at least as good as the IIRPP best feasible solutions on an average of 0.25 out of 10 instances. In Table 5.20, the average IIRPP best feasible solution is 6111.23 and the average best running time is 673.01 seconds. The average percentage gap is 10.26% and the average running time is 0.20 seconds for RPP-H. The RPP-H solutions are at least as good as the IIRPP best feasible solutions on an average of 0.63 out of 10 instances. The average percentage gap is 20.86% and the average running time is 2.87 seconds for IIRPP-F1-H. The IIRPP-F1-H solutions are at least as good as the IIRPP best feasible solutions on an average of 0.31 out of 10 instances. The average percentage gap is 20.90% and the average running time is 0.67 seconds for IIRPP-F2-H. The IIRPP-F2-H solutions are

at least as good as the IIRPP best feasible solutions on an average of 0.25 out of 10 instances. IIRPP-F1-H and IIRPP-F2-H produce similar average percentage gaps. However, IIRPP-F2-H is faster than IIRPP-F1-H. RPP-H is the fastest and the best of the three heuristics. The average percentage gap for RPP-H is around 10% from the IIRPP best feasible solution on instances up to 64 nodes within 0.20 seconds, which is half of the average percentage gaps for IIRPP-F1-H and IIRPP-F2-H.

5.3.3 Summary of Results

In Table 5.21, we summarize the results shown in Tables 5.1 to 5.4. As the size of the grid increases from 25 to 64, the average number of arcs increases from 4.00 to 10.00 in G . The average number of nodes increases from 45.33 to 126.88 and the average number of arcs increases from 63.01 to 194.76 in G^1 . The average number of nodes increases from 27.37 to 72.64 and the average number of arcs increases from 8.63 to 26.86 in G^2 . This shows that the graph transformation required to solve IIRPP-F2 is significantly smaller than the graph transformation required to solve IIRPP-F1. IIRPP-F1 produced optimal solutions on 580 instances, feasible solutions on 52 instances, and did not generate a feasible solution to the remaining eight instances. IIRPP-F2 produced optimal solutions on 621 instances, feasible solutions on 18 instances, and did not generate a feasible solution to the remaining one instance. This clearly shows the difference in solution quality between IIRPP-F1 and IIRPP-F2. The smaller size of G^2 computationally helps IIRPP-F2 to produce better solutions compared to IIRPP-F1 within a reasonable time limit.

In Table 5.22, we summarize the results shown in Tables 5.5 to 5.8. As the size of the grid increases from 25 to 64, the average percentage optimality gap increases from 0.00% to 2.45% for instances with $p_i = 0.2$ and from 0.00% to 1.72% for instances with $p_i = 0.4$ for IIRPP-F1. All instances both with $p_i = 0.2$ and $p_i = 0.4$ are optimal up to the grid size of 49 for IIRPP-F2. Instances with $p_i = 0.2$ and $p_i = 0.4$ have an average percentage optimality gap of 0.50% and 0.78%, respectively, for the grid size of 64 for IIRPP-F2. This shows that the IIRPP-F2 produces better quality solutions compared to IIRPP-F1.

In Table 5.23, we summarize the results shown in Tables 5.9 to 5.12. As the size of the grid increases from 25 to 64, the average running time increases from 0.02 seconds to 0.09 seconds for RPP. The average running time increases from 0.10 seconds to 1076.64 seconds for instances with $p_i = 0.2$ and from 0.28 seconds to 1338.47 seconds for instances with $p_i = 0.4$ for IIRPP-F1. The average running time increases from 0.06 seconds to 320.72 seconds for instances with $p_i = 0.2$ and from 0.07 seconds to 747.15 seconds for instances with $p_i = 0.4$ for IIRPP-F2. The average running times for RPP is significantly smaller than IIRPP-F1 and IIRPP-F2 indicating the difference in complexity between RPP and IIRPP. IIRPP-F2 is faster than IIRPP-F1 by 3.4 times for instances with $p_i = 0.2$ and 1.8 times for instances with $p_i = 0.4$. This shows that IIRPP-F2 can find solutions faster than IIRPP-F1 because of the smaller size of G^2 . Instances with higher p_i values takes more time to be solved.

In Table 5.24, we summarize the results shown in Tables 5.13 to 5.16. As the size of the grid increases from 25 to 64, the average optimal solution increases from

2088.01 to 5695.42 for RPP. The average percentage gap increases from 3.86% to 6.04% for instances with $p_i = 0.2$ and from 7.41% to 8.92% for instances with $p_i = 0.4$ for IIRPP-F1. The average percentage gap remains about 3.80% for instances with $p_i = 0.2$ and increases from 7.41% to 8.51% for instances with $p_i = 0.4$ for IIRPP-F2. The average route lengths for RPP increases between 4% and 9% to cover the intersections that need to be photographed. When all instances in a group are optimally solved by both IIRPP-F1 and IIRPP-F2, the average percentage gap is equal for IIRPP-F1 and IIRPP-F2. Otherwise, the average percentage gap is lower for IIRPP-F2 than IIRPP-F1 because for most of the instances IIRPP-F2 produces better quality solutions compared to IIRPP-F1.

In Table 5.25, we summarize the results shown in Tables 5.17 to 5.20. As the size of the grid increases from 25 to 64, the average IIRPP best feasible solution increases from 2205.38 to 6111.23 and the average best running time increases from 0.06 seconds to 673.01 seconds. The average percentage gap remains between 8.30% to 11.14% and the average running time increases from 0.03 seconds to 0.20 seconds for RPP-H. The number of RPP-H solutions that are at least as good as the IIRPP best feasible solutions decreases from 2.94 to 0.63 out of 10 instances. The average percentage gap remains between 16.82% to 22.21% and the average running time increases from 0.13 seconds to 2.87 seconds for IIRPP-F1-H. The number of IIRPP-F1-H solutions that are at least as good as the IIRPP best feasible solutions decreases from 2.00 to 0.31 out of 10 instances. The average percentage gap remains between 16.42% to 21.43% and the average running time increases from 0.13 seconds to 2.87 seconds for IIRPP-F2-H. The number of IIRPP-F2-H solutions that are at least

as good as the IIRPP best feasible solutions decreases from 1.81 to 0.31 out of 10 instances. IIRPP-F1-H and IIRPP-F2-H have similar average percentage gaps between 16.50% and 22.50%. However, IIRPP-F2-H is about 3.5 times faster on average than IIRPP-F1-H indicating that the underlying graph transformation for IIRPP-F2 is computationally easier than the underlying graph transformation for IIRPP-F1. RPP-H is about 3.3 times faster on average than IIRPP-F2-H and has half the average percentage gap of IIRPP-F2-H. On average, RPP-H comes within 11% of the IIRPP best feasible solution in 0.20 seconds for instances with grid size of 64.

