POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Robust Optimization for Supply Chain Applications: Facility Location and Drone Delivery Problems

CHUN CHENG
Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Mathématiques

Août 2020

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Robust Optimization for Supply Chain Applications: Facility Location and Drone Delivery Problems

présentée par **Chun CHENG**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Michel GENDREAU, président
Louis-Martin ROUSSEAU, membre et directeur de recherche
Yossiri ADULYASAK, membre et codirecteur de recherche
Erick DELAGE, membre
Pascal VAN HENTENRYCK, membre externe

DEDICATION

To the uncertain future

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors, Professors Louis-Martin Rousseau and Yossiri Adulyasak, for their constant supports and patient guidance. Their supports make it possible for me to explore different research areas during my Ph.D. study. They are knowledgeable—they can always give me helpful feedbacks even though we have worked on various research topics. Four years ago, when I was admitted to the Ph.D. program at Polytechnique, Louis-Martin said that I would have a great time in Montreal in one of our emails. Time flies and it comes out that he was right—the four years here are the happiest time of my life. However, I do realize that my happiness mainly owes to the smooth progress of my research projects, and to the supports my supervisors have provided. Besides guidance in research, they also give me wise suggestions for academic career preparation. In a word, I could not have imagined better supervisors for my Ph.D. study.

I am also grateful to Professors Melvyn Sim and Patrick Jaillet for hosting me as a visiting student at NUS and MIT respectively. These experiences are unique and valuable. It is a great honor to work with them. In addition, I would like to thank Professors Michel Gendreau, Erick Delage, and Pascal Van Hentenryck for accepting to be members of my dissertation committee.

Moreover, I acknowledge the Institute for data valorization (IVADO) for providing financial support for my Ph.D. study, the Fonds de recherche du Québec − Nature et technologies (FRQNT) for supporting my visit at NUS, the Fondation et Alumni de Polytechnique Montréal and Mitacs for supporting my visit at MIT, and the Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT) for providing me several scholarships. I also thank my supervisor Louis-Martin Rousseau for writing several recommendation letters to support my applications for these fundings and scholarships.

Lastly, I would like to thank my parents for their endless love. I also thank my younger sister and brother-in-law for taking care of my parents such that I can study abroad with no worries. My sincere thanks also go to my friends in China and Montreal for their support and company.

RÉSUMÉ

Les décisions concernant la localisation des infrastructures dans les chaînes d'approvisionnement sont d'une importance stratégique: la construction d'une nouvelle infrastructure est généralement coûteuse et l'impact de cette décision est durable. Une fois qu'une nouvelle installation sera ouverte, elle devrait rester opérationnelle pendant plusieurs années. Cependant, des facteurs environnementaux, tels que les déplacements de population et les catastrophes naturelles, peuvent affecter le fonctionnement des installations. Par exemple, le déplacement de la population peut modifier les modèles de demande, ce qui influence davantage les décisions d'allocation entre les clients et les installations. Les catastrophes naturelles peuvent diminuer partiellement ou complètement la capacité d'une installation, entraînant des décisions de réaffectation ou des pertes de ventes. Toutes ces incertitudes peuvent faire en sorte qu'une décision optimale d'aujourd'hui ne donne pas de bons résultats à l'avenir. Ainsi, il est important de considérer les incertitudes potentielles dans la phase de conception des chaînes d'approvisionnements, tout en prenant explicitement en compte les réaffectations possibles des clients comme décisions de recours dans la phase d'exécution.

Dans la première moitié de cette thèse, nous étudions trois problèmes de localisation d'établissements sous risques de perturbations, où chaque travail a un objectif différent. Plus précisément, l'étude du chapitre 3 se concentre principalement sur l'amélioration des algorithmes; le travail du chapitre 4 considère simultanément plusieurs types d'incertitudes; et le chapitre 5 étudie un problème de conception de réseau à trois échelons soumis à des perturbations. Nous adoptons des méthodes d'optimisation robuste (OR) en deux étapes, où les décisions de localisation des installations sont prises *ici et maintenant* et les décisions de recours pour réaffecter les clients sont prises après que les informations d'incertitude sur la disponibilité des installations et la demande des clients ont été révélées. Nous implémentons des méthodes exactes et approximatives pour résoudre les modèles robustes. Les résultats démontrent que le cadre OR proposé peut améliorer la fiabilité des systèmes de chaîne d'approvisionnement avec seulement une légère augmentation du coût normal (le coût du scénario sans interruption). Les différents modèles construits dans cette thèse peuvent également être utilisés comme outils d'aide à la décision pour voir le compromis entre coût et fiabilité.

Outre la planification stratégique, nous avons étudié également les problèmes de niveau opérationnel : problèmes de livraison à l'aide de drones. La livraison par drone est connue comme

contributeur potentiel à l'amélioration de l'efficacité et à la résolution des problèmes de livraison du dernier kilomètre. Pour cette raison, le routage des drones est devenu un domaine de recherche très actif ces dernières années. Contrairement au problème de routage des véhicules, cependant, la conception des itinéraires des drones est difficile en raison de multiples caractéristiques opérationnelles, notamment les opérations multi-voyages, la planification de la recharge et le calcul de la consommation d'énergie. Pour combler certaines lacunes importantes dans la littérature, le chapitre 6 résout un problème de routage de drone multi-voyages, où la consommation d'énergie des drones est affectée par la charge utile et la distance de déplacement alors que de telles relations sont non linéaires. Pour aborder la fonction d'énergie non linéaire (convexe), nous proposons deux types de coupes (cuts) qui sont incorporées dans le schéma de branchement et de coupes (branch-and-cut). Nous utilisons une formulation à 2 indices pour modéliser le problème et également générer des instances de référence pour l'évaluation d'algorithmes. Les tests numériques indiquent que même si le modèle d'origine est non linéaire, notre approche est efficace à la fois en termes d'algorithme et de qualité de solution.

La livraison par drones peut également être affectée par diverses incertitudes, telles que des conditions de vent incertaines et des obstacles imprévisibles. Motivé par les problèmes de retard des drones résultant de l'incertitude du vent, notre travail dans le chapitre 7 vise à optimiser de manière robuste le risque de retard pour un problème de programmation de drones avec des temps de voyage incertains. À cette fin, nous utilisons un cadre d'optimisation robuste aux distributions pour modéliser le problème. Comme les données historiques sur le vent sont souvent disponibles, nous utilisons des techniques d'apprentissage automatique pour partitionner les données pour la construction de l'ensemble d'ambiguïté. À partir des données météorologiques réelles, nous observons que les conditions de vent l'après-midi dépendent des conditions de vent du matin. Par conséquent, nous proposons une description de l'ambiguïté en ensemble à deux périodes pour modéliser la distribution conjointe des temps de parcours incertains. Nous proposons également un modèle de planification des drones à deux périodes, où les décisions de programmation dans l'après-midi s'adapteraient aux résultats des informations météorologiques observées le matin. En utilisant des données météorologiques réelles, nous validons que le modèle d'optimisation robuste adaptatif peut réduire efficacement le retard dans les tests hors échantillon par rapport à d'autres méthodes de référence.

ABSTRACT

Facility location decision is strategic: The construction of a new facility is typically costly and the impact of the decision is long-lasting. Once a new facility is opened, it is expected to remain in operation for several years. However, environmental factors, such as population shift and natural disasters, may affect facilities' operations. For example, population shift may change demand patterns, which further influence the allocation decisions between customers and facilities. Natural disasters may diminish a facility's capacity partially or completely, resulting in reassignment decisions or lost sales. All these uncertainties may cause today's optimal decision to perform poorly in the future. Thus, it is important to consider potential uncertainties in the supply chain design phase, while explicitly taking into account the possible customer reassignments as recourse decisions in the execution phase.

In the first half of this thesis, we study three facility location problems under disruption risks, where each work has a different focus. Specifically, the study in Chapter 3 mainly focuses on algorithmic improvement; the work in Chapter 4 considers multiple types of uncertainties simultaneously; and Chapter 5 studies a three-echelon network design problem under disruptions. We adopt the two-stage robust optimization (RO) method for these problems, where facility location decisions are made *here-and-now* and recourse decisions to reassign customers are made after the uncertainty information on the facility availability and customer demand has been revealed. We implement both exact and approximate methods to solve the robust models. Results demonstrate that the proposed RO framework can improve supply chain systems' reliability with only a slight increase in the nominal cost (the cost of the disruption-free scenario). The various robust models constructed in this thesis can also be used as decision support tools to see the trade-off between cost and reliability.

Besides strategic planning, we also study operational level problems in this thesis—drone delivery problems. Drone delivery is known as a potential contributor in improving efficiency and alleviating last-mile delivery problems. For this reason, drone routing and scheduling has become a highly active area of research in recent years. Unlike the vehicle routing problem, however, designing drones' routes is challenging due to multiple operational characteristics including multi-trip operations, recharge planning, and energy consumption calculation. To fill some important gaps in the literature, Chapter 6 solves a multi-trip drone routing problem, where drones' energy consumption is affected by payload and travel distance whereas such relationships are nonlinear. To tackle the nonlinear (convex) energy function, we propose two types of cuts that are incorporated into the branch-and-cut scheme. We use a

2-index formulation to model the problem and also generate benchmark instances for algorithm evaluation. Numerical tests indicate that even though the original model is nonlinear, our approach is effective in both computational efficiency and solution quality.

Drone delivery can also be affected by various uncertainties, such as uncertain wind conditions and unpredictable obstacles. Motivated by the drone lateness issues resulting from wind uncertainty, our work in Chapter 7 aims to robustly optimize the lateness risk for a drone scheduling problem with uncertain travel times. To that end, we use a distributionally robust optimization framework to model the problem. As historical wind data is often available, we use machine learning techniques to partition the data for the construction of the ambiguity set. From the actual weather data, we observe that the wind conditions in the afternoon are dependent on the wind conditions in the morning. Accordingly, we propose a two-period cluster-wise ambiguity set to model the joint distribution of uncertain travel times. We also propose a two-period drone scheduling model, where the scheduling decisions in the afternoon would adapt to the outcome of the weather information observed in the morning. Using actual weather data, we validate that the adaptive robust optimization model can effectively reduce lateness in out-of-sample tests in comparison with other benchmark methods.

Keyword: Facility location; disruption risk; demand uncertainty; two-stage robust optimization; column-and-constraint generation; drone delivery; nonlinear energy consumption; branch-and-cut; uncertain weather condition; cluster-wise ambiguity set; distributionally robust optimization

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

RO	Robust Optimization
BD	Benders Decomposition
C&CG	Column-and-Constraint Generation
FLP	Fixed-Charge Location Problem
LNDP	Logistics Network Design Problem
UAVs	Unmanned Aerial Vehicles
MTDRP	Multi-trip Drone Routing Problem
B&C	Branch-and-Cut
PMP	p-median Problem
UFLP	Uncapacitated Fixed-Charge Location Problem
MIP	Mixed-Integer Programming
DRO	Distributionally Robust Optimization
LTP	Location-Transportation Problem
AP	Affine Policy
MILP	Mixed-Integer Linear Programming
DC	Distribution Center
MTVRP	Multi-trip Vehicle Routing Problem
VRP	Vehicle Routing Problem
SA	Simulated Annealing
CO_2	Carbon Dioxide
FSTSP	Flying Sidekick Traveling Salesman Problem
PDSTSP	Parallel Drone Scheduling Traveling Salesman Problem
DP	Dynamic Programming
B&B	Branch-and-Bound
B&P	Branch-and-Price
MTVRPTW	MTVRP with Time Windows
DRP	Drone Routing Problem
CFLP	Capacitated Fixed-Charge Location Problem
LP	Linear Programming
ARC	Adjustable Robust Counterpart
MCFP	Minimum Cost Flow Problem
KKT	Karush–Kuhn–Tucker
LDR	Linear Decision Rule

AARC	Affinely Adjustable Robust Counterpart
UPMP	Uncapacitated p-median Problem
CPMP	Capacitated p-median Problem
MP	Master Problem
SP	Stochastic Programming
G-LNDP	Generic LNDP
SECs	Subtour Elimination Constraints
ERI	Essential Riskiness Index
VRPTW	Vehicle Routing Problem with Time Window

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Supply chain management involves three levels of decisions: strategic, tactical, and operational. Strategic planning includes product development, system design, and customer identification, among others. Tactical level management includes material procurements, production schedules, inventory management and so forth. Operational level decisions involve detailed management of production operations, logistics activities, etc. In this thesis, we study both strategic and operational level problems. Specifically, in the first half of this thesis, we consider three facility location problems under uncertainty. In the second half, we examine both the deterministic and stochastic drone delivery problems.

1.1 Facility Location Problem Under Uncertainty

A facility location problem looks for the optimal placement of facilities to satisfy requests of service from a given set of customers. Several questions should be addressed in this problem. For example, how many facilities should be located? Where should these facilities be located? How large should each facility be? How should we assign customers to opened facilities? The answers to these questions depend on the optimization objective (e.g., minimization of cost or maximization of profit) of the considered system and the available resources. Facility location is an important aspect of strategic planning for both private companies and public sectors [3]. The construction and acquisition of a new facility is typically a costly and time-consuming process. Therefore, once a facility is built, it is expected to remain in operation for several years. However, during the lifetime of a facility, it may face various uncertainties, among which some can interrupt its operation and make it fail to serve assigned customers. For example, events such as power outages and natural disasters may destroy a facility partially or completely such that its residual capacity is not sufficient to serve its assigned customers. Other environmental changes, such as population shift and economical issues, may cause customers' demand to deviate significantly from their nominal values such that opened facilities are not able to meet increased demand or some facilities' capacities are wasted if the demand decreases. To conclude, uncertain factors can turn today's optimal location decisions into tomorrow's poor performance. It is therefore critical to consider potential uncertainties proactively at the planning stage, to avoid high recourse costs at the operational stage.

