

2017

Robust Vehicle Routing in Disaster Relief and Ride-Sharing: Models and Algorithms

Yinglei Li

Binghamton University--SUNY, yli90@binghamton.edu

Follow this and additional works at: https://orb.binghamton.edu/dissertation_and_theses



Part of the [Industrial Engineering Commons](#)

Recommended Citation

Li, Yinglei, "Robust Vehicle Routing in Disaster Relief and Ride-Sharing: Models and Algorithms" (2017). *Graduate Dissertations and Theses*. 49.

https://orb.binghamton.edu/dissertation_and_theses/49

This Dissertation is brought to you for free and open access by the Dissertations, Theses and Capstones at The Open Repository @ Binghamton (The ORB). It has been accepted for inclusion in Graduate Dissertations and Theses by an authorized administrator of The Open Repository @ Binghamton (The ORB). For more information, please contact ORB@binghamton.edu.

ROBUST VEHICLE ROUTING IN DISASTER RELIEF AND RIDE-SHARING:
MODELS AND ALGORITHMS

BY

YINGLEI LI

BS, South China Agricultural University, 2012
MS, State University of New York at Binghamton, 2014

DISSERTATION

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Industrial and Systems Engineering
in the Graduate School of
Binghamton University
State University of New York
2017

© Copyright by Yinglei Li 2017

All Rights Reserved

Accepted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Industrial and Systems Engineering
in the Graduate School of
Binghamton University
State University of New York
2017

December 8th, 2017

Sung Hoon Chung, Chair and Faculty Advisor
Department of Systems Science and Industrial Engineering, Binghamton University

Nagen Nagarur, Member
Department of Systems Science and Industrial Engineering, Binghamton University

Sarah Lam, Member
Department of Systems Science and Industrial Engineering, Binghamton University

Daehan Won, Member
Department of Systems Science and Industrial Engineering, Binghamton University

Weiyi Meng, Outside Examiner
Department of Computer Science, Binghamton University

Abstract

In this dissertation, the variants of vehicle routing problems (VRPs) are specifically considered in two applications: disaster relief routing and ride-sharing. In disaster relief operations, VRPs are important, especially in the immediate response phase, as vehicles are an essential part of the supply chain for delivering critical supplies. This dissertation addresses the capacitated vehicle routing problem (CVRP) and the split delivery vehicle routing problem (SDVRP) with uncertain travel times and demands when planning vehicle routes for delivering critical supplies to the affected population in need after a disaster. A robust optimization approach is used for the CVRP and the SDVRP considering the five objective functions: minimization of the total number of vehicles deployed (minV), the total travel time/travel cost (minT), the summation of arrival times (minS), the summation of demand-weighted arrival times (minD), and the latest arrival time (minL), out of which we claim that minS, minD, and minL are critical for deliveries to be fast and fair for relief efforts, while minV and minT are common cost-based objective functions in the traditional VRP. In ride-sharing problem, the participants' information is provided in a short notice, for which driver-rider matching and associated routes need to be decided quickly. The uncertain travel time is considered explicitly when matching and route decisions are made, and a robust optimization approach is proposed to handle it properly. To achieve computational tractability, a new two-stage heuristic method that combines the extended insertion algorithm and tabu search (TS) is proposed to solve the models for large-scale problems. In addition, a new hybrid algorithm named scoring tabu search with variable neighborhood (STSVN) is proposed to solve the models and compared with TS. The solutions of the CVRP and the SDVRP are compared for different examples using five different metrics in which the results show that the latter is not only capable of accommodating the demand greater than the vehicle capacity but also is quite effective to mitigate demand and travel time uncertainty, thereby outperforms CVRP in the disaster relief routing perspective. The results of ride-sharing problem show the influence of parameters and uncertain travel time on the solutions. The performance of TS and STSVN are compared in terms of solving the models for disaster relief routing and ride-sharing problems and the results show that STSVN outperforms TS in searching the near-optimal/optimal solutions within the same CPU time.

Keywords: Robust Optimization, Vehicle Routing Problems, Tabu Search, Insertion Algorithm, Scoring Tabu Search with Variable Neighborhood

Contents

List of Tables	vii
List of Figures	ix
1 Introduction	1
1.1 Disaster Relief Routing	5
1.2 Ride-sharing	7
1.3 Algorithms	9
2 Literature Review	12
2.1 Vehicle Routing Problem with Uncertain Travel Time and Demand	13
2.2 Disaster Relief Routing Problem	15
2.3 Ride-sharing Problem	18
2.4 Tabu Search and Variable Neighborhood Search in VRP	21
3 Models	28
3.1 Models in Disaster Relief Routing	28
3.1.1 Deterministic CVRP	28
3.1.2 Deterministic SDVRP	31
3.1.3 Robust Counterparts of the CVRP	33
3.1.4 Robust Counterparts of the SDVRP	37
3.2 Models in Ride-sharing	41
3.2.1 Deterministic Model for Dynamic Ride-sharing	41
3.2.2 Robust Counterpart	45
4 Algorithms	49
4.1 Insertion Algorithms	50
4.1.1 Insertion Algorithms for Disaster Relief Routing	50
4.1.2 Insertion Algorithm for Ride-sharing	54
4.2 Tabu Search	55
4.2.1 Tabu Search for Disaster Relief Routing	56
4.2.2 Tabu Search for Ride-sharing	60
4.3 Algorithms for Solving Robust Counterparts	61
4.4 Scoring Tabu Search with Variable Neighborhood	64
4.4.1 Main Steps of STSVN	65
4.4.2 STSVN in Ride-sharing	67

5	Results	73
5.1	Results for Disaster Relief Routing	73
5.1.1	Simple Examples	73
5.1.2	Large Examples	76
5.2	Results for Ride-sharing	88
5.3	Results for Comparison of TS and STSVN	95
6	Conclusion and Future Work	107
	Bibliography	111

List of Tables

1.1	Key Features of Problems	2
3.1	Notation for Disaster Relief Routing	29
3.2	Notation for Ride-sharing Models	48
5.1	Results of Example 1 and Example 2	75
5.2	Results of Example 3	76
5.3	CVRP and SDVRP Results: $A - n32 - k5$	77
5.4	CVRP and SDVRP Results: $A - n38 - k5$	78
5.5	CVRP and SDVRP Results: $A - n44 - k6$	79
5.6	CVRP and SDVRP Results: $A - n55 - k9$	80
5.7	CVRP and SDVRP Results: $A - n69 - k9$	81
5.8	CVRP and SDVRP Results: $E - n76 - k14$	82
5.9	CVRP and SDVRP Results: $A - n80 - k10$	83
5.10	CVRP and SDVRP Results: $E - n101 - k8$	84
5.11	CVRP and SDVRP Results: $E - n101 - k14$	85
5.12	Comparison of CVRP and SDVRP Models	86
5.13	Results of SD2 in CVRP and SDVRP	87
5.14	Number of Drivers and Customer Requests in Examples	90
5.15	Results for Example $E - n22 - k4$ with Different Settings	92
5.16	Results for Example $E - n22 - k4$ with Different Γ_k	94
5.17	Results for Example $A - n32 - k5$ with Different Γ_k	95
5.18	Results for Example $A - n36 - k5$ with Different Γ_k	95
5.19	Results for Example $A - n44 - k6$ with Different Γ_k	95
5.20	Results for Example $A - n48 - k7$ with Different Γ_k	95
5.21	Results for Example $A - n54 - k7$ with Different Γ_k	96
5.22	Results for Example $A - n60 - k9$ with Different Γ_k	96
5.23	Results for Example $A - n64 - k9$ with Different Γ_k	96
5.24	Results for Example $E - n76 - k7$ with Different Γ_k	96
5.25	Results for Example $A - n80 - k10$ with Different Γ_k	97
5.26	Comparison of TS and STSVN for CVRP and SDVRP Models, Part 1	98
5.27	Comparison of TS and STSVN for CVRP and SDVRP Models, Part 2	99
5.28	Comparison of TS and STSVN for CVRP and SDVRP Models, Part 3	100
5.29	Comparison of TS and STSVN for CVRP and SDVRP Models, Part 4	101
5.30	Comparison of TS and STSVN for Ride-sharing Models, $E - n22 - k4$	103
5.31	Comparison of TS and STSVN for Ride-sharing Models, $A - n32 - k5$	103
5.32	Comparison of TS and STSVN for Ride-sharing Models, $A - n36 - k5$	103
5.33	Comparison of TS and STSVN for Ride-sharing Models, $A - n44 - k6$	104

5.34	Comparison of TS and STSVN for Ride-sharing Models, $A - n48 - k7$. . .	104
5.35	Comparison of TS and STSVN for Ride-sharing Models, $A - n54 - k7$. . .	104
5.36	Comparison of TS and STSVN for Ride-sharing Models, $A - n60 - k9$. . .	104
5.37	Comparison of TS and STSVN for Ride-sharing Models, $A - n64 - k9$. . .	105
5.38	Comparison of TS and STSVN for Ride-sharing Models, $E - n76 - k7$. . .	105
5.39	Comparison of TS and STSVN for Ride-sharing Models, $A - n80 - k10$. .	105

List of Figures

5.1	Simple Examples	75
5.2	Robust Model Results of minT with Γ_T	78
5.3	Robust Model Results of minD with Γ_Q	79
5.4	Scenario 1 Results for Various Γ_T	87
5.5	Scenario 2 Results for Various Γ_T	88
5.6	Scenario 2 Results for Various Γ_Q	89
5.7	Results for Example $E - n22 - k4$ with Different Settings	93
5.8	Normal Probability Plots for Disaster Relief Routing Results of TS	102
5.9	Normal Probability Plots for Disaster Relief Routing Results of STSVN	102
5.10	Normal Probability Plots for Ride-sharing Results of TS	106
5.11	Normal Probability Plots for Ride-sharing Results of STSVN	106

Chapter 1

Introduction

Vehicle routing problems (VRP) are a type of problems aim to design optimal delivery or collection routes from one or several depots to a number of geographically scattered customers, subject to some constraints (Laporte, 1992). Based on problem characteristics, there are different variants of VRPs (Eksioglu et al., 2009; Toth and Vigo, 2002b). In the capacitated vehicle routing problem (CVRP), the demands are not split, which means that each customer (node) is visited only once (Lysgaard et al., 2004; Toth and Vigo, 2002a). Different from CVRP, split delivery vehicle routing problem (SDVRP) allows split delivery and each customer (node) can be visited more than one vehicle (Ho and Haugland, 2004; Archetti et al., 2006). For the vehicle routing problem with uncertain demands and/or uncertain travel time, the demands and/or travel time are not deterministic, which means that the information is not fixed and may vary within the range (Gendreau et al., 1996; Allahviranloo et al., 2014; Braaten et al., 2017). The vehicle routing problem with time window can be considered as the extended CVRP with time window constraints, which means that each customer (node) is visited only once by exactly one vehicle within a given time interval (Potvin and Bengio, 1996; Taillard et al., 1997; Bräysy and Gendreau, 2002; Cordeau et al., 2001). Based on the number of depots, there are single-depot vehicle routing problem (Barbarosoglu and Ozgur, 1999) and multi-depot vehicle routing problem (Polacek et al., 2004; Kuo and Wang, 2012; Paraskevopoulos et al., 2008). When there are more than one type of vehicles, the problem can be considered as heterogeneous fleet vehicle routing problem (Gendreau et al., 1999; Wassan and Osman, 2002; Brandão, 2011). When the vehicles do not return to the depots after serving the last customer (node), the

problem is called open vehicle routing problem (Brandão, 2004; Fleszar et al., 2009). For more details of VRPs and its variants, see Eksioglu et al. (2009).

In this dissertation, variants of VRP are specifically considered in two applications: disaster relief routing and ride-sharing. Based on the setting of each application, the models consist of different features from the variants of VRP. In the disaster relief routing, CVRP and SDVRP are considered and compared. The deterministic models of CVRP and SDVRP are used as the base models and the robust counter-parts are proposed to consider the uncertainty. In the ride-sharing model, each customer (node) is visited at most once by exactly one vehicle within a given time interval (CVRP with time windows) and vehicles may depart from different origins and arrive at different destinations (open vehicle routing problem with multiple depots). In addition, the capacity of the vehicles in ride-sharing model may vary (heterogeneous fleet). The robust counter-part is proposed to consider the uncertainty in the ride-sharing model. The key features of the models in disaster relief routing and ride-sharing are summarized in Table 1.1 and the details are described in Chapters 1 – 3.

Table 1.1: Key Features of Problems

Feature	Disaster relief routing	Ride-sharing
VRP	CVRP, SDVRP	CVRP
Depot	Single	Multiple
Time window	No	Yes
Heterogeneous fleet	No	Yes
Uncertain travel time	Yes	Yes
Uncertain demand	Yes	No

In disaster relief operations, vehicle routing problems (VRPs) are important, especially in the immediate response phase, as vehicles are an essential part of the supply chain for delivering critical supplies. In disaster relief operations, vehicle routing problems are involved in detailed information collection, medical aid delivery, medical supply delivery, food supply delivery, etc (Luis et al., 2012). After a disaster occurs, the travel time and demand are uncertain and can be only estimated within a range. If the uncertainty is not presumed in planning a vehicle route, the actual travel time of the route may be very different from the expected travel time after the route is applied in the chaotic situation

of disaster. The optimal deterministic routes could be infeasible due to a perturbation in parameters caused by uncertainties. Therefore, it is essential to mitigate the impact of uncertainty in planning a vehicle route and before applying it. This research aims to enhance disaster relief operations by considering uncertainties explicitly before applying the routing decisions.

Ride-sharing is a transportation mode where travelers having similar itineraries share a vehicle to, in general, reduce costs. Ride-sharing has attracted researchers' attention not only because of its economical benefits, but also thanks to its positive environmental and societal impacts such as reducing air pollution, traffic congestions, etc (Ferguson, 1997; Kelly, 2007; Morency, 2007; Chan and Shaheen, 2012; Agatz et al., 2012; Furuhata et al., 2013; Pelzer et al., 2015; Nourinejad and Roorda, 2016; Stiglic et al., 2016; Alonso-Mora et al., 2017). Herein, a dynamic ride-sharing problem is considered, in which volunteer drivers (servers) and riders (clients) may have different origins and destinations. Here, the "dynamic" means that drivers and riders are not fixed among different ride-sharing arrangements: some drivers can be riders in the next ride-sharing arrangement and vice versa; and they may even opt out of ride-sharing temporarily and rejoin whenever they want. The origin and destination of a rider can be various among different ride-sharing arrangements. We assume that such information is updated continuously via electronic bulletin in social media or smart phone applications. Therefore, the information in each ride-sharing arrangement is deterministic but the information among different ride-sharing arrangements is dynamic. The riders' requests and the available drivers' information are collected and stored in a centralized system. We consider not only the one-to-one match between a driver and a single rider but also the one-to-multiple match in which a single driver may serve multiple riders during his/her trip. Given all the drivers' and riders' information in the same pool (e.g., same service area), multiple matches (each match consisting of one driver and one or more riders) are made at the same time in such a way that the overall travel cost of all trips can be minimized while the matching rate is maximized. In this setting, the goal is to increase the matching rate between drivers and riders and to provide a robust route quickly for each volunteer driver based on continuously updated information under travel time uncertainty between locations. In this research, vehicle routing problems

are used as a basis for finding routes for drivers.

The common characteristics of these two applications are: (1) The information of travel time and demand are provided in a short notice. In disaster relief routing, the travel time and demand are uncertain and can be only estimated within a range based on limited information. In ride-sharing, the demand (number of people at each location) are known and certain, but the travel time are uncertain due to traffic congestion. (2) The events are non-recurring. In disaster relief routing, every disaster may occur in different areas, so the locations of shelters are different and the information of travel time and demands are various. Therefore, most solutions of routing decisions are one-time use because of the change of information in the next event. Similarly, the information among different ride-sharing arrangements varies, so the solution for matching and routing should be provided for each ride-sharing arrangement and may not be re-usable for next arrangement. (3) The routing decisions should be made quickly. In disaster relief routing, the decisions should be made within several hours since the critical supplies should be delivered to the people as soon as possible. In ride-sharing, the decisions should be made before the earliest departure time of the participants, which is generally several minutes to several hours after they post the information of requests. While these two applications share several common characteristics, there are specific settings in each application. In the disaster relief routing, the decisions focus on routing, while the decisions in ride-sharing are for both matching and routing. Based on the settings of each application, the VRP based models are modified specifically.

The remainder of this dissertation is organized as follows. The detailed introduction of each problem is described in Section 1.1 and Section 1.2. In Chapter 2, the literature review is provided. In Chapter 3, the deterministic CVRP and SDVRP, and their robust counterparts are introduced for disaster relief routing. The deterministic model and its robust counterpart are introduced for ride-sharing. The brief introduction of proposed algorithms is shown in Section 1.3 and the details are described in Chapter 4. The results are shown in Chapter 5. Conclusion and future work follow in Chapter 6.

1.1 Disaster Relief Routing

There are significant devastating effects of natural and man-made disasters. For example, the Hurricane Katrina in August 2005, a well-known disaster, resulted in damage estimates exceeding 200 billion U.S. dollars (Burby, 2006). More recently, two severe earthquakes occurred in Nepal on April 25th and May 12th of 2015, which caused at least 8,000 deaths, 25,000 injuries, and approximately two million homeless people (Binns and Low, 2015). As of August 31, 2017, at least 39 casualties have been reported in Texas due to Hurricane Harvey, and the resulting, catastrophic floods inundated hundreds of thousands of homes, affecting more than 30,000 people. Therefore, the importance of effective management of disasters cannot be overemphasized as it is directly relevant to human life, health, and welfare.

Disaster management is typically divided into three phases: preparation, immediate response, and reconstruction (Kovács and Spens, 2007) or four phases: mitigation added after preparation (Pearce, 2003). This dissertation focuses on the immediate response phase in the context of disaster relief operations and humanitarian logistics, which takes part in the aftermath of disasters. Specifically, the relief routing problem is tackled to effectively and equitably deliver critical supplies to the affected population.

In the disaster relief, it may not be practical to assume that the vehicle capacity is always sufficient to meet all the demand from a customer location and, therefore, the location may need to be visited multiple times (Yi and Kumar, 2007), which implies split delivery. Wang et al. (2014) also state that an affected area can be served more than one time when the demand of the disaster area is greater than the capacity of the vehicle. Therefore, the split delivery vehicle routing problem (SDVRP) should play an important role in disaster relief operations to handle large demand.

The SDVRP, which was introduced in Dror and Trudeau (1989), is relatively new compared with the capacitated vehicle routing problem (CVRP). As aforementioned, the SDVRP allows a demand node to be visited by more than one vehicle while the CVRP requires that a demand node be visited exactly once. The SDVRP has attracted researchers' interest because of the potential cost savings (Dror and Trudeau, 1989; Archetti et al., 2006), and

the variants of the SDVRP and algorithms to solve them have been extensively studied in recent years. However, very few papers have tackled the SDVRP with uncertain travel time and/or demand. Dealing with uncertain demands in the SDVRP may be quite challenging. In this dissertation, the SDVRP with uncertain travel times and demands is addressed in the context of disaster relief operations, and is compared with the CVRP counterpart.

Uncertainty in travel time and demand is a critical factor in planning a vehicle route after a disaster because the optimal deterministic routes could be even infeasible due to a perturbation in parameters caused by uncertainties. Therefore, it is essential to mitigate the impact of uncertainty in planning a vehicle route and we aim to enhance disaster relief operations by considering uncertainties explicitly. To do so, robust counterparts of the CVRP and SDVRP models are proposed to consider different objectives in disaster relief operations and eventually contribute to the goals of the humanitarian logistics. In the humanitarian logistics literature, very few research addresses the robustness in the SDVRP and discuss its impact on the relief operations.

Stochastic programming and robust optimization are two major modeling approaches that address uncertainty in operations research (Bertsimas et al., 2011a; Sim, 2004). Stochastic programming has shortcomings to be used for the VRP for disaster relief operations because it requires known probability distribution functions for uncertain parameters and generally needs heavy computations (Bertsimas et al., 2011a; Sim, 2004). It is hard to know the exact information or even the probability distributions of uncertain travel times and demands due to the chaotic situation in the aftermath of disasters. In light of the above, robust optimization can be an appropriate alternative in context of disaster relief operations since it only requires the ranges of uncertain parameters instead of the exact functions, which are much more tractable.

Minimization of the total number of vehicles deployed (minV) and the total travel time/travel cost (minT) are common cost-based objective functions in the traditional VRP. Contrary to the conventional VRP objectives, the objectives of disaster relief vehicle routing should be different. Minimization of the summation of arrival times (minS), the summation of demand-weighted arrival times (minD), and the latest arrival time (minL) are critical for deliveries to be fast and fair for relief efforts. The optimal solutions of minS and minL can

provide earlier service times to customers than the ones of minV and minT (Campbell et al., 2008a). In humanitarian relief operations, routing and relief supplies allocation decisions should be quick and sufficient with a focus on equitable service to all aid recipients (Huang et al., 2012). Therefore, it is important to consider minS, minL, and minD in disaster relief routing. In this dissertation, five objectives are considered and compared: minimization of the total number of vehicles deployed (minV), the total travel time/travel cost (minT), the summation of arrival times (minS), the summation of demand-weighted arrival times (minD), and the latest arrival time (minL).

1.2 Ride-sharing

Ride-sharing has attracted researchers' attention because of its economic benefits, positive environmental and societal impacts. Despite such various beneficial impacts, the widespread adoption of ride-sharing has been limited due to its disorganized coordination, safety concerns, and lack of effective methods to encourage participation (Furuhata et al., 2013). The coordination for dynamic ride-sharing is particularly difficult because the participants' schedules vary on a daily basis and the relevant information is usually provided in a very short notice (Agatz et al., 2012).

Ride-sharing is not a new idea but it has gained a resurgence of interest over the last decade along with the recent development of information technology that may help the coordination of dynamic ride-sharing in a variety of ways including smart phone apps and social media services (Agatz et al., 2012). However, there is still a barrier for dynamic ride-sharing to overcome for its wide-adoption: the lack of computationally efficient algorithms that can provide a real-time match for drivers (servers) and riders (clients) (Agatz et al., 2012). Consequently, there has been a growing need to address the computation issues, especially for dynamic ride-sharing that requires immediate matching upon request. Unfortunately, however, the number of specific contributions has been limited perhaps due to its complex nature of mixed integer programming formulation. Therefore, the development of computationally tractable algorithms for large-scale ride-sharing problems is one of key topics in current research.

It may not be reasonable to consider fixed value of travel time due to the traffic con-

gestion. Basically, the travel time is assumed to be known within a range instead of the fixed value. In the dynamic ride-sharing problem considered in this research, one assumption is that a driver may visit several different locations to pick up riders on the way to a final destination and, therefore, the probability that the driver will be subject to uncertain travel time gets higher. The consideration of uncertain travel time plays a very important role in coordinating dynamic ride-sharing, since in most cases participants have time constraints. In particular, travel time uncertainty is greatly related to the effectiveness of driver-rider matching. According to the recent study about the role of participants' flexibility to matching (Stiglic et al., 2016), the willingness of participants to depart from the origin and arrive at the destination a little earlier or later, respectively (the flexibility of time window), is the main factor influencing the success rate of matching. In addition, the willingness of drivers and/or riders to detour to pick up additional riders (flexible maximum driving/riding time) also influence the rate of matches. Therefore, the explicit consideration of travel time uncertainty is important since it may significantly impact the time window and detour constraints, which are related to matching and the cost for ride-sharing.

Turning our attention to the implementation of dynamic ride-sharing, there are already commercial firms that provide dynamic ride-sharing service in the market via various means such as mobile phone applications. For example, Fliinc and Carma are the dynamic ride-sharing providers that offer centralized matching service. Using mobile phones and/or desktop applications, users can enter a ride offer or ride search with desired start location, destination, and time. The system will automatically find someone nearby with a similar destination and the users will receive the ride-sharing information from mobile phone/desktop applications if someone wants to share a ride as well as providing estimated cost. However, most services are somehow limited (e.g., no national coverage in the U.S., or limited number of participants) mainly due to the well-known "critical mass" issue (Stiglic et al., 2016), which we attempt to solve by means of our proposed robust optimization and heuristic algorithms that may increase the matching success rate: the robust optimization is capable of providing the near-optimal feasible solutions despite unexpected traffic congestion and the heuristic algorithms help to achieve computational tractability.

In ride-sharing problems, there are several constraints that must be considered. First,

drivers may have their own preferred time to depart from their origins. If they are flexible and willing to pick up riders, it is reasonable to assume that there exists an upper limit on the travel time that drivers are willing to spend. Likewise, riders may have their preferred departure time from their origins as well. Additionally, drivers and riders need to arrive at the destinations before their respective deadlines. Furthermore, vehicles have limited capacity (in general 4 to 5 people including the driver). Therefore, the drivers' routes need to be determined based on those constraints and some riders cannot be picked up considering the number of drivers available and the constraints mentioned above. Travel time uncertainty may have a significant impact on ride-sharing, as most constraints mentioned above involve time as a primary factor. In some unfortunate cases, a feasible route satisfying all the constraints may become infeasible if the time between pick-up locations deviates from its nominal value. To overcome such an issue, a robust optimization (RO) approach is proposed. The robust optimization has gained a surge of interest over the last twenty years and it has been used in a variety of application areas including transportation/logistics, supply chains management, finance, etc. For a survey of robust optimization, see, e.g., Ben-Tal and Nemirovski (2002a); Beyer and Sendhoff (2007); Bertsimas et al. (2011b).

In this dissertation, vehicle routing problems are used as a basis for finding routes for drivers, which is known to be an *NP-hard* problem. It is impractical to find an exact optimal route for the ride-sharing problem especially when the size of the problem is large. Indeed, computational tractability is of particular interest because we wish to provide a robust route as quickly as possible using continuously updated information. The ride-sharing platform we propose will not be viable if it takes more than a few hours to find routes for drivers. To achieve computational tractability, heuristic algorithms are proposed to solve the problems.

1.3 Algorithms

Small problems in this dissertation can be solved by using the commercial packages such as Gurobi and CPLEX. However, for large-scale problems, it is by no means practical to utilize the solvers as the VRPs are NP-hard. Considering the settings of disaster relief routing and dynamic ride-sharing, the routing decisions need to be made quickly, it is desirable to obtain the near-optimal solutions in a relatively short period of time. In light of this, heuristic

approaches are proposed to solve the models.

The first approach is a two-stage heuristic algorithm for which the well-known insertion algorithm is extended and used in conjunction with a tabu search method. The overall heuristic scheme is as follows: the maximum CPU time allowed is set; the extended insertion algorithm is used to find a good feasible solution for a tabu search method; a tabu search is implemented repeatedly; and provide the best-so-far solution at the end of the given CPU time. In particular, we extend the insertion algorithm in Campbell and Savelsbergh (2004) to consider the capacity constraints of the CVRP and the SDVRP with different objective functions. For the SDVRP, we further extend the insertion algorithm to consider the split delivery. For the ride-sharing model, we extend (Campbell and Savelsbergh, 2004) to consider the capacity constraints (3.96), the maximum requests constraints (3.97), the maximum time constraints (3.104), and the riders that are left unserved. The details of the extended insertion algorithms are shown in Section 4.1. Tabu search (TS) is used to search for optimal or near-optimal solutions. In TS, the initial solution is found by implementing the proposed insertion algorithms. To solve different models, the move operators are specific and adjusted for each model. For the CVRP, five types of move operators, which do not consider split delivery, are used to find the neighbor solutions. These move operators can be used in SDVRP to find neighbor solutions without the consideration of split delivery. In addition to these move operators, four types of move operators that are specific for searching the neighbor solutions with the consideration of split delivery are used to solve the SDVRP. For ride-sharing model, four types of move operators are used to search neighbor solutions in TS. The details of TS is shown in Section 4.2. To solve the robust counterparts in the robust models, two algorithms are proposed to consider the increased demand and travel time due to change in the robust parameters that control the budget of uncertainty in CVRP and SDVRP. In addition, an algorithms is proposed to consider the increased travel time due to change in the robust parameters that control the budget of uncertainty in ride-sharing. The details of these algorithms are shown in Section 4.3.

In addition, a new hybrid algorithm named scoring tabu search with variable neighborhood (STSVN) is proposed to solve the models and compared with TS. In STSVN, a new scoring strategy and the features extracted from TS and variable neighborhood search

(VNS) are integrated to enable the adaptiveness of this hybrid algorithm as well as the ability to escape from local optima. The initial solution used in STSVN is constructed by the insertion algorithm. The algorithms in Section 4.3 are used to solve the robust counterparts in the robust models. The performance of STSVN and TS are compared by testing the same examples for different models.

Chapter 2

Literature Review

As mentioned in Chapter 1, VRP is a very broad area and there are many variants of VRP such as capacitated vehicle routing problem (Lysgaard et al., 2004), split delivery vehicle routing problem (Ho and Haugland, 2004; Archetti et al., 2006), vehicle routing problem with uncertain demands and/or uncertain travel time (Gendreau et al., 1996), vehicle routing problem with time window (Potvin and Bengio, 1996; Taillard et al., 1997; Bräysy and Gendreau, 2002; Cordeau et al., 2001), vehicle routing problem with backhauls (Osman and Wassan, 2002; Wassan, 2007; Duhamel et al., 1997), single-depot vehicle routing problem (Barbarosoglu and Ozgur, 1999), multi-depot vehicle routing problem (Polacek et al., 2004; Kuo and Wang, 2012; Paraskevopoulos et al., 2008), heterogeneous fleet vehicle routing problem (Gendreau et al., 1999; Wassan and Osman, 2002; Brandão, 2011), capacitated arc routing problem (Hertz et al., 2000; Polacek et al., 2008), open vehicle routing problem (Brandão, 2004; Fleszar et al., 2009), the vehicle routing problem with simultaneous pick-up and delivery service (Montané and Galvao, 2006; Polat et al., 2015), vehicle routing problem with two-dimensional loading constraints (Gendreau et al., 2008; Wei et al., 2015), stochastic and dynamic vehicle routing problem (Sarasola et al., 2016). For more details of VRPs and its variants, see Eksioglu et al. (2009).

In Chapter 2, the focus is to review the papers that are related to VRP with uncertain demands and/or uncertain travel time, especially related to the disaster relief routing and ride-sharing. The papers related to the robust optimization and stochastic programming in VRP are discussed in Section 2.1. The papers related to disaster relief routing and ride-sharing are summarized in Sections 2.2 and 2.3. Through the discussion in Sections 2.1 – 2.3,

the contributions of this dissertation are: (1) The robust models of SDVRP are proposed to consider travel time and demand uncertainty. (2) A robust model of ride-sharing is proposed to consider uncertain travel time. (3) The managerial insights are explored and provided for decision making when considering CVRP, SDVRP, different objectives, and uncertainty in disaster relief routing. In Section 2.4, the papers related to the algorithms to solve VRP, especially tabu search (TS) and variable neighborhood search (VNS), are reviewed. The differences between the algorithms in literature and the new hybrid algorithm STSVN proposed in this dissertation are discussed. To the best of my knowledge, this dissertation is the first paper to propose a hybrid algorithm combining TS and VNS to solve the models in disaster relief routing that consider uncertainty and the models of ride-sharing.

2.1 Vehicle Routing Problem with Uncertain Travel Time and Demand

Uncertain demands and travel times are important and frequently lead to issues in the VRP (Allahviranloo et al., 2014; Braaten et al., 2017). There have been a variety of methods proposed to properly handle and mitigate the impact of uncertainty in the literature. For example, two-stage stochastic programming has been used to model the uncertainty of the damage caused by disasters and its effect on supply or demand (Barbarosolu and Arda, 2004; Mete and Zabinsky, 2010; Zhu et al., 2008; Salmerón and Apte, 2010; Shen et al., 2009; Rawls and Turnquist, 2010; Van Hentenryck et al., 2010). In addition, several two-stage stochastic programming approaches have been proposed to model the uncertainty in travel time (Shen et al., 2009; Mete and Zabinsky, 2010; Rawls and Turnquist, 2010; Salmerón and Apte, 2010; Van Hentenryck et al., 2010), where travel times are scenario-dependent. However, stochastic programming has some disadvantages in VRP. It requires the known probability distribution function, and generally needs heavy computations (Bertsimas et al., 2011a; Sim, 2004). However, we may not know the exact information or even the probability distribution of the uncertainty of travel time in disaster relief routing and ride-sharing. When we only know the range of uncertainty, we still need to find relatively efficient and effective solution to VRP for immediate response. In that case, stochastic programming may not perform well. To address this issue, robust optimization is a good alternative to

solve the VRP with uncertainty in context of disaster relief routing and ride-sharing.

Robust optimization (RO) is a modeling methodology in which part of (or all) data are uncertain and only known to belong to some uncertainty sets (Ben-Tal and Nemirovski, 2002b). Without the probability distribution information regarding such uncertain data, a solution constructed by robust optimization can be feasible for any realization of the uncertainty in a given set (Bertsimas et al., 2011a). Because of its advantages, robust optimization has been applied in a variety of areas such as emergency logistics planning (Ben-Tal et al., 2011; Najafi et al., 2013) and value-based performance and risk management in supply chains (Hahn and Kuhn, 2012). In particular, robust optimization is used to address the demand uncertainty in the CVRP (Sungur et al., 2008; Erera et al., 2010; Ben-Tal et al., 2011; Gounaris et al., 2013; Allahviranloo et al., 2014). Regarding the uncertain travel times in the CVRP, Braaten et al. (2017) and Solano-Charris et al. (2014) consider a robust version of the CVRP with time windows. In addition, Han et al. (2013) address the CVRP with uncertain travel times in which a penalty is incurred for each vehicle that exceeds a given time limit while Agra et al. (2013) tackle the CVRP with time windows and travel times that belong to an uncertainty polytope. Furthermore, Chen et al. (2016) apply robust optimization for the road network daily maintenance routing problem with uncertain service times. Note that Lee et al. (2012) consider uncertain travel times and demands at the same time in the CVRP, while most other papers focus on only one of the two. We also note that most papers mentioned above consider the objective of minimizing the total travel time (or travel cost), which may not be relevant to the humanitarian logistics. Applying RO in the objectives such as minimizing the summation of arrival times and summation of demand-weighted arrival times can be challenging.

In the literature, only limited number of papers, e.g., Bouzaiene-Ayari et al. (1992), Yu et al. (2012), and Lei et al. (2012), that focus on the SDVRP with stochastic demands are found. Bouzaiene-Ayari et al. (1992) propose a heuristic algorithm for the SDVRP with stochastic demands. Yu et al. (2012) address the large scale stochastic inventory routing problem with split delivery and service level constraints. Lei et al. (2012) present a paired vehicle recourse policy for the SDVRP with stochastic demands. An adaptive large neighborhood search heuristic is applied for solving this problem. To the best of

our knowledge, there are no robust models of the SDVRP with uncertain travel times and demands for different objective functions in the literature.

In this dissertation, travel time and demand uncertainty are considered explicitly in CVRP and SDVRP, and I explore several objectives that may better suit the purpose of humanitarian logistics such as minimizing the summation of arrival times and the latest arrival time. In terms of ride-sharing, there are no robust models of ride-sharing problem with uncertain travel times (The detailed review of ride-sharing is shown in Section 2.3.).

In terms of VRP with uncertainty, the contributions in this research are (1) the proposed robust models of the SDVRP with uncertain travel times and demands for different objective functions and (2) the proposed robust model of the ride-sharing problem with uncertain travel times.

2.2 Disaster Relief Routing Problem

One of the most important questions in disaster relief is how to respond to these emergencies in an efficient manner to minimize the loss of life and maximize the efficiency of the rescue operations (Haghani and Oh, 1996). Disaster management is typically divided into three phases: preparation, immediate response, and reconstruction (Kovács and Spens, 2007) or four phases: mitigation added after preparation (Pearce, 2003). For more details for humanitarian logistics in disaster relief operations, see Kovács and Spens (2007).