5.4 Conclusions and Future Directions

We introduced an important variant of the RPP involving turns (IIRPP) which is relevant for road inspections. We gave two formulations IIRPP-F1 and IIRPP-F2, based on two different graph transformations. Using running times and optimality gaps, we showed that IIRPP-F2 is a faster and stronger formulation compared to IIRPP-F1 because of the smaller size of the transformed graph G^2 compared to G^1 . With increasing size of the instances, the running times and the number of instances that could not be solved optimally with the formulations increased and are substantial for IIRPP-F1. Even IIRPP-F2 could not solve 18 out of 160 instances with a grid size of 64 within one hour. We also showed that with a larger number of intersections to be inspected (larger values of p_i), the route lengths increase, and the instances take longer to solve. We developed a RPP based heuristic and two

IIRPP based heuristics. RPP-H performed better than IIRPP-F1-H and IIRPP-F2-H, coming within 11% of the IIRPP solutions in less than a fraction of a second for instances with a grid size of 64. In future work, we expect to develop branch-and-cut algorithms to solve larger instances optimally within a reasonable amount of time. We also plan to develop smarter heuristics to reduce the gap between heuristic solutions and optimal IIRPP solutions. We have some ideas on how to accomplish this, but leave it for further study.

IIRPP-F1												IIRPP-F2											
$ V $	p_a	p_d	$ E $	$ A $	p_r	p_i	$ V^1 $	$ E^1 $	$ A^1 $	NOS	NFS	$ V^2 $	$ E^2 $	$ A^2 $	NOS	NFS							
25	0.05	0.05	37	2	0.2	0.2	36.20	30.60	37.30	10	0	25.00	37.00	2.00	10	0							
25	0.05	0.05	37	2	0.2	0.4	49.40	24.50	75.20	10	0	28.30	35.40	8.50	10	0							
25	0.05	0.05	37	2	0.4	0.2	42.70	27.60	56.00	10	0	26.60	36.20	5.20	10	0							
25	0.05	0.05	37	2	0.4	0.4	61.80	18.10	111.40	10	0	30.40	34.40	12.60	10	0							
25	0.05	0.15	33	2	0.2	0.2	35.00	27.30	32.40	10	0	25.40	32.80	2.80	10	0							
25	0.05	0.15	33	2	0.2	0.4	44.40	21.90	60.90	10	0	26.00	32.50	4.00	10	0							
25	0.05	0.15	33	2	0.4	0.2	40.20	24.80	46.30	10	0	26.40	32.30	4.80	10	0							
25	0.05	0.15	33	2	0.4	0.4	59.30	16.60	99.80	10	0	32.40	29.40	16.60	10	0							
25	0.15	0.05	33	6	0.2	0.2	35.10	27.80	35.60	10	0	25.40	32.80	6.80	10	0							
25	0.15	0.05	33	6	0.2	0.4	44.30	23.30	60.10	10	0	26.90	32.10	9.70	10	0							
25	0.15	0.05	33	6	0.4	0.2	42.20	23.90	57.10	10	0	26.20	32.40	8.40	10	0							
25	0.15	0.05	33	6	0.4	0.4	60.10	15.90	104.80	10	0	30.20	30.60	16.00	10	0							
25	0.15	0.15	29	6	0.2	0.2	33.50	24.70	28.80	10	0	25.30	29.00	6.30	10	0							
25	0.15	0.15	29	6	0.2	0.4	43.90	19.00	60.40	10	0	26.30	28.40	8.50	10	0							
25	0.15	0.15	29	6	0.4	0.2	41.70	20.40	53.80	10	0	26.50	28.40	8.70	10	0							
25	0.15	0.15	29	6	0.4	0.4	55.50	14.30	88.30	10	0	30.60	26.20	17.20	10	0							
Average			33	4			45.33	22.54	63.01	10.00	0.00	27.37	31.87	8.63	10.00	0.00							

Table 5.1: Comparison of the transformed graphs G^1 and G^2 with the original graph G on 25 nodes.

IIRPP-F1												IIRPP-F2											
$ V $	p_a	p_d	$ E $	$ A $	p_r	p_i	$ V^1 $	$ E^1 $	$ A^1 $	NOS	NFS	$ V^2 $	$ E^2 $	$ A^2 $	NOS	NFS							
36	0.05	0.05	55	3	0.2	0.2	51.90	46.30	51.30	10	0	37.00	54.50	5.00	10	0							
36	0.05	0.05	55	3	0.2	0.4	70.80	37.50	107.30	10	0	41.40	52.30	13.80	10	0							
36	0.05	0.05	55	3	0.4	0.2	65.30	39.80	90.60	10	0	39.50	53.30	9.90	10	0							
36	0.05	0.05	55	3	0.4	0.4	99.80	24.20	192.40	10	0	48.30	49.00	27.30	10	0							
36	0.05	0.15	49	3	0.2	0.2	50.60	40.70	47.80	10	0	36.60	48.70	4.20	10	0							
36	0.05	0.15	49	3	0.2	0.4	64.90	33.80	86.50	9	1	39.40	47.30	9.80	10	0							
36	0.05	0.15	49	3	0.4	0.2	60.60	35.70	76.00	10	0	37.70	48.20	6.30	10	0							
36	0.05	0.15	49	3	0.4	0.4	88.20	23.30	154.20	9	1	45.70	44.30	22.10	10	0							
36	0.15	0.05	49	9	0.2	0.2	51.70	41.50	53.90	10	0	37.10	48.50	11.10	10	0							
36	0.15	0.05	49	9	0.2	0.4	67.00	33.50	99.00	10	0	39.90	47.10	16.70	10	0							
36	0.15	0.05	49	9	0.4	0.2	61.60	35.90	82.70	10	0	38.30	47.90	13.50	10	0							
36	0.15	0.05	49	9	0.4	0.4	91.90	22.50	169.20	10	0	45.40	44.60	27.20	10	0							
36	0.15	0.15	44	8	0.2	0.2	48.80	37.10	45.30	10	0	36.50	43.80	8.90	10	0							
36	0.15	0.15	44	8	0.2	0.4	62.50	30.40	83.00	10	0	38.80	42.70	13.40	10	0							
36	0.15	0.15	44	8	0.4	0.2	59.20	31.70	74.40	10	0	37.70	43.20	11.30	10	0							
36	0.15	0.15	44	8	0.4	0.4	87.20	19.40	152.70	10	0	44.90	40.00	24.90	10	0							
Average			49.25	5.75			67.63	33.33	97.89	9.88	0.13	40.26	47.21	14.09	10.00	0.00							

Table 5.2: Comparison of the transformed graphs G^1 and G^2 with the original graph G on 36 nodes.