In the literature, many probabilistic models have been developed for the facility location problem under disruptions, where the failure probability of each facility is known in advance. The sum of the facility location cost and the expected transportation cost is minimized

[4–7]. However, for rare events, it may be impossible to obtain or predict precise probability information because of insufficient historical data or inaccurate forecasting methods. In such circumstances, robust optimization (RO) methods can be used to find a solution that protects the decision-makers against parameter ambiguity and stochastic uncertainty without depending on probability information [8]. RO uses uncertainty sets to represent the random data; therefore, any identified solution is immune to all the possible realizations within an uncertainty set. In addition, whereas the static RO method determines only *here-and-now* decisions, the two-stage adjustable RO method is capable of generating less conservative solutions, because it allows *wait-and-see* decisions that can adapt to the realized observations. However, this flexibility comes with significant computational challenges. Several solution methods, such as the Benders decomposition (BD) method [9–13] and the column-and-constraint generation (C&CG) algorithm [14,15], have been developed to solve two-stage RO models exactly. Approximation schemes, such as affine decision rules [16,17] and piecewise affine decision rules [18], are also used.

In this thesis, we study three facility location problems under uncertainties, where two-stage RO methods are used to solve these problems. The first one is a fixed charge location problem (FLP) under disruptions, where our main focus is algorithm improvement for the robust models. The second one considers both disruption risk and uncertain demand in the FLP. For this variant, we study the impacts of multiple uncertainties and further extend the proposed modeling and solution schemes to other types of reliable facility location problems. The third one is a three-echelon logistics network design problem (LNDP), where both supply and transshipment nodes are subject to disruptions. For this problem, we propose three two-stage RO models with different objective functions and performance bounds to reduce the conservativeness of robust solutions.

1.2 Drone Delivery Problem

In recent years, unmanned aerial vehicles (UAVs) or drones have attracted people's attention, especially since 2013 when Amazon announced their Prime Air UAV [19]. Other companies, like DHL, Google, and Alibaba also began developing their own drones, because they believe drones have the potential to reduce cost and waiting time for last-mile delivery. The development of technology has made this idea possible. For example, carbon fiber manufacturing costs have decreased dramatically during the past few years, which enable stronger and lighter air frames [20]; lithium polymer batteries with high energy density are also now available, which help extend drones' flight range [21]. Different companies have designed different drone models, notably, the multirotor drones used by UPS and DHL, and the hy-

brid drones developed by Amazon and Alphabet. Being similar to the multirotor helicopters, multirotor drones are lifted and propelled by rotors. Hybrid drones can take off and land vertically (like helicopters), but use wing or wing-like surfaces to generate lift. Meanwhile they can also perform horizontal maneuvers like airplanes. On October 18, 2019, Alphabet's drone unit Wing launched the first commercial drone delivery flight in the United States [22].

Compared to trucks, drones have some specific advantages: (i) They can save labor, because no drivers (or pilots) are needed. (ii) They can often travel faster than trucks. (iii) They are not restricted to road networks [23]. These merits enable logistics companies and on-line stores to use drones for rapid parcel delivery. Humanitarian organizations are also considering using drones in disaster scenarios. For example, in the immediate aftermath of a disaster, drones can provide support with risk assessment, mapping, and temporary communication network creation [24]. In situations where the transportation network is severely compromised by natural disasters, drones can deliver emergency supplies to affected regions. In addition, by taking traffic off the roads, drone might reduce negative implications on congestion, safety, and the environment [25].

On the other hand, some unique characteristics of drones have presented new operational challenges. Limited battery capacity influences a drone's flight duration, which can also be affected by payload and speed [2]. Therefore, how should we represent the relationship between battery energy consumption and various factors which affect it? How should we route drones so that they can safely return after visiting designated sites? Furthermore, drones' payload is also limited, which means that a drone can only visit a small number of customers during a trip. Thus, how should we schedule drones to serve more demands to maximize their use?

In addition, drone delivery can also be affected by weather conditions. For example, cold temperatures might cause drones' energy capacity to drop quickly. Wind may lead to late deliveries or even cancellations of services. In October 2014, DHL planned to use drones to deliver medical supplies from the city of Norden to a remote island off the country's coast, but finally they cancelled the schedule due to wind. They had postponed the first flight schedule on September 25 for the same reason, although they had performed approximately 50 experiments to gather information for the optimization of their drone models before the planned medical delivery [26]. Other authors [27,28] suggest that delays caused by weather conditions should be overcome before drone-based order delivery becomes a reality. Some authors [29] further state that the future of drone delivery depends on its capability to adapt to different scenarios of weather conditions, and drone delivery companies, such as Zipline, attempt to build models which allow drones to safely and efficiently operate under different

weather scenarios.

Although various operational challenges arise, current research about drones still focuses only on engineering issues and studies addressing these other challenges are scarce [30]. Motivated by these issues, the second half of this thesis studies two drone delivery problems. The first one is a multi-trip drone routing problem (MTDRP), where a branch-and-cut (B&C) algorithm is used to solve the problem and various cuts are proposed to calculate the complex nonlinear energy consumption. The second one is a drone scheduling problem with uncertain weather conditions. For this problem, we aim to provide a robust formulation to decide drone delivery schedules that minimize the lateness risk at customers.

1.3 Organization of the Thesis

The studies presented in this thesis have yielded five scientific articles as follows.

1. Cheng, C., Qi, M., Zhang, Y., Rousseau, L.-M., 2018. A Two-Stage Robust Approach for the Reliable Logistics Network Design Problem. *Transportation Research Part B: Methodological*, 111, 185-202.

2. Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2020. Drone Routing with Energy Function: Formulation and Exact Algorithm. *Transportation Research Part B: Methodological*, 139, 364-387.

3. Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2018. Robust Facility Location Under Disruptions. GERAD Technical Report, G-2018-91, 28 pages. Major revision at *INFORMS Journal on Optimization*.

4. Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2019. Robust Facility Location Under Demand Uncertainty and Facility Disruptions. CIRRELT Technical Report, CIRRELT-2019-53, 23 pages. Major revision at *Omega: The International Journal of Management Science*.

5. Cheng, C., Adulyasak, Y., Rousseau, L.-M., Sim, M., 2020. Robust Drone Delivery with Weather Information. Available at `http://www.optimization-online.org/DB_FILE/2020/07/7897.pdf`. Submitted to *Operations Research*.

At the beginning of each chapter, we give details on the chapters that are part of the articles.

The remainder of this thesis is organized as follows. Chapter 2 reviews related literature on facility location and drone delivery problems. Chapter 3 studies the facility location problem

under disruptions. Chapter 4 examines the facility location problem under demand uncertainty and facility disruptions. We further study a three-echelon LNDP under disruptions in Chapter 5. This is followed by the MTDRP in Chapter 6. The drone delivery problem with uncertain weather is studied in Chapter 7. We conclude this thesis in Chapter 8.

CHAPTER 2 LITERATURE REVIEW

This chapter reviews related works on facility location problems under uncertainty and drone delivery problems.

2.1 Facility Location Under Uncertainty

Flexible supply chains are expected to adapt effectively to supply disruptions and demand changes while maintaining customer service levels [31]. To mitigate supply chain risks, several strategies can be utilized to add redundancy to the system, e.g., inventory management, sourcing flexibility, demand substitution, and facility location [32,33]. In this thesis, we use facility location to enhance supply chain flexibility, and thus related works on facility location under uncertainties are reviewed. For deterministic facility location problems, where all the parameters are known perfectly at the time of making decisions, interested readers can refer to the textbook [34].

In facility location problems, the uncertainty can be generally classified into three types: provider-side uncertainty, receiver-side uncertainty, and in-between uncertainty [35]. The provider-side uncertainty includes uncertain facility capacity, status (operational or failed), and supply lead time. The receiver-side uncertainty captures randomness in demand. The in-between uncertainty involves uncertain transportation costs/times and arc status. These three types of uncertainty have been widely considered in the literature.

Provider-side Uncertainty. Supply chain disruption is not a new concept; it has existed as long as the supply chain itself. However, in recent years it has received increasing attention. The authors of [32] give four reasons for the explosion of interest: (1) high-profile events, such as the 9/11 terrorist attack in the United States and the Japanese earthquake, have brought disruption to the forefront of public attention; (2) the "just in time" concept leaves little room for adjustment, and this significantly exacerbates the impact of disruption; (3) with the development of the global supply chain, suppliers are more integrated and some are located in economically or politically unstable regions; (4) as with any other maturing research area, scholars study this topic because of its high profile.

The reference [36] was the first to consider disruptions in facility location models and the author introduced the unreliable p-median problem (PMP) and the (p, q)-center problem. In the former, each facility has a given probability of being inactive; in the latter, p facilities must be built to minimize the maximum cost when at most q facilities are disrupted. The

authors of [4] introduce the reliable PMP and the reliable uncapacitated fixed-charge location problem (UFLP), where all the facilities have the same disruption probability. They minimize the weighted sum of the nominal cost and the expected transportation cost of disruption scenarios. In [5], the authors relax the constraint that all the facilities have the same disruption probability and consider site-dependent probabilities in the UFLP. They propose a mixed-integer programming (MIP) formulation and a continuous approximation model. The MIP model is solved by Lagrangian relaxation. The reference [6] studies a reliable inventory-location problem. The authors assume that each facility has an equal probability of disruption. Each customer may receive service from a sequence of $R \geq 1$ facilities, i.e., in the normal scenario a customer is serviced by its level-1 facility, and when its level-r facility fails, it will be assigned to its level-$(r+1)$ facility. If all R facilities fail, the customer will not be serviced, and there is an associated penalty. The authors of [7] present a reliable location-routing problem. Their problem setting is similar to that of [6], except that they make routing decisions instead of inventory policies. The reference [35] studies a reliable UFLP, which is first formulated as a two-stage stochastic program and then as a nonlinear integer program. Several heuristics are developed for this problem. The authors of [37] study the UFLP in a stochastic optimization framework by incorporating risk preferences. They propose conditional value-at-risk- and absolute semideviation-based models to control the risk of transportation cost at each customer. The authors of [38] study a reliable PMP and a reliable UFLP, where each facility has a heterogeneous failure probability and each customer is allocated to a primary facility and a back-up facility. They propose an evolutionary learning algorithm to solve the problem. The reference [39] studies a reliable service network design problem, where customers must pass certain network access points to reach facilities for services. The authors assume that each network access point is associated with a site-dependent failure probability and minimize the expected system cost. The work in [40] addresses the reliable UFLP with correlated disruptions, which is solved by Lagrangian relaxation based algorithms. The authors of [41] explore the reliable supply chain network design problem with partial disruptions. A two-stage stochastic programming model is developed, which is solved by a Benders decomposition method. The reference [42] studies a reliable PMP. The authors of [43] propose a distributionally RO (DRO) model for the UFLP with correlated disruptions. They assume that the exact probability distribution of disruption scenarios is unknown but lies in a distributional uncertainty set, such that the marginal disruption probability of a site is equal to a given value. The expected cost under the worst-case distribution is minimized. For more details on reliable facility location, see the review papers [32] and [44].

Receiver-side Uncertainty. For early works of facility location under demand uncertainty,

see the review paper [45]. In reference [46], a multi-period facility location problem under demand uncertainty is studied, where a box uncertainty set and an ellipsoid uncertainty set are used. The authors of [47] are the first to study the two-stage robust location-transportation problem (LTP), and a cutting plane algorithm is applied. They compare solutions generated by the two-stage RO method, one-stage (also known as static) RO method, and the two-stage stochastic programming method. For the same problem, the study in [14] focuses on comparing the performance of the C&CG algorithm and the Benders-style cutting plane method. The reference [18] studies a multi-period LTP and develops various approximation schemes to solve the problem based on the affine policy (AP). The authors of [48] study a multi-period LTP with multiple items and integer-valued demand uncertainty. Adjustable RO formulations are developed for the problem. The reference [49] studies the robust facility location under decision-dependent uncertain demand. Specifically, the demand at a customer node increases if there are new facilities opened in the customer's neighborhood. The DRO method is used to model the problem and the resulting model is reformulated to a mixed-integer linear programming (MILP) model, which is further enhanced by valid inequalities. The authors of [50] develop a DRO model for the uncapacitated facility location problem under demand uncertainty, where the distribution of customer demand is assumed to lie in an ambiguity set. The authors of [51] extend the work in [50] to the capacitated cases, where a two-stage DRO method is used.

In-between Uncertainty. The in-between uncertainty captures the randomness of the links between the provider-side and the receiver-side. It might be uncertain travel times or costs between the suppliers and the customers, or the failures of links resulting from unexpected events. The reference [52] studies a p-hub center location problem under uncertain travel times. A chance constrained programming approach is used and the deterministic equivalent model is solved by a genetic algorithm. The authors of [53] consider a two-echelon facility location problem, where products are first delivered to depots and then from depots to customers. The transportation costs in both echelons are uncertain. They use budgeted uncertainty sets and a static RO method for the problem. In [54], a single-commodity flow network design problem with multiple concurrent edge failures is addressed. The authors formulate the problem as a two-stage RO model and solve it with a C&CG algorithm.

Simultaneous Consideration of Multiple Uncertainties. There also exist studies that consider multiple types of uncertainties simultaneously. For example, the reference [55] studies a closed-loop supply chain network design problem, where box uncertainty sets are used to describe the randomness in demand, returns, and transportation costs. The authors of [56] study a network design problem in the post-disaster environment, where demand- and network-related uncertainties are present. They use a two-stage stochastic programming

method to model the problem, and develop a Benders decomposition-based branch-and-cut algorithm to solve it. The reference [57] considers both uncertain demand and transportation costs in uncapacitated hub location problems. The authors use budgeted uncertainty sets to characterize both uncertainties and the duality technique to reformulate the static robust models. For the same types of uncertainties, the work of [58] applies the DRO method to both the uncapacitated and capacitated hub location problems. The authors of [59] study a supply chain network design problem, considering supply-side and demand-side uncertainties simultaneously. They assume demand variables follow a known distribution function and describe supply-side uncertainty (facility disruptions and link failures) through scenarios. The reference [60] studies a humanitarian network design problem, which includes suppliers, relief distribution centers, and affected areas. Installation costs of distribution centers, shortage costs at affected areas, transportation costs at both echelons, supply capacity, and demand are subject to uncertainties. The authors use budgeted uncertainty sets to describe randomnesses and a static RO method to model the problem. The authors of [61] apply the two-stage RO method to optimize the operations of pre- and post- disasters. In particular, decision-makers locate facilities and pre-position inventories before a disaster. After a disaster, they deliver commodities to affected regions. The authors assume the demand of commodities, the residual inventories (caused by partial disruptions of facilities), and the capacities of transportation arcs to be uncertain. The reference [62] considers a facility location problem in global sourcing under arc disruptions and uncertain demand, which is solved by a two-stage RO scheme.