For most disasters, an immediate response is necessary. Supply chains are desirable to be designed and deployed immediately even though the information of the situation is limited (Beamon and Kotleba, 2006; Long and Wood, 1995; Tomasini and Van Wassenhove, 2004). For the immediate response, the research can be classified into several groups. (1) Facility location problem: For example, Widener and Horner (2011) propose a hierarchical approach to decide the efficient placements of facilities for distributing relief services. Horner and Downs (2010) propose a capacitated warehouse location model for managing the flow of goods shipments to people in need. (2) Resources allocation problem: In this type of problems, the main focus is to determine the amount of resource distributed to different locations. For example, Gong and Batta (2007) propose the ambulance allocation and reallocation models for a post-disaster relief operation. Sheu (2007) propose a hybrid

fuzzy clustering-optimization approach to determine resource distribution for the urgent relief demands. (3) Network design problem: Bozorgi-Amiri et al. (2012) study a relief chain design problem and formulate this problem as a mixed-integer nonlinear programming to minimize the summation of the expected cost. (4) Vehicle routing problem (VRP): The VRP in the immediate response phase is called disaster relief routing (Luis et al., 2012). Vehicles are an essential part of the supply chain for delivering critical supplies. In disaster relief operations, vehicle routing problems are involved in detailed information collection, medical aid delivery, medical supply delivery, food supply delivery, etc (Luis et al., 2012). For example, Balcik et al. (2008) consider the VRP in deliver relief supplies from distribution center to different locations. Afshar and Haghani (2012) propose a mathematical model that controls the flow of several relief commodities from the sources to the recipients through the supply chain. Huang et al. (2013) focus on the assessment routing problem for assessing damage and relief needs after a disaster. Ozdamar and Yi (2008); Özdamar and Demir (2012) consider the VRP in evacuation and logistics support. Wang et al. (2014) propose a nonlinear integer open location-routing model for relief distribution problem. This dissertation focuses on the disaster relief routing (Luis et al., 2012) to effectively and equitably deliver critical supplies to the affected population.

In disaster relief routing, the routing decisions should be quick and sufficient with a focus on equitable service to all aid recipients (Huang et al., 2012). Campbell et al. (2008a) explore the objectives to minimize the summation of arrival times (minS) and the latest arrival time (minL) for CVRP, as the optimal solutions of minS and minL can provide earlier service times to customers than the ones of objective to minimize the total number of vehicles deployed (minV) and the total travel cost/time (minT). Huang et al. (2012) consider the objectives to minimize minT, minS, and the summation of demand-weighted arrival times (minD) for SDVRP and explore which solutions can be quick, sufficient, and equitable for all aid recipients (Huang et al., 2012). Therefore, it is important to consider minS, minL, and minD in disaster relief routing. In my research, minV, minT, minS, minD, and minL are considered and compared in both deterministic models and robust models for CVRP and SDVRP.

In terms of split delivery in disaster relief, limited number of papers are found (Özdamar

et al., 2004; Yi and Özdamar, 2007; Yi and Kumar, 2007; Afshar and Haghani, 2012; Lin et al., 2011; Wang et al., 2014). The works in Özdamar et al. (2004); Yi and Özdamar (2007); Yi and Kumar (2007); Afshar and Haghani (2012) are common in several points: they consider multi-commodity network flow problem in disaster relief. In these models, vehicles are regarded as commodity flows, so that split delivery is allowed. The objectives are to minimize the sum of unsatisfied demand of all commodities. They do not consider the uncertainty of travel time and demand. Lin et al. (2011); Wang et al. (2014) focus on multi-objective models that allowed split delivery. However, they also do not consider the uncertainty of travel time and demand. As discussed in Section 2.1, SDVRP has been studied by several researchers (Frizzell and Giffin, 1995; Belenguer et al., 2000; Ho and Haugland, 2004; Archetti et al., 2006; Boudia et al., 2007; Chen et al., 2007; Jin et al., 2007; Mitra, 2008; Archetti et al., 2008; Jin et al., 2008; Archetti et al., 2011; Belenguer et al., 2010; Desaulniers, 2010; Moreno et al., 2010; Berbotto et al., 2011; Salani and Vacca, 2011; Berbotto et al., 2014; Huang et al., 2012; Gulczynski et al., 2008), but no robust model of SDVRP has been proposed yet.

The disaster relief routing problems are different from general business routing problems, as the demand information may be unpredictable using historical data (Sheu, 2007). The varying travel times due to road disruptions and chaotic situations are also critical in planning a vehicle route after a disaster. Therefore, it is essential to mitigate the impact of uncertainty in planning a vehicle route for disaster relief. Bozorgi-Amiri et al. (2013) develop a stochastic programming approach to consider the uncertain demands, supplies, the cost of procurement, and transportation. In Barbarosolu and Arda (2004); Mete and Zabinsky (2010); Zhu et al. (2008); Salmerón and Apte (2010); Shen et al. (2009); Rawls and Turnquist (2010); Van Hentenryck et al. (2010), two-stage stochastic programming has been used to model the uncertainty of the damage caused by disasters and its effect on supply or demand. In addition, several two-stage stochastic programming approaches have been proposed to model the uncertainty in travel time (Shen et al., 2009; Mete and Zabinsky, 2010; Rawls and Turnquist, 2010; Salmerón and Apte, 2010; Van Hentenryck et al., 2010), where travel times are scenario-dependent. However, stochastic programming requires probability distributions of uncertain factors, which is not practical in disaster relief routing, since

the information and impact of each disaster is different from historical data. In addition, stochastic programming requires heavy computation (Bertsimas et al., 2011a; Sim, 2004). As discussed in Section 2.1, robust optimization is a good alternative to solve the VRP with uncertainty in context of disaster relief routing. In the general settings, robust optimization has been used to address the demand uncertainty in the CVRP (Sungur et al., 2008; Erera et al., 2010; Ben-Tal et al., 2011; Gounaris et al., 2013; Allahviranloo et al., 2014) and the uncertain travel times in the CVRP (Braaten et al., 2017; Solano-Charris et al., 2014; Han et al., 2013; Agra et al., 2013). However, only Ben-Tal et al. (2011) consider the application in disaster relief routing. In addition, these papers only consider CVRP with the objective of minimizing the total travel time (or travel cost), which may not be relevant to the humanitarian logistics. Applying RO in the objectives such as minimizing the summation of arrival times and summation of demand-weighted arrival times can provide some insights for disaster relief routing. Moreover, to the best of our knowledge, there are no robust models of the SDVRP with uncertain travel times and demands for different objective functions in the literature.

In this dissertation, travel time and demand uncertainty are considered explicitly in CVRP and SDVRP, and explore several objectives that may better suit the purpose of humanitarian logistics such as minimizing the summation of arrival times and the latest arrival time. In terms of disaster relief routing, the contributions of this dissertation are as follows. (1) The robust models of SDVRP are proposed to consider travel time and demand uncertainty. (2) The robust models of CVRP with different objectives are proposed in the context of disaster relief. (3) The managerial insights are explored and provided for decision making when considering CVRP, SDVRP, different objectives, and uncertainty in disaster relief routing.

2.3 Ride-sharing Problem

Ride-sharing problems have attracted scholars in the field of transportation and operations research. For the review of ride-sharing problems, see Agatz et al. (2012); Chan and Shaheen (2012); Furuhata et al. (2013). Among others, a dynamic ride-sharing problem finds riders with similar itineraries and time schedules on a short notice and aims to provide a best route

for a driver who is willing to pick-up the riders (Agatz et al., 2012). Features of such dynamic ride-sharing problem include dynamic drivers' and riders' information, non-recurring trips, and prearranged routes. That is, the ride-sharing should be established (prearranged) on a short notice that can range from a few minutes to a few hours before departure time (dynamic driver and rider information), which will be a single (non-recurring) trip. This distinguishes it from traditional carpooling or vanpooling, both of which require a long-term commitment among two or more people to travel together on recurring trips for a particular purpose.

A significant number of studies have been conducted for various aspects of ride-sharing including matching between drivers and riders, routing for the drivers, and pricing. Some studies put the emphasis on the matching. Pelzer et al. (2015) propose a partition-based matching algorithm for dynamic ride-sharing. Nourinejad and Roorda (2016) formulate the centralized matching model for one-to-one match of driver to passenger and decentralized matching model based on agent based simulation. Stiglic et al. (2016) show the impact of different types of participants' flexibility on the performance of a single-driver, single-rider ride-sharing system based on quantitative results. Their results indicate that small increases in flexibility can significantly increase the expected matching rate. Recently, Wang et al. (2017) present several mathematical programming methods to establish stable or nearly-stable matches.

In terms of the routing, Fanelli and Greco (2015) consider the ride-sharing with a vehicle of unlimited capacity. However, the assumption of unlimited capacity is not realistic. As one vehicle is considered, the matching between drivers and riders is not considered in Fanelli and Greco (2015). Using taxi data in New York City, Alonso-Mora et al. (2017) consider the real-time high-capacity ride-sharing that scales to large numbers of passengers and trips and dynamically generates optimal routes with respect to online demand and vehicle locations. The algorithm starts from a greedy assignment and improves it through a constrained optimization, quickly returning solutions of good quality and converging to the optimal assignment over time.

As ride-sharing involves dynamic demand and price, Zhang et al. (2016) propose a discounted trade reduction mechanism for dynamic ride-sharing pricing. In addition, Shen

et al. (2016) propose a posted-price, integrated online ride-sharing mechanism in autonomous mobility-on-demand systems. Cangialosi et al. (2016) propose a generalized ride sharing system that allow users to schedule multi-modal trips in a single task.

Researchers have proposed various modeling and solution methodologies for ride-sharing and its variants. In terms of exact methods, Baldacci et al. (2004) propose an exact method for the carpooling problem based on Lagrangian column generation. Deleplanque et al. (2014) propose a branch-and-price method for a reliability oriented dial-a-ride model. Hosni et al. (2014) formulate shared-taxi problem as a mixed integer program and present a Lagrangian decomposition approach to solve the model. Huang et al. (2014) compare three methods to solve the ride-sharing problem: branch-and-bound, integer programming, and kinetic tree. Their results indicate that kinetic tree algorithm is better than the other two methods in their example. Armant and Brown (2014) propose a mixed integer programming model for ride-sharing to minimize the total travel time. Linearization and symmetry breaking are used to solve the model. Bistaffa et al. (2014) propose a coalition formulation algorithm for ride-sharing and use a bounding technique to obtain approximate solutions. Cangialosi et al. (2014) propose a mathematical model to minimize the summation of the difference between the desired departure and arrival times with respect to the computed ones. Yousaf et al. (2014) formulate the ride-sharing problem as a multi-source-destination path planning problem. They propose a ride matching algorithm to calculate the matching score using all parameters such as detour distance, range factor, similarity of interests and personalized preference between drivers and riders. The path optimization is implemented by the Bellman-Ford algorithm.

In terms of heuristic methods, Manna and Prestwich (2014) use greedy algorithm and local search to solve the problem of online stochastic ride-sharing and taxi sharing. Santos and Xavier (2015) formulate a dynamic ride-sharing problem with the objective to maximize the number of served requests and minimize the total cost. The model is solved by a greedy randomized adaptive search procedure. Pinson et al. (2016) apply a generalized dial-a-ride problem to ride sharing case. A mathematical model is formulated to minimize the total difference between departure time from origin and arrival time at the destination of the driver and riders. Variable neighborhood descent is used to search good solutions.

As ride-sharing and carpooling require efficient information processing and communication support systems, Calvo et al. (2004) propose a distributed geographic information system (GIS) using several information and communication technologies. Winter and Nittel (2006) use mobile geosensor networks for ride-sharing. Fu et al. (2008) use traffic information grid to provide information for dynamic ride-sharing community service. Lin and Shen (2016) use hierarchical cloud architecture to develop a wireless social network aided vehicle sharing system called VShare. When a user send a travel request to VShare, it will identify the carpooler in nearby locations. If no carpool is found within nearby locations, the travel requests will be matched by the hierarchical cloud server architecture.

Some case studies of ride-sharing have been implemented. Heinrich (2010) implements real-time ride-sharing in the San Francisco Bay Area. Amey (2011) proposes a methodology for estimating ride-sharing viability within an organization and applies it to the MIT Community. Ghoseiri et al. (2011) present a dynamic ride-sharing matching optimization model that aims at identifying suitable matches between passengers requesting ride-sharing services with appropriate ride-sharing drivers available to carpool for credits and HOV lane privileges. Kleiner et al. (2011) propose a mechanism for dynamic ride-sharing based on parallel auctions.

However, there has been no paper addressing the uncertainty in travel times in dynamic ride-sharing problems and computational tractability for solving the problems with uncertainty. In this dissertation, a robust optimization based model is proposed to generate the optimal solution to handle the uncertain travel time.

2.4 Tabu Search and Variable Neighborhood Search in VRP

Vehicle Routing Problem (VRP) is the problem of designing optimal routes from one or several depots to a number of locations, subject to several constraints (Laporte, 1992). VRP plays an important role in logistics, therefore, a wide variety of VRPs are studied in literature. Based on different various VRPs, several heuristic algorithms and exact algorithms are proposed in literature. The exact algorithms used in VRPs are branch-and-bound (Fischetti et al., 1994), k-trees (Fisher, 1994), dynamic programming (Mahmoudi and Zhou, 2016), branch-and-cut-and-price (Lysgaard and Wøhlk, 2014), set partitioning and column

generation (Gendreau et al., 1994). Due to the limited success of exact methods in handling large size problems, most research on VRPs use heuristic approaches (Sze et al., 2016), such as tabu search (Potvin and Bengio, 1996; Taillard et al., 1997; Bräysy and Gendreau, 2002; Cordeau et al., 2001), variable neighborhood search (Crispim and Brandão, 2005; Belhaiza et al., 2014; Paraskevopoulos et al., 2008; Escobar et al., 2014), simulated annealing (Baños et al., 2013), and genetic algorithm (Karakatič and Podgorelec, 2015). For more details of algorithms used in VRPs, please review Toth and Vigo (2014).

As mentioned previously, the common characteristics of disaster relief routing and ride-sharing in this dissertation are: (1) The information of travel time and demand are provided in a short notice. (2) The routing decisions should be made quickly. (3) The trips are non-recurring, which means most solutions are one-time use. Small problems can be solved by using the commercial packages such as Gurobi and CPLEX. However, for large-scale problems, it is desirable to obtain the near-optimal solutions in a relatively short period of time using heuristic algorithms.

Tabu search (TS) is one of the most widely used heuristic algorithms to solve VRPs. TS is a single solution based, deterministic method to search for an optimal solution. For tutorials, we refer readers to (Glover, 1990). The general framework of TS is as follows. In TS, the initial solution is assumed to be given by some construction algorithms. The tabu list is used to record all the previous moves during the searching procedure. The best-so-far solution is updated during TS. The set of neighbor solutions are found by the defined move operators. In each iteration, the best neighbor solution that is not in the tabu list is used as the current solution in the next iteration (Potvin and Bengio, 1996; Taillard et al., 1997; Bräysy and Gendreau, 2002; Cordeau et al., 2001). TS and its variants have been applied in different types of VRPs, such as capacitated vehicle routing problem (Gendreau et al., 1994), split delivery vehicle routing problem (Ho and Haugland, 2004; Archetti et al., 2006), vehicle routing problem with uncertain demands and/or uncertain travel time (Gendreau et al., 1996), vehicle routing problem with time window (Potvin and Bengio, 1996; Taillard et al., 1997; Bräysy and Gendreau, 2002; Cordeau et al., 2001), vehicle routing problem with backhauls (Osman and Wassan, 2002; Wassan, 2007; Duhamel et al., 1997), single-depot vehicle routing problem (Barbarosoglu and Ozgur, 1999), multi-depot

vehicle routing problem (Paraskevopoulos et al., 2008), heterogeneous fleet vehicle routing problem (Gendreau et al., 1999; Wassan and Osman, 2002; Brandão, 2011), capacitated arc routing problem (Hertz et al., 2000), open vehicle routing problem (Brandão, 2004), the vehicle routing problem with simultaneous pick-up and delivery service (Montané and Galvao, 2006; Polat et al., 2015), vehicle routing problem with two-dimensional loading constraints (Gendreau et al., 2008; Wei et al., 2015).

One advantage of TS is exploring the search space by moving from a solution to its best neighbor solution that is not in the tabu list, which can escape from local optima (Renaud et al., 1996). To enhance the performance to solve the problems, various strategies have been proposed to be embedded in TS in literature (Osman, 1993; Gendreau et al., 1996; Potvin and Bengio, 1996; Duhamel et al., 1997; Taillard et al., 1997; Gendreau et al., 1999; Laporte et al., 2000; Ghiani et al., 2003; Tang and Miller-Hooks, 2005; Wassan, 2007). There are two common strategies to select the next move in TS (Osman, 1993): First strategy selects the best neighbor solution, which is not in tabu list, from all neighbor solutions. The first strategy is very popular in most TS in VRP problems (Gendreau et al., 1996; Potvin and Bengio, 1996; Duhamel et al., 1997). The second strategy is a greedy approach. It selects the first neighbor solution that provides an improvement in the objective value and is not in tabu list. If all neighbor solutions are not better than the current solution, the best neighbor solution, which is not in tabu list, is selected from all neighbor solutions (Osman, 1993). In most TS, all the move operators are applied in the current solution to search all corresponding neighbor solutions. In Duhamel et al. (1997), TS randomly chooses a move operator in each iteration. In most TS, the tabu list is used to record all the previous moves. In Gendreau et al. (1994), random tabu tags are used instead of tabu list. In Taillard et al. (1997), decomposition is used to divide a whole solution into several parts. For each part, TS is used to search the corresponding improved part. One of the interesting features added into TS is adaptive memory (Gendreau et al., 1999; Laporte et al., 2000; Ghiani et al., 2003; Tang and Miller-Hooks, 2005; Wassan, 2007). In Gendreau et al. (1999), different initial solutions are constructed by a stochastic insertion method. For each initial solution, TS is applied to search for better solutions. TS for all initial solutions are parallel implementations. During each TS, the routes of the best solutions visited during the

search are stored in an adaptive memory. Combining routes taken from different solutions in this memory, new solutions can be created and used as new initial solutions for each TS. However, as these routes are generated from different initial solutions, the selection of these routes should avoid including the same customer (node) twice in a solution (Laporte et al., 2000). In addition, the specific constraints of the VRPs also limit the success of combining different routes from the memory. Due to these restrictions, the selection process often terminates with a partial solution that have to be completed using a construction heuristic (Laporte et al., 2000). Reactive TS (RTS) is another interesting variant of TS (Nanry and Barnes, 2000; Osman and Wassan, 2002; Wassan, 2007; Paraskevopoulos et al., 2008). Based on the previously visited solutions and the quality of exploration, RTS adjusts its parameters. When a solution is revisited within a specified number of iterations, RTS will increase the length of the tabu list. If no solutions are repeated during a specified number of iterations, RTS will decrease the length of tabu list.

While heuristic algorithms can provide satisfactory results within a reasonable time, researchers aim to enhance the algorithms by including more attracting features from other algorithms (Sze et al., 2016). Developing hybrid approaches that include the advantages of two or more heuristic algorithms may provide better performance than a single approach. In this dissertation, TS is first modified to solve the corresponding deterministic and robust models in disaster relief routing and ride-sharing. In addition, a new hybrid algorithm based on TS, named scoring tabu search with variable neighborhood (STSVN), is proposed to solve the models and compared with the first approach (TS). In STSVN, a new scoring strategy, a new selection strategy, and the features extracted from variable neighborhood search (VNS) are integrated in TS to enhance the adaptiveness of STSVN as well as its ability to escape from local optima.

Before the discussion of STSVN, a brief introduction of VNS is as follows. VNS is another heuristic algorithm for solving different variants of the VRPs (Mladenović and Hansen, 1997; Polacek et al., 2004; Kuo and Wang, 2012; Sze et al., 2016, 2017), such as VRP with time windows (Polacek et al., 2004; Kuo and Wang, 2012), VRP with simultaneous pickup and delivery (Polat et al., 2015), open VRP (Fleszar et al., 2009), VRP with two-dimensional loading constraints (Wei et al., 2015), dynamic VRP (Sarasola et al., 2016),

capacitated arc routing problem (Hertz et al., 2000; Polacek et al., 2008), VRP with objective to minimize the sum of arrival times at nodes (Sze et al., 2017). The basic idea of VNS is to systematically change the neighborhood structure with local search (Mladenović and Hansen, 1997; Polacek et al., 2004; Kuo and Wang, 2012). Therefore, several neighborhood structures are used instead of a single one, as VNS contains local search implementations within each neighborhood structure. Here, neighborhood structure is defined by a move operator. For example, a move operator will find a neighbor solution by exchanging the positions of two nodes in the current solution. The set of all neighbor solutions found by this move operator is considered as a neighborhood structure. In VNS, the systematic change of neighborhood is applied during both a descent phase and an exploration phase to escape from the local optima (Polacek et al., 2004; Kuo and Wang, 2012). The basic VNS is as follows. Select the set of neighborhood structures (move operators) used in VNS. Generate an initial solution x and choose a stopping condition. Repeat the following Step (1) and (2) until the stopping condition is met. Step (1) set $k = 1$. Step (2) repeat Step (a) to Step (c) until $k =$ the total number of neighborhood structures (move operators). Step (a) shaking. Generate solution x' randomly from the neighborhood k . Step (b) local search. Use x' as initial solution in local search and find the best neighbor solution x'' , which is considered as a local optima in the neighborhood k based on x' . Step (c) If x'' is better than x , then $x \leftarrow x''$ and $k = 1$; otherwise, $k = k + 1$ (Mladenović and Hansen, 1997; Polacek et al., 2004; Kuo and Wang, 2012). The readers can review Hansen et al. (2010) for more details of VNS. The main purpose of the shaking phase in VNS is to provide a good initial solution for the local search. The initial solution is generated randomly from the neighborhood of the current solution. However, there is no strategy in the local search of VNS to avoid re-visiting the same solutions (Paraskevopoulos et al., 2008; Escobar et al., 2014).

As TS and VNS are two popular heuristic algorithms used in VRP and they use different strategies to escape from local optima, several papers propose the hybrid algorithms to include the advantages of TS and VNS in order to obtain better performance (Crispim and Brandão, 2005; Belhaiza et al., 2014; Paraskevopoulos et al., 2008; Escobar et al., 2014). Crispim and Brandão (2005); Belhaiza et al. (2014); Escobar et al. (2014) propose to add a tabu list into variable neighborhood search to avoid revisiting the same solutions. During

searching, the current solution is updated using a neighbor solution, which is not in tabu list, from one of the neighborhood structures. The neighborhood structure is changed in a consecutive way until the stop condition is met. Paraskevopoulos et al. (2008) mainly use the VNS structure and only use TS in local search phase of VNS. In Paraskevopoulos et al. (2008), the current solution may deteriorate from one iteration to another. However, the disadvantage of this process is that the search can be stuck in a bad neighborhood structure and a lot of time is spent to evaluate the worse solutions within a neighborhood structure, leading to less time to reach better solutions in other neighborhood structures. In addition, Paraskevopoulos et al. (2008) use TS in local search in each neighborhood structure, but do not use tabu list between neighborhood structures.

Comparing the proposed scoring tabu search with variable neighborhood (STSVN) in this dissertation with Crispim and Brandão (2005); Belhaiza et al. (2014); Paraskevopoulos et al. (2008); Escobar et al. (2014), STSVN not only combines the advantage of TS and VNS, but includes two new strategies to enhance the efficiency converging to local optima and the adaptiveness to determine diversification. The proposed new selection strategy in STSVN determines the next move. To enable this effectiveness of this selection strategy, a list of marks are generated in each iteration to record the routes that can be modified for current solution for improvement. This selection strategy is similar to the concept of adaptive memory (Gendreau et al., 1999; Laporte et al., 2000; Ghiani et al., 2003; Tang and Miller-Hooks, 2005; Wassan, 2007), but it addresses the limitation of adaptive memory. In adaptive memory of TS, solutions are generated from different initial solutions and the good solutions are kept in a pool. Then several routes are combined together. However, the new solution generated is generally incomplete based on the constraints (Gendreau et al., 1999; Laporte et al., 2000; Ghiani et al., 2003; Tang and Miller-Hooks, 2005; Wassan, 2007). In a new selection strategy proposed in this dissertation, the selection and combination processes of the neighbor solutions are based on the current solution. Only the neighbor solutions that are feasible and better than the current solution will be used to replace the routes of current solution. The marks generated in each selection will ensure the new combined solution is feasible and the improvement of objective function value is cumulated. In addition, this selection strategy is integrated with the tabu list. If a new combined solution is already

in the tabu list, the last neighbor solution that was used for the combined solution is removed from the candidate list. Then the selection strategy is implemented again until a new combined solution is found that is not in the tabu list. The selection strategy is also connected to the process of changing the move operator to ensure there are new neighbor solutions in the candidate list if the candidate list becomes empty. This is a new feature compared with Crispim and Brandão (2005); Belhaiza et al. (2014); Paraskevopoulos et al. (2008); Escobar et al. (2014).

The new scoring strategy feature allows the algorithm to adapt to the status of searching. The new scoring mechanism records the number of changes of each route during the search. The scores are considered as the memory of the algorithm and dynamically updated. The scores follow this rule: recent memory takes larger weights and previous memory takes less weights. When a local optima is reached after trying all move operators, the route with the least score is used for the diversification process to escape from the local optima. The advantage of the new scoring strategy is that the algorithm will automatically decide on the route for diversification based on the scores. This is another new feature compared with Crispim and Brandão (2005); Belhaiza et al. (2014); Paraskevopoulos et al. (2008); Escobar et al. (2014). The details of the algorithm is shown in Chapter 4.

In addition, the proposed hybrid algorithms combining TS and VNS in literature are used for VRP with backhauls (Crispim and Brandão, 2005), VRP with time windows (Belhaiza et al., 2014; Paraskevopoulos et al., 2008), heterogeneous fleet VRP (Paraskevopoulos et al., 2008), capacitated location-routing problem Escobar et al. (2014). To the best of my knowledge, this dissertation is the first paper to propose a hybrid algorithm combining TS and VNS to solve the models in disaster relief routing that consider uncertainty and the models of ride-sharing.

Chapter 3

Models

3.1 Models in Disaster Relief Routing

In this section, the deterministic models for CVRP and SDVRP with different objectives are presented as the bases for the robust counterparts. The robust counterparts of CVRP and SDVRP are then proposed. The notation used in Section 3.1 is summarized in Table 3.1.

3.1.1 Deterministic CVRP

The notation used in this section is as follows. In the deterministic CVRP, let the depot, the distribution center for critical supplies near a disaster area, be located at node 0 and the set of nodes except the depot be denoted by $N = \{1, \dots, n\}$. Here, we consider a node as the geographical locus where a distribution center or shelter can be located. The set of all nodes is then $N_0 = \{0\} \cup \{1, \dots, n\}$. The set of all arcs is denoted by A , where an arc is a road segment connecting nodes, such that $G = (N_0, A)$ represents the road network. The travel time between nodes i and j in N_0 is denoted by $t_{ij}, \forall (i, j) \in A$. In our assumption, the travel cost between nodes i and j is proportional to its travel time. The demand, the amount of critical supply needed, from node i is denoted by q_i , which needs to be served as soon as possible. We let k be the index of a vehicle and K be the set of vehicles, i.e., $k \in K = \{1, \dots, |K|\}$. All vehicles are assumed to be homogeneous and C denotes the capacity of a vehicle. The decision variables $x_{ij}, \forall (i, j) \in A$ are binary variables that indicate whether a vehicle travels from i to j .

Similar to the well-known Miller-Tucker-Zelmin (MTZ) formulation of the VRP (Miller

Table 3.1: Notation for Disaster Relief Routing

Symbols	Description
a_i	Arrival time of a vehicle at node i
a_{ik}	Arrival time of vehicle k at node i
A	Set of arcs in the network
c_i	Flow in the vehicle when it leaves the node i
C	Capacity of a vehicle
(i, j)	Arc between the nodes i and j
K	Set of vehicles, $k \in K$
K_{min}	Minimum number of vehicles needed for the SDVRP
N	Set of nodes except the depot
N_0	Set of all nodes including depot
q_i	Demand from node i
q	Vector $q = (q_i : i \in N)$
\bar{q}_i	Nominal demand of node i
\hat{q}_i	Increased demand of node i
\bar{q}	Vector for nominal demand vector
\hat{q}	Vector for increased demand
t_{ij}	Travel time between nodes i and j
t	Vector $t = (t_{ij} : (i, j) \in A)$
\bar{t}_{ij}	Nominal travel time of arc (i, j)
\hat{t}_{ij}	Maximum travel delay of arc (i, j)
\bar{t}	Vector for nominal travel time vector
\hat{t}	Vector for the maximum travel delay
\hat{t}_{e_i}	i th greatest $\hat{t}_{ij}, (i, j) \in A$ and x_{e_i} corresponds \hat{t}_{e_i}
T	Upper bound on the total travel time for each vehicle
x_{ij}	Binary variable that is equal to 1 only if arc (i, j) is traversed by a vehicle
x_{ijk}	Binary variable that is equal to 1 only if arc (i, j) is traversed by vehicle k
x'_{ij}	Variable to indicate whether the uncertain travel time of arc (i, j) is considered
U	Uncertainty set
U_T	Uncertainty set of travel time, $U_T = \{t \mid \bar{t} \leq t \leq \bar{t} + \hat{t}\}$
U_Q	Uncertainty set of demand, $U_Q = \{q \mid \bar{q} \leq q \leq \bar{q} + \hat{q}\}$
y_{ik}	Amount of demand served by vehicle k to node i
Γ_T	Parameter to control budget of travel time uncertainty
Γ_Q	Parameter to control budget of demand uncertainty

et al., 1960), a continuous variable $c_i, \forall i \in N$ is used, denoting the flow in the vehicle when it leaves the node i , to construct constraints that prevent subtours. A continuous variable $a_i, \forall i \in N$ denotes the arrival time of a vehicle at node i , and an upper bound on the total travel time for each vehicle is denoted by T , which can be relaxed if needed in the problem context by assigning a sufficiently large number to it. We do not consider a time window

for delivering critical supplies, as the objectives in the context of humanitarian logistics are prompt deliveries and we assume that all the demand nodes can accept deliveries any time (e.g., shelters open 24 hours/day). However, time window constraints can be easily added when necessary. The deterministic model that minimizes the total number of vehicles deployed (minV) is formulated as follows:

$$\text{(CVRP-minV)} \quad \min \sum_{i=1}^n x_{0i} \quad (3.1)$$

$$\text{s.t.} \quad \sum_{j \in N_0} x_{ij} = 1 \quad \forall i \in N \quad (3.2)$$

$$\sum_{j \in N_0} x_{ij} - \sum_{j \in N_0} x_{ji} = 0 \quad \forall i \in N_0 \quad (3.3)$$

$$t_{ij} \leq a_j - a_i + T(1 - x_{ij}) \quad \forall i, j \in N \quad (3.4)$$

$$t_{0i}x_{0i} \leq a_i \quad \forall i \in N \quad (3.5)$$

$$q_j \leq c_j - c_i + C(1 - x_{ij}) \quad \forall i, j \in N \quad (3.6)$$

$$q_i \leq c_i \leq C \quad \forall i \in N \quad (3.7)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (3.8)$$

The objective (3.1) is to minimize the number of vehicles deployed, constraints (3.2) require that each node be visited once by exactly one vehicle, and equations (3.3) are flow conservation constraints. The variables x_{ij} are associated with arrival times in inequalities (3.4), which also prevent subtours not including the depot. The appropriate minimum arrival time for each node is guaranteed in inequalities (3.5), and constraints (3.6) work in a similar fashion as constraints (3.4) and the capacity constraints are imposed in (3.7). To solve the CVRP-minV efficiently, an additional constraint, $\sum_{i=1}^n x_{0i} \geq \sum_{i=1}^n q_i / C$ may be added to the model, which provides a larger lower bound in the objective function value.

The model to minimize the total travel time/cost (minT) is exactly the same as the CVRP-minV except the objective function. That is,

$$\text{(CVRP-minT)} \quad \min \sum_{(i,j) \in A} t_{ij}x_{ij} \quad (3.9)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.8)$$

To minimize the summation of arrival times (minS), one more constraint to specify the given number of vehicles available, $|K|$, needs to be added; otherwise the optimal solution will be trivial, deploying vehicles as many as the number of nodes. The CVRP-minS can be formulated as:

$$\text{(CVRP-minS)} \quad \min \sum_{i \in N} a_i \quad (3.10)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.8)$$

$$\sum_{i \in N} x_{0i} = |K| \quad (3.11)$$

To minimize the summation of demand-weighted arrival times (minD), only the objective function needs to be changed as follows:

$$\text{(CVRP-minD)} \quad \min \sum_{i \in N} q_i a_i \quad (3.12)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.8), (3.11)$$

where q_i is the amount of demand from node i .

At last, the model to minimize the latest arrival time (minL) is formulated as:

$$\text{(CVRP-minL)} \quad \min a_l \quad (3.13)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.8), (3.11)$$

$$a_i \leq a_l \quad \forall i \in N \quad (3.14)$$

where a_i and a_l are the arrival time at node i and the latest arrival time, respectively.

3.1.2 Deterministic SDVRP

We once again note that the split delivery VRP (SDVRP) can be particularly useful when it comes to the disaster relief routing, as the amount of demand from a single node may exceed the vehicle capacity. The two-index formulation, e.g., x_{ij} , is used in the CVRP formulation and now we introduce the three-index formulation for the SDVRP while the notation for parameters remains the same to be consistent. The new decision variables $x_{ijk}, \forall (i, j) \in A, k \in K$ are binary, indicating whether vehicle k travels from i to j ($x_{ijk} = 1$) or not ($x_{ijk} = 0$). The amount of demand served by vehicle k to node i is denoted by

$y_{ik}, \forall i \in N, k \in K$ and a continuous variable $a_{ik}, \forall i \in N, k \in K$ denotes the arrival time of vehicle k at node i . The minimum number of vehicles needed, K_{min} , for the SDVRP can be calculated by solving $K_{min} = \lceil \sum_{i=1}^n q_i / C \rceil$.

In order to obtain a feasible solution, the number of available vehicles, $|K|$, must be at least K_{min} , i.e., $|K| \geq K_{min}$. We extend the deterministic SDVRP in Berbotto et al. (2014) to include the vehicle specific arrival time variable, a_{ik} in the subtour elimination constraints. The model to minimize the total travel time of the SDVRP (SDVRP-minT) is formulated as follows.

$$\text{(SDVRP-minT)} \quad \min \sum_{(i,j) \in A} \sum_{k \in K} t_{ij} x_{ijk} \quad (3.15)$$

$$\text{s.t.} \quad \sum_{j \in N_0} \sum_{k \in K} x_{ijk} \geq 1 \quad \forall i \in N \quad (3.16)$$

$$\sum_{j \in N_0} \sum_{k \in K} x_{0jk} \leq |K| \quad (3.17)$$

$$\sum_{j \in N_0} \sum_{k \in K} x_{0jk} \geq K_{min} \quad (3.18)$$

$$\sum_{j \in N_0} x_{ijk} - \sum_{j \in N_0} x_{jik} = 0 \quad \forall i \in N_0, k \in K \quad (3.19)$$

$$t_{ij} \leq a_{jk} - a_{ik} + T(1 - x_{ijk}) \quad \forall i, j \in N, k \in K \quad (3.20)$$

$$t_{0i} x_{0ik} \leq a_{ik} \quad \forall i \in N, k \in K \quad (3.21)$$

$$y_{ik} \leq q_i \sum_{j \in N_0} x_{ijk} \quad \forall i \in N, k \in K \quad (3.22)$$

$$\sum_{i \in N} y_{ik} \leq C \quad k \in K \quad (3.23)$$

$$\sum_{k \in K} y_{ik} = q_i \quad \forall i \in N \quad (3.24)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, k \in K \quad (3.25)$$

$$y_{ik} \geq 0 \quad \forall i \in N, k \in K \quad (3.26)$$

The constraints (3.16) require that each node should be visited by at least one vehicle. The inequality (3.17) ensures that at most $|K|$ vehicles depart from the depot while the inequality (3.18) ensures that at least K_{min} vehicles depart from the depot. The equations (3.19) are flow conservation constraints. The variables x_{ijk} are associated with arrival times

in inequalities (3.20), which also prevent subtours not including the depot. The appropriate minimum arrival times for each node are guaranteed in inequalities (3.21). The inequalities (3.22) ensure that the node can only be served if the vehicle visits it. The inequalities (3.23) ensure that the maximum load of each vehicle does not exceed capacity C . The equations (3.24) require that the entire demand of each node is satisfied.