IIRPP-F1												IIRPP-F2											
$ V $	p_a	p_d	$ E $	$ A $	p_r	p_i	$ V^1 $	$ E^1 $	$ A^1 $	NOS	NFS	$ V^2 $	$ E^2 $	$ A^2 $	NOS	NFS							
49	0.05	0.05	76	4	0.2	0.2	72.50	63.10	77.70	10	0	50.20	75.40	6.40	10	0							
49	0.05	0.05	76	4	0.2	0.4	98.70	50.80	156.00	9	1	55.80	72.60	17.60	10	0							
49	0.05	0.05	76	4	0.4	0.2	88.90	54.40	126.60	10	0	52.00	74.50	10.00	10	0							
49	0.05	0.05	76	4	0.4	0.4	131.30	35.90	247.20	10	0	63.80	69.00	32.80	10	0							
49	0.05	0.15	68	4	0.2	0.2	68.70	24.80	62.70	8	2	50.40	67.30	6.80	10	0							
49	0.05	0.15	68	4	0.2	0.4	92.44	16.60	134.44	9	0	54.67	65.22	15.22	9	1							
49	0.05	0.15	68	4	0.4	0.2	85.10	27.80	110.10	10	0	53.60	65.80	13.00	10	0							
49	0.05	0.15	68	4	0.4	0.4	121.40	23.30	214.40	9	1	60.10	62.60	25.90	10	0							
49	0.15	0.05	69	12	0.2	0.2	73.40	23.90	83.40	10	0	50.80	68.10	15.60	10	0							
49	0.15	0.05	69	12	0.2	0.4	99.90	15.90	159.10	9	1	56.30	65.80	25.70	10	0							
49	0.15	0.05	69	12	0.4	0.2	89.70	24.70	133.10	10	0	53.00	67.20	19.60	10	0							
49	0.15	0.05	69	12	0.4	0.4	129.50	19.00	239.60	10	0	65.20	61.60	43.00	10	0							
49	0.15	0.15	62	11	0.2	0.2	71.20	20.40	76.50	7	3	50.60	61.20	14.20	10	0							
49	0.15	0.15	62	11	0.2	0.4	93.10	14.30	139.40	9	1	53.50	60.10	19.30	10	0							
49	0.15	0.15	62	11	0.4	0.2	83.10	46.30	110.50	10	0	52.10	60.50	17.10	10	0							
49	0.15	0.15	62	11	0.4	0.4	118.70	37.50	206.50	10	0	61.30	56.40	34.50	10	0							
Average			68.75	7.75			94.85	31.17	142.33	9.38	0.56	55.21	65.83	19.80	9.94	0.06							

Table 5.3: Comparison of the transformed graphs G^1 and G^2 with the original graph G on 49 nodes.

IIRPP-F1												IIRPP-F2											
$ V $	p_a	p_d	$ E $	$ A $	p_r	p_i	$ V^1 $	$ E^1 $	$ A^1 $	NOS	NFS	$ V^2 $	$ E^2 $	$ A^2 $	NOS	NFS							
64	0.05	0.05	102	5	0.2	0.2	98.78	39.80	114.67	6	3	65.11	101.44	7.22	7	3							
64	0.05	0.05	102	5	0.2	0.4	137.30	24.20	229.10	7	3	73.70	97.30	24.10	8	2							
64	0.05	0.05	102	5	0.4	0.2	121.20	40.70	183.20	7	3	69.60	99.20	16.20	10	0							
64	0.05	0.05	102	5	0.4	0.4	178.10	33.80	347.20	7	3	84.20	92.10	45.00	7	3							
64	0.05	0.15	91	5	0.2	0.2	94.20	35.70	97.30	5	5	65.30	90.40	7.50	8	2							
64	0.05	0.15	91	5	0.2	0.4	125.88	23.30	189.25	4	4	71.38	87.50	19.38	8	1							
64	0.05	0.15	91	5	0.4	0.2	112.38	41.50	149.38	6	2	69.25	88.38	15.50	10	0							
64	0.05	0.15	91	5	0.4	0.4	158.30	33.50	277.10	8	2	80.50	82.90	37.70	10	0							
64	0.15	0.05	92	16	0.2	0.2	98.90	35.90	119.90	7	3	66.10	91.10	19.90	9	1							
64	0.15	0.05	92	16	0.2	0.4	138.20	22.50	235.50	6	4	74.50	87.00	36.50	7	3							
64	0.15	0.05	92	16	0.4	0.2	116.90	37.10	168.70	9	1	70.00	89.20	27.60	10	0							
64	0.15	0.05	92	16	0.4	0.4	173.20	30.40	330.80	10	0	82.70	83.30	52.10	10	0							
64	0.15	0.15	82	14	0.2	0.2	90.00	31.70	88.40	9	1	66.20	81.00	18.20	10	0							
64	0.15	0.15	82	14	0.2	0.4	119.90	19.40	173.10	6	4	71.20	78.80	27.60	9	1							
64	0.15	0.15	82	14	0.4	0.2	109.33	63.10	140.78	8	1	69.56	79.44	24.67	10	0							
64	0.15	0.15	82	14	0.4	0.4	157.44	50.80	271.78	7	2	83.00	73.22	50.56	9	1							
Average			91.75	10			126.88	35.21	194.76	7.00	2.56	72.64	87.64	26.86	8.88	1.06							

Table 5.4: Comparison of the transformed graphs G^1 and G^2 with the original graph G on 64 nodes.

				IIRPP-F1 Gap%		IIRPP-F2 Gap%	
$ V $	p_a	p_d	p_r	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$
25	0.05	0.05	0.2	0.00	0.00	0.00	0.00
25	0.05	0.05	0.4	0.00	0.00	0.00	0.00
25	0.05	0.15	0.2	0.00	0.00	0.00	0.00
25	0.05	0.15	0.4	0.00	0.00	0.00	0.00
25	0.15	0.05	0.2	0.00	0.00	0.00	0.00
25	0.15	0.05	0.4	0.00	0.00	0.00	0.00
25	0.15	0.15	0.2	0.00	0.00	0.00	0.00
25	0.15	0.15	0.4	0.00	0.00	0.00	0.00
Average				0.00	0.00	0.00	0.00

Table 5.5: Comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations on 25 nodes.

				IIRPP-F1 Gap%		IIRPP-F2 Gap%	
$ V $	p_a	p_d	p_r	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$
36	0.05	0.05	0.2	0.00	0.00	0.00	0.00
36	0.05	0.05	0.4	0.00	0.00	0.00	0.00
36	0.05	0.15	0.2	0.00	0.75	0.00	0.00
36	0.05	0.15	0.4	0.00	1.31	0.00	0.00
36	0.15	0.05	0.2	0.00	0.00	0.00	0.00
36	0.15	0.05	0.4	0.00	0.00	0.00	0.00
36	0.15	0.15	0.2	0.00	0.00	0.00	0.00
36	0.15	0.15	0.4	0.00	0.00	0.00	0.00
Average				0.00	0.26	0.00	0.00

Table 5.6: Comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations on 36 nodes.

				IIRPP-F1 Gap%		IIRPP-F2 Gap%	
$ V $	p_a	p_d	p_r	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$
49	0.05	0.05	0.2	0.00	0.44	0.00	0.00
49	0.05	0.05	0.4	0.00	0.00	0.00	0.00
49	0.05	0.15	0.2	0.75	0.00	0.00	0.00
49	0.05	0.15	0.4	0.00	0.37	0.00	0.00
49	0.15	0.05	0.2	0.00	0.89	0.00	0.00
49	0.15	0.05	0.4	0.00	0.00	0.00	0.00
49	0.15	0.15	0.2	1.27	0.42	0.00	0.00
49	0.15	0.15	0.4	0.00	0.00	0.00	0.00
Average				0.25	0.27	0.00	0.00

Table 5.7: Comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations on 49 nodes.

				IIRPP-F1 Gap%		IIRPP-F2 Gap%	
$ V $	p_a	p_d	p_r	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$
64	0.05	0.05	0.2	7.45	1.12	1.62	2.85
64	0.05	0.05	0.4	1.88	2.37	0.00	1.05
64	0.05	0.15	0.2	5.78	3.57	2.29	0.69
64	0.05	0.15	0.4	1.22	0.35	0.00	0.00
64	0.15	0.05	0.2	2.27	2.93	0.08	1.19
64	0.15	0.05	0.4	0.34	0.00	0.00	0.00
64	0.15	0.15	0.2	0.52	2.11	0.00	0.48
64	0.15	0.15	0.4	0.16	1.33	0.00	0.00
Average				2.45	1.72	0.50	0.78

Table 5.8: Comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations on 64 nodes.