Network Design Under Disruptions. Although supply chain disruption has received extensive attention, research into disruptions in the context of network design is scarce [32]. The reliable LNDP extends the reliable facility location problem by considering multiple echelons—supplier nodes, transshipment nodes, and demand nodes, and by allowing both the supplier and transshipment nodes to be destroyed.

The authors of [63] propose several scenario-based models (each scenario has an occurrence probability) for designing supply chains that are resilient to disruptions. They first present a reliable network design model for a network that will be built from scratch. For existing networks, they provide fortification models and indicate that the reliability of the existing facilities can be enhanced by investing in protection and security measures. The reference [64] studies a reliable LNDP with a $p-$robustness criterion, the objective of which is to minimize the nominal cost. The authors propose a scenario-based MIP model and develop a hybrid genetic algorithm. In their numerical tests they randomly generate several scenarios, where each facility has a 10% probability of becoming disabled. The authors of [65] consider a capacitated supply chain network design model in which both the facilities and the trans-

portation network have a given probability of disruptions. They assume that the facilities are partially destroyed when disruptions occur and that the customers of a disrupted distribution center (DC) are not assigned to other DCs; instead, the capacity lost at the disrupted DC is replenished from non-disrupted DCs. They formulate a linear MIP model and propose a modified BD method. The reference [66] studies a reliable facility location/network design problem with a constraint on the maximum allowable failure cost. The facilities and network links are assumed to be uncapacitated. At most one facility fails at a time, and the demand nodes served by the disrupted facility must be reallocated to the nearest surviving facility. The objective is to minimize the transportation cost. The work in [67] studies the influence of disruptions on the competitiveness of supply chains and proposes three policies to mitigate the disruption risk.

From the literature, we can get the following observations:

1. Most studies assume that the probability information of disruptions is known perfectly a priori; only a few papers consider distribution-free methods. However, in realistic applications, it is often difficult to identify the distribution of disruptions, especially those caused by natural disasters. Thus, more distribution-free frameworks are needed to describe the randomness of disruptions and solve the resulting models.

2. Most works consider one type of uncertainty at a time. Although some papers consider multiple types of uncertainties simultaneously, their modeling schemes may produce overly conservative solutions as all the decisions are made *here-and-now* [55,60], or relatively optimistic solutions because it is impossible to enumerate all the disruption scenarios [59]. Thus, adaptive methods with more accurate description of disruptions are needed to model systems with multiple types of uncertainties.

3. Research on reliable network design is relatively scarce, especially in the framework of RO. Therefore, more works are needed for the reliable design of multi-echelon systems, where facilities at different levels may expose to disruptions.

2.2 Drone Delivery Problem

Besides operational challenges like multi-trip planning and energy consumption calculation, a drone delivery system has to mitigate the risks of weather uncertainty. Uncertain wind conditions, i.e., speeds and directions of the wind, could affect the transit times of the drones to their destinations, leading to late deliveries or even cancellations of services [26]. Some reports [27–29] indicate that the future of drone delivery depends on drones' capabilities to

adapt to different weather scenarios. Thus, we divide the literature reviewed here into two categories: drone delivery without and with consideration of weather influence.

2.2.1 Drone Delivery Without Consideration of Weather Influence

Since most studies do not consider the influence of weather conditions, we further divide the literature on drone delivery into two subcategories: drone-only problems and truck-drone problems. For drone-only problems, only drones are used in the delivery system. For truck-drone problems, both trucks (one or multiple) and drones are used simultaneously. A truck can be used either as a tool to carry drones (i.e., the truck does not have delivery tasks) or for both delivery tasks and as a temporary hub to launch/retrieve drones. Trucks and drones can also work in parallel making deliveries. As our work in Chapter 6 is also closely related to the multi-trip vehicle routing problem (MTVRP), we also review related works on MTVRP here.

Drone-only Problems. Studies on drone-only delivery systems normally assume that there are multiple drones and that each drone can cover one or several customers per trip. The authors of [68] study an automated drone delivery system, where all customers' demands are the same. They use the relationship among battery capacity, payload, and flight range to optimize the drone fleet size. The reference [69] describes the implementation steps used to assign a fleet of heterogeneous UAVs to deliver items to target locations. Each order placed by a customer can include one or multiple items. Because of drones' limited payload, one order may not be completely fulfilled in one trip; thus, multiple deliveries might be required. The authors use a genetic algorithm to solve the problem, where a multi-dimensional chromosome representation is introduced. The work in [2] proposes two vehicle routing problem (VRP) variants for drone delivery. The first one minimizes the total operating cost subject to a delivery time limit, and the second one optimizes delivery time subject to a budget constraint. The costs include drone fleet cost and energy cost. Instead of dealing directly with the original form of the power function, which is nonlinear, the authors use a linear approximation function to calculate the power consumption which varies linearly with payload and battery weight. To save cost, each drone can perform multiple trips and visit multiple customers per trip. They use a simulated annealing (SA) heuristic to solve the models. The authors of [70] study a drone delivery problem with time windows and a trip duration limit. They minimize three different objectives: travel distance, the number of drones used, and the number of batteries required. When imposing the linear energy constraints, the battery capacity is reserved at 20% to be a buffer for unusual conditions.

Some works study the impacts of drone delivery on costs and carbon dioxide (CO_2) emissions.

The authors of [71] analyze the feasibility of using drones for package delivery in terms of energy requirement and economics. They approximate power consumption as a linear function of payload and velocity. The work in [72] assesses the potential of drones in reducing CO_2 emissions generated by the electricity supply chain and provides a comparison of this system with delivery using diesel vehicles and electric trucks/tricycles. The authors also consider the emissions from the vehicle production and disposal phases. The study in [73] uses the same battery reservation policy as in [70] when studying the energy use and environmental impacts of drones for last-mile delivery in comparison with medium-duty trucks. The power function in [73] for hovering takes a similar form as that in [2].

There are also studies focusing on drone energy models, where the drones' flying status is considered. The authors of [74] derive a theoretical model to calculate the multirotor drone's power consumption. They identify the model's parameters by performing field tests. In their experiments, they consider different drone statuses in a flight path: ascend/descend, hover, and straight line fight. The study in [75] compares the energy demands of drone-based and ground-based (diesel trucks and electric trucks) parcel delivery services. Factors like drone weight, speed, head wind speed, and other drone parameters are taken into account for energy calculation. The authors of [76] review energy consumption models for drone delivery. They identify key factors that affect drone energy consumption and discuss similarities and differences among various models. For cruising flight, drone power consumption can be modeled as a convex function of a drone's total weight [73–75], while for hovering it is proportional to the weight to the power 1.5 [2].

Truck-drone Problems. The truck-drone tandem system is the most intensively studied area in drone delivery problems. Most papers in this area assume that during each trip a drone can visit only one customer. The reference [30] considers two types of truck-drone delivery problems. The first is the flying sidekick traveling salesman problem (FSTSP), where one truck carries one drone to deliver parcels to a set of customers. As the driver performs deliveries, the UAV is launched from the truck, delivering a parcel for an individual customer, then the truck and the drone rendezvous at a new customer location. The second problem in [30] is the parallel drone scheduling traveling salesman problem (PDSTSP), where multiple drones make single-stop delivery trips from the depot while a single truck serves other customers without carrying any drone. The objective of both problems is to minimize the time required to service all customers and return to the depot. Simple heuristics are used to solve both problems. The work in [77] uses a SA heuristic to solve the FSTSP. The authors of [23] use a route first-cluster second heuristic to solve a variant of the FSTSP where the truck can wait at the start node for the drone to return. The authors of [78] and [79] use a dynamic programming (DP) approach and a branch-and-bound (B&B) algorithm

for the same variant, respectively. The reference [80] extends the FSTSP by allowing the launch and rendezvous operations to be performed not only at a node, but also along a route arc. A greedy randomized adaptive search procedure is developed for the problem. The authors of [81] extend the FSTSP by considering energy consumption and no-fly zones. They use the power consumption linear approximation from [2] and propose an evolutionary-based heuristic solution algorithm that integrates constructive and search heuristics. The authors of [82] use a truck-drone tandem system to minimize latency in a customer-oriented distribution system. They compare the benefits of using drones for a single trip versus multiple trips. The work in [83] extends the PDSTSP by assuming that drones can perform two types of tasks: drop-off and pickup. A constraint programming method is applied. The reference [84] studies a same-day delivery problem with trucks and drones, where customer orders come dynamically during a shift. The authors present a Markov decision model and an approximate DP algorithm to solve the problem.

Some studies consider multiple trucks where each is equipped with one or multiple drones. The authors of [85] and [86] consider a fleet of homogeneous trucks with multiple drones per truck. Their objective is to minimize the maximum duration of the routes, and they focus on the worst-case analysis. The study in [87] extends the problem by considering time window constraints. The work in [88] allows docking hubs where trucks can drop off, and drones can pick up, parcels for delivery maintain backup drones. The authors present an arc-based model and develop a branch-and-price (B&P) algorithm. The authors of [89] study the multiple FSTSP with variable drone speeds. They assume that drone power consumption is a function of speed and payload, which affects flight endurance and range.

Sometimes the truck is only used for carrying drones and packages without making any deliveries itself [90,91]. The authors of [92] use continuous approximation techniques to derive the improvement of service quality (i.e., the completion time of all deliveries) by using a truck-drone system. Unlike other studies, they do not restrict the drone launch/retrieval locations to be customer sites. The study in [93] also uses a continuous approximation approach to derive general insights from the aspect of cost.

Multi-trip Vehicle Routing Problems. The MTVRP extends the classical VRP by allowing each truck to perform multiple trips. The reference [94] is the first to study this problem. The author develops a modification of the saving algorithm and uses a bin packing heuristic to assign routes to vehicles. The authors of [95] develop a B&P algorithm for the MTVRP with time windows (MTVRPTW). Their numerical tests focus on the type 2 instance sets in [96]. The reference [97] proposes a network flow model based on generated trips for the same problem. The work in [98] develops two set-partitioning-like formulations for

the MTVRP. The authors of [99] develop an exact two-phase algorithm for the MTVRPTW with a trip duration limit. In the first phase, they enumerate all feasible trips; in the second phase, they use a B&P algorithm to select the best set of schedules. For the same problem, the authors of [100] and [101] develop an adaptive large neighborhood search and a route pool-based metaheuristic, respectively. In [102], the authors develop two set covering formulations for the MTVRPTW without the trip duration constraint and uses B&P algorithms. The authors compare the two models on instances with the first 25 customers of Solomon's "C2", "R2", and "RC2" instances.

In the review paper [103], the authors suggest that there are four ways to formulate the MTVRP. The first one is the 4-index formulation, which uses both the vehicle index and the trip index. Specifically, a binary variable x_{ij}^{vr} is defined to denote whether trip r of vehicle v travels through arc (i, j). The second and the third ones are the 3-index formulations with either a trip index, or with a vehicle index, respectively. That is, a variable x_{ij}^{r} (x_{ij}^{v}) is used to denote whether trip r (vehicle v) travels through arc (i, j). And the last one is the 2-index formulation using a variable x_{ij}, i.e., neither a vehicle nor a trip index is used. For the 3-index formulation with a trip index, since the number of trips performed by each vehicle is unknown, one has to set a sufficiently large cardinality for the trip set, resulting in a weak model with a large number of variables. Or, we can impose an upper bound on the maximal number of trips each vehicle can perform. For the 3-index formulation with a vehicle index, symmetries resulting from identical vehicles are introduced to the model, which make the formulation weak. The authors of [103] indicate that the only compact formulation for the MTVRP is proposed by [104], where a 2-index formulation is applied. The work in [105] also uses a 2-index formulation for a multi-trip cumulative capacitated VRP, where the objective is to minimize the sum of arrival times at required nodes.

2.2.2 Drone Delivery With Consideration of Weather Influence

One major aspect which makes drone delivery problems particularly challenging, as opposed to the common vehicle delivery problems, is the influence of weather conditions on drones' operational efficiency. Some recent studies attempted to tackle this important issue by incorporating weather conditions either directly or indirectly into the optimization models. The authors of [106] study a drone routing problem considering the impact of wind on energy consumption. They first assume that each drone travels at a fixed speed during a trip and the wind speed and direction remain unchanged (i.e., given parameters) during the time horizon. They further assume that the total weight of a drone is constant during a trip. As all of the influencing factors are given parameters, energy consumption on each arc is

a constant number. Thus, energy constraints are ultimately transformed into generalized resource constraints. The authors of [107] design a decision support tool for the multi-trip drone fleet mission planning problem under different weather conditions by decomposing the problem into several subproblems. They first divide the time period into several flying time slots, and the weather condition in each slot is known and analogous. They then determine the clusters of customers for each slot and the routes for each cluster. Finally, they decide on the sequence of routes within each cluster. The work in [108] considers the influence of wind and obstacles on flight time by setting symmetric and asymmetric flight distances.

The authors of [109] study a drone location routing problem under different weather conditions. They state that weather such as snow, fog, and heavy rain would limit drones' operations in some regions. In their study, weather scenarios are known with an occurrence probability, and each scenario indicates whether a specific region can operate drones. They formulate the problem as a two-stage stochastic programming model. The authors of [110] study a stochastic drone facility location problem. They assume that battery energy consumption can be negatively affected under certain weather conditions, which in turn affects drones' flight range. In order to incorporate the risk due to weather conditions, a chance constraint is used to ensure that the probability that a drone's flight range is larger than or equal to the travel distance between a facility and a customer is acceptable. They specifically assume that the flight range follows an exponential distribution, and reformulate the resulting model as a mixed-integer linear programming model. The reference [111] considers a drone scheduling problem with uncertain flight ranges, resulting from the impact of air temperature on battery duration. The authors use a RO method for the problem and compare the performance of three uncertainty sets—polyhedral, box, and ellipsoidal. They use historical and forecast data to estimate the hourly temperature (a span of 12 hours) over an area, then calculate the flight range deviation using a regression function. These deviations are fixed values and considered as different scenarios with weights. Thus, their uncertainty sets are constructed to represent the randomness of the scenario weights.