To minimize the summation of arrival times, the SDVRP-minS can be formulated as:

$$\begin{aligned}
 \text{(SDVRP-minS)} \quad & \min \sum_{i \in N} \sum_{k \in K} a_{ik} & (3.27) \\
 \text{s.t.} \quad & (3.16)\text{--}(3.26)
 \end{aligned}$$

To minimize the summation of demand-weighted arrival times, the SDVRP-minD can be formulated as:

$$\begin{aligned}
 \text{(SDVRP-minD)} \quad & \min \sum_{i \in N} \sum_{k \in K} y_{ik} a_{ik} & (3.28) \\
 \text{s.t.} \quad & (3.16)\text{--}(3.26)
 \end{aligned}$$

Note that the SDVRP-minD is a mixed integer nonlinear programming (MINLP) model, as y_{ik} and a_{ik} are variables.

The objective to minimize the latest arrival time, a_l , is formulated as:

$$\begin{aligned}
 \text{(SDVRP-minL)} \quad & \min a_l & (3.29) \\
 \text{s.t.} \quad & (3.16)\text{--}(3.26) \\
 & a_{ik} \leq a_l \quad \forall i \in N, k \in K & (3.30)
 \end{aligned}$$

3.1.3 Robust Counterparts of the CVRP

In robust optimization (RO), there are a variety of ways to model uncertainty depending on how to define the sets to which the uncertain parameters belong. In this paper, we assume that uncertainty sets, which may be obtained by analyzing the historical data, are convex, closed, and bounded. Let us denote an uncertainty set by U and the travel times

and demands be subject to uncertainty. That is, $(t, q) \in U$ where t and q are the vectors such that $t = (t_{ij} : (i, j) \in A)$ and $q = (q_i : i \in N)$. Taking into account the uncertainty and considering that RO aims to find the best worst-case solutions, the robust CVRP-minV (RCVRP-minV) can be formulated as:

$$\text{(RCVRP-minV)} \quad \min \sum_{i=1}^n x_{0i} \quad (3.31)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.3), (3.8)$$

$$\max_{(t,q) \in U} t_{ij} \leq a_j - a_i + T(1 - x_{ij}) \quad \forall i, j \in N \quad (3.32)$$

$$\max_{(t,q) \in U} t_{0i} x_{0i} \leq a_i \quad \forall i \in N \quad (3.33)$$

$$\max_{(t,q) \in U} q_j \leq c_j - c_i + C(1 - x_{ij}) \quad \forall i, j \in N \quad (3.34)$$

$$\max_{(t,q) \in U} q_i \leq c_i \leq C \quad \forall i \in N \quad (3.35)$$

Likewise, the robust CVRP-minT (RCVRP-minT) can be formulated as:

$$\text{(RCVRP-minT)} \quad \min_x \max_{(t,q) \in U} \sum_{(i,j) \in A} t_{ij} x_{ij} \quad (3.36)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.3), (3.8), (3.32)\text{--}(3.35)$$

The robust counterparts for the CVRP-minS, CVRP-minD, and CVRP-minL can be formulated in a similar fashion as follows:

$$\text{(RCVRP-minS)} \quad \min \sum_{i \in N} a_i \quad (3.37)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.3), (3.8), (3.11), (3.32)\text{--}(3.35)$$

$$\text{(RCVRP-minD)} \quad \min_x \max_{(t,q) \in U} \sum_{i \in N} q_i a_i \quad (3.38)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.3), (3.8), (3.11), (3.32)\text{--}(3.35)$$

$$\text{(RCVRP-minL)} \quad \min a_l \quad (3.39)$$

$$\text{s.t.} \quad (3.2)\text{--}(3.3), (3.8), (3.11), (3.14), (3.32)\text{--}(3.35)$$

Note that the objective functions of the RCVRP-minT and the RCVRP-minD are subject to uncertainty while the ones of other models are not.

Let us assume that there is no correlation between t and q , then $U = U_T \times U_Q$ where $t \in U_T$ and $q \in U_Q$. Indeed, this assumption can be made without loss of generality in our robust formulations because inequalities (3.32)–(3.35), and functions (3.36) and (3.38) consider only one type of uncertainty.

In RO, all uncertainty sets are assumed to be bounded. Accordingly, let $U_T = \{t \mid \bar{t} \leq t \leq \bar{t} + \hat{t}\}$ and $U_Q = \{q \mid \bar{q} \leq q \leq \bar{q} + \hat{q}\}$ where \bar{t} and \bar{q} are the nominal travel time and demand vectors, respectively, and \hat{t} and \hat{q} are vectors for the maximum travel delay and increased demand caused by the destabilized infrastructure after a disaster. Such uncertainty sets employed in this paper are called box sets and we refer readers interested in a more general notion of uncertainty sets, e.g., ellipsoidal set and convex hull, to Ordóñez (2010) and Ben-Tal and Nemirovski (2002b). Because there is only one uncertain factor per constraint, inequalities (3.32)–(3.35) can be rewritten as follows:

$$\bar{t}_{ij} + \hat{t}_{ij}x_{ij} \leq a_j - a_i + T(1 - x_{ij}) \quad \forall i, j \in N \quad (3.40)$$

$$\bar{t}_{0i} + \hat{t}_{0i}x_{0i} \leq a_i \quad \forall i \in N \quad (3.41)$$

$$\bar{q}_j + \hat{q}_jx_{ij} \leq c_j - c_i + C(1 - x_{ij}) \quad \forall i, j \in N \quad (3.42)$$

$$\bar{q}_i + \hat{q}_i x_{0i} \leq c_i \leq C \quad \forall i \in N \quad (3.43)$$

These new constraints are deterministic with given \bar{t}_{ij} , \hat{t}_{ij} , \bar{q}_i , and \hat{q}_i .

For the objective function of RCVRP-minT (3.36), it has uncertain travel times up to the number of arcs in the road network. By employing the concept of the *budget of uncertainty* (Bertsimas and Sim, 2004), its uncertainty set can be reformulated as:

$$U_T = \left\{ t \mid \bar{t}_{ij} \leq t_{ij} \leq \bar{t}_{ij} + \hat{t}_{ij}x'_{ij}, (i, j) \in A, \sum_{(i,j) \in A} x'_{ij} \leq \Gamma_T, x'_{ij} \in \{0, 1\} \right\} \quad (3.44)$$

where the parameter Γ_T is called the *budget of uncertainty* and it controls the degree of conservatism or robustness of the solution. x'_{ij} is the variable to indicate whether the uncertain travel time of arc (i, j) is considered.

The objective function of RCVRP-minT is then:

$$(\text{RCVRP-minT}) \quad \min_{x \in X} \sum_{(i,j) \in A} \bar{t}_{ij} x_{ij} + \max_{t \in U_T} \sum_{(i,j) \in A} \hat{t}_{ij} x'_{ij} \quad (3.45)$$

where X is the feasible set for x . We may relabel $\hat{t}_{ij}, (i, j) \in A$ in a decreasing order, i.e., $\hat{t}_{e_1} \geq \hat{t}_{e_2} \geq \dots \geq \hat{t}_{e_m} \geq \hat{t}_{e_{m+1}} (= 0)$. Therefore, \hat{t}_{e_i} is the i th greatest $\hat{t}_{ij}, (i, j) \in A$. For the sake of notational convenience, we also employ x_{e_i} that corresponds \hat{t}_{e_i} . The following Theorem 1 shows that the solution of RCVRP-minT can be found by solving multiple deterministic CVRP-minT problems.

Theorem 1. The solution of RCVRP-minT (3.45) can be computed as the minimum of $|A| + 1$ deterministic VRP problems, for $l = 1, 2, \dots, |A| + 1$:

$$Z^l = \Gamma_T \hat{t}_{e_l} + \min_{x \in X} \left(\sum_{(i,j) \in A} \bar{t}_{ij} x_{ij} + \sum_{k=1}^l (\hat{t}_{e_k} - \hat{t}_{e_l}) x_{e_k} \right)$$

where $|A|$ is the number of arcs in the road network. Let $l^* = \arg \min_l Z^l$, then $Z^* = Z^{l^*}$ and $x^* = x^{l^*}$ where x^l is the optimal solution of Z^l .

Proof. See Bertsimas and Sim (2003).

The objective function of RCVRP-minD (3.38) can have uncertain demand nodes up to the number of nodes in the road network. A set $S_Q \subseteq U_Q, |S_Q| = \Gamma_Q$ is introduced, where Γ_Q is the budget of uncertainty, to the degree which the system is protected deterministically. Then the objective function of RCVRP-minD can be written as follows:

$$(\text{RCVRP-minD}) \quad \min \left(\sum_{i \in N} \bar{q}_i a_i + \max_{\{S_Q | S_Q \subseteq U_Q, |S_Q| \leq \Gamma_Q\}} \sum_{i \in S_Q} \hat{q}_i a_i \right) \quad (3.46)$$

This objective function is protected by:

$$\beta(\mathbf{a}, \Gamma_Q) = \max_{\{S_Q | S_Q \subseteq U_Q, |S_Q| \leq \Gamma_Q\}} \sum_{i \in S_Q} \hat{q}_i a_i \quad (3.47)$$

where \mathbf{a} is vector of $a_i, \forall i \in N$.

Proposition 1. Equation (3.47) is equivalent to the following linear optimization problem:

$$\beta(\mathbf{a}, \Gamma_Q) = \max \sum_{i \in N} \hat{q}_i a_i z'_i \quad (3.48)$$

$$\text{s.t. } \sum_{i \in N} z'_i \leq \Gamma_Q \quad (3.49)$$

$$0 \leq z'_i \leq 1 \quad \forall i \in N \quad (3.50)$$

Proof. It is clear that the optimal solution value of function (3.48) consists of $\lfloor \Gamma_Q \rfloor$ variables z'_i at 1. This is equivalent to the selection of subset $\{S_Q \mid S_Q \subseteq U_Q, |S_Q| \leq \Gamma_Q\}$ with corresponding function $\sum_{i \in S_Q} \hat{q}_i a_i$.

Theorem 2. The RCVRP-minD has the equivalent formulation as follows.

$$\text{(RCVRP-minD) } \min \sum_{i \in N} \bar{q}_i a_i + \Gamma_Q g' + \sum_{i \in N} p'_i \quad (3.51)$$

$$\text{s.t. } (3.2)\text{--}(3.3), (3.8), (3.11), (3.40)\text{--}(3.43)$$

$$g' + p'_i \geq \hat{q}_i a_i \quad \forall i \in N \quad (3.52)$$

$$p'_i \geq 0 \quad \forall i \in N \quad (3.53)$$

$$g' \geq 0 \quad (3.54)$$

Proof. Consider the dual of function (3.48):

$$\min \Gamma_Q g' + \sum_{i \in N} p'_i \quad (3.55)$$

$$\text{s.t. } g' + p'_i \geq \hat{q}_i a_i \quad \forall i \in N \quad (3.56)$$

$$p'_i \geq 0 \quad \forall i \in N \quad (3.57)$$

$$g' \geq 0 \quad (3.58)$$

By strong duality, since function (3.48) is feasible and bounded for $\Gamma_Q \in [0, |\mathcal{S}_Q|]$, then the dual problem (3.55) is also feasible and bounded and their objective values coincide. Using Proposition 1, we have that function (3.47) is equal to the objective function value of function (3.55). Substituting (3.55)–(3.58), we obtain that function (3.46) is equivalent to function (3.51). \blacksquare

3.1.4 Robust Counterparts of the SDVRP

The robust models of the SDVRP use the same fashion of the robust models of the CVRP.

$$\text{(RSDVRP-minT)} \min_x \max_{(t,d) \in U} \sum_{(i,j) \in A} \sum_{k \in K} t_{ij} x_{ijk} \quad (3.59)$$

$$\text{s.t.} \sum_{j \in N_0} \sum_{k \in K} x_{ijk} \geq 1 \quad \forall i \in N \quad (3.60)$$

$$\sum_{j \in N_0} \sum_{k \in K} x_{0jk} \leq |K| \quad (3.61)$$

$$\sum_{j \in N_0} x_{ijk} - \sum_{j \in N_0} x_{jik} = 0 \quad \forall i \in N_0, k \in K \quad (3.62)$$

$$\max_{(t,d) \in U} t_{ij} \leq a_{jk} - a_{ik} + T(1 - x_{ijk}) \quad \forall i, j \in N, k \in K \quad (3.63)$$

$$\max_{(t,d) \in U} t_{0i} x_{0ik} \leq a_{ik} \quad \forall i \in N, k \in K \quad (3.64)$$

$$y_{ik} - \max_{(t,d) \in U} q_i \sum_{j \in N_0} x_{ijk} \leq 0 \quad \forall i \in N, k \in K \quad (3.65)$$

$$\sum_{i \in N} y_{ik} \leq C \quad k \in K \quad (3.66)$$

$$\sum_{k \in K} y_{ik} - \max_{(t,d) \in U} q_i = 0 \quad \forall i \in N \quad (3.67)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, k \in K \quad (3.68)$$

$$y_{ik} \geq 0 \quad \forall i \in N, k \in K \quad (3.69)$$

The robust model to minimize the summation of arrival times can be formulated as:

$$\text{(RSDVRP-minS)} \min \sum_{i \in N} \sum_{k \in K} a_{ik} \quad (3.70)$$

$$\text{s.t. (3.60)–(3.69)}$$

The robust model to minimize the summation of demand weighted arrival times can be formulated as:

$$\text{(RSDVRP-minD)} \min \sum_{i \in N} \sum_{k \in K} y_{ik} a_{ik} \quad (3.71)$$

$$\text{s.t. (3.60)–(3.69)}$$

The robust model to minimize the latest arrival time is formulated as:

$$(\text{RSDVRP-minL}) \min a_l \quad (3.72)$$

$$\text{s.t. (3.60)–(3.69), (3.30)}$$

Inequalities (3.63) – (3.65), (3.67) can be written as:

$$\bar{t}_{ij} + \hat{t}_{ij}x_{ijk} \leq a_{jk} - a_{ik} + T(1 - x_{ijk}) \quad \forall i, j \in N, k \in K \quad (3.73)$$

$$(\bar{t}_{0i} + \hat{t}_{0i})x_{0ik} \leq a_{ik} \quad \forall i \in N, k \in K \quad (3.74)$$

$$y_{ik} - (\bar{q}_i + \hat{q}_i) \sum_{j \in N_0} x_{ijk} \leq 0 \quad \forall i \in N, k \in K \quad (3.75)$$

$$\sum_{k \in K} y_{ik} - (\bar{q}_i + \hat{q}_i) = 0 \quad \forall i \in N \quad (3.76)$$

The main difference between the CVRP models and the SDVRP models is whether an arc (i, j) can be used by multiple vehicles or not. In the CVRP, an arc (i, j) can be used at most once, therefore, each t_{ij} is related to one variable x_{ij} . In contrast, an arc (i, j) in the SDVRP can be used by multiple vehicles. Therefore, t_{ij} is related to $x_{ijk}, k \in K$. A set $S \subseteq U_T, |S| = \Gamma_T$ is introduced, where Γ_T is the budget of uncertainty. Then the RSDVRP-minT can be written as follows:

$$(\text{RSDVRP-minT}) \min \left(\sum_{(i,j) \in A} \sum_{k \in K} \bar{t}_{ij}x_{ijk} + \max_{\{S|S \subseteq U_T, |S| \leq \Gamma_T\}} \sum_{(i,j) \in S} \sum_{k \in K} \hat{t}_{ij}x_{ijk} \right) \quad (3.77)$$

The objective function is protected by:

$$\beta(\mathbf{x}, \Gamma_T) = \max_{\{S|S \subseteq U_T, |S| \leq \Gamma_T\}} \sum_{(i,j) \in S} \sum_{k \in K} \hat{t}_{ij}x_{ijk} \quad (3.78)$$

where \mathbf{x} is vector of decision variables.

Proposition 2. Equation (3.78) can be written as

$$\beta(\mathbf{x}, \Gamma_T) = \max_{\{S|S \subseteq U_T, |S| \leq \Gamma_T\}} \sum_{(i,j) \in S} \hat{t}_{ij} \sum_{k \in K} x_{ijk} \quad (3.79)$$

A new type of variable w_{ij} is introduced, which denotes the number of vehicles using arc (i, j) . Therefore,

$$\beta(\mathbf{x}, \Gamma_T) = \max_{\{S | S \subseteq U_T, |S| \leq \Gamma_T\}} \sum_{(i,j) \in S} \hat{t}_{ij} w_{ij} \quad (3.80)$$

$$\text{s.t. } w_{ij} = \sum_{k \in K} x_{ijk} \quad \forall (i, j) \in A \quad (3.81)$$

Equations (3.80) and (3.81) are equivalent to the following linear optimization problem:

$$\beta(\mathbf{x}, \Gamma_T) = \max \sum_{(i,j) \in A} \hat{t}_{ij} w_{ij} z_{ij} \quad (3.82)$$

$$\text{s.t. } (3.81)$$

$$\sum_{(i,j) \in A} z_{ij} \leq \Gamma_T \quad (3.83)$$

$$0 \leq z_{ij} \leq 1 \quad \forall (i, j) \in A \quad (3.84)$$

Proof. Clearly the optimal solution value of function (3.82) consists of $\lfloor \Gamma_T \rfloor$ variables z_{ij} at 1. This is equivalent to the selection of subset $\{S \mid S \subseteq U_T, |S| \leq \Gamma_T\}$ with corresponding function $\sum_{(i,j) \in S} \hat{t}_{ij} w_{ij}$.

Theorem 3. The RSDVRP-minT has the equivalent formulation as follows.

$$\text{(RSDVRP-minT) } \min \sum_{(i,j) \in A} \sum_{k \in K} \bar{t}_{ij} x_{ijk} + \Gamma_T g + \sum_{(i,j) \in A} p_{ij} \quad (3.85)$$

$$\text{s.t. } (3.60)–(3.62), (3.66)–(3.69), (3.73)–(3.76), (3.81)$$

$$g + p_{ij} \geq \hat{t}_{ij} w_{ij} \quad \forall (i, j) \in A \quad (3.86)$$

$$p_{ij} \geq 0 \quad \forall (i, j) \in A \quad (3.87)$$

$$g \geq 0 \quad (3.88)$$

$$0 \leq w_{ij} \leq |K| \quad \forall (i, j) \in A \quad (3.89)$$

Proof. Consider the dual of Problem (3.82):

$$\min \Gamma_T g + \sum_{(i,j) \in A} p_{ij} \quad (3.90)$$

$$\text{s.t. } g + p_{ij} \geq \hat{t}_{ij} w_{ij} \quad \forall (i, j) \in A \quad (3.91)$$

$$p_{ij} \geq 0 \quad \forall (i, j) \in A \quad (3.92)$$

$$g \geq 0 \quad (3.93)$$

By strong duality, since function (3.82) is feasible and bounded for $\Gamma_T \in [0, |S|]$, then the dual function (3.90) is also feasible and bounded and their objective values coincide. Using Proposition 2, we have that function (3.78) is equal to the objective function value of function (3.90). Substituting (3.90)–(3.93), we obtain that function (3.77) is equivalent to function (3.85).

3.2 Models in Ride-sharing

In this section, a dynamic ride-sharing model is proposed to find the optimal matches between riders and drivers and at the same time calculate the optimal routes for drivers in which multiple drivers and multiple riders are considered. The notation of ride-sharing models in Section 3.2 is summarized in Table 3.2.

3.2.1 Deterministic Model for Dynamic Ride-sharing

The dynamic ride-sharing problem considered in this paper contains several common assumptions (Agatz et al., 2012): (1) The ride-sharing is arranged on short-notice (a few minutes or within one hour before the earliest departure time of participants); (2) The ride-sharing is for non-recurring trips, which means that drivers and riders are different in each single arrangement; (3) The ride-sharing is pre-arranged based on participants' constraints, and only the trips that satisfy the drivers' and riders' constraints are conducted; and (4) There must be at least one driver and one rider. Let K be the set of drivers and R be the set of riders. In each arrangement of ride-sharing, $|K| \geq 1$ and $|R| \geq 1$. The origin and destination of the drivers can be different from the ones of riders.

The notation used in the mathematical model is as follows. The origin and destination of driver $k \in K$ are denoted by $b_k \in V_s$ and $w_k \in V'_s$ where V_s and V'_s represent the set of the drivers' origins and destinations, respectively. The origin-destination (o-d) pair of driver k is then denoted by (b_k, w_k) and the set of the drivers' origin-destination pairs is denoted by

P_s . Similarly, the origin, destination, and pair of them of rider r are denoted by $b_r \in V_c$, $w_r \in V'_c$, and $(b_r, w_r) \in P_c$ where V_c , V'_c , and P_c represent the set of the riders' origins, destinations, and o-d pairs, respectively. Let s_{kr} denote the decision variable to determine whether driver k will serve rider r . Driver k will transport the rider r from the origin of rider r to the destination of rider r when $s_{kr} = 1$, otherwise, $s_{kr} = 0$. Let $V = V_s \cup V'_s \cup V_c \cup V'_c$ and $i \in V$ be a location that is defined as either an origin or a destination for drivers and/or riders. Let $V_r = V'_c \cup V_c \setminus \{b_r\}$ denote the set of the riders' origins and destinations excluding b_r and $V'_r = V_c \cup V'_c \setminus \{w_r\}$ denote the set of the riders' origins and destinations excluding w_r . The transportation network considered in this problem is a complete network and $(i, j), i, j \in V$ denotes an arc in the transportation network representing the shortest feasible path between the nodes i and j . Let A represent the set of arcs in the complete network such that $(i, j) \in A$. In addition, let t_{ij} be the travel time between locations i and j , and $x_{ijk}, (i, j) \in A, k \in K$ be a binary variable that is equal to 1 only if arc (i, j) is traversed by driver k . Here, the assumption is that the travel cost is proportional to the travel time, and the coefficient to convert travel time into travel cost is denoted by f' . In addition, let $y_r, r \in R$, be a binary variable that is equal to 1 if the rider r is not served by any drivers and d_r is the associated penalty if the pick-up request of rider r is not served by any driver. Moreover, a_i is a nonnegative variable representing the arrival time at node i . Let e_{b_k} and l_{b_k} denote the earliest time and the latest time that driver k is willing to depart from his origin b_k , respectively. Likewise, let e_{w_k} and l_{w_k} denote the earliest time and latest time that driver k is willing to arrive at his destination. For rider $r \in R$, we can define $e_{b_r}, l_{b_r}, e_{w_r}$, and l_{w_r} in the same fashion. Let C_k denote the number of seats available for driver k , m_k denote the maximum requests that driver k is willing to serve, and t_k denote the maximum driving time driver k is willing to spend, and q_r be the amount of demand (the number of people to be picked up) in the pick-up request of rider location r as there may be more than one rider in the same location (e.g., family, friends, etc). Let T denote a large number that is used in the constraints to eliminate the sub-tours, $T \geq t_k, \forall k \in K$. The notation is summarized in Table 3.2.

The dynamic ride-sharing model can be formulated as a mixed-integer program as fol-

lows:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} f' t_{ij} x_{ijk} + \sum_{r \in R} d_r y_r \quad (3.94)$$

$$\text{s.t.} \quad \sum_{k \in K} s_{kr} + y_r = 1 \quad \forall r \in R \quad (3.95)$$

$$\sum_{r \in R} d_r s_{kr} \leq C_k \quad \forall k \in K \quad (3.96)$$

$$\sum_{r \in R} s_{kr} \leq m_k \quad \forall k \in K \quad (3.97)$$

$$\sum_{j \in \{b_k\} \cup V_r \setminus \{w_r\}} x_{jb_rk} - s_{kr} = 0 \quad \forall k \in K, \forall r \in R \quad (3.98)$$

$$\sum_{j \in V_r} x_{b_rjk} - s_{kr} = 0 \quad \forall k \in K, \forall r \in R \quad (3.99)$$

$$\sum_{j \in V'_r} x_{jw_rk} - s_{kr} = 0 \quad \forall k \in K, \forall r \in R \quad (3.100)$$

$$\sum_{j \in \{w_k\} \cup V'_r \setminus \{b_r\}} x_{w_rjk} - s_{kr} = 0 \quad \forall k \in K, \forall r \in R \quad (3.101)$$

$$\sum_{j \in \{w_k\} \cup V_c} x_{b_kjk} = 1 \quad \forall k \in K \quad (3.102)$$

$$\sum_{j \in \{b_k\} \cup V'_c} x_{jw_kk} = 1 \quad \forall k \in K \quad (3.103)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ijk} \leq t_k \quad \forall k \in K \quad (3.104)$$

$$a_j - a_i \geq t_{ij} - M \left(1 - \sum_{k \in K} x_{ijk} \right) \quad \forall (i,j) \in A \quad (3.105)$$

$$t_{b_k i} x_{b_k i k} - a_i \leq 0 \quad \forall k \in K, \forall i \in V_c \cup \{w_k\} \quad (3.106)$$

$$a_{b_r} - e_{b_r} (1 - y_r) \geq 0 \quad \forall r \in R \quad (3.107)$$

$$a_{b_r} - l_{b_r} (1 - y_r) \leq 0 \quad \forall r \in R \quad (3.108)$$

$$a_{w_r} - e_{w_r} (1 - y_r) \geq 0 \quad \forall r \in R \quad (3.109)$$

$$a_{w_r} - l_{w_r} (1 - y_r) \leq 0 \quad \forall r \in R \quad (3.110)$$

$$e_{b_k} \leq a_{b_k} \leq l_{b_k} \quad \forall k \in K \quad (3.111)$$

$$e_{w_k} \leq a_{w_k} \leq l_{w_k} \quad \forall k \in K \quad (3.112)$$

$$a_{w_k} \geq a_{w_r} + t_{w_r, w_k} - T (1 - s_{kr}) \quad \forall r \in R, \forall k \in K \quad (3.113)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in K \quad (3.114)$$

$$s_{kr} \in \{0, 1\} \quad \forall k \in K, \forall r \in R \quad (3.115)$$

$$y_r \in \{0, 1\} \quad \forall r \in R \quad (3.116)$$

$$a_{b_r}, a_{w_r}, a_{b_k}, a_{w_k} \geq 0 \quad \forall r \in R, \forall k \in K \quad (3.117)$$

The objective (3.94) is to minimize the total cost of the participants utilizing the ride-sharing, including the total travel cost of the drivers and the cost due to the penalties of unserved riders. Constraints (3.95) ensure that rider r is served by at most one driver and the penalty will occur if rider r is not served by any drivers. Constraints (3.96) ensure that the capacity constraints of the vehicles are satisfied during the ride-sharing. Constraints (3.97) ensure that the number of requests assigned to driver k does not exceed the maximum requests that driver k is willing to serve. Constraints (3.98) and (3.99) ensure that driver k needs to visit the origin of rider r if rider r is assigned to driver k . Constraints (3.100) and (3.101) ensure that driver k needs to visit the destination of rider r if rider r is assigned to driver k . Constraints (3.102) and (3.103) ensure that driver k leaves from his origin and arrives at his destination. Constraints (3.104) are the maximum time constraints. Constraints (3.105) and (3.106) ensure the sub-tours are eliminated and the arrival times are presented correctly. Constraints (3.107) and (3.108) ensure that rider r will be picked up at his origin within his given time window if rider r is matched with a driver. Constraints (3.109) and (3.110) ensure that rider r will arrive at his destination within his given time window if rider r is matched with a driver. Constraints (3.111) ensure that driver k will depart from his origin within his given time window. Constraints (3.112) ensure that driver k will arrive at his destination within his given time window. Constraints (3.113) ensure that driver k will send his assigned riders to their destinations before arriving at his destination.

In Stiglic et al. (2016), the matching flexibility is related to the willingness of participants to depart the origin and arrive at the destination earlier or later. When the ranges for departure time and arrival time increase, the matching flexibility increases. If a participant does not have specific earliest departure time or earliest arrival time, the earliest departure time or earliest arrival time for this participant can be set as the time when he posts his ride-sharing request. If a participant does not have specific latest departure time or latest

arrival time, the latest departure time or latest arrival time for this participant can be set as infinite. One assumption of this model is that the rider r is ready to depart his origin within his given time window. It means that the driver k does not need to wait for rider r as long as he arrives at the origin of rider r within the time window given by rider r . If waiting time δ_i at node i is considered in the model, then constraints (3.105) can be modified as

$$a_j - a_i \geq t_{ij} + \delta_i - T \left(1 - \sum_{k \in K} x_{ijk} \right) \quad \forall (i, j) \in A \quad (3.118)$$

Note that the above ride-sharing problem is in the formalism of mixed integer programming, which can be solved using off-the-shelf software such as CPLEX and Gurobi if the size of the problem is small enough. However, the ride-sharing problem is *NP-hard*, therefore, heuristics is required if the problem is not small. We present the insertion algorithm and tabu search method accordingly.

3.2.2 Robust Counterpart

Travel time uncertainty can play a critical role in our dynamic ride-sharing problem, as the optimal solution could be even infeasible if uncertain time is realized. In this section, let us consider how to find robust solution against the travel time uncertainty.

We assume the actual travel time t_{ij} , $(i, j) \in A$ is in the range $[\bar{t}_{ij}, \bar{t}_{ij} + \hat{t}_{ij}]$, where \bar{t}_{ij} is the nominal value of travel time and \hat{t}_{ij} is the maximum delay. Let J_k be the set of arcs subject to travel time uncertainty for driver k and let Γ_k be the parameter to control the budget of uncertainty for driver k (Ordóñez, 2010). Also, let S_k be the subset of J_k and its size is controlled by Γ_k . Now, we are ready to state the robust counterpart of the dynamic ride-sharing problem:

$$\min f' \sum_{k \in K} \left(\sum_{(i,j) \in A} \bar{t}_{ij} x_{ijk} + \max_{\{S_k | S_k \subseteq J_k, |S_k| \leq \Gamma_k\}} \sum_{(i,j) \in S_k} \hat{t}_{ij} x_{ijk} \right) + \sum_{r \in R} d_r y_r \quad (3.119)$$

$$\text{s.t. } (3.95) - (3.103), (3.105), (3.107) - (3.117)$$

$$\sum_{(i,j) \in A} \bar{t}_{ij} x_{ijk} + \max_{\{S_k | S_k \subseteq J_k, |S_k| \leq \Gamma_k\}} \sum_{(i,j) \in S_k} \hat{t}_{ij} x_{ijk} \leq t_k \quad \forall k \in K \quad (3.120)$$

$$a_j - a_i \geq (\bar{t}_{ij} + \hat{t}_{ij}) \sum_{k \in K} x_{ijk} - T \left(1 - \sum_{k \in K} x_{ijk} \right) \quad \forall (i, j) \in A \quad (3.121)$$

$$(\bar{t}_{b_k i} + \hat{t}_{b_k i}) x_{b_k i}^k - a_i \leq 0 \quad \forall k \in K, i \in V_c \cup \{w_k\} \quad (3.122)$$

Note that unlike the Soyster's robust model (Soyster, 1973), our formulation allows controlling the degree of conservatism by varying the parameter Γ_k . Note in particular that the constraints (3.120) are protected by:

$$\beta(x_{ijk}, \Gamma_k) = \max_{\{S_k | S_k \subseteq J_k, |S_k| \leq \Gamma_k\}} \sum_{(i,j) \in S_k} \hat{t}_{ij} x_{ijk} \quad (3.123)$$

Proposition 3. Given a set of x_{ijk} values, the function (3.123) is equivalent to the following optimization problem:

$$\max \sum_{(i,j) \in A} \hat{t}_{ij} x_{ijk} z_{ijk} \quad (3.124)$$

$$\text{s.t.} \sum_{(i,j) \in A} z_{ijk} \leq \Gamma_k \quad (3.125)$$

$$0 \leq z_{ijk} \leq 1 \quad \forall (i, j) \in A \quad (3.126)$$

Proof. Clearly the optimal solution value of function (3.124) consists of $[\Gamma_k]$ variables z_{ijk} at 1. This is equivalent to the selection of subset $\{S_k \mid S_k \subseteq J_k, |S_k| \leq \Gamma_k\}$ with corresponding function $\sum_{(i,j) \in S_k} \hat{t}_{ij} x_{ijk}$.

Theorem 4. The robust counterpart has the equivalent formulation as follows:

$$\min f' \sum_{k \in K} \left(\sum_{(i,j) \in A} \bar{t}_{ij} x_{ijk} + \Gamma_k g_k + \sum_{(i,j) \in A^k} p_{ijk} \right) + \sum_{r \in R} d_r y_r \quad (3.127)$$

$$\text{s.t.} \quad (3.95) - (3.103), (3.105), (3.107) - (3.117), (3.121), (3.122)$$

$$\sum_{(i,j) \in A} \bar{t}_{ij} x_{ijk} + \Gamma_k g_k + \sum_{(i,j) \in A^k} p_{ijk} \leq t_k \quad \forall k \in K \quad (3.128)$$

$$g_k + p_{ijk} \geq \hat{t}_{ij} x_{ijk} \quad \forall (i, j) \in A, k \in K \quad (3.129)$$

$$p_{ijk} \geq 0 \quad \forall (i, j) \in A, k \in K \quad (3.130)$$

$$g_k \geq 0 \qquad \forall k \in K \qquad (3.131)$$

where g_k and p_{ijk} are dual variables.

Proof. The dual of problem (3.124) – (3.126) can be written as:

$$\min \Gamma_k g_k + \sum_{(i,j) \in A} p_{ijk} \qquad (3.132)$$

$$\text{s.t. } g_k + p_{ijk} \geq \hat{t}_{ij} x_{ijk} \qquad \forall (i, j) \in A \qquad (3.133)$$

$$p_{ijk} \geq 0 \qquad \forall (i, j) \in A \qquad (3.134)$$

$$g_k \geq 0 \qquad \forall k \in K \qquad (3.135)$$

It is clear that the problem (3.124) – (3.126) and its dual (3.132) – (3.135) are both feasible and bounded. Then by the strong duality, their objective values must coincide. In addition, the function (3.123) is equivalent to the objective function (3.132) by Proposition 1. Therefore by substituting (3.132) – (3.135) into problem (3.119) – (3.122), we obtain (3.127) – (3.131).