				RPP Time	IIRPP-F1 Time			IIRPP-F2 Time	
$ V $	p_a	p_d	p_r	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$	
25	0.05	0.05	0.2	0.03	0.10	0.20	0.06	0.07	
25	0.05	0.05	0.4	0.02	0.11	0.09	0.08	0.09	
25	0.05	0.15	0.2	0.02	0.13	0.26	0.08	0.05	
25	0.05	0.15	0.4	0.02	0.06	0.13	0.05	0.07	
25	0.15	0.05	0.2	0.02	0.19	0.04	0.05	0.04	
25	0.15	0.05	0.4	0.04	0.05	0.08	0.04	0.06	
25	0.15	0.15	0.2	0.02	0.07	0.04	0.04	0.04	
25	0.15	0.15	0.4	0.02	0.12	1.35	0.05	0.11	
Average				0.02	0.10	0.28	0.06	0.07	

Table 5.9: Comparison of the running time for the RPP and IIRPP formulations on 25 nodes.

				RPP Time	IIRPP-F1 Time			IIRPP-F2 Time	
$ V $	p_a	p_d	p_r	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$	
36	0.05	0.05	0.2	0.05	3.05	1.56	0.85	0.82	
36	0.05	0.05	0.4	0.04	0.28	2.07	0.15	0.80	
36	0.05	0.15	0.2	0.05	2.71	366.90	0.95	16.83	
36	0.05	0.15	0.4	0.04	1.28	361.94	0.15	0.79	
36	0.15	0.05	0.2	0.03	0.52	4.75	0.19	0.86	
36	0.15	0.05	0.4	0.04	1.05	29.55	0.31	1.23	
36	0.15	0.15	0.2	0.03	0.20	0.32	0.12	0.12	
36	0.15	0.15	0.4	0.03	1.86	91.16	0.11	0.52	
Average				0.04	1.37	107.28	0.36	2.75	

Table 5.10: Comparison of the running time for the RPP and IIRPP formulations on 36 nodes.

				RPP Time	IIRPP-F1 Time			IIRPP-F2 Time	
$ V $	p_a	p_d	p_r	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$	
49	0.05	0.05	0.2	0.08	21.16	453.35	2.21	81.86	
49	0.05	0.05	0.4	0.05	5.21	166.64	0.69	2.59	
49	0.05	0.15	0.2	0.05	746.04	211.99	91.92	229.79	
49	0.05	0.15	0.4	0.06	145.51	714.47	0.90	126.83	
49	0.15	0.05	0.2	0.04	86.25	360.86	0.94	4.43	
49	0.15	0.05	0.4	0.04	1.55	9.89	0.53	2.28	
49	0.15	0.15	0.2	0.06	1086.17	363.09	71.98	6.12	
49	0.15	0.15	0.4	0.03	3.33	8.92	0.21	0.67	
Average				0.05	261.90	286.15	21.17	56.82	

Table 5.11: Comparison of the running time for the RPP and IIRPP formulations on 49 nodes.

				RPP Time	IIRPP-F1 Time			IIRPP-F2 Time	
$ V $	p_a	p_d	p_r	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$	
64	0.05	0.05	0.2	0.11	1352.75	1236.98	807.11	1057.13	
64	0.05	0.05	0.4	0.13	1145.00	1600.07	55.99	1183.37	
64	0.05	0.15	0.2	0.11	2096.28	2510.66	1016.27	1439.14	
64	0.05	0.15	0.4	0.08	1479.86	844.08	23.37	331.58	
64	0.15	0.05	0.2	0.10	1243.26	1715.96	427.58	1149.53	
64	0.15	0.05	0.4	0.07	439.77	264.95	7.26	59.44	
64	0.15	0.15	0.2	0.07	403.24	1499.75	164.61	515.47	
64	0.15	0.15	0.4	0.06	452.93	1035.31	63.53	241.57	
Average				0.09	1076.64	1338.47	320.72	747.15	

Table 5.12: Comparison of the running time for the RPP and IIRPP formulations on 64 nodes.

				RPP OS	IIRPP-F1 Diff%			IIRPP-F2 Diff%	
$ V $	p_a	p_d	p_r	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$	
25	0.05	0.05	0.2	1706.97	1.21	4.09	1.21	4.09	
25	0.05	0.05	0.4	2368.13	4.05	7.48	4.05	7.48	
25	0.05	0.15	0.2	1722.68	6.01	8.31	6.01	8.31	
25	0.05	0.15	0.4	2395.98	4.68	9.73	4.68	9.73	
25	0.15	0.05	0.2	1659.78	1.66	13.08	1.66	13.08	
25	0.15	0.05	0.4	2588.80	0.36	5.28	0.36	5.28	
25	0.15	0.15	0.2	1732.24	7.92	2.36	7.92	2.36	
25	0.15	0.15	0.4	2529.53	5.03	8.96	5.03	8.96	
Average				2088.01	3.86	7.41	3.86	7.41	

Table 5.13: Comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations on 25 nodes.

				RPP OS	IIRPP-F1 Diff%			IIRPP-F2 Diff%	
$ V $	p_a	p_d	p_r	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$	
36	0.05	0.05	0.2	2458.47	2.55	8.15	2.55	8.15	
36	0.05	0.05	0.4	3774.18	3.15	5.79	3.15	5.79	
36	0.05	0.15	0.2	2517.13	2.38	9.71	2.38	9.70	
36	0.05	0.15	0.4	3627.48	4.20	8.96	4.20	8.09	
36	0.15	0.05	0.2	2378.62	3.44	9.16	3.44	9.16	
36	0.15	0.05	0.4	3772.47	2.58	6.83	2.58	6.83	
36	0.15	0.15	0.2	2432.52	4.10	10.44	4.10	10.44	
36	0.15	0.15	0.4	3923.57	7.44	9.48	7.44	9.48	
Average				3110.55	3.73	8.56	3.73	8.45	

Table 5.14: Comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations on 36 nodes.

				RPP OS	IIRPP-F1 Diff%			IIRPP-F2 Diff%	
$ V $	p_a	p_d	p_r	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$	
49	0.05	0.05	0.2	3285.66	1.38	5.56	1.38	5.56	
49	0.05	0.05	0.4	5063.34	1.79	6.02	1.79	6.02	
49	0.05	0.15	0.2	3281.47	3.89	4.96	3.88	4.96	
49	0.05	0.15	0.4	5124.26	2.84	7.83	2.84	7.83	
49	0.15	0.05	0.2	3495.18	4.06	8.45	4.06	8.19	
49	0.15	0.05	0.4	5039.98	5.19	10.17	5.19	10.17	
49	0.15	0.15	0.2	3604.03	3.52	6.35	3.37	6.35	
49	0.15	0.15	0.4	5067.68	6.43	12.75	6.43	12.75	
Average				4245.20	3.64	7.76	3.62	7.73	

Table 5.15: Comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations on 49 nodes.