Our work in Chapter 7 considers the impact of weather conditions on drones' flight times, and it falls within the realm of data-driven RO. Thus, we also briefly review related works on data-driven RO here, where the key step is to construct an uncertainty or ambiguity set from historical data. Details on the construction of these sets can be found in [112–115]. Recently, the authors of [116] propose an event-wise ambiguity set, which is rich enough to capture a wide range of ambiguity sets such as statistic-based and machine-learning-based ones. The works in [117] and [118] are novel applications of the framework proposed in [116]. Specifically, the work in [117] addresses a single-period vehicle allocation problem with uncertain demand, which is related to weather conditions (rainy or not rainy). The authors

use a multivariate regression tree to construct the ambiguity set. The reference [118] studies a two-period, multi-item joint pricing and production problem, where the K-means clustering algorithm is utilized to partition the demand residuals and then a cluster-wise ambiguity set is constructed.

Based on the aforementioned studies, we get the following observations:

1. Only a few studies explicitly consider drone energy consumption, among which many use an approximation that is linear in the payload. However, the solutions generated by the linear approximation method may fail to detect infeasible routes due to excess energy consumption.

2. To the best of our knowledge, no benchmark instance is available for algorithm evaluation, and no efficient exact algorithm is developed for the drone routing problem (DRP). These gaps put a limitation on algorithm development for the DRP.

3. The impact of weather on drone delivery is either ignored completely or considered in a simplified way. Specifically, future weather conditions (or flight range/time) are assumed to be deterministic, or follow a known distribution, or belong to some scenarios with known probabilities at the moment of making decisions. Only reference [111] uses a distribution-free method; however, the authors consider limited deviation scenarios and need to specify the budget or radius of the uncertainty sets. Therefore, more research is needed to solve the drone delivery problem under weather uncertainty.

CHAPTER 3 ROBUST FACILITY LOCATION UNDER DISRUPTIONS

This chapter is based on the following article.

- Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2018. Robust Facility Location Under Disruptions. GERAD Technical Report, G-2018-91, 28 pages. Major revision at *INFORMS Journal on Optimization*.

3.1 Introduction

In this chapter, we develop two-stage RO models for the reliable uncapacitated/capacitated fixed-charge location problem (UFLP/CFLP). In the first stage we make location decisions, and in the second stage we make recourse decisions (i.e., we reassign customers to surviving facilities). The goal is to guarantee the system's performance under disruptive scenarios. The contributions of our work are as follows:

1. We compare the numerical efficiency of exhaustive scenario search to the usual MILP reformulation when solving the NP-hard adversarial problem that arises in a step of the C&CG algorithm.

2. We validate the numerical efficiency and quality of solutions obtained when employing affine decision rules on this class of problems.

3. To illustrate the use of a bi-objective approach to better trade-off between the reliability cost and the nominal cost, we impose an upper bound on the nominal cost when robustifying the system. The results demonstrate that the bound constraints can further reduce the conservativeness of the robust solutions and serve as a decision support tool indicating the trade-off between reliability and nominal cost.

Based on the literature review in Chapter 2, we observe that most studies of the facility location problem under disruptions assume that the probability distribution of disruptions is known perfectly a priori. Only the authors of [42] and [43] use distribution-free approaches. We emphasize that our work differs from their studies in the following aspects. The authors of [42] study the reliable PMP and explore the modeling capability of the two-stage RO method by taking into account demand variations. Specifically, they introduce a parameter ϑ (not a random variable) to denote demand change and study its effect by setting it to a negative value, 0, and a positive value in numerical tests. Note that they assume the facility set J

and the customer set I be the same, which makes it possible to incorporate ϑ into the model, because there should be a link between disruptions and demand variations. In their setting, since $J = I$, disruptions may occur at a customer site (which is also a facility site), resulting in demand variations. They evaluate their C&CG algorithm by a comparison with the BD method. However, we study the reliable FLP and propose two solution methods. First, we use a LP-based enumeration method for the subproblem in order to evaluate the worst-case recourse scenario in the C&CG algorithm. This approach does not require to set big-M values, and it also provides information for other potential worst-case scenarios, which can be used to speed up the algorithm. Second, we introduce an approximation scheme based on the affine policy for large instances, and we provide conditions under which this scheme produces optimal solutions. We further introduce an enhancement to the robust formulations, which can effectively reduce the conservativeness of solutions. We emphasize that our modeling scheme is also able to incorporate demand variations for situations with $J = I$. The work in [43] considers the UFLP with correlated disruptions, which are characterized by a joint distribution. Therefore, DRO instead of two-stage RO framework is used. The authors first exploit the structural property of the DRO model (supermodularity) and then reformulate it as a stochastic program, where standard solution methods such as BD can be used. Their numerical tests focus on quantifying the benefits of considering disruptions that are correlated rather than independent.

The rest of this chapter is organized as follows. Section 3.2 presents the deterministic and robust models for the FLP, and Section 3.3 describes the solution methods. Section 3.4 discusses the numerical results, and Section 3.5 provides concluding remarks.

3.2 Mathematical Models

In this section, we introduce the notation and present the robust models for the UFLP and CFLP.

3.2.1 Notation

We consider a two-echelon supply chain system, where I and J are the sets of customer nodes and facility sites, respectively. The parameter f_j is the fixed cost of locating a facility at candidate site $j \in J$, and C_j is the capacity of a facility at candidate site $j \in J$ if we build a facility there. The parameter h_i is the demand at customer $i \in I$, and d_{ij} is the distance from demand node $i \in I$ to candidate location $j \in J$. For customer $i \in I$, the unit penalty cost associated with unmet demand is p_i. We use $y_j = 1$ to indicate that a facility is located

at site $j \in J$, and $y_j = 0$ otherwise. The variable x_{ij} is the fraction of demand from node $i \in I$ that is satisfied by candidate facility $j \in J$, and u_i is the fraction of unsatisfied demand at site $i \in I$.

3.2.2 Robust UFLP

We first give the deterministic UFLP and then present the corresponding robust model. The deterministic UFLP can be formulated as follows:

$$\min_{\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} x_{ij} + \sum_{i\in I} p_i h_i u_i, \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{j\in J} x_{ij} + u_i \geq 1 \qquad \forall i \in I, \tag{3.1b}$$

$$x_{ij} \leq y_j \qquad \forall i \in I, j \in J, \tag{3.1c}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{3.1d}$$

$$x_{ij} \geq 0 \qquad \forall i \in I, j \in J, \tag{3.1e}$$

$$u_i \geq 0 \qquad \forall i \in I. \tag{3.1f}$$

The objective (3.1a) minimizes the total cost, which includes the fixed facility location cost, the demand-weighted transportation cost, and the penalty cost of unmet demand. Constraints (3.1b) indicate that the sum of the satisfied and unsatisfied demand must be greater than or equal to the customer's total demand. These constraints are inequalities rather than equalities because this is a minimization problem and equality always holds at the optimum. The inequality constraints make it easier to reformulate the model based on the affine policy (see Section 3.3.2). Constraints (3.1c) ensure that demand nodes are assigned to opened facilities. Constraints (3.1d)–(3.1f) impose the integrality and non-negativity constraints. Note that, in contrast to the model presented in [34], we incorporate the cost of unmet demand in the objective. This allows us to find a trade-off between reassigning demand or leaving it unmet when considering the robust reformulations and the CFLP variant. For the UFLP where the facilities are uncapacitated, if the demand cannot be left unmet, we can set a sufficiently large value for $p_i, \forall i \in I$ such that each customer's demand is fully satisfied and it is not optimal to pay the penalty cost.

We now introduce the two-stage adjustable robust counterpart (ARC) model for the UFLP under disruptions. The disruption risk is characterized by an uncertainty set ensuring that

at most k facilities will fail simultaneously in a disruptive scenario [42]:

$$\mathbb{Z}(k) = \left\{ \mathbf{z} \in \{0,1\}^{|J|} : \sum_{j \in J} z_j \leq k \right\}, \tag{3.2}$$

where $z_j = 1$ if facility j is disrupted, and $z_j = 0$ otherwise. In Section 3.3.2, we prove that $\mathbb{Z}(k)$ can be replaced by its convex hull, which facilitates us to develop a heuristic method for the robust models.

The ARC model for the reliable UFLP is formulated as follows:

$$\min_{\mathbf{y},\mathbf{x}(\cdot),\mathbf{u}(\cdot)} \sup_{\mathbf{z} \in \mathbb{Z}(k)} \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij}(\mathbf{z}) + \sum_{i \in I} p_i h_i u_i(\mathbf{z}), \tag{3.3a}$$

$$\text{s.t.} \quad \sum_{j \in J} x_{ij}(\mathbf{z}) + u_i(\mathbf{z}) \geq 1 \qquad \forall \mathbf{z} \in \mathbb{Z}(k), i \in I, \tag{3.3b}$$

$$x_{ij}(\mathbf{z}) \leq y_j(1 - z_j) \qquad \forall \mathbf{z} \in \mathbb{Z}(k), i \in I, j \in J, \tag{3.3c}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{3.3d}$$

$$x_{ij}(\mathbf{z}) \geq 0 \qquad \forall \mathbf{z} \in \mathbb{Z}(k), i \in I, j \in J, \tag{3.3e}$$

$$u_i(\mathbf{z}) \geq 0 \qquad \forall \mathbf{z} \in \mathbb{Z}(k), i \in I. \tag{3.3f}$$

The objective function (3.3a) minimizes the worst-case cost. We use $\mathbf{x}(\cdot)$ and $\mathbf{u}(\cdot)$ to indicate that the allocation decisions are implemented once $\mathbf{z}$ is known, while $\mathbf{y}$ is the decision that must be made before any realization of $\mathbf{z}$. Constraints (3.3c) ensure that the demand nodes are assigned to opened and surviving facilities in a disruptive scenario.

We note that our ARC model is also able to incorporate disruption-caused demand deviations presented in [42]. Specifically, we first assume that $I = J$ and then change the objective function (3.3a) to

$$\min_{\mathbf{y},\mathbf{x}(\cdot),\mathbf{u}(\cdot)} \sup_{\mathbf{z} \in \mathbb{Z}(k)} \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} (1 - \vartheta z_i) h_i d_{ij} x_{ij}(\mathbf{z}) + \sum_{i \in I} (1 - \vartheta z_i) p_i h_i u_i(\mathbf{z}). \tag{3.4}$$

That being said, we will still focus on the models with objective (3.3a), due to the fact that ϑ is treated as a parameter instead of a random variable. Moreover, there are many applications, where candidate facility sites and customers are not exactly the same. For example, in global supply chains, upstream factories and downstream customers are normally located in different areas.

3.2.3 Robust CFLP

The formulation for the deterministic CFLP is as follows:

$$\min_{\mathbf{y},\mathbf{x},\mathbf{u}} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} x_{ij} + \sum_{i\in I} p_i h_i u_i, \tag{3.5a}$$

$$\text{s.t.} \quad \text{(3.1b)–(3.1f) and} \tag{3.5b}$$

$$\sum_{i\in I} h_i x_{ij} \le C_j y_j \qquad \forall j \in J. \tag{3.5c}$$

The objective functions of the deterministic UFLP and CFLP are the same. Constraints (3.5c) ensure that once a facility is opened, its capacity is respected. They also ensure that customers are allocated to opened facilities, so constraints (3.1c) become redundant. However, we retain them because they can strengthen the linear programming relaxation [34].

Similarly, we get the robust CFLP from (3.3a)–(3.3f) with the constraints

$$\sum_{i\in I} h_i x_{ij}(\mathbf{z}) \le C_j y_j (1 - z_j) \qquad \forall \mathbf{z} \in \mathbb{Z}(k), j \in J. \tag{3.6}$$

In this robust CFLP, when facility j is disrupted, its capacity becomes 0 and its service capability is totally lost. We refer to this as *complete disruption*. Our framework is also able to incorporate *partial disruption*, where a damaged facility can still satisfy part of the demand [64,119]. The reliable CFLP with partial disruption can be modeled with (3.3a)–(3.3b), (3.3d)–(3.3f), and the constraints

$$\sum_{i\in I} h_i x_{ij}(\mathbf{z}) \le C_j y_j (1 - \omega_j z_j) \qquad \forall \mathbf{z} \in \mathbb{Z}(k), j \in J, \tag{3.7}$$

where parameter ω_j $(0 < \omega_j \le 1)$ is the proportion of lost capacity at location j when a disruption occurs. We give the model here to demonstrate the strong modeling capability of the two-stage RO approach, but in the following sections we focus on complete disruption, i.e., $\omega_j = 1, \forall j \in J$. Because our solution method still applies with minor modifications.

3.2.4 Properties of the Robust Formulations

In this section, we present two properties of the robust formulations, based on which we develop the solution method.

Lemma 3.2.1 *Given the facility location* $\hat{\mathbf{y}}$*, the uncertainty set* $\mathbb{Z}(k)$*, and two potential*

worst-case scenarios $\mathbf{z}^1 \in \mathbb{Z}(k)$ *and* $\mathbf{z}^2 \in \mathbb{Z}(k)$ *with respective recourse costs* B_1 *and* B_2*, if the set of functional facilities (i.e., those with* $\hat{y}_j = 1$ *and* $z_j = 0$*) in scenario* $\mathbf{z}^1$ *is a subset of functional facilities in scenario* $\mathbf{z}^2$*, then* $B_1 \geq B_2$*.*

Proof. See Appendix A.

Lemma 3.2.2 *Given the facility location* $\hat{\mathbf{y}}$ *with* $\sum_{j \in J} \hat{y}_j = m$ *and the uncertainty set* $\mathbb{Z}(k)$*, we have that if* $m > k$*, the worst-case disruptions occur at opened facilities, i.e., those with* $\hat{y}_j = 1$*. If* $m \leq k$*, the worst-case disruptions occur at all the opened facilities, and all the demand in the system will be left unsatisfied.*

Proof. This result follows from the proof of Lemma 3.2.1.