Table 3.2: Notation for Ride-sharing Models

Symbols	Description
a_i	Arrival time of a vehicle at node i
A	Set of arcs in the network
b_k	Origin of driver k
(b_k, w_k)	Origin-destination (o-d) pair of driver k
b_r	Origin of rider r
(b_r, w_r)	Origin-destination (o-d) pair of rider r
C_k	Number of seats available for driver k
d_r	Associated penalty if the pick-up request of rider r is not served by any driver
e_{b_k}	Earliest time that driver k is willing to depart from his origin b_k
e_{w_k}	Earliest time that driver k is willing to arrive at his destination w_k
e_{b_r}	Earliest time that rider r is willing to depart from his origin b_r
e_{w_r}	Earliest time that rider r is willing to arrive at his destination w_r
i, j	Location that is defined as either an origin or a destination for drivers and/or riders
(i, j)	Arc between the nodes i and j
J_k	Set of arcs subject to travel time uncertainty
K	Set of drivers, $k \in K$
l_{b_k}	Latest time that driver k is willing to depart from his origin b_k
l_{w_k}	Latest time that driver k is willing to arrive at his destination w_k
l_{b_r}	Latest time that rider r is willing to depart from his origin b_r
l_{w_r}	Latest time that rider r is willing to arrive at his destination w_r
m_k	Maximum requests that driver k is willing to serve
P_c	Set of the riders' origin-destination pairs
P_s	Set of the drivers' origin-destination pairs
q_r	Demand of rider request r
R	Set of riders, $r \in R$
s_{kr}	Decision variable to determine whether driver k will serve rider r
S_k	Subset of J_k and its size is controlled by Γ_k
t_{ij}	Travel time between nodes i and j
t_k	Maximum driving time driver k is willing to spend
\bar{t}_{ij}	Nominal travel time of arc (i, j)
\hat{t}_{ij}	Maximum travel delay of arc (i, j)
T	Upper bound on the total travel time for each vehicle
x_{ij}	Binary variable that is equal to 1 only if arc (i, j) is traversed by a vehicle
V_c	Set of the riders' origins
V'_c	Set of the riders' destinations
V_r	$V_r = V'_c \cup V_c \setminus \{b_r\}$, set of the riders' origins and destinations excluding b_r
V'_r	$V'_r = V_c \cup V'_c \setminus \{w_r\}$, set of the riders' origins and destinations excluding w_r
V_s	Set of the drivers' origins
V'_s	Set of the drivers' destinations
w_k	Destination of driver k
w_r	Destination of rider r
Γ_k	Parameter to control budget of travel time uncertainty in the route k

Chapter 4

Algorithms

In this section, heuristic algorithms used in this dissertation are described. Small problems in disaster relief routing and ride-sharing models can be solved by using the commercial packages such as Gurobi and CPLEX. However, for large-scale problems, it is not practical to utilize the solvers as the VRPs are NP-hard. Considering the settings of disaster relief routing and dynamic ride-sharing, the routing decisions need to be made quickly, it is desirable to obtain the near-optimal solutions in a relatively short period of time. In light of this, two heuristic approaches are proposed to solve the models.

The first approach is a two-stage heuristic algorithm for which the well-known insertion algorithm is extended and used in conjunction with a tabu search method. The overall heuristic scheme is as follows: the maximum CPU time allowed is set; the extended insertion algorithm is used to find a good feasible solution for a tabu search method; a tabu search is implemented repeatedly; and provide the best-so-far solution at the end of the given CPU time. In particular, the insertion algorithm in Campbell and Savelsbergh (2004) are modified to consider the capacity constraints of the CVRP and the SDVRP with different objective functions. For the SDVRP, the insertion algorithm is further extended to consider the split delivery. For the ride-sharing model, the insertion algorithm is extended to consider the capacity constraints (3.96), the maximum requests constraints (3.97), the maximum time constraints (3.104), and the riders that are left unserved. The details of the extended insertion algorithms are shown in Section 4.1. Tabu search (TS) is used to search for the optimal or near-optimal solutions. In TS, the initial solution is found by implementing the proposed insertion algorithms. To solve different models, the move operators are specific

and adjusted for each model. For the CVRP, five types of move operators, which do not consider split delivery, are used to find the neighbor solutions. These move operators can be used in SDVRP to find neighbor solutions without the consideration of split delivery. In addition to these move operators, four types of move operators that are specific for searching the neighbor solutions with the consideration of split delivery are used to solve the SDVRP. For ride-sharing model, six types of move operators are used to search neighbor solutions in TS. The details of TS are shown in Section 4.2. To solve the robust counterparts in the robust models, two algorithms are proposed to consider the increased demand and travel time due to the change in robust parameters, Γ_Q and Γ_T , in CVRP and SDVRP. In addition, an algorithm is proposed to consider the increased travel time due to change in robust parameters, Γ_k , in ride-sharing. The details of these algorithms are shown in Section 4.3.

In this dissertation, a new hybrid algorithm is proposed as scoring tabu search with variable neighborhood (STSVN). In STSVN, a new scoring strategy and the features extracted from tabu search and variable neighborhood search are integrated to enable the adaptiveness of this hybrid algorithm as well as the ability to escape from local optima. The initial solution used in STSVN is constructed by the insertion algorithm. The algorithms in Section 4.3 are used to solve the robust counterparts in the robust models. The performance of STSVN is compared with TS by testing the same examples for different models.

4.1 Insertion Algorithms

4.1.1 Insertion Algorithms for Disaster Relief Routing

In this dissertation, the insertion algorithm is employed to find a good initial solution and it has been used for various vehicle routing problems, see, e.g., Campbell and Savelsbergh (2004); Campbell et al. (2008b). It starts with an easy, partial solution and creates a reasonably good feasible solution by inserting an unrouted customer node repeatedly. The constructed solution from the insertion algorithm is not guaranteed to be an optimal solution or a near-optimal solution, and it is used as an initial solution for a tabu search method. In this dissertation, the insertion algorithm in Campbell and Savelsbergh (2004) is modified to consider the capacity constraints of the CVRP by including them in the

algorithm (Algorithms 1 and 2) and the insertion algorithm is extended to further consider the split delivery by including the modified capacity constraint (Algorithm 3).

The similar terminology and notation of insertion algorithm used in Campbell and Savelsbergh (2004) is employed in this dissertation. Let us define a route as a set of customer nodes, i.e., route $k = (0, 1, 2, 3, \dots, i, \dots, n + 1)$ where i represents the i^{th} position and n is the number of nodes in the route and we let $n + 1 = 0$ (depot). A route that only contains the depot is defined as an empty route in this dissertation, i.e., an empty route $k = (0, 1)$. Let $k \in K'$ where K' is the set of routes and $|K'|$ is the number of routes. The set of unassigned nodes is denoted by N' and we introduce a new variable, the delivery volume currently assigned to route k , which is denoted by c^k . A variable E is used to keep track of the number of routes that cannot accommodate the additional demand q_j at node j due to the capacity limit, i.e., $C < c^k + q_j$, if an unrouted node j is inserted into the route $k \in K'$. The current objective function value for K' is denoted by $f(K')$. Let $K'_{i,j,k}$ denote the set of routes with inserting node j between position $i - 1$ and position i in route k . The objective function value of $K'_{i,j,k}$ is denoted by $f(K'_{i,j,k})$. Let $\delta = f(K'_{i,j,k}) - f(K')$, so that δ represents the difference between $f(K'_{i,j,k})$ and $f(K')$. Since a node can be visited by multiple vehicles (routes) in split delivery, let the delivery volume served at node j in route k be denoted by y_{jk} , so that $0 \leq y_{jk} \leq q_j$. The remaining demand at node j is denoted by q'_j .

The extended insertion algorithm for the CVRP-minV (minimizing the total number of vehicles or routes) is shown in Algorithm 1, which initially checks whether the demand of a node is greater than the capacity of a vehicle. If such a case exists, then Algorithm 1 will report no feasible solution for CVRP-minV; otherwise it continues to the following steps. Unlike Campbell and Savelsbergh (2004) where the insertion algorithm starts with all empty routes, Algorithm 1 starts with one empty route and keeps inserting nodes into the route. If a node cannot be inserted to any $k \in K'$ due to the capacity limit, then a new empty route is added to K' . Algorithm 1 stops when all nodes are inserted into routes. When the solution of CVRP-minV is found, it means that the minimum number of vehicles K_{min} needed to meet all the demand is decided. We propose Algorithm 2 to solve CVRP-minT, CVRP-minS, CVRP-minD, and CVRP-minL, as the insertion algorithms for

those problems share the same structure, in which we let $|K| \geq K_{min}$ as a given number of available vehicles. In Algorithm 2, K' starts with $|K|$ empty routes, instead of one empty route as in Algorithm 1, and the constraint $c^k + q_j \leq C$ is checked for all j and k . Therefore, the algorithm can guarantee K' is feasible for the capacity constraint.

The insertion algorithms for the SDVRP-minT, SDVRP-minS, SDVRP-minD, and SDVRP-minL share the same structures, as shown in Algorithm 3. In Algorithm 3, split delivery is allowed. The capacity constraint is replaced by $C - c^k \geq 0$ in Algorithm 3 because a vehicle can serve partial demand of a node. Once node j is inserted between position $i - 1$ and position i of route k , y_{jk} , q'_j , and c^k are updated based on the condition $(C - c^k) < q'_j$. If the full demand of a node has been served by the routes, then this node is removed from N' .

```

Extended Insertion Algorithm for CVRP-minV
Initialize feasibility = 1;
for  $j \in N'$  do
  if  $q_j > C$  then
    | feasibility = 0
  end
end
if feasibility = 0 then
  | Report no feasible solution for CVRP-minV
else
  |  $K'$  is initialized to contain one empty route;
  while  $N' \neq \emptyset$  do
    for  $j \in N'$  do
       $E = 0$ ;
      for  $k \in K'$  do
        if  $C - c^k \geq q_j$  then
          |  $k^* = k$ 
        else
          |  $E = E + 1$ 
        end
      end
      if  $E = |K'|$  then
        | add a new empty route  $k'$  in  $K'$ ;
        | insert  $j$  between position 0 and position 1 of  $k'$ ;
        |  $c^{k'} = q_j$ ;
      else
        | insert  $j$  between position 0 and position 1 of  $k^*$ ;
        |  $c^{k^*} = c^{k^*} + q_j$ 
      end
       $N' = N' \setminus j$ ;
    end
  end
  return  $K', |K'|$ 
end

```

Algorithm 1: Insertion Algorithm for CVRP-minV

Extended Insertion Algorithm for CVRP-minT, -minS, -minD, and -minL
 K' is initialized to contain $|K|$ empty routes, $f(K') = 0$;
 $c^k = 0, \forall k \in K'$;
while $N' \neq \emptyset$ **do**
 $\delta^* = \infty$;
 for $j \in N'$ **do**
 for $k \in K'$ **do**
 if $C - c^k \geq q_j$ **then**
 for $i \in \text{route } k$ **do**
 $K'_{i,j,k} = K'$, insert j between position $i - 1$ and position i of route k for $K'_{i,j,k}$;
 $\delta = f(K'_{i,j,k}) - f(K')$;
 if $\delta < \delta^*$ **then**
 $\delta^* = \delta, j^* = j, i^* = i, k^* = k$
 end
 end
 end
 end
 end
 insert j^* between position $i^* - 1$ and position i^* of route k^* for K' ;
 $f(K') = f(K') + \delta^*, c^{k^*} = c^{k^*} + q_{j^*}, N' = N' \setminus j^*$;
end
return $K', f(K')$

Algorithm 2: Insertion Algorithm for CVRP-minT, -minS, -minD, and -minL

Extended Insertion Algorithm for the SDVRP
 K' is initialized to contain $|K|$ empty routes, $f(K') = 0, q'_j = q_j, \forall j \in N'$;
 $y_{jk} = 0, \forall j \in N', \forall k \in K'$;
 $c^k = 0, \forall k \in K'$;
while $N' \neq \emptyset$ **do**
 $\delta^* = \infty$;
 for $j \in N'$ **do**
 for $k \in K'$ **do**
 if $C - c^k \geq 0$ **then**
 for $i \in \text{route } k$ **do**
 $K'_{i,j,k} = K'$, insert j between position $i - 1$ and position i of route k for $K'_{i,j,k}$;
 $\delta = f(K'_{i,j,k}) - f(K')$;
 if $\delta < \delta^*$ **then**
 $\delta^* = \delta, j^* = j, i^* = i, k^* = k$
 end
 end
 end
 end
 end
 insert j^* between position $i^* - 1$ and position i^* of route k^* for K' ;
 $f(K') = f(K') + \delta^*$;
 if $(C - c^k) < q'_{j^*}$ **then**
 $y_{j^*k^*} = y_{j^*k^*} + (C - c^{k^*}), q'_{j^*} = q'_{j^*} - (C - c^{k^*}), c^{k^*} = C$
 else
 $y_{j^*k^*} = y_{j^*k^*} + q'_{j^*}, q'_{j^*} = 0, c^{k^*} = c^{k^*} + q'_{j^*}$
 end
 if $q'_{j^*} = 0$ **then**
 $N' = N' \setminus j^*$
 end
end

Algorithm 3: Insertion Algorithm for SDVRP

4.1.2 Insertion Algorithm for Ride-sharing

The insertion algorithm in this section is used to find the initial feasible solution for the ride-sharing problem. The insertion algorithm in Campbell and Savelsbergh (2004) is extended to consider the capacity constraints (3.96), the maximum requests constraints (3.97), the maximum time constraints (3.104), and the riders that are left unserved. The details of the revised insertion algorithm is shown in Algorithm 4. The notation is the same as the one in the ride-sharing model. A route is defined as a set of locations. The route k for driver k starts with driver' origin b_k and ends with his/her destination w_k . The number of riders' origins and destinations between b_k and w_k is denoted by n_k , $0 \leq n_k \leq m_k$, i.e., route $k = (b_k, \dots, b_r, w_r, \dots, w_k)$. To refer to the position of the location in route k , i is used to represent the i^{th} position in route k . In the insertion algorithm, route k is for driver k , therefore, $|K|$ is the number of routes. Note that b_r and w_r should be in the same route and the position of b_r is prior to the position of w_r , $\forall r \in R$. The set of routes is denoted by K' . The seats have not been taken in the vehicle of driver k is denoted by c^k . A variable E is used to keep track of the number of routes that cannot accommodate the demand q_r of rider r due to the capacity constraints (3.96), the maximum requests constraints (3.97), or the maximum time constraints (3.104). The current objective function value for K' is denoted by $f(K')$. Let $K'_{i,r,k}$ denote the set of routes with inserting origin and destination of rider r between position $i - 1$ and position i in route k . The objective function value of $K'_{i,r,k}$ is denoted by $f(K'_{i,r,k})$. Let $\delta' = f(K'_{i,r,k}) - f(K')$, so that δ' represents the difference between $f(K'_{i,r,k})$ and $f(K')$.

Algorithm 4 starts with $|K|$ routes and route k contains b_k and w_k . Algorithm 4 stops when all nodes are inserted into routes or added into unserved list U' . The capacity constraints (3.96), the maximum requests constraints (3.97), and the maximum time constraints (3.104) are checked during the insertion algorithm. The time window constraints (3.107) – (3.113) are checked after insertion algorithm ends. If the time window constraints are not satisfied, the objective function value for K' is $f(K') + p'\lambda$ where λ represents the number of routes that do not satisfy the time window constraints. Therefore, the solution K' constructed by Algorithm 2 satisfies the capacity constraints (3.96), the maximum requests

constraints (3.97), and the maximum time constraints (3.104), but may violate the time window constraints. As p' is a very large number, $p'\lambda$ is considered as the penalty due to the violation of the time window constraints.

```

Extended Insertion Algorithm for Ride-sharing
 $K'$  is initialized to contain  $|K|$  routes, route  $k = (b_k, w_k)$ ,  $U' = \emptyset$ ,  $f(K') = 0$ ;
while  $R \neq \emptyset$  do
     $\delta^* = \infty$ ;
    for  $r \in R$  do
         $E = 0$ ;
        for  $k \in K'$  do
            if  $c^k \geq q_r$  and  $n_k < m_k$  then
                for  $i \in \text{route } k$  do
                     $K'_{i,r,k} = K'$ , insert  $b_r$  and  $w_r$  between position  $i - 1$  and position  $i$  of route  $k$  for
                     $K'_{i,r,k}$ ;
                     $\delta' = f(K'_{i,r,k}) - f(K')$ ;
                    if  $K'_{i,r,k}$  satisfies the maximum time constraints (3.104) then
                        if  $\delta' < \delta^*$  then
                             $\delta^* = \delta'$ ,  $r^* = r$ ,  $i^* = i$ ,  $k^* = k$ 
                        end
                    else
                         $E = E + 1$ 
                    end
                end
            else
                 $E = E + 1$ 
            end
        end
        if  $E = |K'|$  then
             $U' = U' \cup \{r\}$ ,  $R = R \setminus r$ 
        end
    end
    insert  $b_{r^*}$  and  $w_{r^*}$  between position  $i^* - 1$  and position  $i^*$  of route  $k^*$  for  $K'$ ;
     $f(K') = f(K') + \delta^*$ ,  $c^{k^*} = c^{k^*} - q_{r^*}$ ,  $R = R \setminus r^*$ ;
end
return  $K'$ ,  $U'$ ,  $f(K')$ 

```

Algorithm 4: Extended Insertion Algorithm for Ride-sharing

4.2 Tabu Search

The tabu search (TS) approach, one of the most widely used heuristics in operations research, is a single solution based, deterministic method to search for an optimal solution. One advantage of TS is exploring the search space by moving from a solution to its best neighbor solution that is not in the tabu list, which can escape from local optima (Renaud et al., 1996). The tabu list can record the solutions that have been visited and avoid moving to the same solution again. For tutorials, we refer readers to Glover (1990). The tabu search algorithm used in this dissertation to solve the different models with specific move operators.

4.2.1 Tabu Search for Disaster Relief Routing

In TS, the initial solution is assumed to be given, which can be found by implementing the proposed insertion algorithms. The current solution is denoted by K' and the tabu list is denoted by σ . The best-so-far solution while TS is being implemented is denoted by K^{best} , and the objective function value of K^{best} is denoted by $f(K^{best})$. In addition, the neighbor solution of K' is denoted by K'_h , and $f(K'_h)$ is the objective function value of K'_h . The set of K'_h that satisfies all constraints is denoted by M , and K'_h are sorted from the one with the smallest $f(K'_h)$ to the one with largest $f(K'_h)$. The maximum CPU time that allows the program to run is denoted by B_{max} and the elapsed CPU time is denoted by B . While $B < B_{max}$, TS is implemented iteratively. At each iteration, all K'_h of K' are found according to the move operators. In the move operators, the nodes are denoted by i and i' and the routes are denoted by k and k' .

For the CVRP, five types of move operators, which do not consider split delivery, are used to find the neighbor solutions. These move operators can be used in SDVRP to find neighbor solutions without the consideration of split delivery. In addition to these move operators, four types of move operators that are specific for searching the neighbor solutions with the consideration of split delivery are used to solve the SDVRP.

Five types of move operators that do not consider split delivery are as follows. (1) Exchange-node move operator: Choose two different nodes i and i' in the routes k and k' (k and k' can be same route or different routes) and switch the positions of nodes i and i' in routes k and k' . (2) Relocate-node move operator: Remove node i from route k and relocate i in front of node i' of route k' . (3) Two-insertion move operator (Sze et al., 2016): Take two consecutive nodes out of route k and insert them into route k' , as shown in Algorithm 6. (4) Reverse-sequence move operator: Choose a segment of route k and reverse the sequence of visited nodes in this segment, as shown in Algorithm 7. (5) Exchange-segments move operator: Exchange the segments of routes k and k' , as shown in Algorithm 8. These move operators are applied for $\forall i \in k, \forall i' \in k', \forall k, k' \in K', i \neq i'$.

Four types of move operators that are specific to consider split delivery (Berbotto et al., 2014) are as follows. (1) Add-split-node move operator: Add node i of route k into route

k' if $i \notin k', i \neq i', k \neq k'$. The demand of node i is split and served by route k and route k' , as shown in Algorithm 9. In Algorithms 9 – 12, β represents a set of nodes that are served by more than one vehicle. (2) Delete-split-node move operator: Choose a node that is served by more than one route, and the node from one of the routes is removed, as shown in Algorithm 10. (3) Delete-and-relocate move operator: Choose a node i that is served by more than one route. Then delete node i in k , remove i' from route k' and insert i' into route k , as shown in Algorithm 11. (4) Delete-and-split move operator: Choose a node i that is served by routes k and k' , and a node i' that is served by only one route. Then delete i in route k , and add i' into route k , as shown in Algorithm 12.

In the tabu search for CVRP and SDVRP, the infeasible neighbor solutions are not kept. Only the feasible neighbor solutions are evaluated and ranked according to their objective function values. The best neighbor solution that is not in the tabu list is used as current solution for the next iteration, and added in the tabu list. The best-so-far solution is saved during the whole procedure.

```

Tabu Search Algorithm for CVRP and SDVRP
 $K'$  = set of routes (current solution),  $K^{best} = K'$ ,  $\sigma = \emptyset$ ;
while  $B < B_{max}$  do
     $M = \emptyset$ ;
    find all  $K'_h$  of  $K'$  according to the move operators, and add them into  $M$ ;
    for  $K'_h \in M$  do
        | evaluate  $f(K'_h)$  of  $K'_h$ ;
    end
    rank  $K'_h \in M$  from the smallest  $f(K'_h)$  to the largest  $f(K'_h)$ ;
    for  $K'_h \in M$  do
        | if  $K'_h \notin \sigma$  then
        | |  $K' = K'_h$ ;
        | | add  $K'_h$  to  $\sigma$ ;
        | | if  $f(K'_h) < f(K^{best})$  then
        | | |  $K^{best} = K'_h$ ;
        | | |  $f(K^{best}) = f(K'_h)$ ;
        | | end
        | else
        | | if  $f(K'_h) < f(K^{best})$  then
        | | |  $K' = K'_h$ ;
        | | | add  $K'_h$  to  $\sigma$ ;
        | | |  $K^{best} = K'_h$ ;
        | | |  $f(K^{best}) = f(K'_h)$ ;
        | | end
        | end
        | break the for-loop when  $K'$  is updated;
    end
    update  $\sigma$  based on frequency
end
return  $K^{best}$  and  $f(K^{best})$ 

```

Algorithm 5: Tabu Search Algorithm for CVRP and SDVRP

Two-insertion Move Operator

```

if  $i \neq i'$  and there is at least one customer node in front of  $i$  then
   $K'_h = K'$ ;
  take  $i$  and the node in front of  $i$  out of route  $k$  for  $K'_h$ , insert them in front of  $i'$  in route  $r'$  for  $K'_h$ 
end

```

Algorithm 6: Two-insertion Move Operator**Reverse-sequence Move Operator**

```

 $n_k$  = the number of customer nodes in route  $k$ 
if  $n_k \leq 2$  then
   $l$  = position index of a node in route  $k$ 
  for  $l < n_k$  do
    for  $m \leq (n_k - l)$  do
       $K'_h = K'$ ;
      reverse the segment that starts from the node at position  $l$  and end at the position  $l + m$  in route  $k$  for  $K'_h$ 
    end
  end
end
end

```

Algorithm 7: Reverse-sequence Move Operator**Exchange-segments Move Operator**

```

if  $k \neq k'$  then
   $n_k$  = the number of customer nodes in route  $k$ 
   $n_{k'}$  = the number of customer nodes in route  $k'$ 
  if  $n_k \leq 2$  and  $n_{k'} \leq 2$  then
     $l$  = position index of a node in route  $k$ 
     $l'$  = position index of a node in route  $k'$ 
    for  $l < n_k$  do
      for  $m \leq (n_k - l)$  do
        for  $l' < n_{k'}$  do
          for  $m' \leq (n_{k'} - l')$  do
             $K'_h = K'$ ;
            exchange the segment that starts from the node at position  $l$  and end at the position  $l + m$  in route  $k$  with the segment that starts from the node at position  $l'$  and end at the position  $l' + m'$  in route  $k'$  for  $K'_h$ 
          end
        end
      end
    end
  end
end
end
end

```

Algorithm 8: Exchange-segments Move Operator**Add-split-node Move Operator**

```

for  $i \notin \beta$  do
  given route  $k$  containing  $i$ 
  for  $k' \in K'$  do
    if  $k \neq k'$  then
      for  $i' \in k'$  do
         $K'_h = K'$ 
        add  $i$  before  $i'$  in route  $k'$  for  $K'_h$ , modify  $y_{ik}$  and  $y_{ik'}$  accordingly
      end
    end
  end
end
end

```

Algorithm 9: Add-split-node Move Operator

```

Delete-split-node Move Operator
for  $i \in \beta$  do
  given route  $k$  and route  $k'$  containing  $i$ ,  $k \neq k'$ 
   $K'_h = K'$ 
  remove  $i$  from route  $k$ , modify  $y_{ik}$  and  $y_{ik'}$  accordingly
end

```

Algorithm 10: Delete-split-node Move Operator

```

Delete-and-relocate Move Operator
for  $i \in \beta$  do
  given route  $k$  and route  $k'$  contain  $i$ ,  $k \neq k'$ 
  for  $i' \in \text{route } k'$  do
    if  $i \neq i'$  then
       $K'_h = K'$ ;
      delete  $i$  in route  $k$ , remove  $i'$  from route  $k'$  and insert  $i'$  into route  $k$ ;
      modify  $y_{ik}$ ,  $y_{i'k}$  and  $y_{i'k'}$  accordingly
    end
  end
end

```

Algorithm 11: Delete-and-relocate Move Operator

```

Delete-and-split Move Operator
for  $i \in \beta$  do
  given route  $k$  and route  $k'$  contain  $i$ ,  $k \neq k'$ 
  for  $i' \in \text{route } k'$  do
    if  $i' \notin \beta$  then
       $K'_h = K'$ ;
      delete  $i$  in route  $k$ , and add  $i'$  into route  $k$ ;
      modify  $y_{ik}$  and  $y_{i'k}$  accordingly
    end
  end
end

```

Algorithm 12: Delete-and-split Move Operator

4.2.2 Tabu Search for Ride-sharing

For the ride-sharing problem, the initial solution can be found by implementing Algorithm 4. In the TS to solve the ride-sharing problem, the notation corresponds to the ride-sharing model. Six types of move operators are used to search neighbor solutions in the TS for ride-sharing. (1) Exchange-within-route: Choose node i and node i' in route k and switch the positions of node i and node i' in route k . This move operator is used $\forall i, i' \in \text{route } k, i \neq i', \forall k \in K'$. (2) Exchange-between-routes: If b_r and w_r are in route k and $b_{r'}$ and $w_{r'}$ are in route k' , switch the positions of b_r and $b_{r'}$, and switch the positions of w_r and $w_{r'}$, $\forall r, r' \in R, \forall k, k' \in K'$. (3) Relocate-between-routes: Remove b_r and w_r in route k , insert b_r between position $i - 1$ and position i in route k' , insert w_r between position $i' - 1$ and position i' in route k' . This move operator is applied when $i < i', \forall i \leq n_k + 1, \forall i \leq n_k + 2, \forall r \in R, \forall k \in K'$. (4) Exchange-unserved-customer: If b_r and w_r are in route k and r' is in unserved list U' , replace b_r with $b_{r'}$ and replace w_r with $w_{r'}$, remove r' from U' and add r to U' . This move operator is applied $\forall r \in R, \forall k \in K', \forall r' \in U'$. (5) Insert-unserved-customer-to-route: If r is in unserved list U' , insert b_r between position $i - 1$ and position i in route k , insert w_r between position $i' - 1$ and position i' in route k . This move operator is applied when $i < i', \forall i \leq n_k + 1, \forall i \leq n_k + 2, \forall r \in R, \forall k \in K'$. (6) Add-customer-to-unserved-list: If b_r and w_r are in route k , remove b_r and w_r from route k , add r into unserved list U' . This move operator is applied $\forall r \in R, \forall k \in K'$.

After a solution (K', U') is found, the capacity constraints (3.96), the maximum requests constraints (3.97), the maximum time constraints (3.104) are checked. If the solution (K', U') does not satisfy these constraints, the solution is not considered. If the solution does not satisfy time window constraints, the objective function value for (K', U') is $f(K', U') + p'\lambda$, where λ represents the number of routes that do not satisfy the time window constraints and p' represents the penalty of violation. Here, p' is a large number to ensure that the feasible solutions have more priority to be selected in each iteration of the heuristic algorithm. If p' is not large enough, the heuristic algorithm may use too much time to explore the region of infeasible solutions (Smith et al., 1997). In the setting of ride-sharing, the CPU time used to search feasible good solutions is limited because the

feasible good solutions should be provided in a short time. Therefore, feasible solutions are more important and desired. The penalty method used for ride-sharing is the static penalty function, which is simple and suitable to be used when large priority is given to feasible solutions. In literature, there are other types of penalty functions such as dynamic penalty functions and adaptive penalty functions (Smith et al., 1997; Mezura-Montes and Coello, 2011; Coit et al., 1996). These penalty functions may allow exploring the good feasible solutions near the boundary of infeasible solutions. However, dynamic penalty functions typically require problem specific tuning (Smith et al., 1997). Adaptive penalty functions involve more parameters to control the change of penalty values (Smith et al., 1997). Because the penalty functions are not the main focus in this dissertation, the comparison of different penalty functions can be the future work in this dissertation. For more details of different penalty functions, see Smith et al. (1997); Mezura-Montes and Coello (2011); Coit et al. (1996).

In Algorithm 13, the current solution is denoted by (K', U') and the tabu list is denoted by σ . The best-so-far solution during TS implementation is denoted by $(K', U')^{best}$, and the objective function value of $(K', U')^{best}$ is denoted by $f((K', U')^{best})$. In addition, the neighbor solution of (K', U') is denoted by $(K', U')_h$, and $f((K', U')_h)$ is the objective function value of $(K', U')_h$. The set of $(K', U')_h$ is denoted by M , and $(K', U')_h$ are rearranged from the one with the smallest $f((K', U')_h)$ to the one with largest $f(K'_h)$. The maximum CPU time that allows program to run is denoted by B_{max} and the elapsed CPU time is denoted by B . While $B < B_{max}$, TS is implemented iteratively.

4.3 Algorithms for Solving Robust Counterparts

To solve the robust counterparts of CVRP and SDVRP with different objectives to consider uncertain travel times and demands with the heuristic algorithms, Algorithms 14 and 15 are proposed to consider the total amount of increased demand and travel time in the solution based on parameters Γ_Q and Γ_T for robust models in disaster relief routing. Depending on the degree of robustness (that can be decided by Γ_Q and Γ_T values), a certain uncertain parameter, e.g., \hat{t}_{ij} for some (i, j) in A , may or may not be included in the solution. The algorithms presented in this section deal with such cases. To solve the robust counterparts

```

Tabu Search Algorithm for Ride-sharing
 $(K', U')$  = current solution,  $(K, U)^{best} = (K', U')$ ,  $\sigma = \emptyset$ ;
while  $B < B_{max}$  do
     $M = \emptyset$ ;
    find all  $(K', U')_h$  of  $(K', U')$  according to the move operators;
    add the  $(K', U')_h$  that satisfy constraints (3.96), (3.97), and (3.104) into  $M$ ;
    for  $(K', U')_h \in M$  do
        | evaluate  $f((K', U')_h)$  of  $(K', U')_h$ ;
    end
    rank  $(K', U')_h \in M$  from the smallest  $f((K', U')_h)$  to the largest  $f((K', U')_h)$ ;
    for  $(K', U')_h \in M$  do
        | if  $(K', U')_h \notin \sigma$  then
            |  $(K', U') = (K', U')_h$ ;
            | add  $(K', U')_h$  to  $\sigma$ ;
            | if  $f((K', U')_h) < f((K', U')^{best})$  then
                |  $(K', U')^{best} = (K', U')_h$ ;
                |  $f((K', U')^{best}) = f((K', U')_h)$ ;
            | end
        | else
            | if  $f((K', U')_h) < f((K', U')^{best})$  then
                |  $(K', U') = (K', U')_h$ ;
                | add  $(K', U')_h$  to  $\sigma$ ;
                |  $(K', U')^{best} = (K', U')_h$ ;
                |  $f((K', U')^{best}) = f((K', U')_h)$ ;
            | end
        | end
        | break the for-loop when  $(K', U')$  is updated;
    end
    update  $\sigma$  based on frequency
end
return  $(K', U')^{best}$  and  $f((K', U')^{best})$ 

```

Algorithm 13: Tabu Search Algorithm for Ride-sharing

of the ride-sharing model to consider uncertain travel times with the heuristic algorithms, Algorithm 16 is proposed to consider the total amount of increased travel time in the solution based on parameters $\Gamma_k, \forall k \in K$. Depending on the degree of robustness (that can be decided by values of $\Gamma_k, \forall k \in K$), \hat{t}_{ij} for some (i, j) in A , may or may not be included in the solution.

```

Update Demand Based on  $\Gamma_Q$ 
Given  $\Gamma_Q, \bar{q}_i, i \in N, \hat{q}_i, i \in N$ ;
sort  $\hat{q}_i, i \in N$  in decreasing order;
 $l$  = the position index of  $\hat{q}_i$  in sorted order,  $l = 1, 2, \dots, |N|$ ;
 $S'_Q$  = the set of nodes of which the  $\hat{q}_i$  are considered in the solution based on  $\Gamma_Q$ ;
 $S'_Q = \emptyset$ 
for  $l \leq \Gamma_Q$  do
|   add node index  $i$  of  $\hat{q}_i$  at position  $l$  into  $S'_Q$ 
end
for  $i \in N$  do
|   if  $i \in S'_Q$  then
|   |    $q_i = \bar{q}_i + \hat{q}_i$ 
|   else
|   |    $q_i = \bar{q}_i$ 
|   end
end
return  $q_i, i \in N$ 

```

Algorithm 14: Uncertain Demand Selection Algorithm Based on Γ_Q

```

Update Travel Time Based on  $\Gamma_T$ 
Given  $\Gamma_T, \bar{t}_{ij}, (i, j) \in A, \hat{t}_{ij}, (i, j) \in A$ , and a solution  $K'$ ;
compute  $w_{ij}, (i, j) \in A$ ;
 $v_{ij} = w_{ij}\hat{t}_{ij}, (i, j) \in A$ ;
sort  $v_{ij}, (i, j) \in A$  in decreasing order;
 $l$  = the position index of  $v_{ij}$  in sorted order,  $l = 1, 2, \dots, |A|$ ;
 $S'_T$  = the set of arcs of which the  $v_{ij}$  are considered in the solution based on  $\Gamma_T$ .  $S'_T = \emptyset$ ;
for  $l \leq \Gamma_T$  do
|   add arc  $(i, j)$  of  $v_{ij}$  at position  $l$  into  $S'_T$ 
end
for  $(i, j) \in A$  do
|   if  $(i, j) \in S'_T$  then
|   |    $t'_{ij} = \bar{t}_{ij} + \hat{t}_{ij}$ 
|   else
|   |    $t'_{ij} = \bar{t}_{ij}$ 
|   end
end
return  $t'_{ij}, (i, j) \in A$ 

```

Algorithm 15: Uncertain Travel Time Selection Algorithm Based on Γ_T

Algorithm 14 is to select the maximum potential increased amount of demand based on Γ_Q , which is implemented before the heuristic algorithms are used. As each node is visited at least once, q_i becomes fixed for each solution after determination of set of \hat{q}_i .