				RPP OS	IIRPP-F1 Diff%			IIRPP-F2 Diff%	
$ V $	p_a	p_d	p_r	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$	
64	0.05	0.05	0.2	4579.26	15.82	8.46	1.77	10.26	
64	0.05	0.05	0.4	6820.47	3.90	6.93	2.88	5.67	
64	0.05	0.15	0.2	4526.89	5.87	10.04	3.46	7.95	
64	0.05	0.15	0.4	6499.43	5.25	9.14	4.85	9.11	
64	0.15	0.05	0.2	4773.34	5.08	9.88	4.67	8.67	
64	0.15	0.05	0.4	6888.24	4.71	8.18	4.67	8.18	
64	0.15	0.15	0.2	4527.43	4.73	12.08	4.71	11.93	
64	0.15	0.15	0.4	6948.28	2.97	6.64	2.97	6.28	
Average				5695.42	6.04	8.92	3.75	8.51	

Table 5.16: Comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations on 64 nodes.

RPP-H												IIRPP-F1-H				IIRPP-F2-H			
$ V $	p_a	p_d	p_r	p_i	BFS	Time	Diff%	NBS	Time	Diff%	NBS	Time	Diff%	NBS	Time	Diff%	NBS		
25	0.05	0.05	0.2	0.2	1727.56	0.06	5.27	4	0.04	30.26	0	0.10	30.25	0	0.07	0.07	0		
25	0.05	0.05	0.2	0.4	1776.76	0.07	8.96	4	0.04	21.77	0	0.14	21.96	0	0.07	0.07	0		
25	0.05	0.05	0.4	0.2	2463.96	0.07	4.06	3	0.04	14.10	2	0.12	15.70	2	0.08	0.08	2		
25	0.05	0.05	0.4	0.4	2545.19	0.08	11.85	0	0.04	14.49	1	0.24	11.55	1	0.08	0.08	1		
25	0.05	0.15	0.2	0.2	1826.13	0.08	3.51	4	0.03	20.10	3	0.08	20.04	3	0.05	0.05	3		
25	0.05	0.15	0.2	0.4	1865.83	0.05	10.25	4	0.04	27.27	2	0.12	25.25	2	0.05	0.05	2		
25	0.05	0.15	0.4	0.2	2508.12	0.05	7.58	3	0.03	16.25	2	0.10	16.14	2	0.07	0.07	2		
25	0.05	0.15	0.4	0.4	2629.07	0.06	14.15	0	0.03	20.74	1	0.24	18.16	1	0.08	0.08	1		
25	0.15	0.05	0.2	0.2	1687.28	0.05	6.84	4	0.03	15.14	4	0.08	16.70	4	0.05	0.05	3		
25	0.15	0.05	0.2	0.4	1876.81	0.04	7.94	4	0.03	17.44	1	0.11	13.07	1	0.06	0.06	0		
25	0.15	0.05	0.4	0.2	2598.18	0.04	5.61	3	0.04	6.66	2	0.09	8.64	2	0.07	0.07	2		
25	0.15	0.05	0.4	0.4	2725.48	0.05	18.97	1	0.04	10.25	4	0.20	12.11	4	0.08	0.08	4		
25	0.15	0.15	0.2	0.2	1869.50	0.04	4.80	4	0.03	15.66	3	0.07	13.70	3	0.05	0.05	3		
25	0.15	0.15	0.2	0.4	1773.12	0.04	7.75	3	0.03	13.76	4	0.09	16.06	4	0.04	0.04	3		
25	0.15	0.15	0.4	0.2	2656.84	0.05	3.22	4	0.03	15.88	1	0.13	14.81	1	0.08	0.08	1		
25	0.15	0.15	0.4	0.4	2756.21	0.11	11.98	2	0.03	9.42	2	0.21	8.63	2	0.07	0.07	2		
Average					2205.38	0.06	8.30	2.94	0.03	16.82	2.00	0.13	16.42	2.00	0.07	16.42	1.81		

Table 5.17: Comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations on 25 nodes.

RPP-H												IIRPP-F1-H			IIRPP-F2-H		
$ V $	p_a	p_d	p_r	p_i	BFS	Time	Diff%	Time	NBS	Diff%	Time	NBS	Diff%	Time	NBS		
36	0.05	0.05	0.2	0.2	2521.05	0.85	9.86	0.07	2	26.06	0.17	1	26.92	0.11	1		
36	0.05	0.05	0.2	0.4	2658.84	0.78	15.12	0.07	3	25.17	0.33	0	25.65	0.12	0		
36	0.05	0.05	0.4	0.2	3893.12	0.16	10.33	0.08	0	12.48	0.35	0	13.00	0.16	0		
36	0.05	0.05	0.4	0.4	3992.67	0.80	12.35	0.08	0	9.44	0.98	0	7.95	0.22	1		
36	0.05	0.15	0.2	0.2	2576.93	0.95	11.88	0.07	1	42.17	0.23	0	40.69	0.14	0		
36	0.05	0.15	0.2	0.4	2761.33	16.83	9.39	0.07	0	23.06	0.31	0	21.93	0.10	0		
36	0.05	0.15	0.4	0.2	3779.81	0.15	8.38	0.06	1	13.85	0.29	2	14.96	0.16	1		
36	0.05	0.15	0.4	0.4	3920.81	0.79	11.00	0.06	0	18.49	0.83	0	17.93	0.22	0		
36	0.15	0.05	0.2	0.2	2460.46	0.19	5.22	0.06	5	23.60	0.17	0	23.59	0.09	0		
36	0.15	0.05	0.2	0.4	2596.55	0.86	16.57	0.05	1	17.33	0.27	1	17.95	0.10	1		
36	0.15	0.05	0.4	0.2	3869.84	0.51	6.04	0.06	3	15.71	0.30	0	14.68	0.14	1		
36	0.15	0.05	0.4	0.4	4030.12	1.26	11.61	0.06	0	15.01	0.75	1	14.03	0.16	1		
36	0.15	0.15	0.2	0.2	2532.25	0.12	4.52	0.05	3	21.27	0.15	1	26.09	0.08	0		
36	0.15	0.15	0.2	0.4	2686.45	0.12	9.19	0.06	2	33.93	0.29	0	33.91	0.13	0		
36	0.15	0.15	0.4	0.2	4215.41	0.11	5.70	0.05	0	14.52	0.32	1	12.48	0.14	1		
36	0.15	0.15	0.4	0.4	4295.48	0.51	9.75	0.05	2	13.77	0.67	1	14.35	0.18	1		
Average					3299.45	1.56	9.81	0.06	1.44	20.37	0.40	0.50	20.38	0.14	0.50		

Table 5.18: Comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations on 36 nodes.