Lemma 3.2.2 indicates that when $m > k$, we can enumerate all the potential worst-case scenarios by considering only the set of opened facilities instead of set J. This helps to reduce the number of minimum cost flow problems to be solved in the subproblem of the C&CG framework.

3.3 Solution Methods

In this section, we introduce a new C&CG algorithm and an approximation scheme for the ARC models. For both models, we also implement the duality-based C&CG algorithm [42] and the BD method as benchmarks. We close this section by discussing potential extensions of our modeling and solution scheme to other applications.

3.3.1 Column-and-Constraint-Generation Algorithm

We implement the C&CG algorithm in a master-subproblem framework. At each iteration, in the master problem, we make location decision $\hat{\mathbf{y}}$. In the subproblem, for a given first-stage solution $\hat{\mathbf{y}}$, we identify the worst-case scenario. If the relative gap between the upper and lower bounds satisfies the optimality tolerance, the algorithm terminates; otherwise we create recourse variables and the corresponding constraints for the identified scenario, add them to the master problem, and continue to the next iteration. In this section, we first present the framework of the proposed C&CG algorithm for the two robust models and then introduce an enhancement strategy to improve the computational performance.

C&CG Algorithm for Robust UFLP

We use $\mathbf{x}^l$ and $\mathbf{u}^l$ to represent the allocation variables associated with the lth disruption scenario, and $\mathbf{z}^l$ is the status (disrupted or functional) of the facilities in the lth scenario.

The master problem for the robust UFLP is

$$\phi = \min_{\mathbf{y},s,\{\mathbf{x}^l\}_{l=1}^n,\{\mathbf{u}^l\}_{l=1}^n} s, \tag{3.8a}$$

$$\text{s.t.} \quad s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} x_{ij}^l + \sum_{i\in I} p_i h_i u_i^l \qquad \forall l \in \{1,\dots,n\}, \tag{3.8b}$$

$$\sum_{j\in J} x_{ij}^l + u_i^l \geq 1 \qquad \forall l \in \{1,\dots,n\}, i \in I, \tag{3.8c}$$

$$x_{ij}^l \leq y_j(1 - z_j^l) \qquad \forall l \in \{1,\dots,n\}, i \in I, j \in J, \tag{3.8d}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{3.8e}$$

$$x_{ij}^l \geq 0 \qquad \forall l \in \{1,\dots,n\}, i \in I, j \in J, \tag{3.8f}$$

$$u_i^l \geq 0 \qquad \forall l \in \{1,\dots,n\}, i \in I. \tag{3.8g}$$

We use a LP-based enumeration method derived from Lemma 3.2.2 to solve the subproblem. The details are as follows.

(a) For a given $\hat{\mathbf{y}}$, when $\sum_{j\in J} \hat{y}_i > k$, we enumerate all the potential worst-case scenarios (when $k > 0$, the uncertainty set has multiple extreme points, and each point is potentially the worst-case scenario) and solve a minimum cost flow problem (MCFP) associated with each scenario to identify the actual worst-case scenario. Let $\bar{J}$ be the new facility set in a scenario, which includes only the functional facilities. Then the following MCFP is solved for each scenario:

$$\psi = \min_{\mathbf{x},\mathbf{u}} \sum_{j\in J} f_j \hat{y}_j + \sum_{i\in I}\sum_{j\in \bar{J}} h_i d_{ij} x_{ij} + \sum_{i\in I} p_i h_i u_i, \tag{3.9a}$$

$$\text{s.t.} \quad \sum_{j\in \bar{J}} x_{ij} + u_i \geq 1 \qquad \forall i \in I, \tag{3.9b}$$

$$x_{ij}, u_i \geq 0 \qquad \forall i \in I, j \in \bar{J}. \tag{3.9c}$$

To solve the MCFP, we use the Python NetworkX 2.0 package [120], which applies a primal network simplex algorithm.

(b) If $\sum_{j\in J} \hat{y}_i \leq k$, in the worst-case scenario, all the opened facilities are disrupted and all the customer demand is left unsatisfied.

C&CG Algorithm for Robust CFLP

The master problem for the robust CFLP is defined by (3.8a)–(3.8g) and the constraints

$$\sum_{i \in I} h_i x_{ij}^l \leq C_j y_j (1 - z_j^l) \qquad \forall l \in \{1, \ldots, n\}, j \in J. \tag{3.10}$$

Similarly to the subproblem of the robust UFLP, when $\sum_{j \in J} \hat{y}_j > k$, we solve an MCFP for each potential worst-case scenario in the subproblem of the robust CFLP, which is defined by (3.9a)–(3.9c) and the constraints

$$\sum_{i \in I} h_i x_{ij}^l \leq C_j \qquad \forall j \in \bar{J}. \tag{3.11}$$

If $\sum_{j \in J} \hat{y}_j \leq k$, in the worst-case scenario, all the opened facilities are disrupted and all the demand is left unsatisfied.

For comparison purposes, we give the duality-based C&CG algorithm and the BD method in Appendix A and Appendix A for the robust UFLP and CFLP respectively. The subproblems of both algorithms are obtained by applying duality theory. We note that our LP-based enumeration method has two advantages over solving the dualized subproblem. First, we do not need to set big-M values for the constraints, which helps to avoid numerical issues that can arise with large parameter values. Second, we have cost information for all the potential disruption scenarios, not just the actual worst-case scenario. Based this advantage, we next introduce an enhancement to improve the convergence of the C&CG algorithm.

Multiple Scenario Generation. At each iteration we add multiple scenarios instead of just one. Any replicated scenarios are eliminated before we solve the master problem. In Section 3.4.2, we test four ways of adding the scenarios. The authors of [42] add two scenarios at each iteration. Specifically, after obtaining the worst-case scenario by solving the dualized subproblem, they create another disruption scenario by changing the disrupted facility with the least demand to make it non-disrupted and changing the non-disrupted facility with the greatest demand to make it disrupted. Note that the method for generating the second scenario in [42] applies only to the case where $I = J$. However, our method can be used in any situation. In addition, since the LP-based enumeration method evaluates all the potential scenarios, no extra computational effort is needed to produce and evaluate an alternative scenario in this framework.

Algorithm 1 describes the proposed C&CG algorithm.

Algorithm 1: C&CG algorithm with LP-based enumeration method for subproblem

Step 1: Solve the deterministic model to find its optimal value c_0^* and optimal solution $\mathbf{y}_0^*$. Let $LB = -\infty, UB = \infty, n = 0$. Set the initial solution to $\mathbf{y}_0^*$.

Step 2: Solve the subproblem with respect to $\mathbf{y}_0^*$ and obtain the cost information of all the potential worst-case scenarios. Let $\hat{\psi}$ be the worst-case cost. Update $UB = \min\{UB, \hat{\psi}\}$. Set $n = n + 1$. Create recourse variables and the corresponding constraints associated with the selected scenarios; add them to the master problem.

Step 3: Iterate until the algorithm terminates:

Step 3.1. Solve the master problem to obtain $\hat{\mathbf{y}}$ and $\hat{\phi}$. Update $LB = \hat{\phi}$.

Step 3.2. Solve the subproblem to obtain the cost information of all the potential worst-case scenarios and $\hat{\psi}$. Update $UB = \min\{UB, \hat{\psi}\}$. Set $n = n + 1$.

Step 3.3. if $(UB - LB)/UB \leq \epsilon$: an ϵ–optimal solution is found and the algorithm terminates;

else: create recourse variables and constraints and add them to the master problem; go to *Step 3.1.*

3.3.2 Robust Reformulations with Affine Policy

Another common technique for adjustable RO models is the affine policy, also known as the linear decision rule (LDR), which restricts the adjustable variables to be an affine function of the uncertain parameters [121,122]. This restriction often leads to tractable robust models for realistically sized problems. The LDR is commonly used as a heuristic method and to provide computational insights for exact algorithms. Before introducing the LDR for our ARC models, we first present Lemma 3.3.1, which is a sufficient condition for the adoption of LDR.

Lemma 3.3.1 *For the robust UFLP and CFLP, when the uncertainty budget k is integer, the uncertainty set $\mathbb{Z}(k)$ has an integrality property, that is, it can be replaced with its convex hull*

$$\mathbb{Z}'(k) = \left\{ \mathbf{z} \in \mathbb{R}^{|J|} : 0 \leq \mathbf{z} \leq 1, \sum_{j \in J} z_j \leq k \right\}.$$

Proof. See Appendix A.

The proof process of Lemma 3.3.1 also indicates that when $\sum_{j \in J} \hat{y}_j \geq k$, the worst-case scenario always occurs at the extreme points. And if $\sum_{j \in J} \hat{y}_j < k$, all the opened facilities would be disrupted as indicated in Lemma 3.2.2. Note that our proposed C&CG algorithm does not take advantage from the integrality property here even though it does exist for both the UFLP and the CFLP. Thus, the proposed C&CG framework can generally be applied to the problems where such property does not hold (e.g., see Section 3.3.3).

Affine Policy for Robust UFLP

The integrality property of the uncertainty set makes it possible to reformulate the models based on the LDR. We set $x_{ij}(\cdot)$ to $x_{ij} = \mathbf{W}_{ij}^T\mathbf{z} + w_{ij}$ and $u_i(\cdot)$ to $u_i = \mathbf{A}_i^T\mathbf{z} + a_i$, where $\mathbf{W}_{ij} \in \mathbb{R}^{|J|}, w_{ij} \in \mathbb{R}, \mathbf{A}_i \in \mathbb{R}^{|J|}$, and $a_i \in \mathbb{R}$. Thus, the affinely ARC (AARC) model for the robust UFLP is

$$
\begin{aligned}
&\min_{\mathbf{y},\mathbf{W},\mathbf{w},\mathbf{A},\mathbf{a}} \sup_{\mathbf{z}\in\mathbb{Z}'(k)} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij}(\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) + \sum_{i\in I} p_i h_i(\mathbf{A}_i^T\mathbf{z} + a_i),\\
&\text{s.t.} \quad \sum_{j\in J}(\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) + (\mathbf{A}_i^T\mathbf{z} + a_i) \geq 1 \qquad \forall \mathbf{z} \in \mathbb{Z}'(k), i \in I,\\
&\qquad \mathbf{W}_{ij}^T\mathbf{z} + w_{ij} \leq y_j(1 - z_j) \qquad \forall \mathbf{z} \in \mathbb{Z}'(k), i \in I, j \in J,\\
&\qquad y_j \in \{0,1\} \qquad \forall j \in J,\\
&\qquad \mathbf{W}_{ij}^T\mathbf{z} + w_{ij} \geq 0 \qquad \forall \mathbf{z} \in \mathbb{Z}'(k), i \in I, j \in J,\\
&\qquad \mathbf{A}_i^T\mathbf{z} + a_i \geq 0 \qquad \forall \mathbf{z} \in \mathbb{Z}'(k), i \in I.
\end{aligned}
\tag{3.12}
$$

To solve the AARC model: (1) We can reformulate the model by first writing it in an epigraph form and then applying duality to the robust constraints [122], which produces a MILP model. We then feed the MILP directly to an optimization solver. (2) We can develop a BD method for an equivalent reformulation of the model, following the idea in [18]. We have implemented both methods, and our tests show that for our problem it is more efficient to solve the MILP directly. The reformulation is given in Appendix A.

Affine Policy for Robust CFLP

The AARC model for the robust CFLP is defined by (3.12) and the constraints

$$
\sum_{i\in I} h_i(\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) \leq C_j y_j(1 - z_j) \qquad \forall \mathbf{z} \in \mathbb{Z}'(k), j \in J. \tag{3.13}
$$

After applying duality to each robust constraint, we get the MILP reformulation, which is given in Appendix A.

According to [123], for linear adjustable RO models with only right-hand-side uncertainty, an LDR is optimal if the uncertainty set is a simplex. Therefore, for our problem, the AARC model gives the optimal solution when $k = 1$. When $2 \leq k < |J|$, it produces an upper bound on the true optimal value of the ARC model. When $k = |J|$, the affine policy with $\mathbf{W}_{ij} = 0$, $w_{ij} = 0$, $\mathbf{A}_i = 0$, and $a_i = 1$ achieves the same worst-case cost as the optimal worst-case cost. In this situation, both the exact algorithm and the approximation scheme

identify solutions with no opened facilities.

3.3.3 Extension of Our Modeling and Solution Schemes for Other Applications

Multiple Uncertainty Sets. The authors of [124] suggest that multiple uncertainty sets can be used to further reduce the conservativeness of robust solutions. To be specific, each uncertainty set with different budget is assigned a weight to characterize decision makers' risk preference, and the overall cost of facility location and the weighted sum of the worst-case cost is minimized. We note that our C&CG algorithm and the LDR can still be used in this context. In the former method, one subproblem is solved using the LP-based enumeration method for each uncertainty set. All the identified worst-case scenarios, recourse variables, and corresponding constraints are added to the master problem in each iteration. In the latter method, the constraints containing variables $\mathbf{z}$ in model (3.12) will be required to hold for all the uncertainty sets. Duality theory can still be used to derive the reformulation.

Integer Recourse. We would like to emphasize that our proposed C&CG solution framework allows the flexibility to tackle the problems when the dualization cannot be directly applied. For example, suppose truckload shipping transportation is used to deliver products for a supply chain. Besides facility location and customer allocation decisions, we also need to decide the number of visit on each arc. Let v_{ij} be the number of visit to customer i from facility j. d_{ij} is the fixed transportation cost of each visit to customer i from facility j. Q is truck capacity. Take the robust UFLP for example, it is now formulated as

$$\begin{aligned}
&\min_{\mathbf{y},\mathbf{x}(\cdot),\mathbf{u}(\cdot),\mathbf{v}(\cdot)} \sup_{\mathbf{z}\in\mathbb{Z}(k)} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} d_{ij} v_{ij}(\mathbf{z}) + \sum_{i\in I} p_i h_i u_i(\mathbf{z}),\\
&\text{s.t.}\quad h_i x_{ij}(\mathbf{z}) \le Q v_{ij}(\mathbf{z}) \qquad \forall \mathbf{z}\in\mathbb{Z}(k), i\in I, j\in J,\\
&\qquad v_{ij}(\mathbf{z}) \ge 0,\ \text{integer} \qquad \forall \mathbf{z}\in\mathbb{Z}(k), i\in I, j\in J,\\
&\qquad \text{and}\quad (3.3\text{b})\text{–}(3.3\text{f}).
\end{aligned}$$

For this variant, the recourse variables $v_{ij}, \forall i\in I, j\in J$ take integer values, so the duality-based method cannot be used. However, we can still use the LP-based enumeration method for the subproblem.