Different from evaluating the uncertainty set of demands, the set of selected \hat{t}_{ij} for each solution varies even for the same Γ_T value (same degree of robustness) because there is one

more parameter to be considered, w_{ij} , the number of vehicles using an arc (i, j) . For an arc (i, j) where \hat{t}_{ij} is large and $w_{ij} = 0$ (implying that this arc is not used in the solution), this \hat{t}_{ij} is not to be selected in the uncertainty set of travel times when evaluating the maximum potential increased travel time due to uncertainty. For an arc (i, j) where \hat{t}_{ij} not large but w_{ij} is large, this \hat{t}_{ij} may be selected in the uncertainty set because this arc is used for multiple vehicles. Algorithm 15 considers the effect of \hat{t}_{ij} and w_{ij} together on the maximum potential increased travel time of a solution. Algorithm 15 is implemented for each new constructed solution in the insertion algorithm and tabu search. The objective function value and metrics of the new solution is computed using t'_{ij} .

```

Compute Total Increased Travel Time Based on  $\Gamma_k$ 
Given  $\Gamma_k, \forall k \in K, \bar{t}_{ij}, (i, j) \in A, \hat{t}_{ij}, (i, j) \in A$ , and a solution  $K'$ ;
for  $k \in K$  do
    sort all  $\hat{t}_{ij}$  in route  $k$  in decreasing order;
     $l$  = the position index of  $\hat{t}_{ij}$  in sorted order ;
     $S^k$  = the set of arcs of which the  $\hat{t}_{ij}$  are considered in route  $k$  based on  $\Gamma_k$ .  $S^k = \emptyset$ ;
    for  $l \leq \Gamma_k$  do
        | add arc  $(i, j)$  of  $\hat{t}_{ij}$  at position  $l$  into  $S^k$ 
    end
    total increased travel time in route  $k = \sum_{(i,j) \in S^k} \hat{t}_{ij}$ ;
end
total increased travel time in  $K' = \sum_{k \in K} \sum_{(i,j) \in S^k} \hat{t}_{ij}$ 

```

Algorithm 16: Compute Total Increased Travel Time Based on Γ_k

4.4 Scoring Tabu Search with Variable Neighborhood

As mentioned previously, because of the settings of disaster relief routing and dynamic ride-sharing, the routing decisions need to be made quickly. It is desirable to obtain the near-optimal solutions in a short period of time. Besides TS in Section 4.2, a new hybrid algorithm based on TS, named scoring tabu search with variable neighborhood (STSVN), is proposed to solve the models and compared with TS.

The STSVN consists of two stages: Stage 1 aims to intensively search better solutions and reach the local optima quickly, while Stage 2 aims to escape from a local optima by a diversification approach. The STSVN is based on TS. Similar to TS, a tabu list is used to record the previous visited solutions within the search space to prevent cycling. Different from TS, STSVN does not yield to the worse solutions during searching within a given neighborhood structure. It means when better solutions are not found in the neighborhood

structure h , then search neighborhood structure $h + 1$ instead of staying in neighborhood structure h . Systematically searching the neighbor solutions based on the change of the neighborhood generated by various move operators is the feature from variable neighborhood search (VNS). In addition, a proposed new selection strategy in STSVN determines the next move. To enable the effectiveness of this selection strategy, a list of marks are generated in each iteration to record the routes that can be modified for the current solution for improvement. In Stage 1, tabu search with change of neighborhood and selection strategy enhances the ability to descent to local optima. In STSVN, the new scoring mechanism records the number of changes of each route during search. The scores are considered as the memory of the algorithm and dynamically updated. The scores follow this rule: Recent memory takes larger weights and previous memory takes less weights. When reaching a local optima after trying all move operators, the route with the least score is used for Stage 2 for diversification to escape from the local optima. In Stage 2, all nodes from the route with the least score are removed from this route and re-inserted into any routes based on the least cost. The new scoring strategy allows STSVN to adapt to the status of searching and explore the searching space that has not been explored previously. The detailed explanation of STSVN is shown in Section 4.4.1 and the summary of STSVN is shown in Algorithm 17.

4.4.1 Main Steps of STSVN

Step 0: Initialization. Obtain an initial solution K' from insertion algorithm. Evaluate the objective function value $f(K')$. Let K^{best} record the best-so-far solution during whole searching and $f(K^{best})$ be the objective function value of K^{best} . At the beginning of STSVN, $K^{best} = K'$ and $f(K^{best}) = f(K')$. Initialize tabu list $\sigma = \emptyset$, add K' into σ . Set the maximum CPU time B_{max} allowed. Define a set of move operators H . Let $score_k$ to record the number of changes of route k during search. Initialize $score_k = 0, \forall k \in K'$. Go to Stage 1.

In Stage 1. The main purpose is to converge to a local optima quickly. Stage 1 consists of Step 1 – Step 11, as follows.

Step 1: If the CPU time $B \geq B_{max}$, then STSVN stops and reports K^{best} and $f(K^{best})$. Otherwise, $h = 1$, $M = \emptyset$, and go to Step 2. In Step 1, the first move operator is chosen

and M is used to store some feasible neighbor solutions of current solution K' . In STSVN, the move operators used to solve CVRP and SDVRP models are the same as the ones used in TS described in Section 4.2.1, and the move operators used to solve ride-sharing models are the same as the ones used in TS described in Section 4.2.2. In Step 1, M is initialized as an empty set.

Step 2: If the CPU time $B < B_{max}$ and $h \leq |H|$, go to Step 3. If the CPU time $B > B_{max}$, then STSVN stops and reports K^{best} and $f(K^{best})$. If $h > |H|$, go to Stage 2.

Step 3: Find all feasible neighbor solutions based on move operator h . Let $K'_{h,k,k',i,i'}$ denote a feasible neighbor solution of current solution K' by applying move operator h on route k , route k' , node i , and node i' . $f(K'_{h,k,k',i,i'})$ represents the objective function value of $K'_{h,k,k',i,i'}$. If $f(K'_{h,k,k',i,i'})$ is less than $f(K')$, then add $K'_{h,k,k',i,i'}$ into M . After evaluation of all $K'_{h,k,k',i,i'}$, go to Step 4.

Step 4: If $M = \emptyset$, $h = h + 1$ and go to Step 2. If $M \neq \emptyset$, go to Step 5.

Step 5: Sort all solutions $K'_{h,k,k',i,i'}$ in M from the smallest $f(K'_{h,k,k',i,i'})$ to the largest $f(K'_{h,k,k',i,i'})$. If $f(K'_{h,k,k',i,i'}) < f(K^{best})$, $K^{best} = K'_{h,k,k',i,i'}$ and $f(K^{best}) = f(K'_{h,k,k',i,i'})$. Go to Step 6.

Step 6: If the CPU time $B \geq B_{max}$, then STSVN stops and reports K^{best} and $f(K^{best})$. Otherwise, go to Step 7.

Step 7: Create a copy of K' as K'' . Initialize $mark_k = 0, \forall k \in K''$. $mark_k$ is used to record whether route k can be replaced in K'' . If $mark_k = 0$, then route k can be replaced in K'' . If $mark_k = 1$, then route k cannot be replaced in K'' . Initialize candidate list = \emptyset . Go to Step 8.

Step 8: For each $K'_{h,k,k',i,i'}$ in M , do: If $mark_k = 0$ and $mark_{k'} = 0$, add $K'_{h,k,k',i,i'}$ into candidate list, let route k in K'' become the route k in $K'_{h,k,k',i,i'}$, and route k' in K'' become the route k' in $K'_{h,k,k',i,i'}$, $mark_k = 1$ and $mark_{k'} = 1$. Go to Step 9. In Step 8, using the new selection strategy in STSVN, the new constructed solution K'' is always better or equal to the first best neighbor solution. In addition, the marks can ensure the feasibility of K'' .

Step 9: Evaluate objective function value $f(K'')$ of K'' . If $f(K'') < f(K^{best})$, then $K^{best} = K''$, $f(K^{best}) = f(K'')$. Go to Step 10.

Step 10: If $K'' \notin \sigma$, then $K' = K''$, $f(K') = f(K'')$, add K'' to σ , $score_k = score_k + mark_k, \forall k \in K'$, $h = 1$, and go to Step 2. If $K'' \in \sigma$, find the last $K'_{h,r',i,i'}$ in candidate list, remove it from M , and go to Step 11.

Step 11: If $M = \emptyset$, $h = h + 1$ and go to Step 2. Otherwise, go to Step 6.

In Stage 2, the main purpose is to escape from the local optima by using the diversification strategy. The Stage 2 consists of Step 12 – Step 15.

Step 12: Find the minimum $score_k, \forall k \in K'$ and record the corresponding route as k_{min} . Remove all nodes (except depot) from route k_{min} . Re-insert these nodes into any routes in K' based on the least cost. Evaluate the modified K' . Update $score_k, \forall k \in K'$ based on Equation 4.1, then update $score_{k_{min}}$ of route k_{min} using Equation 4.2, and go to Step 13. These scores are considered as the memory of the algorithm and dynamically updated. The scores follow this rule: recent memory takes larger weights and previous memory takes less weights.

$$score_k = \frac{score_k}{|K'|} \quad (4.1)$$

$$score_{k_{min}} = score_{k_{min}} + |K'| \quad (4.2)$$

Step 13: Evaluate objective function value $f(K')$ of K' . If $f(K') < f(K^{best})$, then $K^{best} = K'$, $f(K^{best}) = f(K')$. Go to Step 14.

Step 14: If $K' \notin \sigma$, add K' to σ , and go to Step 1 of Stage 1. Otherwise, go to Step 15.

Step 15: Find the maximum $score_k, \forall k \in K'$ and record the corresponding route as k_{max} . Reverse the sequence of all nodes from route k_{max} . Evaluate the modified K' . Go to Step 12.

4.4.2 STSVN in Ride-sharing

The STSVN used in ride-sharing shares the same structure as described in Section 4.4.1. Several steps are slightly different. Note that the notation used in STSVN is the same as the models of ride sharing in Section 3.2. First, the move operators used in STSVN to solve the ride-sharing models are the same as the ones in Section 4.2.2. Second, a solution in ride-sharing include the set of routes K' and the list of unserved customers U' , and the objective function value $f(K', U')$ is total cost including routing cost and the penalty of unserved customers. Third, after a neighbor solution is found, the capacity constraints (3.96), the

STSVN for Disaster Relief Routing

K' = set of routes (current solution), $f(K')$ = objective function value of K' , $K^{best} = K'$, $f(K^{best}) =$ objective function value of K^{best} , tabu list $\sigma = \emptyset$, add K' into σ , $score_k = 0, \forall k \in K'$;

while $B < B_{max}$ **do**

 Start Stage 1;

$h = 1, M = \emptyset$;

while $B < B_{max}$ and $h \leq |H|$ **do**

for $k \in K'$ **do**

for $k' \in K'$ **do**

for $i \in k$ **do**

for $i' \in k'$ **do**

 use move operator h to find neighbor solution $K'_{h,k,k',i,i'}$;

if $K'_{h,k,k',i,i'}$ is feasible **then**

 evaluate objective function value $f(K'_{h,k,k',i,i'})$ of $K'_{h,k,k',i,i'}$;

if $f(K'_{h,k,k',i,i'}) < f(K')$ **then**

 add $K'_{h,k,k',i,i'}$ into M

end

end

end

end

end

end

if $M = \emptyset$ **then**

$h = h + 1$

else

 sort $K'_{h,k,k',i,i'} \in M$ from the smallest $f(K'_{h,k,k',i,i'})$ to the largest $f(K'_{h,k,k',i,i'})$;

if $f(K'_{h,k,k',i,i'}) < f(K^{best})$ **then**

$K^{best} = K'_{h,k,k',i,i'}$ and $f(K^{best}) = f(K'_{h,k,k',i,i'})$

end

 findCombination = yes;

while findCombination = yes and $B < B_{max}$ **do**

$K'' = K'$ (Create a copy of K'), $mark_k = 0, \forall k \in K''$;

 candidate list = \emptyset ;

for $K'_{h,k,k',i,i'} \in M$ **do**

if $mark_k = 0$ and $mark_{k'} = 0$ **then**

 add $K'_{h,k,k',i,i'}$ into candidate list, replace route k in K'' with the route k in

$K'_{h,k,k',i,i'}$, and replace route k' in K'' with the route k' in $K'_{h,k,k',i,i'}$,

$mark_k = 1$, and $mark_{k'} = 1$;

end

end

 evaluate objective function value $f(K'')$ of K'' ;

if $f(K'') < f(K^{best})$ **then**

$K^{best} = K''$, $f(K^{best}) = f(K'')$

end

if $K'' \notin \sigma$ **then**

$K' = K''$;

 add K'' to σ ;

 findCombination = no;

for $k \in K'$ **do**

$score_k = score_k + mark_k$

end

else

 find the last $K'_{h,k,k',i,i'}$ in candidate list, and remove it from M ;

if $M = \emptyset$ **then**

 findCombination = no, $h = h + 1$

end

end

end

end

end

 Go to Stage 2 (See Algorithm 18)

end

Return K^{best} , $f(K^{best})$

Algorithm 17: STSVN for Disaster Relief Routing

```

Stage 2 of STSVN for Disaster Relief Routing
diversification = yes;
while diversification = yes do
    find the minimum  $score_k, \forall k \in K'$  and record the corresponding route as  $k_{min}$ ;
    remove all nodes (except depot) from route  $k_{min}$ ;
    re-insert these nodes into any routes in  $K'$  based on the least cost;
    evaluate the modified  $K'$ ;
    update  $score_k, \forall k \in K'$  based on Equation 4.1, then update  $score_{k_{min}}$  of route  $k_{min}$  using Equation 4.2;
    evaluate objective function value  $f(K')$  of  $K'$ ;
    if  $f(K') < f(K^{best})$  then
        |  $K^{best} = K', f(K^{best}) = f(K')$ 
    end
    if  $K' \notin \sigma$  then
        | add  $K'$  to  $\sigma$ , diversification = no
    else
        | find the maximum  $score_k, \forall k \in K'$  and record the corresponding route as  $k_{max}$ ;
        | reverse the sequence of all nodes from route  $k_{max}$ ;
        | evaluate the modified  $K'$ 
    end
end
Go to Stage 1

```

Algorithm 18: Stage 2 of STSVN for Disaster Relief Routing

maximum requests constraints (3.97), the maximum time constraints (3.104) are checked. If the solution does not satisfy these constraints, the solution is not added into M in Step 3 as shown in Algorithm 19. If the solution does not satisfy time window constraints, the objective function value is added with additional penalty $p'\lambda$, where p' represents a large value of penalty and λ represents the number of time window constraints that are violated.

In Stage 2, the diversification strategy used for ride-sharing is different from the one for CVRP and SDVRP models. The diversification strategy used for ride-sharing is as follows: Find the minimum $score_k, k = 1, \dots, |K|$ and record the corresponding route as k_{min} . Remove all nodes (except driver's origin and destination) from route k_{min} . Find the customers' origins based on the sequence of these nodes, and find the customers' destinations based on the sequence of these nodes. Find the last customer's origin b'_r based on the sequence and find this customer's destination w'_r . Re-insert b'_r and w'_r into route k_{min} , then add all the remaining origins after w'_r , and add all the remaining destinations after all the origins based on the sequence. Evaluate the modified (K', U') . Update $score_k, \forall k = 1, \dots, |K'| + 1$ based on Equation 4.1, then update $score_{k_{min}}$ of route k_{min} using Equation 4.2. Evaluate objective function value $f(K', U')$ of (K', U') . To illustrate this strategy, a simple example is shown here: Let route $k_{min} = [1, 2, 3, 4, 5, 6, 7, 8]$, where 1 is the driver's origin, 8 is the driver's destination, 2, 3, 4 are customers' origins, and 5, 6, 7 are customers' destinations, respectively. After remove all nodes (except driver's origin and destination)

from route k_{min} , route $k_{min} = [1, 8]$. In this example, the last customer's origin b'_r based on the sequence and this customer's destination w'_r are 4 and 7, respectively. After re-insert b'_r and w'_r into route k_{min} , the route $k_{min} = [1, 4, 7, 8]$. After add all the remaining origins after w'_r , the route $k_{min} = [1, 4, 7, 2, 3, 8]$. After add all the remaining destinations after all the origins based on the sequence, the route $k_{min} = [1, 4, 7, 2, 3, 5, 6, 8]$.

STSVN for Ride-sharing

K' = set of routes, U' = unserved list, $f(K', U')$ = objective function value of (K', U') ,
 $(K', U')^{best} = (K', U')$, $f((K', U')^{best})$ = objective function value of $(K', U')^{best}$, tabu list $\sigma = \emptyset$, add
 (K', U') into σ , $score_k = 0, \forall k \in K'$, $score_{|K'|+1} = 0$ for U' ;

```

while  $B < B_{max}$  do
  Start Stage 1;
   $h = 1, M = \emptyset$ ;
  while  $B < B_{max}$  and  $h \leq |H|$  do
    use move operator  $h$  to find all neighbor solutions;
    for neighbor solution  $(K', U')_l \in M$  do
      if  $(K', U')_l$  satisfies constraints (3.96), (3.97), and (3.104) then
        | evaluate  $f(K', U')_l$  of  $(K', U')_l$ 
      end
      if  $(K', U')_l$  does not satisfy time window constraints then
        |  $f(K', U')_l = f(K', U')_l + p'\lambda$ 
      end
      if  $f(K', U')_l < f(K', U')$  then
        | add  $f(K', U')_l$  into  $M$ 
      end
    end
    if  $M = \emptyset$  then
      |  $h = h + 1$ 
    else
      sort  $(K', U')_l \in M$  from the smallest  $f(K', U')_l$  to the largest  $f(K', U')_l$ ;
      if  $f(K', U')_l < f((K', U')^{best})$  then
        |  $(K', U')^{best} = (K', U')_l$  and  $f((K', U')^{best}) = f(K', U')_l$ 
      end
      findCombination = yes;
      while findCombination = yes and  $B < B_{max}$  do
         $(K'', U'') = (K', U')$ ,  $mark_k = 0, \forall k = 1, \dots, |K''| + 1$ ;
        candidate list =  $\emptyset$ ;
        for  $(K', U')_l \in M$  do
          if  $mark_k = 0$  and  $mark_{k'} = 0$  then
            | add  $(K', U')_l$  into candidate list, replace route  $k$  in  $(K'', U'')$  with the route  $k$  in
            |  $(K', U')_l$ , and replace route  $k'$  in  $(K'', U'')$  with the route  $k'$  in  $(K', U')_l$ ,
            |  $mark_k = 1$ , and  $mark_{k'} = 1$ ;
          end
        end
        evaluate objective function value  $f(K'', U'')$  of  $(K'', U'')$ ;
        if  $f(K'', U'') < f((K', U')^{best})$  then
          |  $(K', U')^{best} = (K'', U'')$ ,  $f((K', U')^{best}) = f(K'', U'')$ 
        end
        if  $(K'', U'') \notin \sigma$  then
          |  $(K', U') = (K'', U'')$ ;
          | add  $(K'', U'')$  to  $\sigma$ ;
          | findCombination = no;
          | for  $\forall k = 1, \dots, |K''| + 1$  do
          | |  $score_k = score_k + mark_k$ 
          | end
        else
          | find the last  $(K', U')_l$  in candidate list, and remove it from  $M$ ;
          | if  $M = \emptyset$  then
          | | findCombination = no,  $h = h + 1$ 
          | end
        end
      end
    end
  end
  Go to Stage 2 (See Algorithm 20)
end
Return  $(K', U')^{best}, f((K', U')^{best})$ 

```

Algorithm 19: STSVN for Ride-sharing

```

Stage 2 of STSVN for Ride-sharing
diversification = yes;
while diversification = yes and  $B < B_{max}$  do
    find the minimum  $score_k, k = 1, \dots, |K|$  and record the corresponding route as  $k_{min}$ ;
    remove all nodes (except driver's origin and destination) from route  $k_{min}$ ;
    find the customers' origins based on the sequence of these nodes, and find the customers' destinations
    based on the sequence of these nodes;
    find the last customer's origin  $b'_r$  based on the sequence and find this customer's destination  $w'_r$ ;
    re-insert  $b'_r$  and  $w'_r$  into route  $k_{min}$ , then add all the remaining origins after  $w'_r$ , and add all the
    remaining destinations after all the origins based on the sequence;
    evaluate the modified  $(K', U')$ ;
    update  $score_k, \forall k = 1, \dots, |K'| + 1$  based on Equation 4.1, then update  $score_{k_{min}}$  of route  $k_{min}$  using
    Equation 4.2;
    evaluate objective function value  $f(K', U')$  of  $(K', U')$ ;
    if  $f(K', U') < f((K', U')^{best})$  then
        |  $(K', U')^{best} = (K', U'), f((K', U')^{best}) = f(K', U')$ 
    end
    if  $(K', U') \notin \sigma$  then
        | add  $(K', U')$  to  $\sigma$ , diversification = no
    end
end
Go to Stage 1

```

Algorithm 20: Stage 2 of STSVN for Ride-sharing

Chapter 5

Results

In this chapter, there are three main sections. In Section 5.1, numerical results from several examples are presented, out of which we try to derive insights for disaster relief routing. In Section 5.2, numerical results for ride-sharing are presented while the comparison of two algorithms (TS and STSVN) is shown in Section 5.3.

5.1 Results for Disaster Relief Routing

In this section, numerical results from several examples are shown to derive insights for disaster relief routing in terms of models, objective functions, and uncertainty. First, two simple examples adopted from Campbell et al. (2008a) are used to illustrate how different objective functions can influence the solutions for both base models and robust counterparts, and another example adopted from Huang et al. (2012) to show the difference between the CVRP and the SDVRP, especially from the relief routing perspective. For the large-scale examples, the CVRP instances from Augerat et al. (1998) and Christofides and Eilon (1969), and SDVRP instance from Chen et al. (2007, 2016) are used. The results of these examples confirm that the SDVRP models can provide more flexible solutions for relief routing when the demands from nodes are relatively large.

5.1.1 Simple Examples

Three small examples are used in this section, which are shown in Figure 5.1. In these examples, the capacity of a vehicle is set to six. For the robust counterparts, we use the symmetric parameters, i.e., $\hat{t}_{ij} = \hat{t}_{ji}$ ($\hat{t}_{01} = 100, \hat{t}_{02} = 3, \hat{t}_{03} = 5, \hat{t}_{12} = 4, \hat{t}_{13} = 11$, and

$\hat{t}_{23} = 8$). In addition, $\hat{q}_i = 1, \forall i \in N$. The commercial solver, Gurobi, is used to solve the problems.

The results are shown in Table 5.1–5.2. Here, the performance of an optimal solution compared to the other objectives are addressed, for which the abbreviations are used for the total number of vehicles needed (TV), total travel time (TT), summation of arrival times (SA), latest arrival time (LA), and summation of demand-weighted arrival times (DA). For example, the optimal solution of CVRP-minV (minimizing the number of vehicles deployed) in Example 1 is one vehicle deployed with the route (0,3,1,2,0), and this solution is evaluated with the other objectives (22 for TT, 22 for SA, 14 for LA, and 52 for DA) as shown in Table 5.1. In Example 1 and Example 2, the minV optimal solution is one vehicle deployed for both CVRP and SDVRP and we fix this ($|K| = 1$) when we solve the models with other objective functions such as minT, minS, minL, and minD. We note based on results that minS and minL objectives ensure quick deliveries at the potential expense of total travel time, which is usually desirable in disaster management. The minD considers the summation of demand-weighted arrival times, the optimal solutions of which do not necessarily provide good SA (summation of arrival times) and LA (latest arrival time) outcomes. Likewise, the minS and minL optimal solutions do not necessarily provide good solutions for each other.

In terms of the robust counterparts, $\hat{t}_{01} = \hat{t}_{10} = 100$ are relatively very large, which implies that arc (0,1) may no longer function properly in the aftermath of a disaster. From Table 5.1, the results show that edges (0,1) and (1,0) are not selected in the solutions of the RCVRP and the RSDVRP for minT. For minS, minL, and minD, edge (0,1) is still excluded in the solutions, but edge (1,0) can be selected because it does not influence the arrival time at the nodes (except the depot).

Example 3 results are shown in Table 5.2, from which we point out several observations. In this example, the CVRP requires at least three vehicles to provide a feasible solution, while the SDVRP requires as small as two vehicles (i.e., for given $|K| = 2$, a feasible solution can be obtained.) because a node is allowed to be served by more than one vehicle. Therefore, vehicles can cooperate with each other to meet the demand. If three vehicles are available in the SDVRP, i.e., $|K| = 3$, the optimal number of vehicles deployed depends on the objectives: two vehicles are deployed in SDVRP-minT while three vehicles are deployed

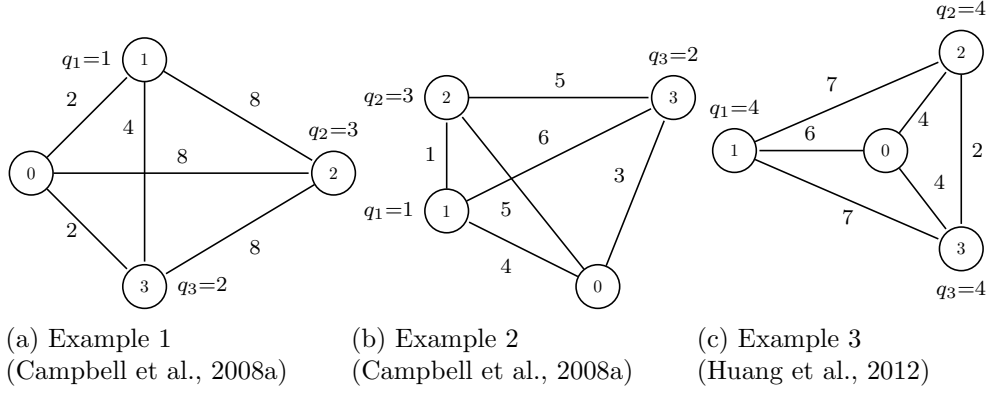


Figure 5.1: Simple Examples

Table 5.1: Results of Example 1 and Example 2

		Base Model								
		Example 1				Example 2				
Model	Route	TT	SA	LA	DA	Route	TT	SA	LA	DA
CVRP-minV	0,3,1,2,0	22	22	14	52	0,2,3,1,0	20	31	16	51
CVRP-minT	0,1,2,3,0	20	30	18	68	0,1,2,3,0	13	19	10	39
CVRP-minS	0,1,3,2,0	22	22	14	56	0,1,2,3,0	13	19	10	39
CVRP-minL	0,1,3,2,0	22	22	14	56	0,3,2,1,0	13	20	9	39
CVRP-minD	0,3,2,1,0	20	30	18	52	0,3,2,1,0	13	20	9	39
SDVRP-minT	0,1,2,3,0	20	30	18	68	0,3,2,1,0	13	20	9	39
SDVRP-minS	0,1,3,2,0	22	22	14	56	0,1,2,3,0	13	19	10	39
SDVRP-minL	0,3,1,2,0	22	22	14	52	0,3,2,1,0	13	20	9	39
SDVRP-minD	0,3,2,1,0	20	30	18	52	0,3,2,1,0	13	20	9	39
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
		Example 1				Example 2				
Model	Route	TT	SA	LA	DA	Route	TT	SA	LA	DA
RCVRP-minV	0,2,1,3,0	45	72	38	204	0,2,1,3,0	38	51	30	148
RCVRP-minT	0,3,1,2,0	45	63	34	201	0,2,1,3,0	38	51	30	148
RCVRP-minS	0,3,1,2,0	45	63	34	201	0,2,1,3,0	38	51	30	148
RCVRP-minL	0,3,1,2,0	45	63	34	201	0,3,2,1,0	130	55	26	160
RCVRP-minD	0,3,2,1,0	137	65	35	183	0,2,1,3,0	38	51	30	148
RSDVRP-minT	0,2,1,3,0	45	72	38	204	0,3,1,2,0	38	63	30	194
RSDVRP-minS	0,3,1,2,0	45	63	34	201	0,2,1,3,0	38	51	30	148
RSDVRP-minL	0,3,1,2,0	45	63	34	201	0,3,2,1,0	130	55	26	160
RSDVRP-minD	0,3,2,1,0	137	65	35	183	0,2,1,3,0	38	51	30	148

in minS, minL, and minD.

The second group in the bottom of Table 5.2 shows the robust counterpart results of Example 3 where we set \hat{t}_{ij} and \hat{q}_i at their maximum values. Unlike the SDVRP, RSDVRP now requires at least three vehicles due to the increased demand from realized uncertainty.

Table 5.2: Results of Example 3

Base Model							
	Model	Routes	TV	TT	SA	LA	DA
$ K = 3$	CVRP-minV	[0,1,0], [0,2,0], [0,3,0]	3	28	14	6	56
	CVRP-minT	[0,1,0], [0,2,0], [0,3,0]	3	28	14	6	56
	CVRP-minS	[0,1,0], [0,2,0], [0,3,0]	3	28	14	6	56
	CVRP-minL	[0,1,0], [0,2,0], [0,3,0]	3	28	14	6	56
	CVRP-minD	[0,1,0], [0,2,0], [0,3,0]	3	28	14	6	56
$ K = 2$	SDVRP-minT	[0,2,1,0], [0,2,3,0]	2	27	25	11	84
	SDVRP-minS	[0,3,2,0], [0,2,1,0]	2	27	25	11	80
	SDVRP-minL	[0,2,1,0], [0,3,1,0]	2	34	30	11	76
	SDVRP-minD	[0,2,1,0], [0,3,1,0]	2	34	30	11	76
$ K = 3$	SDVRP-minT	[0,2,1,0], [0,2,3,0]	2	27	25	11	84
	SDVRP-minS	[0,1,0], [0,2,0], [0,3,0]	3	28	14	6	56
	SDVRP-minL	[0,1,0], [0,2,0], [0,3,0]	3	28	14	6	56
	SDVRP-minD	[0,1,0], [0,2,0], [0,3,0]	3	28	14	6	56
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)							
	Model	Routes	TV	TT	SA	LA	DA
$ K = 3$	RCVRP-minV	[0,1,0], [0,2,0], [0,3,0]	3	244	122	106	610
	RCVRP-minT	[0,1,0], [0,2,0], [0,3,0]	3	244	122	106	610
	RCVRP-minS	[0,1,0], [0,2,0], [0,3,0]	3	244	122	106	610
	RCVRP-minL	[0,1,0], [0,2,0], [0,3,0]	3	244	122	106	610
	RCVRP-minD	[0,1,0], [0,2,0], [0,3,0]	3	244	122	106	610
$ K = 3$	RSDVRP-minT	[0,2,0], [0,2,1,3,0], [0,3,1,2,0]	3	104	142	38	237
	RSDVRP-minS	[0,3,0], [0,2,0], [0,2,1,0]	3	156	41	18	170
	RSDVRP-minL	[0,2,1,0], [0,2,3,0], [0,2,0]	3	164	56	18	210
	RSDVRP-minD	[0,3,0], [0,2,0], [0,2,1,0]	3	156	41	18	170

It is clear from Table 5.2 that the optimal solutions of RSDVRP models can avoid using edge (0, 1) having $\hat{t}_{01} = 100$ by visiting nodes 2 and 3 more than once, while RCVRP has no such capability (the CVRP and RCVRP optimal routes are the same) resulting in the great increase in the objective function values. This is an exemplary case showing the SDVRP and its robust counterpart can be much more effective and useful in disaster relief routing by providing more flexibility.

5.1.2 Large Examples

While using small and simple examples to illustrate the difference across CVRP, SDVRP, and their robust counterparts especially from the disaster relief routing perspective, larger CVRP examples are taken from Augerat et al. (1998) and Christofides and Eilon (1969), and SDVRP example from Chen et al. (2007, 2016) in this section to derive further managerial

insights. In particular, examples named $A - n32 - k5$ (Augerat et al., 1998), $A - n38 - k5$ (Augerat et al., 1998), $A - n44 - k6$ (Augerat et al., 1998), $A - n55 - k9$ (Augerat et al., 1998), $A - n69 - k9$ (Augerat et al., 1998), $E - n76 - k14$ (Christofides and Eilon, 1969), $A - n80 - k10$ (Augerat et al., 1998), $E - n101 - k8$ (Christofides and Eilon, 1969), $E - n101 - k14$ (Christofides and Eilon, 1969), and SD2 (Chen et al., 2007, 2016) are used. The number of nodes in these examples are 32, 38, 44, 55, 69, 76, 80, 101, 101, and 17, respectively.