												RPP-H			IIRPP-F1-H			IIRPP-F2-H		
$ V $	p_a	p_d	p_r	p_i	BFS	Time	Diff%	Time	NBS	Diff%	Time	NBS	Diff%	Time	NBS					
49	0.05	0.05	0.2	0.2	3331.04	2.49	8.21	0.12	2	34.73	0.60	0	34.60	0.25	0					
49	0.05	0.05	0.2	0.4	3468.47	81.83	14.83	0.12	0	27.87	0.85	0	26.78	0.24	0					
49	0.05	0.05	0.4	0.2	5154.11	4.44	5.24	0.11	1	19.44	1.08	0	20.91	0.44	0					
49	0.05	0.05	0.4	0.4	5368.08	165.09	18.08	0.10	0	18.04	2.51	0	17.16	0.51	0					
49	0.05	0.15	0.2	0.2	3408.81	91.92	9.07	0.10	1	32.36	0.35	0	29.79	0.21	0					
49	0.05	0.15	0.2	0.4	3560.67	566.82	14.16	0.10	2	34.33	0.77	0	31.51	0.24	0					
49	0.05	0.15	0.4	0.2	5269.80	83.14	8.78	0.10	0	20.23	0.96	0	20.97	0.49	0					
49	0.05	0.15	0.4	0.4	5525.44	133.39	19.31	0.11	0	19.63	2.24	0	18.95	0.46	0					
49	0.15	0.05	0.2	0.2	3637.25	6.88	7.13	0.10	1	32.23	0.51	0	31.75	0.27	0					
49	0.15	0.05	0.2	0.4	3781.32	4.42	14.93	0.10	1	25.03	1.02	0	23.56	0.26	1					
49	0.15	0.05	0.4	0.2	5301.63	0.52	5.03	0.10	2	9.85	0.74	2	10.57	0.32	1					
49	0.15	0.05	0.4	0.4	5552.69	2.08	11.43	0.09	0	10.95	2.14	1	10.55	0.35	1					
49	0.15	0.15	0.2	0.2	3725.42	71.98	6.30	0.11	2	22.79	0.39	0	22.35	0.19	0					
49	0.15	0.15	0.2	0.4	3833.02	6.95	11.71	0.11	0	21.15	0.80	0	21.02	0.20	0					
49	0.15	0.15	0.4	0.2	5393.60	0.21	8.83	0.08	0	11.25	0.67	1	9.40	0.25	1					
49	0.15	0.15	0.4	0.4	5713.95	0.69	15.24	0.09	0	15.51	1.93	1	12.97	0.33	0					
Average					4501.58	76.43	11.14	0.10	0.75	22.21	1.10	0.31	21.43	0.31	0.25					

Table 5.19: Comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations on 49 nodes.

RPP-H												IIRPP-F1-H				IIRPP-F2-H			
$ V $	p_a	p_d	p_r	p_i	BFS	Time	Diff%	Time	NBS	Diff%	Time	NBS	Diff%	Time	NBS				
64	0.05	0.05	0.2	0.2	4677.95	1086.42	5.48	0.22	1	30.12	1.22	0	30.58	0.57	0				
64	0.05	0.05	0.2	0.4	4966.62	1059.66	13.32	0.20	0	23.45	2.61	0	23.54	0.59	0				
64	0.05	0.05	0.4	0.2	7017.13	81.13	11.22	0.23	0	18.72	2.46	0	18.40	0.92	0				
64	0.05	0.05	0.4	0.4	7206.99	1183.42	16.25	0.25	0	14.11	7.22	0	17.04	1.30	0				
64	0.05	0.15	0.2	0.2	4683.73	1016.27	6.16	0.22	3	27.80	0.94	0	27.34	0.41	0				
64	0.05	0.15	0.2	0.4	4871.86	1435.14	13.97	0.21	1	20.94	1.85	2	23.73	0.45	1				
64	0.05	0.15	0.4	0.2	6798.94	381.67	7.96	0.20	0	23.19	2.11	0	22.63	0.68	0				
64	0.05	0.15	0.4	0.4	7091.78	711.80	14.22	0.20	0	15.53	4.87	1	15.10	0.76	0				
64	0.15	0.05	0.2	0.2	4996.10	426.70	9.67	0.23	0	20.44	1.08	0	20.85	0.42	0				
64	0.15	0.05	0.2	0.4	5187.06	1149.29	12.78	0.22	0	24.60	2.65	0	25.48	0.56	0				
64	0.15	0.05	0.4	0.2	7210.03	78.18	7.50	0.19	0	14.30	2.31	1	13.96	0.77	1				
64	0.15	0.05	0.4	0.4	7451.56	61.45	15.97	0.19	0	12.86	6.74	0	11.33	0.76	1				
64	0.15	0.15	0.2	0.2	4740.75	164.61	4.99	0.17	2	32.17	0.91	0	31.73	0.48	0				
64	0.15	0.15	0.2	0.4	5067.47	874.15	11.54	0.18	1	25.28	1.79	0	24.38	0.49	0				
64	0.15	0.15	0.4	0.2	7253.00	479.97	7.01	0.16	1	17.87	1.97	0	17.13	0.66	0				
64	0.15	0.15	0.4	0.4	8558.74	578.24	6.13	0.17	1	12.33	5.25	1	11.20	0.87	1				
Average					6111.23	673.01	10.26	0.20	0.63	20.86	2.87	0.31	20.90	0.67	0.25				

Table 5.20: Comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations on 64 nodes.

												IIRPP-F1						IIRPP-F2					
$ V $	$ E $	$ A $	$ V^1 $	$ E^1 $	$ A^1 $	NOS	NFS	$ V^2 $	$ E^2 $	$ A^2 $	NOS	NFS											
25	33	4	45.33	22.54	63.01	10.00	0.00	27.37	31.87	8.63	10.00	0.00											
36	49.25	5.75	67.63	33.33	97.89	9.88	0.13	40.26	47.21	14.09	10.00	0.00											
49	68.75	7.75	94.85	31.17	142.33	9.38	0.56	55.21	65.83	19.80	9.94	0.06											
64	91.75	10	126.88	35.21	194.76	7.00	2.56	72.64	87.64	26.86	8.88	1.06											

Table 5.21: Summary of the comparison of the transformed graphs G^1 and G^2 with the original graph G .

V	IIRPP-F1 Gap%		IIRPP-F2 Gap%	
	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$
25	0.00	0.00	0.00	0.00
36	0.00	0.26	0.00	0.00
49	0.25	0.27	0.00	0.00
64	2.45	1.72	0.50	0.78

Table 5.22: Summary of the comparison of the percentage optimality gap between the best feasible solution and the best lower bound for the IIRPP formulations.

V	RPP Time	IIRPP-F1 Time		IIRPP-F2 Time	
	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$
25	0.02	0.10	0.28	0.06	0.07
36	0.04	1.37	107.28	0.36	2.75
49	0.05	261.90	286.15	21.17	56.82
64	0.09	1076.64	1338.47	320.72	747.15

Table 5.23: Summary of the comparison of the running time for the RPP and IIRPP formulations.

V	RPP OS	IIRPP-F1 Diff%		IIRPP-F2 Diff%	
	$p_i = 0$	$p_i = 0.2$	$p_i = 0.4$	$p_i = 0.2$	$p_i = 0.4$
25	2088.01	3.86	7.41	3.86	7.41
36	3110.55	3.73	8.56	3.73	8.45
49	4245.20	3.64	7.76	3.62	7.73
64	5695.42	6.04	8.92	3.75	8.51

Table 5.24: Summary of the comparison of the percentage gap between the RPP optimal solution and the best feasible solutions from the IIRPP formulations.