3.4 Numerical Results

In this section, we present the instances and explore: (i) The efficiency of the multiple-scenario technique and the performance of the proposed C&CG algorithm, compared to existing exact

algorithms. We also compare the C&CG algorithm with other variants of facility location problems under disruptions (Section 3.4.2). (ii) The impact of the LDR on the computational complexity and solution quality (Section 3.4.3). (iii) The trade-off between the nominal cost and worst-case performance. We enhance our robust formulations with an additional set of constraints to evaluate this trade-off (Section 3.4.4).

3.4.1 Instances

We consider a 49-site data set from reference [34], available at `https://daskin.engin.umich.edu/network-discrete-location/`. It is derived from 1990 census data. The 49 sites include the state capitals of the continental United States plus Washington, D.C. Based on this set, we generate other instances using the first 10, 15, ..., 30 nodes as the candidate facility sites and the first 10, 15, ..., 45 and 49 nodes as the customer sites. There are 35 instances in total. The demand $h_i = \lfloor P_i/10^5 \rfloor$, where P_i is the population at node i. The transportation cost $d_{ij} = \lfloor E_{ij} \times 20 \rfloor$, where E_{ij} is the Euclidean distance between nodes i and j. For simplicity, we use the same unit penalty cost p_i for all the customers, i.e., $p = p_i, \forall i \in I$. To represent systems with different penalty costs, we set two values for p. For each instance, we first calculate the transportation costs $d_{ij}, \forall i \in I, j \in J$ and then rank them in nondecreasing order. The two values for p are the maximal value and the $(\lceil 0.8 \times |I| \times |J| \rceil)th$ value in the order. For convenience, we denote these values p^{max} and $p^{0.8}$. The meaning of $p = p^{max}$ is that after a disruption all the demand must be fully satisfied for the UFLP. For the capacitated models, we let the facility capacity $C_j = \lceil \max\{h_j, r_j\} \rceil$, where r_j is a randomly generated number between $[D/10, 3D/10]$, and D is the total demand of all the customers. We label the instances *Fy-Cx-p*^{d} to indicate that there are y candidate facility sites and x customers, and the unit penalty cost is p^d. The details of the instances and our results are available at: `https://sites.google.com/view/chengchun/instances`.

All the algorithms and models are implemented in Python using Gurobi 7.5.1 as the solver. The computations are executed on a cluster of Intel Xeon X5650 CPUs with 2.67 GHz and 24 GB RAM under Linux 6.3. Each experiment is conducted on a four-core processor of one node. The computational time limit is set to 24 hours, which is in line with other studies considering facility location under uncertainty (48 hours in [18] and 96 hours in [57]). The optimality tolerance ϵ is set to 0.01% for all the exact algorithms unless otherwise specified.

3.4.2 Comparison of Exact Algorithms

In this section, we first evaluate the impact of the multiple-scenario technique and then compare the performance of the exact algorithms for the UFLP and CFLP. In the tables,

Gap is the percentage difference between the best upper and lower bounds; *#Opt* is the number of instances solved to optimality; *#Iter* is the number of iterations; *CPU* is the computing time in seconds to solve the instance. Bold font is used to indicate the best results. Specifically, if an instance can be solved to optimality, the best computing time is in bold; otherwise, the best gap is in bold. If *#Opt* is different for different algorithms, the largest value is in bold.

Performance of Multiple-Scenario Technique

Each time, after solving the subproblem, we consider four options for adding the scenarios, corresponding variables, and constraints: (i) only the worst-case scenario; (ii) both the worst-case scenario and the second-worst scenario; (iii) the worst-case, the second-worst, and the third-worst scenarios; (iv) the worst-case scenario and a randomly chosen scenario. The experiments are performed on instances with $k = 2$ and $k = 3$, and the average results are reported in Table 3.1.

Table 3.1 Performance of multiple-scenario technique ($p = p^{0.8}$)

	Only worst-case				Worst-case + second-worst				Worst-case + second + third-worst				Worst-case + random scenario			
Model	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU
UFLP	2.0	54/70⋆	91.5	27904.6	**1.7**	54/70	66.5	24614.7	2.0	54/70	59.3	23984.4	2.1	54/70	69.2	24924.2
CFLP	1.2	59/70	42.6	17577.0	0.9	**60/70**	26.4	16658.2	**0.8**	**60/70**	20.1	15729.8	**0.8**	**60/70**	26.6	16137.7

⋆ indicates the number of instances (out of 70) that are solved to optimality.

Table 3.1 shows that for the robust UFLP, adding the worst-case and second-worst scenarios gives the best optimality gap. For the robust CFLP, the multiple-scenario technique can solve one more instance to optimality, and the average gap generated by the three implementations of the technique is relatively close. Our tests for the robust PMP also give similar conclusions; therefore, in the following sections, we use the worst and second-worst option to enhance the C&CG algorithm.

Exact Algorithms for Robust UFLP

In the following sections, *C&CG-E* indicates the proposed C&CG algorithm (where the LP-based enumeration method is applied for the subproblem) and *C&CG-D* indicates the C&CG algorithm with the dualized subproblem. Table 3.2 presents the average results, and the detailed results are given in Tables A.1–A.3.

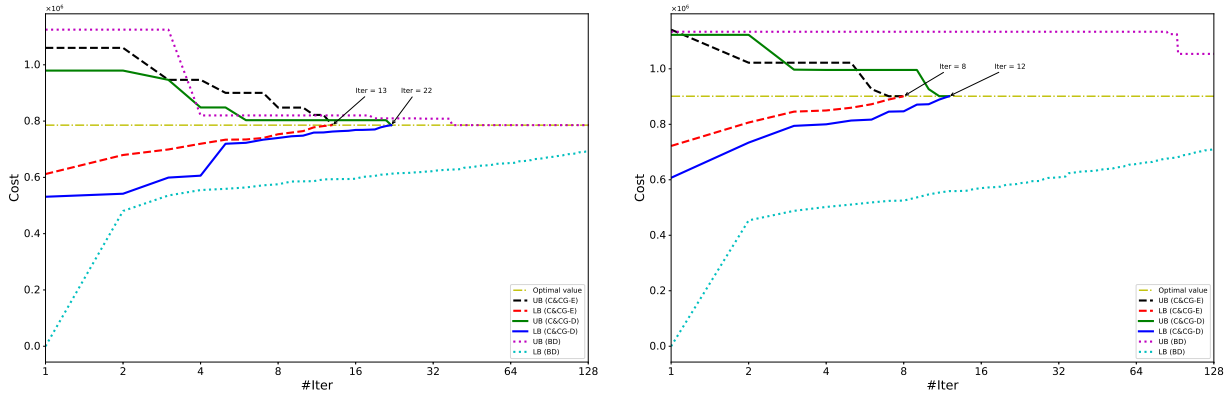
When $k = 2$, both the C&CG algorithms significantly outperform the BD method, solving more instances to optimality in a shorter time. Specifically, both C&CG algorithms can solve

Table 3.2 Average results for the robust UFLP

		C&CG-E				C&CG-D				BD			
k	p	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU
2	$p^{0.8}$	**0.0**	35/35	38.5	**5298.8**	**0.0**	35/35	61.5	9479.7	14.1	18/35	2731.0	49430.9
	p^{max}	**0.0**	35/35	24.8	**1299.9**	**0.0**	35/35	45.7	3815.5	15.9	16/35	3247.7	56230.1
3	$p^{0.8}$	**3.4**	19/35	94.6	43930.6	4.0	19/35	121.3	44616.9	N/A			
	p^{max}	**3.3**	**18/35**	87.3	46206.2	3.9	17/35	120.6	46726.7	N/A			
4	$p^{0.8}$	**9.8**	**14/35**	153.2	57359.4	10.9	13/35	165.0	58191.8	N/A			
	p^{max}	11.0	17/35	122.9	52245.8	**10.4**	17/35	150.0	53551.9	N/A			

N/A: No further experiments are performed.

all the instances to optimality, while the BD method can solve only 18 and 16 instances for $p = p^{0.8}$ and $p = p^{max}$ respectively. Therefore, no experiments are performed for $k = 3$ and $k = 4$ with the BD method. Compared to C&CG-D, the average CPU time of C&CG-E is shorter and there are fewer iterations. Figure 3.1(a) plots the convergence curves of the three algorithms for F10-C49-$p^{0.8}$. It shows that C&CG-E finds the optimal solution after 13 iterations and that C&CG-D takes 22 iterations. However, the optimality gap of BD is significant (around 12%) and it actually requires 364 iterations. When $k = 3$, one more instance can be optimally solved by C&CG-E. Moreover, the average gap is smaller, there are fewer iterations, and the CPU time is shorter. For $k = 4$ and $p = p^{max}$, C&CG-D provides a smaller optimality gap while the CPU time and the number of iterations are greater.



(a) Robust UFLP (b) Robust CFLP

Figure 3.1 Convergence curves after 128 iterations for F10-C49-$p^{0.8}$ with $k = 2$

Table 3.2 also indicates that the value of p has an influence on the computational efficiency. In general, for the UFLP, the instances with $p = p^{0.8}$ are more complex. In addition, from Table A.1, we observe that for some instances, indicated by a black square, even though the values of p are different, the worst-case cost is the same. However, the computational time

varies.

Exact Algorithms for Robust CFLP

We present the summarized results in Table 3.3 and the detailed results in Tables A.4–A.6. Table 3.3 shows that when $k = 2$, all the instances can be solved to optimality by both C&CG algorithms. However, C&CG-E consumes less time on average. Similarly to the results for the robust UFLP, BD takes the most time, and only a small number of the instances can be solved to optimality. Figure 3.1(b) displays the convergence curves of the three algorithms. It shows that for the robust CFLP, C&CG-E has the lowest number of iterations and BD has the highest.

The results for $k = 3$ and $k = 4$ further demonstrate the superiority of C&CG-E, i.e., one more instance can be solved when $k = 3$ and $p = p^{max}$, and the average gap and CPU time are better for both budgets.

Table 3.3 Average results for the robust CFLP

		C&CG-E				C&CG-D				BD			
k	p	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU
2	$p^{0.8}$	**0.0**	35/35	17.4	**3308.4**	**0.0**	35/35	28.7	4060.2	19.5	11/35	3303.6	62286.2
	p^{max}	**0.0**	35/35	15.9	**4653.4**	**0.0**	35/35	27.0	5979.5	31.5	9/35	3455.8	64683.6
3	$p^{0.8}$	**1.7**	25/35	35.5	30008.0	2.2	25/35	57.5	31516.5	N/A			
	p^{max}	**2.4**	**25/35**	31.4	28919.8	4.1	24/35	48.0	29649.4	N/A			
4	$p^{0.8}$	**4.4**	20/35	42.3	41172.4	4.7	20/35	64.9	42041.0	N/A			
	p^{max}	**6.4**	21/35	38.6	39382.0	7.6	21/35	57.6	40966.4	N/A			

Table 3.4 Statistics of average CPU time for master and sub problems

			C&CG-E		C&CG-D		Time reduction factor					C&CG-E		C&CG-D		Time reduction factor	
Model	k	p	Master	Sub	Master	Sub	Master	Sub	Model	k	p	Master	Sub	Master	Sub	Master	Sub
UFLP	2	$p^{0.8}$	5295.7	0.5	9463.1	12.2	1.79	23.76	CFLP	2	$p^{0.8}$	3306.0	1.6	4050.0	9.0	1.23	5.82
		p^{max}	1298.4	0.4	3803.0	10.1	2.93	23.37			p^{max}	4650.7	1.9	5967.4	10.8	1.28	5.67
	3	$p^{0.8}$	43919.4	1.1	44583.3	21.2	1.02	18.55		3	$p^{0.8}$	29998.2	6.7	31490.8	21.6	1.05	3.21
		p^{max}	46193.4	1.8	46689.0	24.6	1.01	13.89			p^{max}	28908.2	8.9	29623.3	22.7	1.02	2.56
	4	$p^{0.8}$	57341.0	1.0	58148.3	24.9	1.01	24.18		4	$p^{0.8}$	41155.5	13.1	42011.5	24.6	1.02	1.88
		p^{max}	52229.5	1.7	53508.4	26.5	1.02	15.80			p^{max}	39355.7	22.7	40930.1	31.7	1.04	1.39

Statistics of CPU Time for Master and Sub Problems

Table 3.4 presents the statistical results of the CPU time spent on the master and sub problems respectively. The *time reduction factor* is calculated as the time spent by the

C&CG-D algorithm divided by the time spent by the C&CG-E algorithm. A time factor equal to 2 indicates that the problem spends only half the time using the C&CG-E compared to using the C&CG-D. Table 3.4 shows that the main time reduction comes from the computing time for the master problem, which indicates that the cuts generated from the second-worst scenarios can improve the convergence of the C&CG algorithm. When $k = 3$ and $k = 4$, the time factor of the master problem is only a bit larger than 1, this is because some instances are not solved to optimality and the time limit is reached.