Table 5.3: CVRP and SDVRP Results: $A - n32 - k5$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	5	865	3228	228	44154	5	784	3257	231	39164
CVRP-minS	5	976	2213	126	28775	5	975	2192	126	28451
CVRP-minL	5	1032	2555	134	32147	5	964	2571	131	35862
CVRP-minD	5	957	2382	157	28567	5	975	2306	164	28109
SDVRP-minT	5	796	2799	200	36272	5	796	2807	174	34516
SDVRP-minS	5	923	2226	134	28489	5	975	2192	126	28451
SDVRP-minL	5	1011	2635	138	33896	5	988	2584	130	34465
SDVRP-minD	5	938	2233	135	27921	5	938	2233	135	27921
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	5	1070	3530	226	47546	5	1058	3802	255	53259
RCVRP-minS	5	1178	2994	188	41322	5	1206	2943	186	40495
RCVRP-minL	5	1317	3798	215	54047	5	1304	3316	173	46389
RCVRP-minD	5	1184	3144	200	40563	5	1182	3130	206	39745
RSDVRP-minT	5	1032	3738	250	48645	5	1051	3469	244	47172
RSDVRP-minS	5	1189	2999	174	41048	5	1162	2950	196	41631
RSDVRP-minL	5	1257	3220	169	43929	5	1185	3192	164	43909
RSDVRP-minD	5	1240	3148	205	40192	5	1189	3117	206	39230

For these examples, heuristic algorithms, TS and STSVN presented in Chapter 4, are used to obtain near-optimal solutions within the given time limit, respectively. The capacity of each vehicle for $A - n32 - k5$, $A - n38 - k5$, $A - n44 - k6$, $A - n55 - k9$, $A - n69 - k9$, $E - n76 - k14$, and $A - n80 - k10$ is set to 100 while the one for $E - n101 - k8$ is set to 200 and the one for $E - n101 - k14$ is set to 112. For each example, the deterministic models and robust models of CVRP-minT, CVRP-minS, CVRP-L, CVRP-minD, SDVRP-minT, SDVRP-minS, SDVRP-minL, and SDVRP-minD are solved. Note that CVRP-minV and SDVRP-minV are not discussed for these examples because many solutions are the optimal

Table 5.4: CVRP and SDVRP Results: $A - n38 - k5$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	5	733	2798	155	35153	5	735	2664	157	31980
CVRP-minS	5	856	2138	152	27363	5	825	2084	152	26867
CVRP-minL	5	1000	3479	172	43657	5	806	2506	116	33302
CVRP-minD	5	876	2488	164	29925	5	856	2244	158	25978
SDVRP-minT	5	745	2846	162	35410	5	734	2651	192	30843
SDVRP-minS	5	884	2259	156	30493	5	850	2192	132	29028
SDVRP-minL	5	909	2764	129	35571	5	893	2664	123	34512
SDVRP-minD	5	832	2382	141	27837	5	872	2262	153	26609

Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	6	1088	3623	221	49686	6	1066	3538	221	46459
RCVRP-minS	6	1222	3005	166	40731	6	1220	2921	155	40953
RCVRP-minL	6	1547	4749	230	66234	6	1344	3538	162	48933
RCVRP-minD	6	1242	3217	187	39911	6	1274	3116	169	38874
RSDVRP-minT	6	1108	3653	223	45675	6	1104	3745	196	46295
RSDVRP-minS	6	1225	2934	161	40416	6	1241	2943	159	39620
RSDVRP-minL	6	1253	3228	147	43612	6	1253	3208	143	42603
RSDVRP-minD	6	1293	3236	174	40110	6	1247	3147	166	38080

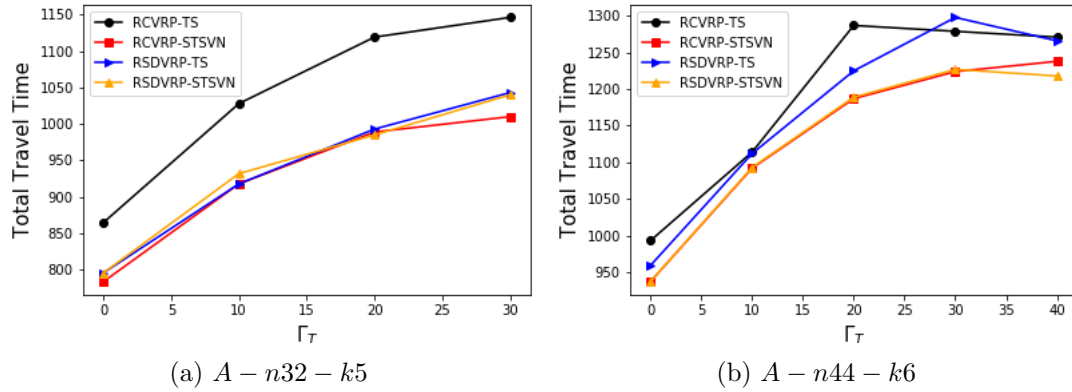


Figure 5.2: Robust Model Results of minT with Γ_T

solutions for these two objective functions in large examples and these solutions can be easily obtained from insertion algorithms. For the examples with number of nodes less than 100, we set the CPU time limit = 3000 seconds for solving each model with an objective. For the examples with number of nodes more than 100, we set CPU time limit = 6000 seconds for solving each model with an objective. For the robust counterpart parameters, we randomly

Table 5.5: CVRP and SDVRP Results: $A - n44 - k6$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	6	993	3508	189	43566	6	937	3322	164	45299
CVRP-minS	6	1166	2741	159	36311	6	1097	2579	158	33117
CVRP-minL	6	1178	3911	194	54150	6	1196	3412	140	44700
CVRP-minD	6	1251	2841	182	34595	6	1121	2687	145	33083
SDVRP-minT	6	959	3316	197	41292	6	937	3896	227	48273
SDVRP-minS	6	1102	2723	146	35355	6	1051	2578	122	32504
SDVRP-minL	6	1201	3496	140	45812	6	1099	2877	118	38112
SDVRP-minD	6	1136	2806	161	34116	6	1148	2940	137	33183

Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	7	1354	4588	221	63910	7	1308	3937	203	53028
RCVRP-minS	7	1514	3414	154	47851	7	1454	3406	151	48160
RCVRP-minL	7	1606	5079	299	69953	7	1608	3963	160	55412
RCVRP-minD	7	1524	3535	161	47331	7	1499	3553	161	46734
RSDVRP-minT	7	1315	4507	257	58171	7	1310	4511	221	58275
RSDVRP-minS	7	1590	3500	166	50124	7	1500	3418	165	47431
RSDVRP-minL	7	1619	3875	160	54103	7	1573	3810	149	52556
RSDVRP-minD	7	1560	3547	165	48105	7	1494	3553	161	46738

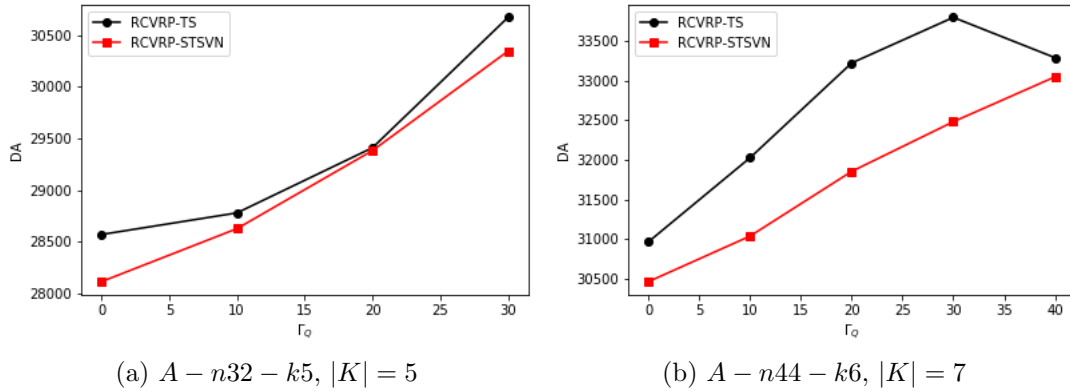


Figure 5.3: Robust Model Results of minD with Γ_Q

generate \hat{t}_{ij} following $U(0, 20)$ where U represents the uniform distribution, and set $\hat{q}_i = 1$. The results of these examples are summarized in Tables 5.3 – 5.11. As the solutions are obtained by heuristic algorithms TS and STSVN within limited time, the solutions are not always optimal solutions. In this section, the analyses of results focus on the insights for disaster relief routing in terms of models, objective functions, and uncertainty. The

Table 5.6: CVRP and SDVRP Results: $A - n55 - k9$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	9	1112	3877	193	55748	9	1074	3453	120	53545
CVRP-minS	9	1237	2743	105	42322	9	1241	2639	149	43040
CVRP-minL	9	1363	3780	133	60968	9	1367	3095	102	49670
CVRP-minD	9	1286	2767	129	41685	9	1220	2662	98	41278
SDVRP-minT	9	1103	3580	144	52587	9	1073	3451	118	51112
SDVRP-minS	9	1193	2647	92	42859	9	1194	2619	94	42042
SDVRP-minL	9	1270	3087	101	48473	9	1264	3255	97	47120
SDVRP-minD	9	1308	2833	112	40963	9	1295	2833	121	41838
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	10	1591	4613	182	72182	10	1562	4597	209	74335
RCVRP-minS	10	1783	3915	151	67387	10	1836	3856	146	65647
RCVRP-minL	10	2053	5796	240	93173	10	2014	4941	152	77687
RCVRP-minD	10	1888	4140	143	64352	10	1886	4013	150	60937
RSDVRP-minT	10	1588	4606	227	73412	10	1542	4479	189	73755
RSDVRP-minS	10	1856	3882	143	64911	10	1720	3790	135	63788
RSDVRP-minL	10	1795	4197	132	71536	10	1829	4342	130	77650
RSDVRP-minD	10	1823	4150	147	61546	10	1743	4064	145	61743

performance of heuristic algorithms TS and STSVN are compared in the third section in this chapter.

The impacts of CVRP and SDVRP models with the same objective function are evaluated and summarized in Table 5.12. For the sake of convenience, the objective function value of a solution from CVRP with objective function b is denoted by $b(CVRP)$ and the objective function value of a solution from SDVRP with objective function b is denoted by $b(SDVRP)$. For the same example, the difference between $b(CVRP)$ and $b(SDVRP)$ is computed as $b(CVRP) - b(SDVRP)$, which is represented by Diff in Table 5.12. The percentage of the difference between $b(CVRP)$ and $b(SDVRP)$ is computed as $\frac{b(CVRP) - b(SDVRP)}{b(CVRP)}$, which is represented by % in Table 5.12. Based on the results summarized in Table 5.12, the SDVRP results are on average better than the ones of CVRP with respect to objective function values. The solutions obtained from TS show larger difference between CVRP and SDVRP models, while the solutions obtained from STSVN show smaller difference between CVRP and SDVRP models. These results arise from the limitation of heuristics and the

Table 5.7: CVRP and SDVRP Results: $A - n69 - k9$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	9	1168	4879	176	58623	9	1169	4846	189	57087
CVRP-minS	9	1286	3663	106	43773	9	1288	3732	118	46977
CVRP-minL	9	1621	5346	154	66643	9	1701	5865	150	71341
CVRP-minD	9	1329	3822	133	44372	9	1291	3817	116	42226
SDVRP-minT	9	1174	4562	156	57358	9	1199	4749	172	60032
SDVRP-minS	9	1293	3668	114	46930	9	1297	3708	134	45882
SDVRP-minL	9	1567	5400	136	62234	9	1567	5430	134	59100
SDVRP-minD	9	1380	4033	131	44550	9	1305	3938	124	43411
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	10	1820	6913	206	89400	10	1779	6381	188	86597
RCVRP-minS	10	2044	5793	181	78672	10	2022	5681	199	77425
RCVRP-minL	10	2316	8300	293	107556	10	2351	7778	189	105796
RCVRP-minD	10	2105	6240	207	71211	10	2122	6282	182	71759
RSDVRP-minT	10	1824	7296	231	91244	10	1786	7038	209	89559
RSDVRP-minS	10	2071	5891	174	76173	10	2045	5707	171	76698
RSDVRP-minL	10	2183	7683	191	95166	10	2245	7873	185	96868
RSDVRP-minD	10	2224	6628	224	72124	10	2048	5933	176	70197

characteristics of the example (not intended to compare the SDVRP vs CVRP).

Figure 5.2 shows how the total travel time increases as the degree of robustness, which can be adjusted by changing Γ_T values, increases. As more arcs are subject to uncertain travel times with larger Γ_T , more realized \hat{t}_{ij} 's are considered, resulting in more robust solutions with increased travel times. For the solutions obtained using TS, the RSDVRP results are up to 11% (10% on average) smaller than those of the RCVRP in $A - n32 - k5$, up to 5% (1% on average) smaller in $A - n44 - k6$. For the solutions obtained using STSVN, there is no significant difference between the RSDVRP results and the RCVRP results in $A - n32 - k5$ and $A - n44 - k6$.

Figure 5.3 shows how the summation of demand-weighted arrival times (DA) increases as the degree of robustness, which can be adjusted by changing Γ_Q values, increases. As more nodes are subject to uncertain demand with larger Γ_Q , more realized \hat{q}_i 's are considered, resulting in more robust solutions with increased demands. As shown in Figure 5.3, the solutions obtained from STSVN are better than the ones from TS in terms of DA. For

Table 5.8: CVRP and SDVRP Results: $E - n76 - k14$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	14	1072	2897	79	51903	14	1046	2968	98	53002
CVRP-minS	14	1230	2125	62	37681	14	1194	2041	62	35924
CVRP-minL	14	1292	3230	107	56525	14	1405	3077	82	55770
CVRP-minD	14	1216	2185	66	36762	14	1193	2117	55	35998
SDVRP-minT	14	1057	3187	93	51573	14	1046	2922	103	49462
SDVRP-minS	14	1191	2327	70	41821	14	1193	2353	70	40447
SDVRP-minL	14	1156	3173	81	50320	14	1513	4691	91	64855
SDVRP-minD	14	1205	2453	68	40026	14	1328	2731	91	39670
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	15	1722	4567	130	83115	15	1678	4223	129	78037
RCVRP-minS	15	1862	3609	117	67041	15	1723	3503	100	64980
RCVRP-minL	15	2051	5074	152	96307	15	2056	5372	120	98943
RCVRP-minD	15	1963	3922	121	66665	15	1780	3598	95	62143
RSDVRP-minT	15	1669	4867	179	75618	15	1589	4412	133	76235
RSDVRP-minS	15	1837	3580	110	66570	15	1801	3622	106	67572
RSDVRP-minL	15	1966	4822	121	77023	15	2102	5531	114	85854
RSDVRP-minD	15	1894	3926	110	64322	15	1889	3799	106	64308

example $A - n32 - k5$, the number of vehicles used (TV) is the same as TV from deterministic model. Using the same TV, DA increases 1.83%, 4.53%, 7.97% when $\Gamma_Q = 10, 20, 30$ for the solutions obtained from STSVN. For example $A - n44 - k6$, the number of vehicles used (TV) is set as seven in Figure 5.3. Using the same TV, DA increases 1.88%, 4.56%, 6.61%, and 8.47% when $\Gamma_Q = 10, 20, 30$, and 40 for the solutions obtained from STSVN. As show in Table 5.5, more realized \hat{q}_i 's are considered in the model, the TV is expected to increase to ensure the feasibility of the solutions.

To show how the solutions of the SDVRP are influenced by different objective functions and different values of Γ_T and Γ_Q , a benchmark instance of the SDVRP called SD2 in Chen et al. (2007, 2016) is used in this section, where there are 16 customer nodes and 1 depot. The capacity of each vehicle is set to 100. The demand from each customer is set to 60 for the nodes having an odd number index and 90 for the ones with an even number index. If this benchmark instance is considered as CVRP, then there is only one feasible solution: each customer node is served by one vehicle and the number of vehicles needed is 16 for all

Table 5.9: CVRP and SDVRP Results: $A - n80 - k10$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	10	1931	8344	226	94622	10	1791	7318	219	80443
CVRP-minS	10	2249	6509	157	73868	10	2214	6048	135	68488
CVRP-minL	10	2262	8406	181	97686	10	2288	8356	169	96144
CVRP-minD	10	2232	6299	171	68617	10	2148	6211	143	67696
SDVRP-minT	10	1850	8005	199	86300	10	1776	7526	182	91620
SDVRP-minS	10	2116	6672	211	76687	10	2168	6160	141	69221
SDVRP-minL	10	2171	8058	160	85178	10	2232	8327	172	87103
SDVRP-minD	10	2183	6703	175	68801	10	2185	6140	142	67365
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	11	2601	10038	271	131795	11	2569	9823	262	119681
RCVRP-minS	11	2943	8293	190	101292	11	2848	8153	184	99998
RCVRP-minL	11	3236	11177	246	146671	11	3190	10086	215	129589
RCVRP-minD	11	3128	8897	225	102904	11	3001	8428	209	96504
RSDVRP-minT	11	2524	10713	314	123318	11	2593	10784	299	127272
RSDVRP-minS	11	3042	8783	212	108314	11	2991	8298	202	101999
RSDVRP-minL	11	3106	10311	200	116203	11	3141	11325	209	131987
RSDVRP-minD	11	3068	9935	284	103769	11	3027	8604	205	98117

objectives. If this benchmark instance is considered as SDVRP, then the minimum number of vehicles needed can be reduced to as small as 12, depending on the objectives. As shown in Table 5.13, the optimal solutions for the different objectives in the SDVRP are better than or equal to those of corresponding CVRP.

The optimal solution of SDVRP-minV for SD2 is 12 vehicles while the CVRP-minV optimal solution is 16. That is, it needs at least 12 vehicles to find a feasible solution for SDVRP and 16 vehicles for CVRP. Accordingly, we consider two scenarios for SDVRP-minS, -minL, and -minD: scenario 1 with 12 available vehicles and scenario 2 with 16 available vehicles. Other than the number of available vehicles, the two scenarios are the same. For each scenario, the impacts of different robust parameter values for Γ_T and Γ_Q on different objective functions are evaluated and the metrics (TT, SA, LA, DA) are computed for each obtained solution. For the sake of convenience, an optimal solution for objective function b , number of available vehicles $|K|$, and robust parameter Γ_T is denoted by $R(b, |K|, \Gamma_T)$. In addition, $R(b, |K|, \Gamma_T)$ evaluated with respect to TT, SA, LA,

Table 5.10: CVRP and SDVRP Results: $E - n101 - k8$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	8	939	6329	150	88313	8	823	5223	130	74013
CVRP-minS	8	1061	5091	158	76397	8	1063	4683	110	69136
CVRP-minL	8	1176	6194	123	88814	8	1176	6194	123	88814
CVRP-minD	8	1122	5165	149	75669	8	1101	4916	139	67114
SDVRP-minT	8	884	5057	140	76529	8	848	5250	119	76532
SDVRP-minS	8	1043	4569	135	67462	8	954	4252	115	62619
SDVRP-minL	8	1100	5405	128	79524	8	1060	6133	122	88272
SDVRP-minD	8	984	4682	134	64764	8	1009	4388	111	59304
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	8	1594	10093	244	155502	8	1590	10095	234	152899
RCVRP-minS	8	1732	9367	204	147389	8	1758	8849	203	138941
RCVRP-minL	8	1849	11598	217	180870	8	1849	11598	217	180870
RCVRP-minD	8	2050	10025	295	135054	8	1892	9305	221	123905
RSDVRP-minT	8	1532	8760	230	133472	8	1520	9013	287	139502
RSDVRP-minS	8	1693	7953	206	123040	8	1666	7741	183	120723
RSDVRP-minL	8	1740	9182	206	140480	8	1921	11476	209	180572
RSDVRP-minD	8	1711	8650	222	119206	8	1697	8346	186	113103

and DA are denoted by $TT(b, |K|, \Gamma_T)$, $SA(b, |K|, \Gamma_T)$, $LA(b, |K|, \Gamma_T)$, and $DA(b, |K|, \Gamma_T)$, respectively. Using $R(b, |K|, 0)$ as the reference, the percentage of the difference between $R(b, |K|, 0)$ and $R(b, |K|, \Gamma_T)$, $\Gamma_T > 0$ for TT is computed as follows.

$$p(TT, b, |K|, \Gamma_T) = \frac{TT(b, |K|, \Gamma_T) - TT(b, |K|, 0)}{TT(b, |K|, 0)} \times 100 \quad (5.1)$$

Likewise, we define $p(SA, b, |K|, \Gamma_T)$, $p(LA, b, |K|, \Gamma_T)$, and $p(DA, b, |K|, \Gamma_T)$ in the same fashion.

The scenario 1 results are shown in Figure 5.4. For the minT objective (shown in the upper left), TT increases when Γ_T increases as expected. The SA, LA, and DA results vary as they are not directly relevant to minT but LA (latest arrival time) increases abruptly at $\Gamma_T = 4$, implying that the minT objective may not be suitable for relief routing purposes even for SDVRP if travel time uncertainty is considered. For minS (shown in the upper right), SA, LA, and DA remain the same values when $\Gamma_T = 0, \dots, 4$. SA increases up to 8.9% as Γ_T increases. For minL (shown in the bottom left), LA remains the same value

Table 5.11: CVRP and SDVRP Results: $E - n101 - k14$

Base Model										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
CVRP-minT	14	1163	4296	90	59881	14	1157	4615	116	64854
CVRP-minS	14	1400	3720	107	57649	14	1296	3055	73	45323
CVRP-minL	14	1500	4852	101	70081	14	1599	5007	99	73076
CVRP-minD	14	1440	3810	102	50224	14	1351	3460	84	45483
SDVRP-minT	14	1165	4272	129	60758	14	1119	4554	99	62353
SDVRP-minS	14	1272	3386	74	49379	14	1264	3128	68	45625
SDVRP-minL	14	1288	4415	108	58601	14	1414	4987	119	65413
SDVRP-minD	14	1348	3786	123	48929	14	1322	3311	82	43726
Robust Counterpart, Worst Case (\hat{t}_{ij} and \hat{q}_i at their max values)										
Model	TS					STSVN				
	TV	TT	SA	LA	DA	TV	TT	SA	LA	DA
RCVRP-minT	15	2207	7419	196	116564	15	1934	6964	172	102808
RCVRP-minS	15	2148	5528	118	82725	15	2016	5332	118	81794
RCVRP-minL	15	2207	7419	196	116564	15	2412	7396	151	113604
RCVRP-minD	15	2315	6120	140	84993	15	2089	5656	123	77374
RSDVRP-minT	15	1880	6178	133	90715	15	1824	5952	137	91902
RSDVRP-minS	15	2046	5373	120	82458	15	2093	5056	125	78960
RSDVRP-minL	15	2206	6786	129	94314	15	2274	6919	143	93269
RSDVRP-minD	15	2161	6034	148	80335	15	2034	5481	112	76513

when $\Gamma_T = 0, \dots, 8$. LA increases 7.5% when $\Gamma_T = 10$, and then remains the same value when $\Gamma_T = 10, \dots, 16$. TT, SA, and DA vary as Γ_T increases but they all remain within 10% range, which implies that the minL objective provides overall effective solution for all the metrics for this particular example. Interestingly, TT and SA at $\Gamma_T = 2$ are less than the ones at $\Gamma_T = 0$, respectively. For minD (shown in the bottom right), SA and LA increase up to 8%, and decrease significantly at some Γ_T values, e.g., at $\Gamma_T = 6, 8$. In scenario 1 where 12 vehicles are available, all feasible solutions utilize the full capacity of vehicles, which implies that no feasible solution can be found with $\Gamma_Q > 0$.

The scenario 2 SDVRP results are shown in Figure 5.5. For minT objective (upper left), the optimal solutions obtained with various Γ_T are the same as the ones in scenario 1, which implies that the number of available vehicles does not influence the minT objective in this example. For minS, minL, and minD, the increased number of available vehicles can be used to reduce the delivery time. For minS (upper right), SA increases up to 5%, and SA, LA, and DA remain the same when $\Gamma_T = 0, \dots, 4$. However, there is an abrupt increase

Table 5.12: Comparison of CVRP and SDVRP Models

		Base Model				Robust Counterpart, Worst Case			
		TS		STSVN		TS		STSVN	
		Diff	%	Diff	%	Diff	%	Diff	%
<i>A – n32 – k5</i>	minT	69	7.98	-12	-1.53	38	3.55	7	0.66
	minS	-13	-0.59	0	0.00	-5	-0.17	-7	-0.24
	minL	-4	-2.99	1	0.76	46	21.40	9	5.20
	minD	646	2.26	188	0.67	371	0.91	515	1.30
<i>A – n38 – k5</i>	minT	-12	-1.64	1	0.14	-20	-1.84	-38	-3.56
	minS	-121	-5.66	-108	-5.18	71	2.36	-22	-0.75
	minL	43	25.00	-7	-6.03	83	36.09	19	11.73
	minD	2088	6.98	-631	-2.43	-199	-0.50	794	2.04
<i>A – n44 – k6</i>	minT	34	3.42	0	0.00	39	2.88	-2	-0.15
	minS	18	0.66	1	0.04	-86	-2.52	-12	-0.35
	minL	54	27.84	22	15.71	139	46.49	11	6.88
	minD	479	1.38	-100	-0.30	-774	-1.64	-4	-0.01
<i>A – n55 – k9</i>	minT	9	0.81	1	0.09	3	0.19	20	1.28
	minS	96	3.50	20	0.76	33	0.84	66	1.71
	minL	32	24.06	5	4.90	108	45.00	22	14.47
	minD	722	1.73	-560	-1.36	2806	4.36	-806	-1.32
<i>A – n69 – k9</i>	minT	-6	-0.51	-30	-2.57	-4	-0.22	-7	-0.39
	minS	-5	-0.14	24	0.64	-98	-1.69	-26	-0.46
	minL	18	11.69	16	10.67	102	34.81	4	2.12
	minD	-178	-0.40	-1185	-2.81	-913	-1.28	1562	2.18
<i>E – n76 – k14</i>	minT	15	1.40	0	0.00	53	3.08	89	5.30
	minS	-202	-9.51	-312	-15.29	29	0.80	-119	-3.40
	minL	26	24.30	-9	-10.98	31	20.39	6	5.00
	minD	-3264	-8.88	-3672	-10.20	2343	3.51	-2165	-3.48
<i>A – n80 – k10</i>	minT	81	4.19	15	0.84	77	2.96	-24	-0.93
	minS	-163	-2.50	-112	-1.85	-490	-5.91	-145	-1.78
	minL	21	11.60	-3	-1.78	46	18.70	6	2.79
	minD	-184	-0.27	331	0.49	-865	-0.84	-1613	-1.67
<i>E – n101 – k8</i>	minT	55	5.86	-25	-3.04	62	3.89	70	4.40
	minS	522	10.25	431	9.20	1414	15.10	1108	12.52
	minL	-5	-4.07	1	0.81	11	5.07	8	3.69
	minD	10905	14.41	7810	11.64	15848	11.73	10802	8.72
<i>E – n101 – k14</i>	minT	-2	-0.17	38	3.28	327	14.82	110	5.69
	minS	334	8.98	-73	-2.39	155	2.80	276	5.18
	minL	-7	-6.93	-20	-20.20	67	34.18	8	5.30
	minD	1295	2.58	1757	3.86	4658	5.48	861	1.11
Average		372.11	4.35	105.64	-0.65	708.50	9.02	316.19	2.52

of LA after $\Gamma_T = 6$. For minL (bottom left), LA does not increase until $\Gamma_T = 10$, and has the same increased value of 7.5% for $\Gamma_T = 12, \dots, 16$. SA decreases up to 17% with different values of Γ_T , which indicates that minL objective may not provide good solutions for the

Table 5.13: Results of SD2 in CVRP and SDVRP

Base Model					
Model	TV	TT	SA	LA	DA
CVRP (all objective functions)	16	800	400	40	30000
SDVRP-minT	12	708.28	788.28	50	35131.2
SDVRP-minS	16	800	400	40	30000
SDVRP-minL	14	920	700	40	30000
SDVRP-minD	14	920	700	40	30000

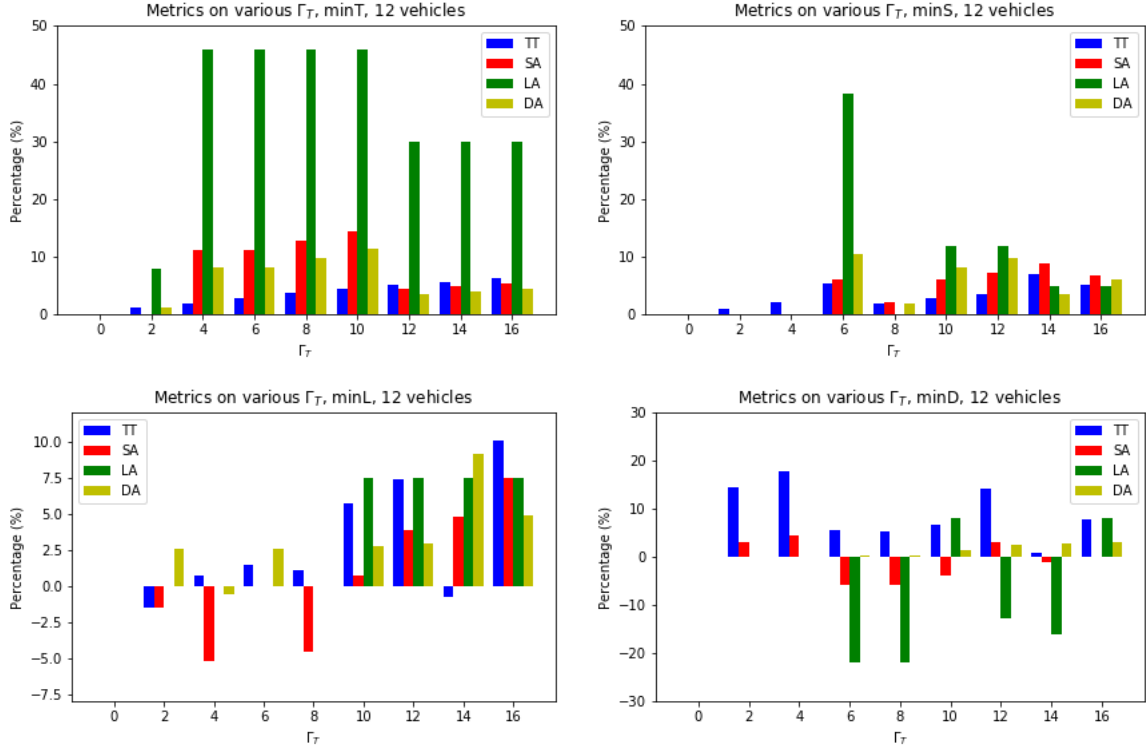


Figure 5.4: Scenario 1 Results for Various Γ_T

SA metric. The minD results (bottom right) show that DA increase very little even for $\Gamma_T = 16$, which is consistent with scenario 1 results.

In scenario 2, feasible solutions can be found even when $\Gamma_Q > 0$ (implying larger realized demand) due to the increased number of vehicles available. Figure 5.6 summarizes how the metrics change as Γ_Q increases for SDVRP-minT, -minS, -minL, and -minD, which shows that TT, SA, LA are not sensitive to Γ_Q because only travel time is considered in such metrics. As expected, DA increases (up to 10%) as Γ_Q increases for minS, minL, and min D but, interestingly, not for minT (as shown in the upper left corner of Figure 5.6).

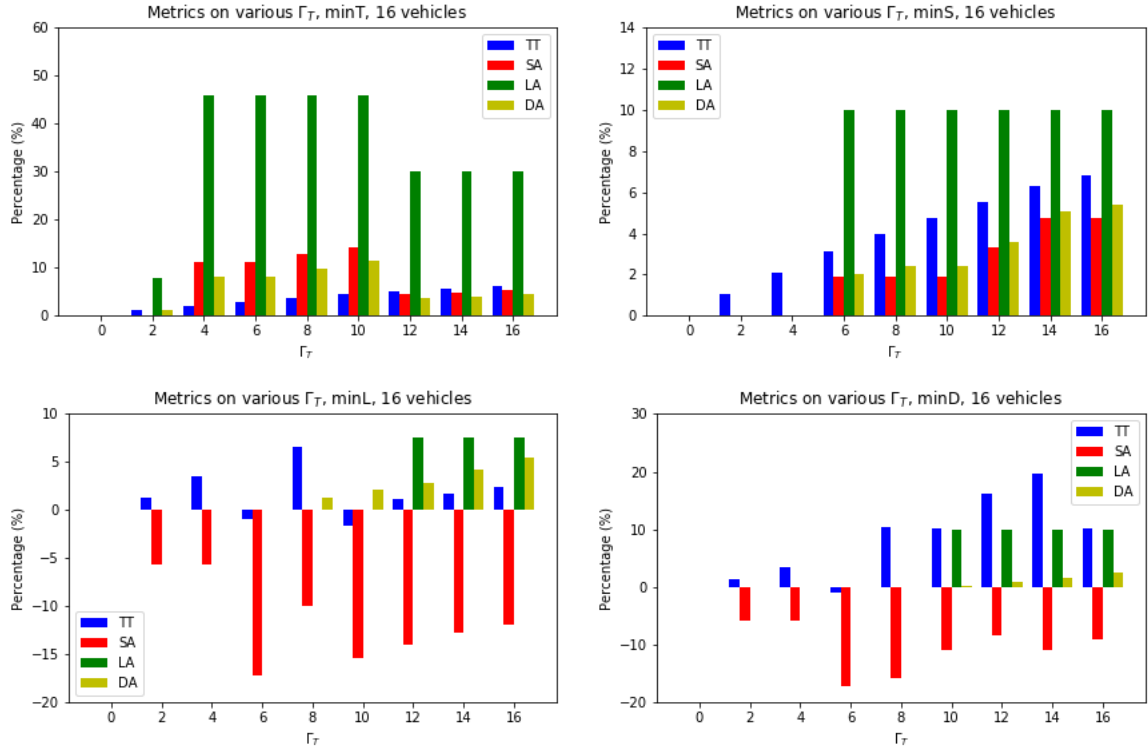


Figure 5.5: Scenario 2 Results for Various Γ_T

This again implies that the traditional minT objective does not provide good solutions for humanitarian relief related metrics such as SA, LA, and DA.

In summary, SDVRP is a better choice than CVRP in terms of the metrics for humanitarian relief vehicle routing, especially when the demand of a node is relatively large compared with the vehicle capacity.

5.2 Results for Ride-sharing

In this section, the numerical results of several examples from Augerat et al. (1998) and Christofides and Eilon (1969) are used to derive insights for ride-sharing in terms of the influence of different parameters and uncertain travel time on the solution obtained from the models. First, example $E - n22 - k4$ from Christofides and Eilon (1969) is used to illustrate how maximum travel time that a driver is willing to spend (t_k), maximum customer requests that a driver is willing to serve (m_k), number of available seats in a vehicle (C_k), time window parameters ($e_{b_r}, l_{b_r}, e_{w_r}, l_{w_r}, e_{b_k}, l_{b_k}, e_{w_k}, l_{w_k}$) influence the matching and routing decisions

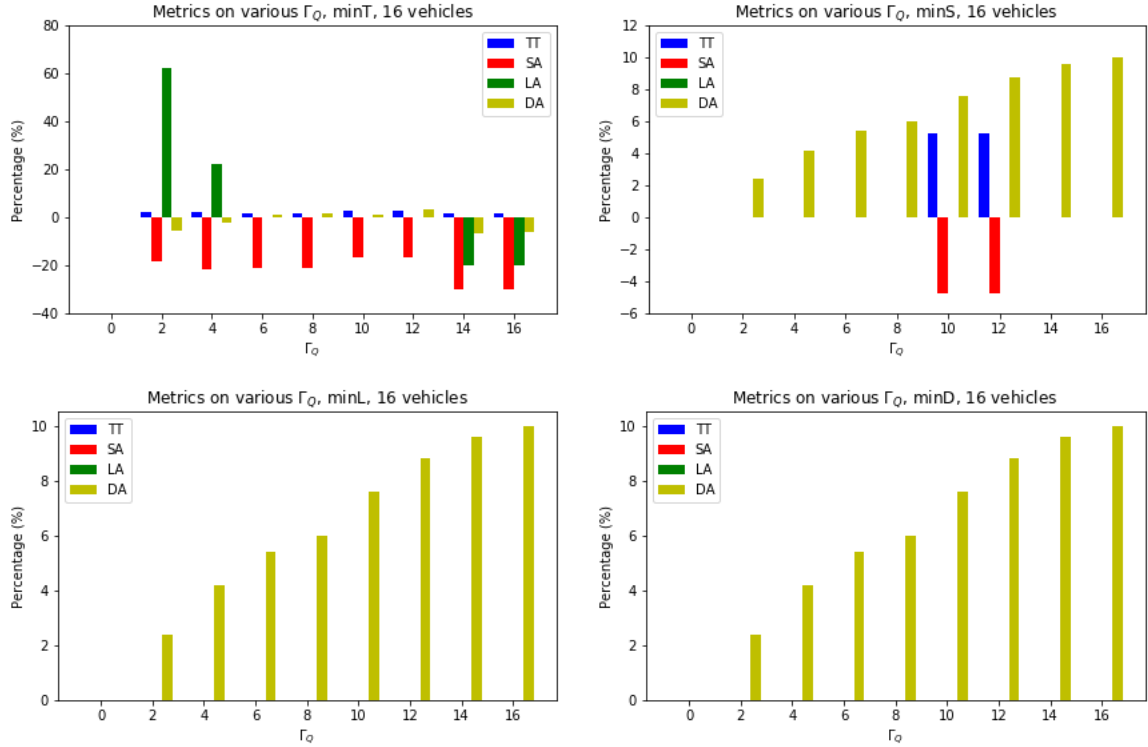


Figure 5.6: Scenario 2 Results for Various Γ_Q

for ride-sharing. Then different examples are used to show how the objective value changes in solutions as the degree of uncertainty, Γ_k , increases in the robust counterpart. Here Γ_k represents, for each driver k , the number of arcs subject to uncertainty. Therefore, if $\Gamma_1 = 3$, it implies that driver 1 will face up to 3 arcs that are subject to uncertainty during his/her course from home to the destination. If $\Gamma_k = 0, \forall k$, then the problem will become deterministic (same as the nominal case). If we allow greater Γ_k , then there will be more arcs with a delay. Consequently, the objective function value will increase. For the larger sizes of examples, heuristic algorithms, TS and STSVN presented in Chapter 4, are used to obtain near-optimal solutions within the given time limit, respectively. In this section, the analyses of results focus on the insights for ride-sharing in terms of the influence of different parameters. The performance of heuristic algorithms TS and STSVN for ride-sharing are compared in Section 5.3. The benchmark instances used in this section are summarized in Table 5.14. For each benchmark instance, the coordinates of the nodes are used to calculate the travel distance between each pair of nodes (in the unit of kilometers), then let the

nominal travel time, \bar{t}_{ij} , be the same as travel distance by assuming the average vehicle speed is fixed as 60 km/hour (Then, it takes 1 minute to travel 1 kilometer.) for the sake of simplicity. The capacity of each vehicle (excluding driver seat) is 4. In each example, the origins and destinations of drivers and riders are different. The parameters used in this section are corresponding to the ride-sharing models in Section 3.2.1. For an example with number of drivers $|K|$ and number of customer requests $|R|$, the origin of driver k is node k , the destination of driver k is node $k + |K|$, the origin of customer request r is node $r + 2|K|$, the destination of customer request r is node $r + 2|K| + |R|$. In this scenario, $p_r = 1000$, $e_{b_r} = 0$, $l_{b_r} = 1000$, $e_{w_r} = 0$, $l_{w_r} = 1000, \forall r \in R$. $q_r = 2$, when $r = 1, 2, 3$. $q_r = 1$, when $r = 4, \dots, |R|$. For drivers, $C_k = 4$, $m_k = 4$, $t_k = 1000$, $e_{b_k} = 0$, $l_{b_k} = 1000$, $e_{w_k} = 0$, $l_{w_k} = 1000, \forall k \in K$. The coefficient f to convert travel time into travel cost is assumed to be 1.