V	RPP-H				IIRPP-F1-H				IIRPP-F2-H				
	BFS	Time	Diff%	NBS	Time	Diff%	NBS	Time	Diff%	NBS	Time	Diff%	NBS
25	2205.38	0.06	8.30	2.94	0.03	16.82	2.00	0.13	16.42	2.00	0.07	16.42	1.81
36	3299.45	1.56	9.81	1.44	0.06	20.37	0.50	0.40	20.38	0.50	0.14	20.38	0.50
45	4501.58	76.43	11.14	0.75	0.10	22.21	0.31	1.10	21.43	0.31	0.31	21.43	0.25
64	6111.23	673.01	10.26	0.63	0.20	20.86	0.31	2.87	20.90	0.31	0.67	20.90	0.25

Table 5.25: Summary of the comparison of the RPP based and IIRPP based heuristics with the IIRPP formulations.

Chapter 6: Concluding Remarks

In this dissertation, we made an effort to bridge the gap between academic research and its practical applicability in the field of logistics. Each chapter addressed a different real-world logistics problem. We used several decision-making techniques, including data-driven optimization and statistical modeling, to produce practical and easy-to-implement solutions for these real-world problems. This should help industry practitioners improve their decision making.

In Chapter 2, we used an iterative methodology to generate robust vehicle routes to read uncertain RFID meters from a distance. Using a large, real-world data set, we demonstrated that it is practically possible to model an inherently stochastic problem with a continuous source of incoming data using deterministic optimization techniques while updating the unknown variables before every decision point. This avoids using an intractable stochastic optimization model and brings together mathematical programming models and statistical modeling. We demonstrated that the choice of statistical model to update the meter reading probabilities is an important consideration. We showed that Bayesian updating works in a practical setup and that it avoids the drawbacks of regression. Furthermore, we developed a hierarchical Bayesian updating model specific to the meter reading

problem to take into account the heterogeneity of the signal transmission behavior of individual meters. We hope to extend this work using Bayesian decision theory and route optimization to help utility companies maximize their information gain.

In Chapter 3, we showed that route balance is very important, and it directly affects the total operating and delivery costs incurred in the CVRP. Under random traffic conditions, route times increase from the original solution produced by the routing algorithm using an objective that minimizes the total route time for a fleet of vehicles. When starting with routes that are already imbalanced, longer routes will take even more time to complete in the presence of heavy traffic. Routes that are not balanced directly and substantially affect the total cost because longer routes lead to a higher chance of a driver working more than the regular work hours. Thus, when routes are not balanced, a delivery company may pay more overtime wages to its drivers. In a practical setting, we can apply a simple randomized routing algorithm repeatedly to obtain many solutions. We can then select a solution with a small total route length and a small route length standard deviation to achieve low operating and delivery costs. It might be helpful for delivery companies to use quantifiable metrics and determine the types of instances that would have a larger impact of reducing route variability on total cost.

In Chapter 4, we showed that route lengths can be estimated using regression models for a hard-to-solve routing problem such as the CETSP. Route length estimation is useful for scenarios where it is important to quickly estimate the route length when decisions need to be made. We estimated the route length for the Steiner zone variable neighborhood search (SZVNS) heuristic on CETSP instances

with node locations generated randomly, and all customers having the same radius for the service regions. We established the performance of regression models using different quantitative and qualitative statistical measures. The independent variables in the regression models captured the geometric properties of an instance, the spread of the customer service regions, and the number of Steiner zones. The variable for the number of Steiner zones captured the feature of an instance exploited by the SZVNS heuristic. Similar regression models could be built for estimating the route lengths using different heuristics by having variables that capture the specific feature of a heuristic. It would also be important to have fast route length prediction models for CETSP instances with node locations that are not generated randomly.

In Chapter 5, we introduced the IIRPP, a variant of the RPP involving turns, to help local governments in road inspections. Along with street segments, intersections need to be inspected for proper road quality management. It is important to make at least one straight or left turn at each intersection. We formulated two integer programs, IIRPP-F1 and IIRPP-F2, based on two different graph transformations to solve the IIRPP optimally. The computational experiments showed that IIRPP-F2 was faster and able to produce better quality solutions within a reasonable amount of time compared to IIRPP-F1 because of the significantly smaller size of its transformed graph. We developed three heuristics for the IIRPP. RPP-H was the best performing heuristic compared to IIRPP-F1-H and IIRPP-F2-H. RPP-H used a simple modification to the optimal RPP routes and produced good quality solutions within a very short amount of time. In the future, we hope to develop branch-and-cut algorithms to solve larger instances optimally within a reasonable

amount of time, and smarter heuristics to reduce the gap between heuristic solutions and optimal IIRPP solutions.

From the real-world logistics problems studied in this dissertation, we learned that it is very important to look at a problem from the perspective of a decision maker trying to solve it in practice. The methodologies to solve practical problems should be designed with applicability in the mind. It is also important to use large data sources to generate novel insights and to build data-driven, decision-making tools that improve business decision making in real-world settings.

Bibliography

- Albert JH, Chib S (1993) Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association* 88(422):669–679.
- Ávila T, Corberán Á, Plana I, Sanchis JM (2016) A new branch-and-cut algorithm for the generalized directed rural postman problem. *Transportation Science* 50(2):750–761.
- Beardwood J, Halton JH, Hammersley JM (1959) The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society* 55(4):299–327.
- Behdani B, Smith JC (2014) An integer-programming-based approach to the close-enough traveling salesman problem. *INFORMS Journal on Computing* 26(3):415–432.
- Benavent E, Soler D (1999) The directed rural postman problem with turn penalties. *Transportation Science* 33(4):408–418.
- Bodin L, Levy L (1991) The arc partitioning problem. *European Journal of Operational Research* 53:393–401.
- Capacitated Vehicle Routing Problem Library (2014). <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>.
- Carrabs F, Cerrone C, Cerulli R, Gaudioso M (2017) A novel discretization scheme for the

- close-enough traveling salesman problem. *Computers & Operations Research* 78:163–171.
- Cavdar B, Sokol J (2015) A distribution-free TSP tour length estimation model for random graphs. *European Journal on Operational Research* 243(2):588–598.
- Cerrone C, Dussault B, Wang X, Golden B, Wasil E (2019) A two-stage solution approach for the directed rural postman problem with turn penalties. *European Journal of Operational Research* 272:754–765.
- Chien (1992) Operational estimators for the length of a traveling salesman tour. *Computers & Operations Research* 19(6):469–478.
- Christofides N, Eilon S (1969a) An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society* 20(3):309–318.
- Christofides N, Eilon S (1969b) Expected distances in distribution problems. *Journal of the Operational Research Society* 20(4):437–443.
- Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12(4):568–581.
- Clossey J, Laporte G, Soriano P (2001) Solving arc routing problems with turn penalties. *Journal of the Operational Research Society* 52(4):433–439.
- Colorado Department of Regulatory Agencies (2018) Public Utilities Commission. Accessed May 4, 2018, <https://www.pueblo.us/DocumentCenter/View/6596/Your-Rights-as-an-Electric-or-Natural-Gas-Utility-Customer>.