Table 3.5 Results for uncapacitated p-median problem under disruptions

				Unit penalty cost = 15						Unit penalty cost = p^{max}					
				C&CG-E			C&CG-D in [42]			C&CG-E			C&CG-D in [42]		
$\|J\|$	ϱ	p	k	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU
25	0.2	8	1	0.0	1	**0.3**	0.0	4	2.6	0.0	1	**0.3**	0.0	4	2.0
			2	0.0	5	3.7	0.0	5	**2.8**	0.0	5	7.6	0.0	5	**3.2**
			3	0.0	21	**58.8**	0.0	58	578.2	0.0	35	**322.5**	0.0	70	1364.8
		10	1	0.0	1	**0.2**	0.0	1	0.6	0.0	1	**0.4**	0.0	1	0.6
			2	0.1	3	**2.4**	0.0	5	3.8	0.1	3	**2.9**	0.0	5	3.3
			3	0.0	5	**6.4**	0.0	9	10.4	0.0	7	15.0	0.0	11	**12.1**
	0.4	8	1	0.0	3	**1.0**	0.0	4	2.0	0.0	3	3.7	0.0	4	**2.2**
			2	0.0	8	**7.3**	0.0	13	12.7	0.0	8	**9.7**	0.0	13	15.7
			3	0.0	46	**492.5**	0.0	96	1718.8	**0.0**	64	**4159.7**	6.9	92	7359.1
		10	1	0.0	1	**0.3**	0.0	1	0.5	0.0	1	1.0	0.0	1	**0.6**
			2	0.0	4	**3.3**	0.0	6	3.5	0.0	4	16.4	0.0	6	**4.4**
			3	0.0	13	**27.2**	0.0	22	36.3	0.0	14	**55.6**	0.0	25	59.5
49	0.2	8	1	0.0	1	**1.2**	0.0	3	4.9	0.0	1	**3.8**	0.0	3	9.1
			2	0.0	6	**22.4**	0.0	9	29.0	**0.0**	27	**4604.8**	5.0	44	7429.3
			3	**1.6**	38	7618.4	3.6	61	7426.4	**9.0**	33	7544.5	10.2	43	7661.3
		10	1	0.0	1	**0.9**	0.0	1	1.5	0.0	1	**1.4**	0.0	1	2.2
			2	0.0	6	**36.3**	0.0	11	71.9	0.0	6	**58.9**	0.0	11	110.0
			3	0.0	27	**4280.2**	0.0	40	7241.1	**7.1**	28	7223.8	12.1	52	7272.8
	0.4	8	1	0.0	2	**4.1**	0.0	3	4.8	0.0	2	**5.5**	0.0	3	11.2
			2	0.0	17	**747.0**	0.0	29	1334.7	**8.5**	24	7661.8	17.5	46	7458.9
			3	**11.0**	32	7227.2	15.6	44	7821.3	**21.5**	27	7507.7	23.1	33	7834.5
		10	1	0.0	2	**3.8**	0.0	3	5.0	0.0	2	**4.8**	0.0	3	5.1
			2	0.0	13	**683.0**	0.0	30	1468.7	0.0	13	**788.0**	0.0	30	3069.0
			3	11.7	24	7355.0	**10.4**	39	7512.4	**15.9**	22	8057.3	18.4	44	7753.2
		Average		**1.0**$^{(3)}$	11.7	1190.9	1.2$^{(3)}$	20.7	1470.6	**2.6**$^{(5)}$	13.8	2002.4	3.9$^{(7)}$	22.9	2393.5

$^{(-)}$: indicates the number of instances (out of 24) that are not solved to optimality.

C&CG-E for Other Variants of the Facility Location Problem Under Disruptions

This section applies C&CG-E to the uncapacitated/capacitated PMP (UPMP/CPMP). For both models, the objective function optimizes the weighted sum of the nominal cost and the worst-case cost. The detailed models [42] are given in Appendix A. Since our preliminary experiments as well as those of [42] have shown that C&CG-D performs better than BD, we compare only the performance of C&CG-E and C&CG-D. The parameters, the optimality tolerance ($\epsilon = 0.1\%$), and the time limit (2 hours, and we allow the last iteration before reaching the time limit to terminate) take the same values as in [42]. The results for the two models are given in Tables 3.5 and 3.6, where $J = I$ and ϱ is the weight of the worst-case

Table 3.6 Results for capacitated p-median problem under disruptions

				Unit penalty cost = 15						Unit penalty cost = p^{max}					
				C&CG-E			C&CG-D in [42]			C&CG-E			C&CG-D in [42]		
$\|J\|$	ϱ	p	k	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU
25	0.2	8	1	0.0	2	**1.5**	0.0	3	3.4	0.0	2	3.5	0.0	3	**3.0**
			2	0.0	5	**7.2**	0.0	7	9.4	0.0	10	**51.5**	0.0	20	104.1
			3	0.0	25	**236.5**	0.0	41	354.4	**15.4**	43	7521.6	19.8	72	7296.4
		10	1	0.0	1	**0.4**	0.0	1	0.5	0.0	1	**0.3**	0.0	1	0.5
			2	0.0	2	**2.2**	0.0	3	4.0	0.0	3	5.4	0.0	4	**4.1**
			3	0.0	7	**17.9**	0.0	12	23.2	0.0	13	111.4	0.0	19	**82.9**
	0.4	8	1	0.0	3	**3.2**	0.0	4	4.3	0.0	3	**3.4**	0.0	4	3.9
			2	0.0	8	**19.2**	0.0	15	39.8	0.0	12	**99.0**	0.0	23	168.0
			3	0.0	37	**1051.7**	0.0	74	2451.2	**28.4**	43	7352.8	36.8	63	7272.3
		10	1	0.0	1	0.8	0.0	1	**0.6**	0.0	1	**0.4**	0.0	1	0.6
			2	0.0	4	**5.5**	0.0	10	13.8	0.0	5	**9.8**	0.0	10	17.7
			3	0.0	12	**51.9**	0.0	21	63.7	0.0	21	**398.2**	0.0	35	460.6
49	0.2	8	1	0.0	1	**1.5**	0.0	3	7.5	0.0	2	**6.2**	0.0	3	9.3
			2	0.0	10	**310.5**	0.0	15	414.4	**5.8**	22	7819.2	12.4	37	7370.9
			3	**9.0**	25	7884.2	9.8	41	7676.8	24.4	16	8390.7	**24.3**	25	7396.5
		10	1	0.0	1	**1.9**	0.0	2	5.1	0.0	1	**2.1**	0.0	2	4.9
			2	0.0	8	**213.0**	0.0	21	840.2	0.0	11	**1121.3**	0.0	21	1307.6
			3	**2.2**	23	7258.5	3.0	35	8083.0	14.6	17	7292.3	**14.0**	27	9613.7
	0.4	8	1	0.0	2	8.0	0.0	3	**6.2**	0.0	2	**7.0**	0.0	3	9.6
			2	**0.7**	26	7815.3	9.5	35	7858.8	**18.3**	18	8751.8	27.0	29	7295.0
			3	**20.5**	18	8668.8	26.3	30	7347.6	**44.1**	15	7852.8	46.4	23	7379.5
		10	1	0.0	3	17.2	0.0	4	**11.5**	0.0	3	16.5	0.0	4	**14.1**
			2	**4.6**	21	8289.5	4.7	34	7257.8	10.9	19	7993.0	**6.8**	36	8071.1
			3	16.2	22	8742.2	**14.6**	31	7343.0	**28.7**	17	8477.9	29.9	27	7410.8
		Average		**2.2**$^{(6)}$	11.1	2108.7	2.8$^{(6)}$	18.6	2075.8	**7.9**$^{(9)}$	12.5	3053.7	9.1$^{(9)}$	20.5	2970.7

cost.

Tables 3.5 and 3.6 show that for both the UPMP and CPMP, C&CG-E has better performance, in terms of average gap and number of iterations. In particular, for the UPMP, when unit penalty cost is p^{max}, two more instances can be solved to optimality by C&CG-E with less than 4700 seconds; however, C&CG-D produces solutions with optimality gaps over 5.0% when reaching the time limit. For the UPMP, the average CPU time of C&CG-E is also lower. For the CPMP, the average CPU time of the two algorithms is quite close, while C&CG-E provides better optimality gaps. From both tables, we observe that C&CG-E generally works better for instances with a large budget. Take $|J| = 25, \varrho = 0.4, p = 8, k = 3$ and unit penalty cost = 15 for example, for the UPMP, C&CG-E consumes 492.5 seconds while C&CG-D takes 1718.8 seconds. Similarly, for the CPMP, the computing time is 1051.7 seconds versus 2451.2 seconds.

Conclusions: (i) For both the UFLP and the CFLP, C&CG-E is the most efficient of the three exact algorithms. (ii) For both the UPMP and the CPMP, C&CG-E generates solutions with better optimality gaps. (iii) The computational complexity is influenced by several factors: problem size, budget of uncertainty, and unit penalty cost. The bottleneck of the C&CG algorithm is the resolution of the master problem.

3.4.3 Evaluation of Linear Decision Rule

We evaluate the LDR in terms of the computational efficiency and the optimality gap. Before presenting the results, we give the following definitions.

• **Achieved worst-case cost $f_C^*(y_L^*)$:** The actual worst-case cost achieved by the LDR. For a location decision y_L^* generated by the LDR, we calculate $f_C^*(y_L^*)$ by fixing the location decision and solving the subproblem of the C&CG algorithm.

• **Optimal worst-case cost f_C^*:** The best worst-case cost that can be achieved for an instance, which is obtained by using exact algorithms to solve the ARC models.

• **Relative suboptimality (Opt. gap):** The relative difference between $f_C^*(y_L^*)$ and f_C^*, computed as $(f_C^*(y_L^*) - f_C^*)/f_C^*(y_L^*) \times 100\%$.

We consider all the instances with $k = 1, \ldots, 4$ and $p = p^{0.8}$, p^{max}, which are solved to optimality by C&CG-E. There are 208 instances for the UFLP and 231 instances for the CFLP. The average results are reported in Table 3.7, and the detailed results are given in Tables A.7–A.8.

Table 3.7 Average results of the LDR for the instances solved to optimality by C&CG-E

				CPU time							CPU time		
Model	p	k	#Opt	C&CG-E	LDR	Opt. gap	Model	p	k	#Opt	C&CG-E	LDR	Opt. gap
UFLP	$p^{0.8}$	1	35	**16.2**	41.5	0.00	CFLP	$p^{0.8}$	1	35	**46.9**	127.4	0.00
		2	35	**5298.8**	10831.8	3.69			2	35	**3308.4**	5111.0	4.25
		3	19	7313.0	**269.9**	3.27			3	25	6659.8	**943.4**	4.43
		4	14	12533.1	**110.0**	4.81			4	20	5394.0	**452.7**	3.59
	p^{max}	1	35	**14.5**	41.0	0.00		p^{max}	1	35	**52.6**	174.0	0.00
		2	35	**1299.9**	21361.7	7.10			2	35	**4653.4**	12416.6	8.23
		3	18	6771.5	**295.9**	3.86			3	25	5194.2	**2012.4**	7.16
		4	17	14613.3	**387.8**	4.34			4	21	6645.0	**485.1**	6.52

Table 3.7 shows that for both models, the average time of C&CG-E is shorter for instances with $k = 1$ and $k = 2$. For $k = 2$, the average CPU time of LDR is significantly higher because the MILP model based on the LDR is not solved to optimality within the time limit for some large instances, making the average CPU time longer (note that we considered only the instances solved to optimality by C&CG-E here). The LDR, however, could efficiently solve the instances when $k = 3$ and $k = 4$ and the average computing times are much shorter than those of the C&CG-E. From Tables A.1–A.8, we can see that the budget of uncertainty has a significant influence on the CPU time of C&CG-E, while this is not the case for the LDR model. In terms of relative suboptimality, when $k = 1$, the gaps are 0 since the LDR is optimal for $k = 1$. When k varies from 2 to 4, the average gap varies between 3.27% and 4.81% for $p = p^{0.8}$ and 3.86% and 8.23% for $p = p^{max}$. In general, the LDR generates

solutions with smaller gaps for the robust UFLP and for instances with $p = p^{0.8}$.

3.4.4 Trade-Off between Reliability and Nominal Cost

In this section, we first evaluate the impact of considering disruptions on a system's nominal cost, i.e., the price of robustness. We then introduce an enhancement to the robust formulations that allows the decision-makers to control the trade-off between the reliability and the nominal cost.

Impact of Reliability

We conduct experiments as follows. (i) *Worst-case cost of the deterministic model*: We solve the deterministic model and obtain the location decision. Then we fix the location decision and identify the worst-case cost. (ii) *Nominal cost of the ARC model*: We solve the ARC model and get the location decision. Then we fix the location decision and solve an MCFP to find the system's nominal cost. Table 3.8 presents the results for four randomly selected instances, where the penultimate column is the increase in the nominal cost compared to the result of the deterministic model. The last column is the increase in the worst-case cost compared to the solution of the ARC model.

From Table 3.8, we can make the following two observations: (a) *Sometimes the reliability of a system can be substantially improved with only a slight increase in the nominal cost.* This shows that considering disruptions indeed increases the system's nominal cost. However, this increase is generally less than the increase in the worst-case cost when disruptions are ignored at the design phase but must be handled when they occur. For example, in F20-C49, when $p = p^{0.8}$ and $k = 2$, the nominal cost generated by the ARC model has a 9.1% increase, whereas the worst-case cost produced by the deterministic solution increases by 31.1%. (b) *The improvement over the worst-case cost is even greater for systems with a higher penalty cost.* When $p = p^{max}$, the difference in the worst-case cost is larger than that for $p = p^{0.8}$. This indicates that for systems with a higher penalty cost, where the customer demand must be met to the greatest extent under disruptive scenarios, it is worth considering disruptions at the design stage. This observation can provide guidelines for the location of public facilities, such as fire stations, where recourse operations are related to the safety of life and property.