Table 5.14: Number of Drivers and Customer Requests in Examples

Example	Number of nodes	$ K $	$ R $
$E - n22 - k4$	22	4	7
$A - n32 - k5$	32	5	11
$A - n36 - k5$	36	5	13
$A - n44 - k6$	44	6	16
$A - n48 - k7$	48	6	18
$A - n54 - k7$	54	7	20
$A - n60 - k9$	60	9	21
$A - n64 - k9$	64	9	23
$E - n76 - k7$	76	10	28
$A - n80 - k10$	80	10	30

To illustrate how different maximum travel time that a driver is willing to spend (t_k), maximum customer requests that a driver is willing to serve (m_k), number of available seats in a vehicle (C_k), time window parameters ($e_{b_r}, l_{b_r}, e_{w_r}, l_{w_r}, e_{b_k}, l_{b_k}, e_{w_k}, l_{w_k}$) influence the matching and routing decisions, example $E - n22 - k4$ is used. In this example, there are four drivers and seven customer requests. The origins of drivers are 1, 2, 3, 4, respectively. The destinations of drivers are 5, 6, 7, 8, respectively. The origins of customer requests are 9, 10, 11, 12, 13, 14, 15, respectively. The destinations of customer requests are 16, 17, 18, 19, 20, 21, 22, respectively. For the size of example $E - n22 - k4$, it is better to solve the model directly using Gurobi because the solution is an optimal solution with a very short CPU

time. Eleven combinations of parameters values are used to test example $E - n22 - k4$ and the corresponding optimal solutions are compared to illustrate how parameters influence the matching and routing decisions, as shown in Table 5.15 and Figure 5.7. Setting 1 provides the maximum flexibility for matching and routing, as the range of time windows are large, $m_k = 4$, $C_k = 4$, $t_k = 1000$. In setting 1, all customer requests can be served by the drivers and the objective function value is only travel cost. In setting 2, $t_k = 100, \forall k \in K$, which indicate that drivers are not willing to spend too much travel time during their trips. In setting 2, six customer requests are not served due to limited travel time that drivers are willing to spend and the objective function value increases for the cost of unserved customers. In setting 3, when $t_k = 1000, k = 3, 4$, six customer requests can be served by drivers 3 and 4, and the objective function value decreases. Compared with setting 1, each driver only want to serve at most two customer requests during his/her trip in setting 4. In setting 4, travel cost increases compared with the travel cost in setting 1. In setting 5, when $m_k = 4, k = 3, 4$, the solution is more similar to the solution in setting 1. Similarly, the objective function value increases in setting 6 and 7 due to the limited number of available seats. Setting 8 – 11 show that the narrow range of time windows for departure and arrival will increase the number of unserved customers in different levels.

Table 5.15: Results for Example $E - n22 - k4$ with Different Settings

Setting	e_{b_r}	l_{b_r}	l_{w_r}	e_{b_k}	l_{b_k}	l_{w_k}	C_k	m_k	t_k	Obj	Solution
1	0	1000	1000	0	1000	1000	4	4	1000	428	[1,11,13,15,18,22,20,5],[2,6], [3,10,17,7],[4,9,12,14,21,19,16,8]
2	0	1000	1000	0	1000	1000	4	4	100	6173	[1,5],[2,6], [3,7],[4,9,16,8]
3	0	1000	1000	0	1000	1000	4	4	$t_k = 100, k = 1, 2$ $t_3 = 1000$ $t_4 = 1000$	1373	[1,5],[2,6], [3,10,13,15,22,20,17,7], [4,9,12,14,21,19,16,8]
4	0	1000	1000	0	1000	1000	4	2	1000	505	[1,13,15,22,20,5],[2,9,16,6], [3,10,11,18,17,7],[4,12,14,21,19,8]
5	0	1000	1000	0	1000	1000	4	$m_k = 2, k = 1, 2$ $m_3 = 4$ $m_4 = 4$	1000	435	[1,13,15,22,20,5],[2,6], [3,10,11,18,17,7], [4,9,12,14,21,19,16,8]
6	0	1000	1000	0	1000	1000	2	4	1000	1473	[1,13,15,22,20,5],[2,9,16,6], [3,10,17,7],[4,12,14,21,19,8]
7	0	1000	1000	0	1000	1000	$C_k = 2, k = 1, 2$ $C_3 = 4$ $C_4 = 4$	4	1000	435	[1,13,15,22,20,5],[2,6], [3,10,11,18,17,7], [4,9,12,14,21,19,16,8]
8	0	1000	1000	50	70	1000	4	4	1000	1473	[1,13,15,22,20,5],[2,9,16,6], [3,10,17,7],[4,12,14,21,19,8]
9	50	70	1000	0	1000	1000	4	4	1000	1473	[1,13,15,22,20,5],[2,9,16,6], [3,10,17,7],[4,12,14,21,19,8]
10	0	1000	1000	0	1000	100	4	4	1000	6173	[1,5],[2,6], [3,7],[4,9,16,8]
11	0	1000	70	0	1000	1000	4	4	1000	2452	[1,13,15,22,20,5],[2,9,16,6], [3,10,17,7],[4,11,18,8]

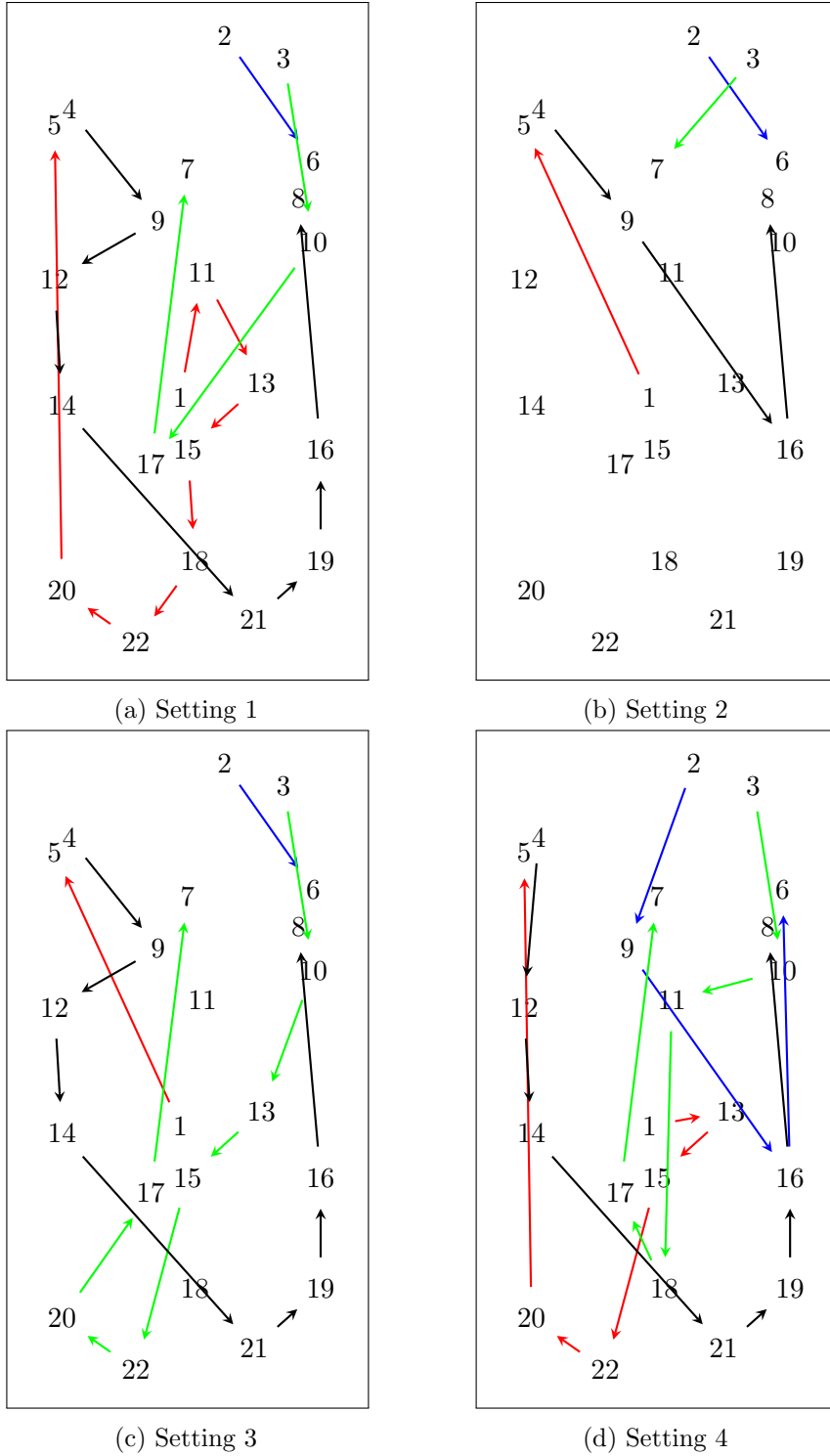


Figure 5.7: Results for Example $E - n22 - k4$ with Different Settings

For example $E - n22 - k4$, it is better to solve the model directly using Gurobi because the solution is an optimal solution with a very short CPU time. The result is reasonable

as the size of this example is small. However, when solving the model with larger example such as $A - n36 - k5$, Gurobi fails to provide the solution after 56600 seconds. For the larger examples, the heuristic algorithms proposed in Chapter 4 are used. The benchmark instances summarized in Table 5.14 are used to show how the objective value changes in solutions as the degree of uncertainty, Γ_k , increases in the robust counterpart. For each example, the CPU time limit = 3000 seconds for solving each model. For the robust counterpart parameters, we randomly generate \hat{t}_{ij} following $U(0, 20)$ where U represents the uniform distribution. The results of these examples are summarized in Tables 5.16 – 5.25. As the solutions are obtained by heuristic algorithms TS and STSVN within limited time, the solutions are not always optimal solutions. In this section, the analyses of results focus on how the objective value changes in solutions as the degree of uncertainty, Γ_k , increases in the robust counterpart.

For the sake of convenience, the objective function value of a solution with Γ_k is denoted by $b(\Gamma_k)$. For the same example, the difference between $b(\Gamma_k)$ and $b(\Gamma_0)$ is computed as $b(\Gamma_k) - b(\Gamma_0)$, which is represented by Diff in Tables 5.16 – 5.25. The percentage of the difference between $b(\Gamma_k)$ and $b(\Gamma_0)$ is computed as $\frac{b(\Gamma_k) - b(\Gamma_0)}{b(\Gamma_0)}$, which is represented by % in Tables 5.16 – 5.25. When $\Gamma_k = 1$, the expected travel cost (related to travel time) increases 3.41% – 19.49% in these examples. When $\Gamma_k = 2$, the expected travel cost (related to travel time) increases 14.9% – 26.39% in these examples. Similarly, when the degree of uncertainty, Γ_k , increases in the robust counterpart, the expected travel cost (related to travel time) increases in these examples.

Table 5.16: Results for Example $E - n22 - k4$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	449	0	0.00	437	0	0.00
1	507	58	12.92	498	61	13.96
2	550	101	22.49	540	103	23.57
3	562	113	25.17	552	115	26.32
4	567	118	26.28	556	119	27.23
5	571	122	27.17	560	123	28.15

Table 5.17: Results for Example $A - n32 - k5$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	838	0	0.00	838	0	0.00
1	921	83	9.90	921	83	9.90
2	974	136	16.23	974	136	16.23
3	1017	179	21.36	1014	176	21.00
4	1033	195	23.27	1033	195	23.27
5	1046	208	24.82	1046	208	24.82

Table 5.18: Results for Example $A - n36 - k5$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	917	0	0.00	933	0	0.00
1	1057	140	15.27	1016	83	8.90
2	1117	200	21.81	1072	139	14.90
3	1163	246	26.83	1114	181	19.40
4	1206	289	31.52	1150	217	23.26
5	1238	321	35.01	1175	242	25.94

Table 5.19: Results for Example $A - n44 - k6$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	1043	0	0.00	939	0	0.00
1	1163	120	11.51	1064	125	13.31
2	1224	181	17.35	1127	188	20.02
3	1283	240	23.01	1193	254	27.05
4	1326	283	27.13	1235	296	31.52
5	1356	313	30.01	1258	319	33.97

Table 5.20: Results for Example $A - n48 - k7$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	1046	0	0.00	1045	0	0.00
1	1158	112	10.71	1145	100	9.57
2	1322	276	26.39	1248	203	19.43
3	1338	292	27.92	1312	267	25.55
4	1438	392	37.48	1339	294	28.13
5	1471	425	40.63	1383	338	32.34

5.3 Results for Comparison of TS and STSVN

In this section, the performance of TS and STSVN are compared in terms of solving CVRP, SDVRP, and ride-sharing models in this dissertation, respectively. The performance of TS

Table 5.21: Results for Example $A - n54 - k7$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	1376	0	0.00	1126	0	0.00
1	1501	125	9.08	1317	191	16.96
2	1589	213	15.48	1345	219	19.45
3	1669	293	21.29	1416	290	25.75
4	1735	359	26.09	1479	353	31.35
5	1789	413	30.01	1485	359	31.88

Table 5.22: Results for Example $A - n60 - k9$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	1310	0	0.00	1239	0	0.00
1	1457	147	11.22	1454	215	17.35
2	1545	235	17.94	1467	228	18.40
3	1617	307	23.44	1562	323	26.07
4	1683	373	28.47	1701	462	37.29
5	1705	395	30.15	1672	433	34.95

Table 5.23: Results for Example $A - n64 - k9$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	1334	0	0.00	1169	0	0.00
1	1594	260	19.49	1296	127	10.86
2	1686	352	26.39	1462	293	25.06
3	1763	429	32.16	1468	299	25.58
4	1822	488	36.58	1525	356	30.45
5	1869	535	40.10	1527	358	30.62

Table 5.24: Results for Example $E - n76 - k7$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	1197	0	0.00	1137	0	0.00
1	1340	143	11.95	1263	126	11.08
2	1415	218	18.21	1327	190	16.71
3	1539	342	28.57	1454	317	27.88
4	1596	399	33.33	1447	310	27.26
5	1636	439	36.68	1581	444	39.05

and STSVN are compared based on the objective function values of the obtained best-so-far solutions within the same given CPU time.

The objective function value of a solution provided by TS is denoted by $b(TS)$ and the

Table 5.25: Results for Example $A - n80 - k10$ with Different Γ_k

Γ_k	TS			STSVN		
	Obj	Diff	%	Obj	Diff	%
0	1725	0	0.00	1762	0	0.00
1	1923	198	11.48	1822	60	3.41
2	2096	371	21.51	2054	292	16.57
3	2119	394	22.84	2129	367	20.83
4	2248	523	30.32	2198	436	24.74
5	2191	466	27.01	2182	420	23.84

objective function value of a solution provided by STSVN is denoted by $b(STSVN)$. For the same example, the difference between $b(TS)$ and $b(STSVN)$ is computed as $b(TS) - b(STSVN)$, which is represented by Diff in Tables 5.26 – 5.29. The percentage of the difference between $b(TS)$ and $b(STSVN)$ is computed as $\frac{b(TS) - b(STSVN)}{b(TS)}$, which is represented by % in Tables 5.26 – 5.29. For the examples of CVRP and SDVRP models, the objective function value of the obtained best-so-far solutions are summarized in Tables 5.26 – 5.29. As shown in Tables 5.26 – 5.29, the objective function values of the obtained best-so-far solutions by STSVN are on average 5.02% less than the ones by TS. Let a case represent solving a model with a specific objective function for an example. In Tables 5.26 – 5.29, there are total 144 cases. In 121 cases, the objective function values of the obtained best-so-far solutions by STSVN are up to 46.49% (6.28% on average) less than the ones by TS. In 4 cases, the objective function values of the obtained best-so-far solutions by STSVN are the same as the ones by TS. In 19 cases, the objective function values of the obtained best-so-far solutions by STSVN are 3.3% on average more than the ones by TS.

After the simple comparison, the statistical tests are implemented for CVRP and SDVRP results. As shown in Tables 5.26 – 5.29, TS and STSVN provide the results for 144 cases, respectively. First, the normality tests (Massey Jr, 1951) are implemented for the data of TS and the data of STSVN, respectively. As shown in Figures 5.8 and 5.9, the data of TS and the data of STSVN do not follow a normal distribution. Therefore, the Wilcoxon-Mann-Whitney test (Fay and Proschan, 2010) is used to test whether the objective function value provided by TS is statistically different from the one provided by STSVN. The Wilcoxon-Mann-Whitney test is suitable for these data because two groups of data are compared (Fay and Proschan, 2010). These two groups of data are independent

Table 5.26: Comparison of TS and STSVN for CVRP and SDVRP Models, Part 1

Example	Model	TS-obj	STSVN-obj	Diff	%
$A - n32 - k5$	CVRP-minT	865	784	81	9.36
	CVRP-minS	2213	2192	21	0.95
	CVRP-minL	134	131	3	2.24
	CVRP-minD	28567	28109	458	1.60
	SDVRP-minT	796	796	0	0.00
	SDVRP-minS	2226	2192	34	1.53
	SDVRP-minL	138	130	8	5.80
	SDVRP-minD	27921	27921	0	0.00
	RCVRP-minT	1070	1058	12	1.12
	RCVRP-minS	2994	2943	51	1.70
	RCVRP-minL	215	173	42	19.53
	RCVRP-minD	40563	39745	818	2.02
	RSDVRP-minT	1032	1051	-19	-1.84
	RSDVRP-minS	2999	2950	49	1.63
	RSDVRP-minL	169	164	5	2.96
RSDVRP-minD	40192	39230	962	2.39	
$A - n38 - k5$	CVRP-minT	733	735	-2	-0.27
	CVRP-minS	2138	2084	54	2.53
	CVRP-minL	172	116	56	32.56
	CVRP-minD	29925	25978	3947	13.19
	SDVRP-minT	745	734	11	1.48
	SDVRP-minS	2259	2192	67	2.97
	SDVRP-minL	129	123	6	4.65
	SDVRP-minD	27837	26609	1228	4.41
	RCVRP-minT	1088	1066	22	2.02
	RCVRP-minS	3005	2921	84	2.80
	RCVRP-minL	230	162	68	29.57
	RCVRP-minD	39911	38874	1037	2.60
	RSDVRP-minT	1108	1104	4	0.36
	RSDVRP-minS	2934	2943	-9	-0.31
	RSDVRP-minL	147	143	4	2.72
RSDVRP-minD	40110	38080	2030	5.06	
$A - n44 - k6$	CVRP-minT	993	937	56	5.64
	CVRP-minS	2741	2579	162	5.91
	CVRP-minL	194	140	54	27.84
	CVRP-minD	34595	33083	1512	4.37
	SDVRP-minT	959	937	22	2.29
	SDVRP-minS	2723	2578	145	5.33
	SDVRP-minL	140	118	22	15.71
	SDVRP-minD	34116	33183	933	2.73

and not normally distributed (Fay and Proschan, 2010). For these data, the p-value = 0.69 (> 0.05) in the Wilcoxon-Mann-Whitney test. Since p-value is greater than 0.05, we

Table 5.27: Comparison of TS and STSVN for CVRP and SDVRP Models, Part 2

Example	Model	TS-obj	STSVN-obj	Diff	%
$A - n44 - k6$	RCVRP-minT	1354	1308	46	3.40
	RCVRP-minS	3414	3406	8	0.23
	RCVRP-minL	299	160	139	46.49
	RCVRP-minD	47331	46734	597	1.26
	RSDVRP-minT	1315	1310	5	0.38
	RSDVRP-minS	3500	3418	82	2.34
	RSDVRP-minL	160	149	11	6.88
	RSDVRP-minD	48105	46738	1367	2.84
$A - n55 - k9$	CVRP-minT	1112	1074	38	3.42
	CVRP-minS	2743	2639	104	3.79
	CVRP-minL	133	102	31	23.31
	CVRP-minD	41685	41278	407	0.98
	SDVRP-minT	1103	1073	30	2.72
	SDVRP-minS	2647	2619	28	1.06
	SDVRP-minL	101	97	4	3.96
	SDVRP-minD	40963	41838	-875	-2.14
	RCVRP-minT	1591	1562	29	1.82
	RCVRP-minS	3915	3856	59	1.51
	RCVRP-minL	240	152	88	36.67
	RCVRP-minD	64352	60937	3415	5.31
	RSDVRP-minT	1588	1542	46	2.90
	RSDVRP-minS	3882	3790	92	2.37
	RSDVRP-minL	132	130	2	1.52
	RSDVRP-minD	61546	61743	-197	-0.32
$A - n69 - k9$	CVRP-minT	1168	1169	-1	-0.09
	CVRP-minS	3663	3732	-69	-1.88
	CVRP-minL	154	150	4	2.60
	CVRP-minD	44372	42226	2146	4.84
	SDVRP-minT	1174	1199	-25	-2.13
	SDVRP-minS	3668	3708	-40	-1.09
	SDVRP-minL	136	134	2	1.47
	SDVRP-minD	44550	43411	1139	2.56
	RCVRP-minT	1820	1779	41	2.25
	RCVRP-minS	5793	5681	112	1.93
	RCVRP-minL	293	189	104	35.49
	RCVRP-minD	71211	71759	-548	-0.77
	RSDVRP-minT	1824	1786	38	2.08
	RSDVRP-minS	5891	5707	184	3.12
	RSDVRP-minL	191	185	6	3.14
	RSDVRP-minD	72124	70197	1927	2.67

conclude that the objective function values between TS and STSVN are not significantly different.

Table 5.28: Comparison of TS and STSVN for CVRP and SDVRP Models, Part 3

Example	Model	TS-obj	STSVN-obj	Diff	%
$E - n76 - k14$	CVRP-minT	1072	1046	26	2.43
	CVRP-minS	2125	2041	84	3.95
	CVRP-minL	107	82	25	23.36
	CVRP-minD	36762	35998	764	2.08
	SDVRP-minT	1057	1046	11	1.04
	SDVRP-minS	2327	2353	-26	-1.12
	SDVRP-minL	81	91	-10	-12.35
	SDVRP-minD	40026	39670	356	0.89
	RCVRP-minT	1722	1678	44	2.56
	RCVRP-minS	3609	3503	106	2.94
	RCVRP-minL	152	120	32	21.05
	RCVRP-minD	66665	62143	4522	6.78
	RSDVRP-minT	1669	1589	80	4.79
	RSDVRP-minS	3580	3622	-42	-1.17
	RSDVRP-minL	121	114	7	5.79
RSDVRP-minD	64322	64308	14	0.02	
$A - n80 - k10$	CVRP-minT	1931	1791	140	7.25
	CVRP-minS	6509	6048	461	7.08
	CVRP-minL	181	169	12	6.63
	CVRP-minD	68617	67696	921	1.34
	SDVRP-minT	1850	1776	74	4.00
	SDVRP-minS	6672	6160	512	7.67
	SDVRP-minL	160	172	-12	-7.50
	SDVRP-minD	68801	67365	1436	2.09
	RCVRP-minT	2601	2569	32	1.23
	RCVRP-minS	8293	8153	140	1.69
	RCVRP-minL	246	215	31	12.60
	RCVRP-minD	102904	96504	6400	6.22
	RSDVRP-minT	2524	2593	-69	-2.73
	RSDVRP-minS	8783	8298	485	5.52
	RSDVRP-minL	200	209	-9	-4.50
RSDVRP-minD	103769	98117	5652	5.45	

Based on these results, the performance of STSVN is on average better than the one of TS for solving the CVRP and SDVRP models for these examples within the same CPU time. However, the performance of STSVN is not significantly better than the one of TS based on the Wilcoxon-Mann-Whitney test.

For the examples of ride-sharing models, the objective function value of the obtained best-so-far solutions are summarized in Tables 5.30 – 5.39. As shown in Tables 5.30 – 5.39, the objective function values of the obtained best-so-far solutions by STSVN are on average

Table 5.29: Comparison of TS and STSVN for CVRP and SDVRP Models, Part 4

Example	Model	TS-obj	STSVN-obj	Diff	%
$E - n101 - k8$	CVRP-minT	939	823	116	12.35
	CVRP-minS	5091	4683	408	8.01
	CVRP-minL	123	123	0	0.00
	CVRP-minD	75669	67114	8555	11.31
	SDVRP-minT	884	848	36	4.07
	SDVRP-minS	4569	4252	317	6.94
	SDVRP-minL	128	122	6	4.69
	SDVRP-minD	64764	59304	5460	8.43
	RCVRP-minT	1594	1590	4	0.25
	RCVRP-minS	9367	8849	518	5.53
	RCVRP-minL	217	217	0	0.00
	RCVRP-minD	135054	123905	11149	8.26
	RSDVRP-minT	1532	1520	12	0.78
	RSDVRP-minS	7953	7741	212	2.67
	RSDVRP-minL	206	209	-3	-1.46
	RSDVRP-minD	119206	113103	6103	5.12
$E - n101 - k14$	CVRP-minT	1163	1157	6	0.52
	CVRP-minS	3720	3055	665	17.88
	CVRP-minL	101	99	2	1.98
	CVRP-minD	50224	45483	4741	9.44
	SDVRP-minT	1165	1119	46	3.95
	SDVRP-minS	3386	3128	258	7.62
	SDVRP-minL	108	119	-11	-10.19
	SDVRP-minD	48929	43726	5203	10.63
	RCVRP-minT	2207	1934	273	12.37
	RCVRP-minS	5528	5332	196	3.55
	RCVRP-minL	196	151	45	22.96
	RCVRP-minD	84993	77374	7619	8.96
	RSDVRP-minT	1880	1824	56	2.98
	RSDVRP-minS	5373	5056	317	5.90
	RSDVRP-minL	129	143	-14	-10.85
	RSDVRP-minD	80335	76513	3822	4.76
Average		15961.39	15246.60	714.80	5.02

less than the ones by TS. In example $E - n22 - k4$, the objective function values of the obtained best-so-far solutions by STSVN are up to 2.67% (1.99% on average) less than the ones by TS. In example $A - n32 - k5$, the objective function values of the obtained best-so-far solutions by STSVN are up to 0.29% (0.05% on average) less than the ones by TS. In example $A - n36 - k5$, the objective function values of the obtained best-so-far solutions by STSVN are up to 5.09% (3.35% on average) less than the ones by TS. In this example, the

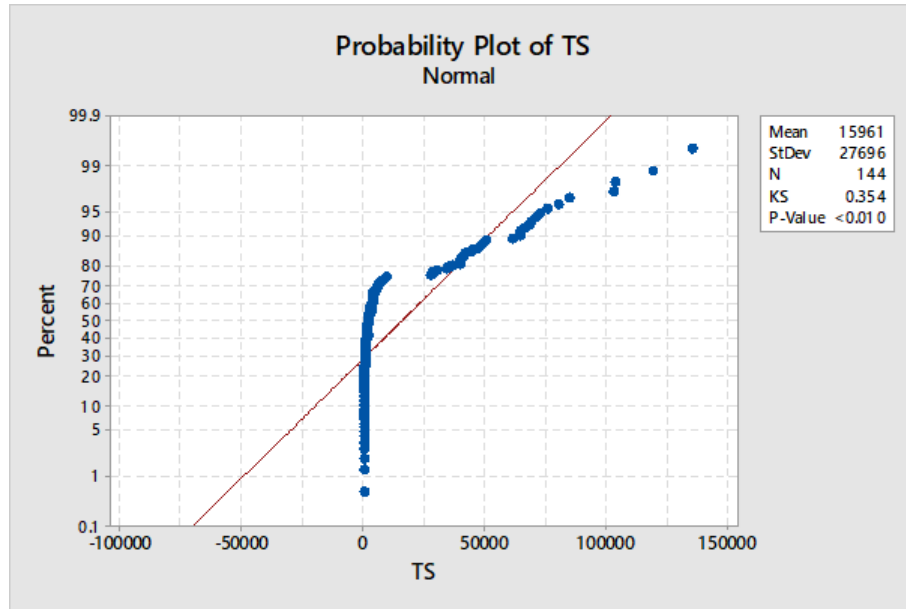


Figure 5.8: Normal Probability Plots for Disaster Relief Routing Results of TS

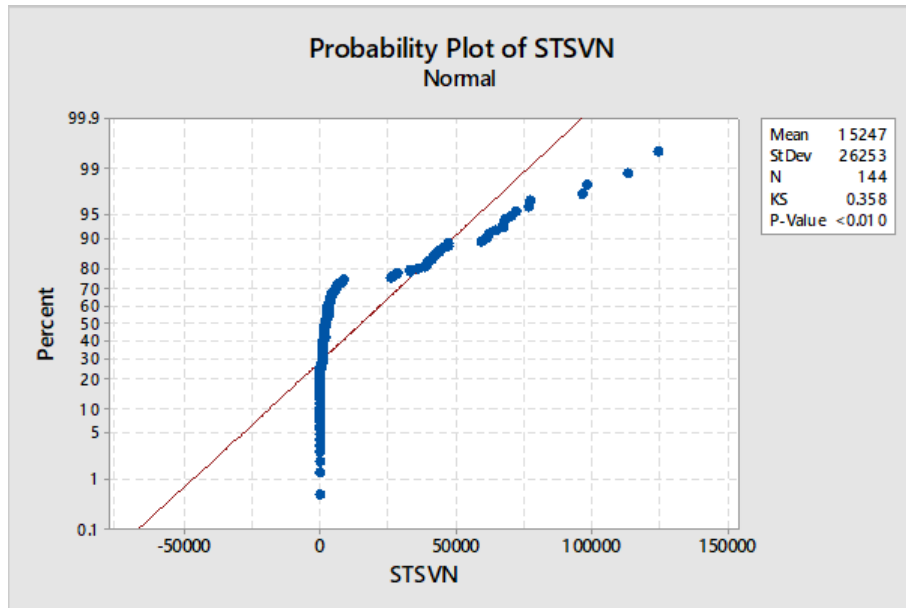


Figure 5.9: Normal Probability Plots for Disaster Relief Routing Results of STSVN

performance of TS is better than STSVN only when $\Gamma_k = 0$. For examples $A - n44 - k6$, $A - n48 - k7$, $A - n54 - k7$, $A - n64 - k9$, and $E - n76 - k7$, the objective function values of the obtained best-so-far solutions by STSVN are up to 9.97% (7.92% on average), 6.88% (3.6% on average), 18.17% (15.45% on average), 18.70% (15.95% on average), 9.34% (5.87% on average) less than the ones by TS, respectively. In example $A - n60 - k9$, the objective

function values of the obtained best-so-far solutions by STSVN are up to 5.42% (2.49% on average) less than the ones by TS. In this example, the performance of TS is better than STSVN only when $\Gamma_k = 4$. In example $A - n80 - k10$, the objective function values of the obtained best-so-far solutions by STSVN are up to 5.25% (1.21% on average) less than the ones by TS. In this example, the performance of TS is better than STSVN only when $\Gamma_k = 0$ and $\Gamma_k = 3$. Based on these results, the performance of STSVN is on average better than the one of TS for solving the ride-sharing models for these examples within the same CPU time.

Table 5.30: Comparison of TS and STSVN for Ride-sharing Models, $E - n22 - k4$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	449	437	12	2.67
1	507	498	9	1.78
2	550	540	10	1.82
3	562	552	10	1.78
4	567	556	11	1.94
5	571	560	11	1.93
Average	534	523	11	1.99

Table 5.31: Comparison of TS and STSVN for Ride-sharing Models, $A - n32 - k5$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	838	838	0	0
1	921	921	0	0
2	974	974	0	0
3	1017	1014	3	0.29
4	1033	1033	0	0
5	1046	1046	0	0
Average	972	971	0.5	0.05

Table 5.32: Comparison of TS and STSVN for Ride-sharing Models, $A - n36 - k5$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	917	933	-16	-1.74
1	1057	1016	41	3.88
2	1117	1072	45	4.03
3	1163	1114	49	4.21
4	1206	1150	56	4.64
5	1238	1175	63	5.09
Average	1116	1077	40	3.35

Table 5.33: Comparison of TS and STSVN for Ride-sharing Models, $A - n44 - k6$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	1043	939	104	9.97
1	1163	1064	99	8.51
2	1224	1127	97	7.92
3	1283	1193	90	7.01
4	1326	1235	91	6.86
5	1356	1258	98	7.23
Average	1233	1136	97	7.92

Table 5.34: Comparison of TS and STSVN for Ride-sharing Models, $A - n48 - k7$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	1046	1045	1	0.10
1	1158	1145	13	1.12
2	1322	1248	74	5.60
3	1338	1312	26	1.94
4	1438	1339	99	6.88
5	1471	1383	88	5.98
Average	1296	1245	50.2	3.60

Table 5.35: Comparison of TS and STSVN for Ride-sharing Models, $A - n54 - k7$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	1376	1126	250	18.17
1	1501	1317	184	12.26
2	1589	1345	244	15.36
3	1669	1416	253	15.16
4	1735	1479	256	14.76
5	1789	1485	304	16.99
Average	1610	1361	249	15.45

Table 5.36: Comparison of TS and STSVN for Ride-sharing Models, $A - n60 - k9$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	1310	1239	71	5.42
1	1457	1454	3	0.21
2	1545	1467	78	5.05
3	1617	1562	55	3.40
4	1683	1701	-18	-1.07
5	1705	1672	33	1.94
Average	1553	1516	37.0	2.49

After the simple comparison, the statistical tests are implemented for ride-sharing results. As shown in Tables 5.30 – 5.39, TS and STSVN provide the results for 60 cases, respectively. First, the normality tests (Massey Jr, 1951) are implemented for the data of

Table 5.37: Comparison of TS and STSVN for Ride-sharing Models, $A - n64 - k9$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	1334	1169	165	12.37
1	1594	1296	298	18.70
2	1686	1462	224	13.29
3	1763	1468	295	16.73
4	1822	1525	297	16.30
5	1869	1527	342	18.30
Average	1678	1408	270	15.95

Table 5.38: Comparison of TS and STSVN for Ride-sharing Models, $E - n76 - k7$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	1197	1137	60	5.01
1	1340	1263	77	5.75
2	1415	1327	88	6.22
3	1539	1454	85	5.52
4	1596	1447	149	9.34
5	1636	1581	55	3.36
Average	1454	1368	86	5.87

Table 5.39: Comparison of TS and STSVN for Ride-sharing Models, $A - n80 - k10$

Γ_k	TS-obj	STSVN-obj	Diff	%
0	1725	1762	-37	-2.14
1	1923	1822	101	5.25
2	2096	2054	42	2.00
3	2119	2129	-10	-0.47
4	2248	2198	50	2.22
5	2191	2182	9	0.41
Average	2050	2025	26	1.21

TS and the data of STSVN, respectively. As shown in Figures 5.10 and 5.11, the data of TS and the data of STSVN follow a normal distribution. Therefore, the Two Sample t-test (Cressie and Whitford, 1986) is used to test whether the objective function value provided by TS is statistically different from the one provided by STSVN for the ride-sharing results. The Two Sample t-test is suitable for these data because two groups of data are compared (Cressie and Whitford, 1986). These two groups of data are independent and approximately follow a normal distribution (Cressie and Whitford, 1986). For these data, the p-value = 0.25 (> 0.05) in the Two Sample t-test. Since p-value is greater than 0.05, we conclude that the objective function values between TS and STSVN are not significantly different for the

ride-sharing results.

Based on these results, the performance of STSVN is on average better than the one of TS for solving the ride-sharing models for these examples within the same CPU time. However, the performance of STSVN is not significantly better than the one of TS based on Two Sample t-test.