- Corberán Á, Plana I, Sanchis JM (2014) The rural postman problem on directed, mixed, and windy graphs. *Arc Routing: Problems, Methods, and Applications* (SIAM) 101–127.
- Coutinho WP, Subramanian A, do Nascimento RQ, Pessoa AA (2016) A branch-and-bound algorithm for the close-enough traveling salesman problem. *INFORMS Journal on Computing* 28(4):752–765.
- Defryn C, Sörensen K (2017) A fast two-level variable neighborhood search for the clustered vehicle routing problem. *Computers & Operations Research* 83:78–94.
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1):269–271.
- Domencich T, McFadden DL (1975) *Urban Travel Demand: A Behavioral Analysis* (Elsevier).
- Dong J, Yang N, Chen M (2007) Heuristic approaches for a TSP variant: The automatic meter reading shortest tour problem. *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies* (Springer Verlag) 145–163.
- Dumitrescu A, Mitchell J (2003) Approximation algorithms for TSP with neighborhoods in the plane. *Journal of Algorithms* 48:135–159.
- Eglese R, Golden B, Wasil E (2014) Route optimization for meter reading and salt spreading. *Arc Routing: Problems, Methods, and Applications* (SIAM) 300–320.
- Frederickson GN, Hecht MS, Kim CE (1978) Approximation algorithms for some routing

- problems. *SIAM Journal on Computing* 7(2):178–193.
- Golden B, Alt F (1979) Interval estimation of a global optimum for large combinatorial problems. *Naval Research Logistics Quarterly* 26(1):69–77.
- Golden B, Wasil E, Kelly J, Chao IM (1998) The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results. *Fleet Management and Logistics* (Springer) 33–56.
- Groër C, Golden B, Wasil E (2009) The balanced billing cycle vehicle routing problem. *Networks* 54(4):243–254.
- Gulczynski D, Heath J, Price C (2006) The close enough traveling salesman problem: A discussion of several heuristics. *Perspectives in Operations Research: Papers in Honor of Saul Gass' 80th Birthday* (Springer Verlag) 271–283.
- Hà MH (2012) Modélisation et résolution de problèmes généralisés de tournées de véhicules, Ph.D. dissertation, Automatique, Ecole des Mines de Nantes, France.
- Hà MH, Bostel N, Langevin A, Rousseau L-M (2014) Solving the close-enough arc routing problem. *Networks* 63(1):107–118.
- Hansen P, Mladenović N, Todosijević R, Hanafi S (2017) Variable neighborhood search: Basics and variants. *EURO Journal on Computational Optimization* 5(3):423–454.
- Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57:97–109.
- Hindle A, Worthington D (2004) Models to estimate average route lengths in different

- geographical environments. *Journal of the Operational Research Society* 55(6):662–666.
- Illinois Administrative Code (2018) Title 83: Public Utilities, Section 280.90: Estimated Bills. Accessed May 4, 2018, <ftp://www.ilga.gov/JCAR/AdminCode/083/083002800F00900R.html>.
- Irving, Texas - Code of Ordinances (2018) Chapter 31: Public Utilities, Article II: Natural Gas, Section 31-11: Estimated bills prohibited. Accessed May 4, 2018, https://library.municode.com/tx/Irving/codes/code_of_ordinances?nodeId=PTIITHCO_CH31PUUT_ARTIINAGA_S31-11ESBIPRCOEXTESE.
- Kara I, Guden H, Koc ON (2012) New formulations for the generalized traveling salesman problem. *Proceedings of the 6th international conference on Applied Mathematics, Simulation, Modelling* 60–65.
- Kenyon AS, Morton DP (2003) Stochastic vehicle routing with random travel times. *Transportation Science* 37(1):69–82.
- Kwon O, Golden B, Wasil E (1995) Estimating the length of the optimal TSP tour: An empirical study using regression and neural networks. *Computers & Operations Research* 22(10):1039–1046.
- Laporte G, Louveaux F, Mercure H (1992) The vehicle routing problem with stochastic travel times. *Transportation Science* 26(3):161–170.
- Levy L (2018) Personal communication. RouteSmart Technologies, Inc.

- Levy L, Sniezek J, Cox B (2002) Utility meter route management and optimization using GIS, Tech. report, Electric and Gas Utilities User Group, Coeur d'Alene, Idaho.
- Louviere JJ, Hensher DA, Swait JD (2000) *Stated choice methods: Analysis and applications* (Cambridge University Press).
- Lum O, Golden B, Wasil E (2018) An open-source desktop application for generating arc-routing benchmark instances. *INFORMS Journal on Computing* 30(2):361-370.
- Marin J-M, Robert CP (2014) *Bayesian Essentials with R* (Springer).
- Mennell WK (2009) Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem, Ph.D. dissertation, Decision, Operations & Information Technologies, University of Maryland, College Park, USA.
- Mennell WK, Golden B, Wasil E (2011) A Steiner-zone heuristic for solving the close-enough traveling salesman problem. *Operations Research, Computing, and Homeland Defense* (INFORMS) 162-183.
- Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21:1087-1091.
- Michigan Department of Labor and Economic Growth (2018) Public Service Commission, Consumer Standards and Billing Practices, Part 4. Accessed May 4, 2018, <https://www.consumersenergy.com/~media/CE/Documents/mpsc-billing-rules.ashx?la=en>.

- Nicola D, Vetschera R, Dragomir A (2019) Total distance approximations for routing solutions. *Computers & Operations Research* 102:67–74.
- Press SJ (2003) *Subjective and Objective Bayesian Statistics: Principles, Models, and Applications* (John Wiley & Sons).
- Renaud A, Absi N, Feillet D (2017) The stochastic close-enough arc routing problem. *Networks* 69(2):205–221.
- Rossi PE, Allenby GM, McCulloch R (2005) *Bayesian Statistics and Marketing* (John Wiley & Sons).
- Rostami B, Desaulniers G, Errico F, Lodi A (2017) The vehicle routing problem with stochastic and correlated travel times. *Data Science for Real-Time Decision-Making* (Canada Excellence Research Chair) 1–51.
- Shuttleworth R, Golden B, Smith S, Wasil E (2008) Advances in meter reading: Heuristic solution of the close enough traveling salesman problem over a street network. *The Vehicle Routing Problem: Latest Advances and New Challenges* (Springer Verlag) 487–501.
- Silberholz J, Golden B (2007) The generalized traveling salesman problem: A new genetic algorithm approach. *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies* (Springer Verlag) 165–181.
- Steiniger S, Hunter AJS (2013) The 2012 free and open source GIS software map - A guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems* 39:136–150.

- Stern D, Dror M (1979) Routing electric meter readers. *Computers & Operations Research* 6:209–223.
- Wang X, Golden B, Wasil E (2019) A Steiner zone variable neighborhood search heuristic for the close-enough traveling salesman problem. *Computers & Operations Research* 101:200–219.
- Wassan N, Wassan N, Nagy G, Salhi S (2017) The multiple trip vehicle routing problem with backhauls: Formulation and a two-level variable neighbourhood search. *Computers & Operations Research* 78:454–467.
- Yang Z, Xiao M-Q, Ge Y-W, Feng D-L, Zhang L, Song H-F, Tang X-L (2018) A double-loop hybrid algorithm for the traveling salesman problem with arbitrary neighbourhoods. *European Journal on Operational Research* 265(1):65–80.
- Yuan B, Orłowska M, Sadiq S (2007) On the optimal robot routing problem in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering* 19(9):1252–1261.