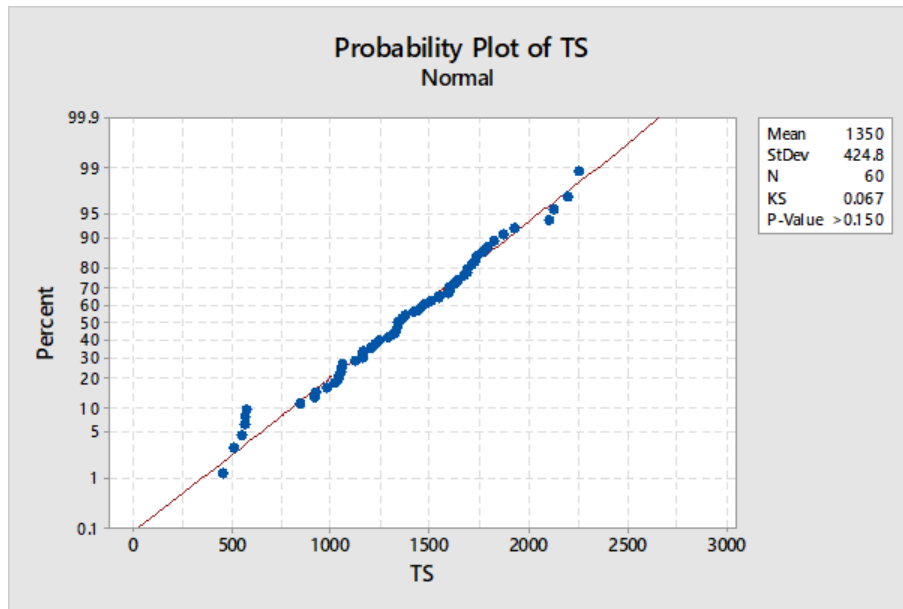


Figure 5.10: Normal Probability Plots for Ride-sharing Results of TS

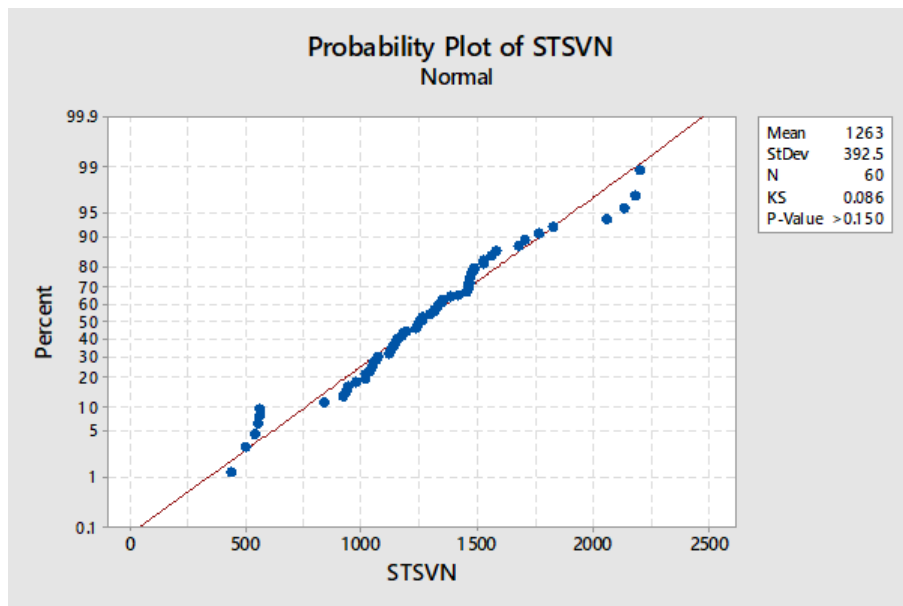


Figure 5.11: Normal Probability Plots for Ride-sharing Results of STSVN

Chapter 6

Conclusion and Future Work

In this dissertation, VRP and its variants are specifically considered in two applications: disaster relief routing and ride-sharing. Based on the settings of each application, the VRP based models are modified specifically.

In disaster relief routing, uncertain travel times and demands are considered when planning vehicle routes for delivering critical supplies to the affected population in need after a disaster. In particular, a robust optimization approach is proposed for CVRP and SDVRP to handle such uncertainty in an effective manner. In addition, CVRP and SDVRP are examined for disaster relief routing purposes using different objective functions, which was evaluated by five different metrics. The results of the examples presented that the impact of uncertainty can be better mitigated in SDVRP than CVRP. The results indicated that minS, minL, and minD objectives can provide better solutions for the route planning in humanitarian relief routing, as these objectives can ensure faster deliveries to the people in need. In addition, minT is more sensitive to the uncertainty of travel times than other objective functions, while minD is more sensitive to the uncertainty of demands. The results also indicated that the SDVRP can provide more flexible solutions, especially when the demands are relatively large: the SDVRP can better utilize the number of vehicles to meet the demand and realize objectives. For minT, the SDVRP provides the solutions with less vehicles required and minimizes the total travel time. For minS, minL, and minD, the SDVRP provides the solutions to utilize all available vehicles to minimize the (summation of, latest, and demand-weighted) arrival times. In the RCVRP and RSDVRP, more realized increased travel times due to uncertainty are considered with larger Γ_T , resulting in

more robust solutions with increased total travel times. The results show that the increased metric values can be within a reasonable range even for the most conservative cases (at the biggest robust parameter values) in the SDVRP. The RSDVRP can avoid the arcs with large uncertain travel time in the optimal (or near-optimal) solutions by visiting a node multiple times from different arcs having less increased travel times, while RCVRP has no such capability. In summary, the examples show that SDVRP outperforms CVRP both for the base model and robust counterparts with respect to the metrics for humanitarian relief vehicle routing, implying that SDVRP and RSDVRP can be more effective and useful in disaster relief routing by providing more flexibility.

For the ride-sharing, a ride-sharing model is proposed to determine the optimal match between drivers and riders and the optimal routes. A robust model is proposed to address the travel time uncertainty. Different examples are used to derive insights for ride-sharing in terms of the influence of different parameters and uncertain travel time on the solution obtained from the models. The results show that flexibility of customer requests and drivers will influence the matching and routing decisions. If drivers are not willing to spend too much travel time during their trips, then less customer requests can be assigned to the drivers and the cost due to the unserved customers increases. Similarly, if drivers only want to serve a limited number of customer requests during his/her trip, the less customer requests can be assigned to the drivers and the cost due to the unserved customers increases. In addition, the narrow range of time windows for departure and arrival will increase the number of unserved customers in different levels. The examples show that the expected travel cost increases as the degree of uncertainty increases in the robust counterpart.

Small problems for these models can be solved by using the commercial packages such as Gurobi and CPLEX. However, for large-scale problems, it is not practical to utilize the solvers as the VRPs are NP-hard. Considering the settings of disaster relief routing and dynamic ride-sharing, the routing decisions need to be made quickly, it is desirable to obtain the near-optimal solutions in a relatively short period of time. Therefore, heuristic approaches are used to solve the models. Insertion algorithms are specifically modified to construct the initial solutions of CVRP, SDVRP, and ride-sharing models in this dissertation. To solve the robust counterparts in the robust models, two algorithms are proposed

to consider the increased demand and travel time due to the changes in robust parameters, Γ_Q and Γ_T , in CVRP and SDVRP. In addition, an algorithm is proposed to consider the increased travel time due to the changes in robust parameter, Γ_k , in ride-sharing. Tabu search (TS) is used to search for optimal or near-optimal solutions. In addition, a new hybrid algorithm called scoring tabu search with variable neighborhood (STSVN) is proposed to solve the models and compared with tabu search. The performance of TS and STSVN are compared based on the objective function values of the obtained best-so-far solutions within the same given CPU time. Based on the results, the performance of STSVN is on average better than the one of TS for solving the CVRP, SDVRP, ride-sharing models for the examples within the same CPU time.

The contribution of this dissertation is as follows. (1) The robust models of SDVRP are proposed to consider travel time and demand uncertainty. (2) A robust model of ride-sharing is proposed to consider uncertain travel time. (3) To solve the models, the insertion algorithms are modified for SDVRP and ride-sharing problem. (4) Uncertain Demand Selection Algorithm and Uncertain Travel Time Selection Algorithm are proposed to be integrated in heuristic algorithms to solve the robust counterpart. (5) A new hybrid algorithm, STSVN, is proposed to solve the models in disaster relief routing and ride-sharing problem efficiently within the given CPU time. (6) The managerial insights are explored and provided for decision making when considering CVRP, SDVRP, different objectives, and uncertainty in disaster relief routing.

For the future work, the proposed robust models of CVRP and SDVRP can be extended to include more types of uncertainty that exist in disaster relief routing such as loading time and unloading time in each node. In this dissertation, the uncertain travel times and demands are assumed to be independent, the scenario, in which travel time, demand, loading time, and unloading time are correlated, is worth researching for the future work. Furthermore, more constraints can be considered in the model, such as the maximum travel distance/travel time for each vehicle and capacity constraint of heterogeneous vehicles. Since the results have shown that the different objectives have different implications in disaster relief routing, the study of multi-objective optimization can be another good research focus. In terms of robust optimization, a statistical learning approach can be applied to better

estimate the uncertainty sets, which will impact the quality of solutions. In terms of ride-sharing, the future work can focus on how to increase the matching rate by increasing the flexibility. As the performance of STSVN is better than TS in solving the models in this dissertation, it would be good to apply STSVN in other optimization problems and compare it with different algorithms. In addition, the comparison of different penalty functions for infeasible solutions can be explored in the ride-sharing problem, which may enhance exploring the good feasible solutions near the boundary of infeasible solutions.

Bibliography

- Afshar, A. and A. Haghani (2012). Modeling integrated supply chain logistics in real-time large-scale disaster relief operations. *Socio-Economic Planning Sciences* 46(4), 327–338.
- Agatz, N., A. Erera, M. Savelsbergh, and X. Wang (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research* 223(2), 295–303.
- Agra, A., M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, and C. Requejo (2013). The robust vehicle routing problem with time windows. *Computers & Operations Research* 40(3), 856–866.
- Allahviranloo, M., J. Y. Chow, and W. W. Recker (2014). Selective vehicle routing problems under uncertainty without recourse. *Transportation Research Part E: Logistics and Transportation Review* 62, 68–88.
- Alonso-Mora, J., S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 201611675.
- Amey, A. (2011). A proposed methodology for estimating rideshare viability within an organization, applied to the mit community. In *TRB Annual Meeting Proceedings*, pp. 1–16.
- Archetti, C., M. Bouchard, and G. Desaulniers (2011). Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. *Transportation Science* 45(3), 285–298.
- Archetti, C., M. W. Savelsbergh, and M. G. Speranza (2006). Worst-case analysis for split delivery vehicle routing problems. *Transportation science* 40(2), 226–234.
- Archetti, C., M. G. Speranza, and A. Hertz (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation science* 40(1), 64–73.
- Archetti, C., M. G. Speranza, and M. W. Savelsbergh (2008). An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science* 42(1), 22–31.
- Armant, V. and K. N. Brown (2014). Minimizing the driving distance in ride sharing systems. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, pp. 568–575. IEEE.
- Augerat, P., J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi (1998). Computational results with a branch-and-cut code for the capacitated vehicle routing problem.
- Balcik, B., B. M. Beamon, and K. Smilowitz (2008). Last mile distribution in humanitarian relief. *Journal of Intelligent Transportation Systems* 12(2), 51–63.
- Baldacci, R., V. Maniezzo, and A. Mingozzi (2004). An exact method for the car pooling problem based on lagrangean column generation. *Operations Research* 52(3), 422–439.

- BañOs, R., J. Ortega, C. Gil, A. FernáNdez, and F. De Toro (2013). A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications* 40(5), 1696–1707.
- Barbarosoglu, G. and D. Ozgur (1999). A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research* 26(3), 255–270.
- Barbarosolu, G. and Y. Arda (2004). A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the operational research society* 55(1), 43–53.
- Beamon, B. M. and S. A. Kotleba (2006). Inventory modelling for complex emergencies in humanitarian relief operations. *International Journal of Logistics: Research and Applications* 9(1), 1–18.
- Belenguer, J., M. Martinez, and E. Mota (2000). A lower bound for the split delivery vehicle routing problem. *Operations research* 48(5), 801–810.
- Belenguer, J.-M., E. Benavent, N. Labadi, C. Prins, and M. Reghioui (2010). Split-delivery capacitated arc-routing problem: Lower bound and metaheuristic. *Transportation Science* 44(2), 206–220.
- Belhaiza, S., P. Hansen, and G. Laporte (2014). A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Computers & Operations Research* 52, 269–281.
- Ben-Tal, A., B. Do Chung, S. R. Mandala, and T. Yao (2011). Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. *Transportation research part B: methodological* 45(8), 1177–1189.
- Ben-Tal, A. and A. Nemirovski (2002a). Robust optimization–methodology and applications. *Mathematical Programming* 92(3), 453–480.
- Ben-Tal, A. and A. Nemirovski (2002b). Robust optimization–methodology and applications. *Mathematical Programming* 92(3), 453–480.
- Berbotto, L., S. García, and F. J. Nogales (2011). A vehicle routing model with split delivery and stop nodes.
- Berbotto, L., S. García, and F. J. Nogales (2014). A randomized granular tabu search heuristic for the split delivery vehicle routing problem. *Annals of Operations Research* 222(1), 153–173.
- Bertsimas, D., D. B. Brown, and C. Caramanis (2011a). Theory and applications of robust optimization. *SIAM review* 53(3), 464–501.
- Bertsimas, D., D. B. Brown, and C. Caramanis (2011b). Theory and applications of robust optimization. *SIAM review* 53(3), 464–501.
- Bertsimas, D. and M. Sim (2003). Robust discrete optimization and network flows. *Mathematical programming* 98(1), 49–71.
- Bertsimas, D. and M. Sim (2004). The price of robustness. *Operations research* 52(1), 35–53.
- Beyer, H.-G. and B. Sendhoff (2007). Robust optimization—a comprehensive survey. *Computer methods in applied mechanics and engineering* 196(33), 3190–3218.
- Binns, C. and W. Y. Low (2015). Nepal disaster: a public health response needed. *Asia-Pacific journal of public health* 27(5), 484–485.
- Bistaffa, F., A. Farinelli, and S. D. Ramchurn (2014). Sharing rides with friends: a coalition formation algorithm for ridesharing.

- Boudia, M., C. Prins, and M. Reghioui (2007). An effective memetic algorithm with population management for the split delivery vehicle routing problem. In *Hybrid Metaheuristics*, pp. 16–30. Springer.
- Bouzaiene-Ayari, B., M. Dror, and G. Laporte (1992). Vehicle routing with stochastic demand and split deliveries.
- Bozorgi-Amiri, A., M. Jabalameli, and S. M. Al-e Hashem (2013). A multi-objective robust stochastic programming model for disaster relief logistics under uncertainty. *OR spectrum* 35(4), 905–933.
- Bozorgi-Amiri, A., M. S. Jabalameli, M. Alinaghian, and M. Heydari (2012). A modified particle swarm optimization for disaster relief logistics under uncertain environment. *The International Journal of Advanced Manufacturing Technology* 60(1), 357–371.
- Braaten, S., O. Gjønnnes, L. M. Hvattum, and G. Tirado (2017). Heuristics for the robust vehicle routing problem with time windows. *Expert Systems with Applications* 77, 136–147.
- Brandão, J. (2004). A tabu search algorithm for the open vehicle routing problem. *European Journal of Operational Research* 157(3), 552–564.
- Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers & Operations Research* 38(1), 140–151.
- Bräysy, O. and M. Gendreau (2002). Tabu search heuristics for the vehicle routing problem with time windows. *Top* 10(2), 211–237.
- Burby, R. J. (2006). Hurricane katrina and the paradoxes of government disaster policy: Bringing about wise governmental decisions for hazardous areas. *The Annals of the American Academy of Political and Social Science* 604(1), 171–191.
- Calvo, R. W., F. de Luigi, P. Haastrup, and V. Maniezzo (2004). A distributed geographic information system for the daily car pooling problem. *Computers & Operations Research* 31(13), 2263–2278.
- Campbell, A. M. and M. Savelsbergh (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation science* 38(3), 369–378.
- Campbell, A. M., D. Vandenbussche, and W. Hermann (2008a). Routing for relief efforts. *Transportation Science* 42(2), 127–145.
- Campbell, A. M., D. Vandenbussche, and W. Hermann (2008b). Routing for relief efforts. *Transportation Science* 42(2), 127–145.
- Cangialosi, E., A. Di Febbraro, and N. Sacco (2016). Designing a multimodal generalised ride sharing system. *IET Intelligent Transport Systems* 10(4), 227–236.
- Cangialosi, E., A. Di Febbraro, N. Sacco, and A. Baudà (2014). Generalized ride-sharing: an enhanced model and new results. In *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 284–290. IEEE.
- Chan, N. D. and S. A. Shaheen (2012). Ridesharing in north america: Past, present, and future. *Transport Reviews* 32(1), 93–112.
- Chen, L., M. Gendreau, M. H. Hà, and A. Langevin (2016). A robust optimization approach for the road network daily maintenance routing problem with uncertain service time. *Transportation research part E: logistics and transportation review* 85, 40–51.
- Chen, P., B. Golden, X. Wang, and E. Wasil (2016). A novel approach to solve the split delivery vehicle routing problem. *International Transactions in Operational Research*.

- Chen, S., B. Golden, and E. Wasil (2007). The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks* 49(4), 318–329.
- Christofides, N. and S. Eilon (1969). An algorithm for the vehicle-dispatching problem. *Or*, 309–318.
- Coit, D. W., A. E. Smith, and D. M. Tate (1996). Adaptive penalty methods for genetic optimization of constrained combinatorial problems. *INFORMS Journal on Computing* 8(2), 173–182.
- Cordeau, J.-F., G. Laporte, A. Mercier, et al. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society* 52(8), 928–936.
- Cressie, N. and H. Whitford (1986). How to use the two sample t-test. *Biometrical Journal* 28(2), 131–148.
- Crispim, J. and J. Brandão (2005). Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. *Journal of the Operational Research Society* 56(11), 1296–1302.
- Deleplanque, S., A. Quilliot, and B. Bernay (2014). Branch and price for a reliability oriented darp model. In *Control, Decision and Information Technologies (CoDIT), 2014 International Conference on*, pp. 081–086. IEEE.
- Desaulniers, G. (2010). Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations research* 58(1), 179–192.
- Dror, M. and P. Trudeau (1989). Savings by split delivery routing. *Transportation Science* 23(2), 141–145.
- Duhamel, C., J.-Y. Potvin, and J.-M. Rousseau (1997). A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transportation science* 31(1), 49–59.
- Eksioglu, B., A. V. Vural, and A. Reisman (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering* 57(4), 1472–1483.
- Erera, A. L., J. C. Morales, and M. Savelsbergh (2010). The vehicle routing problem with stochastic demand and duration constraints. *Transportation Science* 44(4), 474–492.
- Escobar, J. W., R. Linfati, M. G. Baldoquin, and P. Toth (2014). A granular variable tabu neighborhood search for the capacitated location-routing problem. *Transportation Research Part B: Methodological* 67, 344–356.
- Fanelli, A. and G. Greco (2015). Ride sharing with a vehicle of unlimited capacity. *arXiv preprint arXiv:1507.02414*.
- Fay, M. P. and M. A. Proschan (2010). Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics surveys* 4, 1.
- Ferguson, E. (1997). The rise and fall of the american carpool: 1970–1990. *Transportation* 24(4), 349–376.
- Fischetti, M., P. Toth, and D. Vigo (1994). A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Operations Research* 42(5), 846–859.
- Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. *Operations research* 42(4), 626–642.

- Fleszar, K., I. H. Osman, and K. S. Hindi (2009). A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research* 195(3), 803–809.
- Frizzell, P. W. and J. W. Giffin (1995). The split delivery vehicle scheduling problem with time windows and grid network distances. *Computers & Operations Research* 22(6), 655–667.
- Fu, Y., Y. Fang, C. Jiang, and J. Cheng (2008). Dynamic ride sharing community service on traffic information grid. In *Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on*, Volume 2, pp. 348–352. IEEE.
- Furuhata, M., M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological* 57, 28–46.
- Gendreau, M., F. Guertin, J.-Y. Potvin, and E. Taillard (1999). Parallel tabu search for real-time vehicle routing and dispatching. *Transportation science* 33(4), 381–390.
- Gendreau, M., A. Hertz, and G. Laporte (1994). A tabu search heuristic for the vehicle routing problem. *Management science* 40(10), 1276–1290.
- Gendreau, M., M. Iori, G. Laporte, and S. Martello (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks* 51(1), 4–18.
- Gendreau, M., G. Laporte, C. Musaraganyi, and É. D. Taillard (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research* 26(12), 1153–1173.
- Gendreau, M., G. Laporte, and R. Séguin (1996). A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research* 44(3), 469–477.
- Ghiani, G., F. Guerriero, G. Laporte, and R. Musmanno (2003). Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research* 151(1), 1–11.
- Ghoseiri, K., A. E. Haghani, M. Hamedi, and M.-A. U. T. Center (2011). *Real-time rideshare matching problem*. Mid-Atlantic Universities Transportation Center.
- Glover, F. (1990). Tabu search: A tutorial. *Interfaces* 20(4), 74–94.
- Gong, Q. and R. Batta (2007). Allocation and reallocation of ambulances to casualty clusters in a disaster relief operation. *IIE Transactions* 39(1), 27–39.
- Gounaris, C. E., W. Wiesemann, and C. A. Floudas (2013). The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research* 61(3), 677–693.
- Gulczynski, D. J., B. Golden, and E. Wasil (2008). Recent developments in modeling and solving the split delivery vehicle routing problem. *Tutorials in Operations Research*, 170–180.
- Haghani, A. and S.-C. Oh (1996). Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. *Transportation Research Part A: Policy and Practice* 30(3), 231–250.
- Hahn, G. J. and H. Kuhn (2012). Value-based performance and risk management in supply chains: A robust optimization approach. *International Journal of Production Economics* 139(1), 135–144.
- Han, J., C. Lee, and S. Park (2013). A robust scenario approach for the vehicle routing problem with uncertain travel times. *Transportation Science* 48(3), 373–390.

- Hansen, P., N. Mladenović, and J. A. M. Pérez (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research* 175(1), 367–407.
- Heinrich, S. (2010). *Implementing real-time ridesharing in the San Francisco Bay Area*. Ph. D. thesis, San Jose State University.
- Hertz, A., G. Laporte, and M. Mittaz (2000). A tabu search heuristic for the capacitated arc routing problem. *Operations research* 48(1), 129–135.
- Ho, S. C. and D. Haugland (2004). A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research* 31(12), 1947–1964.
- Horner, M. W. and J. A. Downs (2010). Optimizing hurricane disaster relief goods distribution: model development and application with respect to planning strategies. *Disasters* 34(3), 821–844.
- Hosni, H., J. Naoum-Sawaya, and H. Artail (2014). The shared-taxi problem: Formulation and solution methods. *Transportation Research Part B: Methodological* 70, 303–318.
- Huang, M., K. Smilowitz, and B. Balcik (2012). Models for relief routing: Equity, efficiency and efficacy. *Transportation research part E: logistics and transportation review* 48(1), 2–18.
- Huang, M., K. R. Smilowitz, and B. Balcik (2013). A continuous approximation approach for assessment routing in disaster relief. *Transportation Research Part B: Methodological* 50, 20–41.
- Huang, Y., F. Bastani, R. Jin, and X. S. Wang (2014). Large scale real-time ridesharing with service guarantee on road networks. *Proceedings of the VLDB Endowment* 7(14), 2017–2028.
- Jin, M., K. Liu, and R. O. Bowden (2007). A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem. *International Journal of Production Economics* 105(1), 228–242.
- Jin, M., K. Liu, and B. Eksiöglu (2008). A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters* 36(2), 265–270.
- Karakatič, S. and V. Podgorelec (2015). A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing* 27, 519–532.
- Kelly, K. L. (2007). Casual carpooling-enhanced. *Journal of Public Transportation* 10(4), 6.
- Kleiner, A., B. Nebel, and V. Ziparo (2011). A mechanism for dynamic ride sharing based on parallel auctions.
- Kovács, G. and K. M. Spens (2007). Humanitarian logistics in disaster relief operations. *International Journal of Physical Distribution & Logistics Management* 37(2), 99–114.
- Kuo, Y. and C.-C. Wang (2012). A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Systems with Applications* 39(8), 6949–6954.
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59(3), 345–358.
- Laporte, G., M. Gendreau, J.-Y. Potvin, and F. Semet (2000). Classical and modern heuristics for the vehicle routing problem. *International transactions in operational research* 7(4-5), 285–300.

- Lee, C., K. Lee, and S. Park (2012). Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society* 63(9), 1294–1306.
- Lei, H., G. Laporte, and B. Guo (2012). The vehicle routing problem with stochastic demands and split deliveries. *INFOR* 50(2), 59.
- Lin, Y. and H. Shen (2016). Vshare: A wireless social network aided vehicle sharing system using hierarchical cloud architecture. In *Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on*, pp. 37–48. IEEE.
- Lin, Y.-H., R. Batta, P. A. Rogerson, A. Blatt, and M. Flanigan (2011). A logistics model for emergency supply of critical items in the aftermath of a disaster. *Socio-Economic Planning Sciences* 45(4), 132–145.
- Long, D. C. and D. F. Wood (1995). The logistics of famine relief. *Journal of Business Logistics* 16(1), 213.
- Luis, E., I. S. Dolinskaya, and K. R. Smilowitz (2012). Disaster relief routing: Integrating research and practice. *Socio-economic planning sciences* 46(1), 88–97.
- Lysgaard, J., A. N. Letchford, and R. W. Eglese (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming* 100(2), 423–445.
- Lysgaard, J. and S. Wøhlk (2014). A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. *European Journal of Operational Research* 236(3), 800–810.
- Mahmoudi, M. and X. Zhou (2016). Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transportation Research Part B: Methodological* 89, 19–42.
- Manna, C. and S. Prestwich (2014). Online stochastic planning for taxi and ridesharing. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, pp. 906–913. IEEE.
- Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association* 46(253), 68–78.
- Mete, H. O. and Z. B. Zabinsky (2010). Stochastic optimization of medical supply location and distribution in disaster management. *International Journal of Production Economics* 126(1), 76–84.
- Mezura-Montes, E. and C. A. C. Coello (2011). Constraint-handling in nature-inspired numerical optimization: past, present and future. *Swarm and Evolutionary Computation* 1(4), 173–194.
- Miller, C. E., A. W. Tucker, and R. A. Zemlin (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)* 7(4), 326–329.
- Mitra, S. (2008). A parallel clustering technique for the vehicle routing problem with split deliveries and pickups. *Journal of the operational Research Society* 59(11), 1532–1546.
- Mladenović, N. and P. Hansen (1997). Variable neighborhood search. *Computers & operations research* 24(11), 1097–1100.
- Montané, F. A. T. and R. D. Galvao (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research* 33(3), 595–619.
- Morency, C. (2007). The ambivalence of ridesharing. *Transportation* 34(2), 239–253.

- Moreno, L., M. P. De AragãO, and E. Uchoa (2010). Improved lower bounds for the split delivery vehicle routing problem. *Operations Research Letters* 38(4), 302–306.
- Najafi, M., K. Eshghi, and W. Dullaert (2013). A multi-objective robust optimization model for logistics planning in the earthquake response phase. *Transportation Research Part E: Logistics and Transportation Review* 49(1), 217–249.
- Nanry, W. P. and J. W. Barnes (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological* 34(2), 107–121.
- Nourinejad, M. and M. J. Roorda (2016). Agent based model for dynamic ridesharing. *Transportation Research Part C: Emerging Technologies* 64, 117–132.
- Ordóñez, F. (2010). Robust vehicle routing. *TUTORIALS in Operations Research*, 153–178.
- Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research* 41(4), 421–451.
- Osman, I. H. and N. A. Wassan (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls. *Journal of Scheduling* 5(4), 263–285.
- Özdamar, L. and O. Demir (2012). A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transportation Research Part E: Logistics and Transportation Review* 48(3), 591–602.
- Özdamar, L., E. Ekinçi, and B. Küçükyazici (2004). Emergency logistics planning in natural disasters. *Annals of operations research* 129(1), 217–245.
- Ozdamar, L. and W. Yi (2008). Greedy neighborhood search for disaster relief and evacuation logistics. *IEEE Intelligent Systems* 23(1).
- Paraskevopoulos, D. C., P. P. Repoussis, C. D. Tarantilis, G. Ioannou, and G. P. Prastacos (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics* 14(5), 425–455.
- Pearce, L. (2003). Disaster management and community planning, and public participation: how to achieve sustainable hazard mitigation. *Natural hazards* 28(2-3), 211–228.
- Pelzer, D., J. Xiao, D. Zehe, M. H. Lees, A. C. Knoll, and H. Aydt (2015). A partition-based match making algorithm for dynamic ridesharing. *IEEE Transactions on Intelligent Transportation Systems* 16(5), 2587–2598.
- Pinson, C., H. M. Afsar, and C. Prodhon (2016). Heuristic approaches to solve a generalized dial-a-ride problem applied to car-pooling. *IFAC-PapersOnLine* 49(12), 1187–1191.
- Polacek, M., K. F. Doerner, R. F. Hartl, and V. Maniezzo (2008). A variable neighborhood search for the capacitated arc routing problem with intermediate facilities. *Journal of Heuristics* 14(5), 405–423.
- Polacek, M., R. F. Hartl, K. Doerner, and M. Reimann (2004). A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of heuristics* 10(6), 613–627.
- Polat, O., C. B. Kalayci, O. Kulak, and H.-O. Günther (2015). A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. *European Journal of Operational Research* 242(2), 369–382.
- Potvin, J.-Y. and S. Bengio (1996). The vehicle routing problem with time windows part ii: genetic search. *INFORMS journal on Computing* 8(2), 165–172.

- Rawls, C. G. and M. A. Turnquist (2010). Pre-positioning of emergency supplies for disaster response. *Transportation research part B: Methodological* 44(4), 521–534.
- Renaud, J., G. Laporte, and F. F. Boctor (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research* 23(3), 229–235.
- Salani, M. and I. Vacca (2011). Branch and price for the vehicle routing problem with discrete split deliveries and time windows. *European Journal of Operational Research* 213(3), 470–477.
- Salmerón, J. and A. Apte (2010). Stochastic optimization for natural disaster asset prepositioning. *Production and Operations Management* 19(5), 561–574.
- Santos, D. O. and E. C. Xavier (2015). Taxi and ride sharing: a dynamic dial-a-ride problem with money as an incentive. *Expert Systems with Applications* 42(19), 6728–6737.
- Sarasola, B., K. F. Doerner, V. Schmid, and E. Alba (2016). Variable neighborhood search for the stochastic and dynamic vehicle routing problem. *Annals of Operations Research* 236(2), 425–461.
- Shen, W., C. V. Lopes, and J. W. Crandall (2016). An online mechanism for ridesharing in autonomous mobility-on-demand systems. *arXiv preprint arXiv:1603.02208*.
- Shen, Z., M. M. Dessouky, and F. Ordóñez (2009). A two-stage vehicle routing model for large-scale bioterrorism emergencies. *Networks* 54(4), 255–269.
- Sheu, J.-B. (2007). An emergency logistics distribution approach for quick response to urgent relief demand in disasters. *Transportation Research Part E: Logistics and Transportation Review* 43(6), 687–709.
- Sim, M. (2004). *Robust optimization*. Ph. D. thesis, Massachusetts Institute of Technology.
- Smith, A. E., D. W. Coit, T. Baeck, D. Fogel, and Z. Michalewicz (1997). Penalty functions. *Handbook on Evolutionary Computation, pages C 5*, 1–6.
- Solano-Charris, E. L., C. Prins, and A. C. Santos (2014). A robust optimization approach for the vehicle routing problem with uncertain travel cost. In *Control, Decision and Information Technologies (CoDIT), 2014 International Conference on*, pp. 098–103. IEEE.
- Soyster, A. L. (1973). Technical noteconvex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research* 21(5), 1154–1157.
- Stiglic, M., N. Agatz, M. Savelsbergh, and M. Gradisar (2016). Making dynamic ride-sharing work: The impact of driver and rider flexibility. *Transportation Research Part E: Logistics and Transportation Review* 91, 190–207.
- Sungur, I., F. Ordóñez, and M. Dessouky (2008). A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions* 40(5), 509–523.
- Sze, J. F., S. Salhi, and N. Wassan (2016). A hybridisation of adaptive variable neighbourhood search and large neighbourhood search: Application to the vehicle routing problem. *Expert Systems with Applications* 65, 383–397.
- Sze, J. F., S. Salhi, and N. Wassan (2017). The cumulative capacitated vehicle routing problem with min-sum and min-max objectives: An effective hybridisation of adaptive variable neighbourhood search and large neighbourhood search. *Transportation Research Part B: Methodological* 101, 162–184.
- Taillard, É., P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science* 31(2), 170–186.

- Tang, H. and E. Miller-Hooks (2005). A tabu search heuristic for the team orienteering problem. *Computers & Operations Research* 32(6), 1379–1407.
- Tomasini, R. M. and L. N. Van Wassenhove (2004). Pan-american health organization’s humanitarian supply management system: de-politicization of the humanitarian supply chain by creating accountability. *Journal of public procurement* 4, 437–449.
- Toth, P. and D. Vigo (2002a). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics* 123(1), 487–512.
- Toth, P. and D. Vigo (2002b). *The vehicle routing problem*. SIAM.
- Toth, P. and D. Vigo (2014). *Vehicle routing: problems, methods, and applications*. SIAM.
- Van Hentenryck, P., R. Bent, and C. Coffrin (2010). Strategic planning for disaster recovery with stochastic last mile distribution. In *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*, pp. 318–333. Springer.
- Wang, H., L. Du, and S. Ma (2014). Multi-objective open location-routing model with split delivery for optimized relief distribution in post-earthquake. *Transportation Research Part E: Logistics and Transportation Review* 69, 160–179.
- Wang, X., N. Agatz, and A. Erera (2017). Stable matching for dynamic ride-sharing systems. *Transportation Science*.
- Wassan, N. (2007). Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls. *Journal of the Operational Research Society* 58(12), 1630–1641.
- Wassan, N. A. and I. H. Osman (2002). Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society*, 768–782.
- Wei, L., Z. Zhang, D. Zhang, and A. Lim (2015). A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research* 243(3), 798–814.
- Widener, M. J. and M. W. Horner (2011). A hierarchical approach to modeling hurricane disaster relief goods distribution. *Journal of Transport Geography* 19(4), 821–828.
- Winter, S. and S. Nittel (2006). Ad hoc shared-ride trip planning by mobile geosensor networks. *International Journal of Geographical Information Science* 20(8), 899–916.
- Yi, W. and A. Kumar (2007). Ant colony optimization for disaster relief operations. *Transportation Research Part E: Logistics and Transportation Review* 43(6), 660–672.
- Yi, W. and L. Özdamar (2007). A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research* 179(3), 1177–1193.
- Yousaf, J., J. Li, L. Chen, J. Tang, and X. Dai (2014). Generalized multipath planning model for ride-sharing systems. *Frontiers of Computer Science* 8(1), 100–118.
- Yu, Y., C. Chu, H. Chen, and F. Chu (2012). Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research* 197(1), 135–158.
- Zhang, J., D. Wen, and S. Zeng (2016). A discounted trade reduction mechanism for dynamic ridesharing pricing. *IEEE Transactions on Intelligent Transportation Systems* 17(6), 1586–1595.
- Zhu, J., J. Huang, D. Liu, and J. Han (2008). Resources allocation problem for local reserve depots in disaster management based on scenario analysis. In *The 7th International Symposium on Operations Research and its Applications*. Lijiang, China, pp. 395–407.