An Enhancement for Trade-Off between Reliability and Nominal Cost

Sometimes, decision-makers want to both reduce the cost of mitigation under disruption scenarios and control the increase in the nominal cost when robustifying the system. To

Table 3.8 Impact of reliability

				Nominal cost		Worst-case cost		Cost gap (%)	
Model	Instance	p	k	Deterministic	ARC	ARC	Deterministic	Nominal	Worst-case
UFLP	F10-C49	$p^{0.8}$	1	469866	491532	688065	700000	4.4	1.7
			2	469866	602896	785576	1358803	22.1	42.2
			3	469866	701430	880912	1630340	33.0	46.0
			4	469866	691679	953512	1630340	32.1	41.5
		p^{max}	1	469866	491532	689301	724356	4.4	4.8
			2	469866	522163	827587	1718571	10.0	51.8
			3	469866	636200	928582	2756563	26.1	66.3
			4	469866	692915	1018168	2756563	32.2	63.1
	F15-C35	$p^{0.8}$	1	449828	560906	632464	637819	19.8	0.8
			2	449828	573660	711992	1176328	21.6	39.5
			3	449828	658643	792141	1372400	31.7	42.3
			4	449828	697187	863341	1372400	35.5	37.1
		p^{max}	1	449828	469557	657668	691751	4.2	4.9
			2	449828	518018	774758	1646822	13.2	53.0
			3	449828	676307	868055	2626438	33.5	66.9
			4	449828	715603	939255	2626438	37.1	64.2
CFLP	F20-C49	$p^{0.8}$	1	525896	539788	699197	849150	2.6	17.7
			2	525896	578284	824165	1196629	9.1	31.1
			3	525896	793530	935123	1444697	33.7	35.3
			4	525896	1006888	1016404	1639981	47.8	38.0
		p^{max}	1	525896	616558	709395	1085727	14.7	34.7
			2	525896	584478	865055	1775142	10.0	51.3
			3	525896	641227	995549	2312389	18.0	56.9
			4	525896	758230	1094096	2854152	30.6	61.7
	F25-C35	$p^{0.8}$	1	492289	542154	658623	762498	9.2	13.6
			2	492289	616750	772713	1039998	20.2	25.7
			3	492289	653531	858800	1304999	24.7	34.2
			4	492289	680005	680005	1467994	27.6	53.7
		p^{max}	1	492289	643908	667030	986188	23.5	32.4
			2	492289	629601	791860	1606012	21.8	50.7
			3	492289	635840	916247	2150413	22.6	57.4
			4	492289	779098	779098	2645538	36.8	70.6

reflect this, we introduce another group of constraints:

$$
\begin{aligned}
& f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} x_{ij}^0 + \sum_{i\in I} p_i h_i u_i^0 \le (1+q)c_0^*, && \\
& \sum_{j\in J} x_{ij}^0 + u_i^0 \ge 1 && \forall i \in I, \\
& x_{ij}^0 \le y_j && \forall i \in I, j \in J, \\
& \sum_{i\in I} h_i x_{ij}^0 \le C_j y_j && \forall j \in J, \\
& x_{ij}^0 \ge 0 && \forall i \in I, j \in J, \\
& u_i^0 \ge 0 && \forall i \in I,
\end{aligned}
\tag{3.14}
$$

where c_0^* is the optimal objective value of the normal scenario, obtained by solving the deterministic model. Correspondingly, $\mathbf{x}^0$ and $\mathbf{u}^0$ are the allocation decisions. The parameter $q \geq 0$ indicates the decision-makers' tolerance of increased nominal cost when robustifying the system.

We study the impact of constraints (3.14) by varying the value of q. The experiments are conducted on two randomly selected instances, and the results are presented in Table 3.9, where the penultimate column is the increase in the nominal cost compared to that of the deterministic model. The last column is the increase in the worst-case cost compared to the solution of the ARC model without the bound constraints. Note that the value of q does not vary with an equal step length, because we report only the value where the location decision changes. In addition, the first row for each instance corresponds to the result of the deterministic model, and the last row is the result for the ARC model without bound constraints.

Table 3.9 Impact of imposing an upper bound on the nominal cost ($k = 2, p = p^{0.8}$)

			ARC model with constraints (3.14)			Deterministic model	Cost gap (%)	
Model	Instance	q	Location decision	Worst-case cost	Nominal cost	Nominal cost	Nominal	Worst-case
UFLP	F10-C30	0.00	[1, 5, 6]	1208972	435528	435528	0.00	39.99
		0.06	[1, 3, 5, 6]	892768	459163		5.15	18.74
		0.08	[1, 5, 6, 8]	759502	466159		6.57	4.48
		0.26	[1, 5, 6, 7]	743641	475435		8.39	2.44
		0.30	[3, 5, 6, 8]	725463	555996		21.67	0.00
	F10-C49	0.00	[1, 5, 6]	1358803	469866	469866	0.00	42.19
		0.06	[1, 3, 5, 6]	957321	491532		4.41	17.94
		0.08	[1, 5, 6, 8]	828318	500497		6.12	5.16
		0.10	[1, 5, 6, 7]	821814	508753		7.64	4.41
		0.12	[1, 3, 5, 6, 8]	816383	522163		10.02	3.77
		0.28	[1, 3, 5, 6, 7]	811487	530419		11.42	3.19
		0.30	[3, 5, 6, 8]	785576	602896		22.07	0.00
CFLP	F10-C30	0.00	[1, 3, 6, 7, 9]	1043678	548704	548704	0.00	19.97
		0.02	[1, 3, 5, 6, 8, 9]	925575	555901		1.29	9.76
		0.04	[1, 3, 5, 7, 8, 9]	898898	569228		3.61	7.09
		0.06	[1, 3, 6, 7, 8, 9]	869826	580916		5.55	3.98
		0.30	[1, 5, 6, 7, 8, 9]	862084	588988		6.84	3.12
		0.32	[3, 4, 5, 6, 7, 8, 9]	835208	718517		23.63	0.00
	F10-C49	0.00	[1, 3, 4, 8, 9]	1141237	571443	571443	0.00	21.03
		0.02	[1, 4, 5, 8, 9]	1052860	581790		1.78	14.40
		0.04	[1, 3, 4, 5, 8, 9]	969084	590197		3.18	7.00
		0.06	[1, 4, 5, 6, 8, 9]	932736	594405		3.86	3.38
		0.30	[1, 3, 4, 5, 6, 8, 9]	901229	612904		6.76	0.00

Table 3.9 shows that imposing an upper bound on the nominal cost impacts the location decision of the ARC models, i.e., different facilities are chosen or different numbers of sites are opened. We also observe that the bound constraints can help the decision-makers to further control the conservativeness of the robust solutions. For the given instances, sometimes the

nominal cost can be significantly decreased with a slight increase in the worst-case cost. For example, for the UFLP with F10-C49, when q changes from 0.30 to 0.08, the increase in the nominal cost drops from 22.07% to 6.12%; however, the worst-case cost increases by only 5.16%. Similarly, for the CFLP with F10-C30, when q changes from 0.32 to 0.30, the increase in the nominal cost drops from 23.63% to 6.84%, and the worst-case cost increases by only 3.12%. Managers can also use the bound constraints as a decision support tool to see the trade-off between reliability and nominal cost, and to decide the extent to which the nominal cost can be controlled when robustifying the system.

3.5 Conclusions

In this chapter, we solve the reliable fixed-charge location problems, where each facility is exposed to disruption risks. We use a budgeted uncertainty set to characterize the disruptions and apply a two-stage RO method to model the problems. To solve the ARC models exactly, we develop a C&CG algorithm where a LP-based enumeration method is used for the subproblem. This approach can tackle cases with integer recourses where the dualization technique cannot be applied. We also use the LDR to approximate the ARC models, in order to solve large instances in a reasonable time. Our numerical experiments show that the proposed C&CG algorithm outperforms the C&CG algorithms in the literature and that the LDR is capable of providing good first-stage solutions in a shorter time. The results also indicate that the robust models are able to improve the system's reliability without significantly increasing the nominal cost, and that imposing an upper bound on the nominal cost can further control the conservativeness of the robust solutions.

CHAPTER 4 ROBUST FACILITY LOCATION UNDER DEMAND UNCERTAINTY AND FACILITY DISRUPTIONS

This chapter is based on the following article.

- Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2019. Robust Facility Location Under Demand Uncertainty and Facility Disruptions. CIRRELT Technical Report, CIRRELT-2019-53, 23 pages. Major revision at *Omega: The International Journal of Management Science*.

4.1 Introduction

From the literature review in Chapter 2, we can see that most works consider one type of uncertainty at a time. Although some studies consider multiple types of uncertainties simultaneously, their modeling schemes may produce overly conservative solutions as all the decisions are made *here-and-now* [55,60], or relatively optimistic solutions because it is impossible to enumerate all the disruption scenarios [59]. Thus, in this chapter we use a two-stage RO method for the CFLP under uncertain demand and facility disruptions, which utilizes revealed uncertainty information to make recourse decisions, to generate less conservative solutions. To solve the adjustable robust models exactly, we develop a C&CG algorithm based on a decomposition scheme. We benchmark this method with the other C&CG algorithm proposed in [1] for the two-stage robust network flow problem, a general problem that often arises from various supply chain applications. We consider this study to make the following contributions:

1. To the best of our knowledge, this work is the first one to study the CFLP with simultaneous provider-side and receiver-side uncertainties in a two-stage RO framework. The corresponding model generalizes the problems with only demand uncertainty and with only facility disruptions.

2. We implement the C&CG method proposed by the authors of [14] and demonstrate empirically that it can solve the adjustable robust models efficiently in a reasonable runtime. Moreover, it shows better performance than the one proposed in [1]. We further extend our modeling and solution schemes to facility fortification problems with uncertainties, where facility fortification decisions are made for already existing supply chain systems to protect facilities from disruptions and against demand uncertainty.

3. We conduct extensive numerical tests to study the differences in solutions produced by the robust models, and the impact of uncertainty on solution configuration and algorithm efficiency. Managerial insights are also drawn from numerical tests.

The rest of this chapter is organized as follows. Section 4.2 presents the deterministic and the robust models. Section 4.3 describes the solution method and extends the proposed modeling and solution schemes to facility fortification problems under uncertainties. Section 4.4 discusses the numerical results. Section 4.5 concludes this chapter.

4.2 Models

Notation. We denote $\mathbb{R}$ as the space of real numbers and $\mathbb{R}_+$ as the space of positive real numbers. $|I|$ is the cardinality of set I. Other notation is the same as those in Section 3.2.1 expect for $x_{ij}, \forall i \in I, j \in J$ and $u_i, \forall i \in I$. Here, variables x_{ij} and u_i represent product *quantity* instead of *fraction* of demand.

4.2.1 The Deterministic Model

The deterministic CFLP can be formulated as

$$\text{CFLP:} \quad \min_{\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} + \sum_{i\in I} p_i u_i \tag{4.1a}$$

$$\text{s.t.} \sum_{j\in J} x_{ij} + u_i \geq h_i \qquad \forall i \in I, \tag{4.1b}$$

$$\sum_{i\in I} x_{ij} \leq C_j y_j \qquad \forall j \in J, \tag{4.1c}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{4.1d}$$

$$x_{ij} \geq 0 \qquad \forall i \in I, j \in J, \tag{4.1e}$$

$$u_i \geq 0 \qquad \forall i \in I. \tag{4.1f}$$

The objective function (4.1a) minimizes the total cost, which includes the facility location cost, transportation cost, and the penalty cost of unsatisfied demand. Constraints (4.1b) denote that the sum of met and unmet demand must be greater than or equal to a customer's demand. Inequalities (4.1c) impose that customers can only be allocated to opened facilities and that a facility's capacity constraint must be respected. Constraints (4.1d)–(4.1f) impose the integrality and non-negativity constraints. Note that model (4.1) is a quantity-based model, i.e., variables $x_{ij}, \forall i \in I, j \in J$ and $u_i, \forall i \in I$ denote product quantity instead of the fraction of customer demand as those in model (3.5).

4.2.2 The Robust Model Under Uncertain Demand and Facility Disruptions

Uncertainty Set. We use a budgeted uncertainty set to characterize uncertain demand [12,14]:

$$\mathcal{U}_h = \left\{ \boldsymbol{h} \in \mathbb{R}_+^{|I|} : h_i = \bar{h}_i + \theta_i h_i^{\Delta}, 0 \le \theta_i \le 1, \sum_{i \in I} \theta_i \le \Gamma_h \right\}, \tag{4.2}$$

where $\bar{h}_i$ is the nominal (or basic) demand at customer i and $h_i^{\Delta} \ge 0$ is the maximal demand deviation. Γ_h is the uncertainty budget which bounds the maximal number of demand parameters by which values are allowed to deviate from their nominal values.

We characterize disruption risks as in [42]

$$\mathcal{Z}_k = \left\{ \boldsymbol{z} \in \{0,1\}^{|J|} : \sum_{j \in J} z_j \le k \right\}, \tag{4.3}$$

where $z_j = 1$ if facility j is disrupted, and $z_j = 0$ otherwise. Equation (4.3) means that at most k facilities are allowed to fail simultaneously in a disruption scenario.

We use the following uncertainty set to represent simultaneous demand uncertainty and facility disruptions

$$\mathcal{W} = \left\{ (\boldsymbol{h}, \boldsymbol{z}) \in \mathbb{R}_+^{|I|} \times \{0,1\}^{|J|} : \boldsymbol{h} \in \mathcal{U}_h, \boldsymbol{z} \in \mathcal{Z}_k \right\}. \tag{4.4}$$

Robust Model. The adjustable robust counterpart model for CFLP is

$$\text{CFLP-DR:} \quad \min_{\boldsymbol{y}} \sum_{j \in J} f_j y_j + \max_{(\boldsymbol{h}, \boldsymbol{z}) \in \mathcal{W}} g(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{z}) \tag{4.5a}$$

$$\text{s.t.} \quad y_j \in \{0,1\} \qquad \forall j \in J, \tag{4.5b}$$

where $g(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{z})$ is the optimal value of the second-stage problem defined as

$$\min_{\boldsymbol{x}, \boldsymbol{u}} \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} + \sum_{i \in I} p_i u_i \tag{4.6a}$$

$$\text{s.t.} \sum_{j \in J} x_{ij} + u_i \ge h_i \qquad \forall i \in I, \tag{4.6b}$$

$$\sum_{i \in I} x_{ij} \le C_j y_j (1 - z_j) \qquad \forall j \in J, \tag{4.6c}$$

$$x_{ij} \ge 0 \qquad \forall i \in I, j \in J, \tag{4.6d}$$

$$u_i \ge 0 \qquad \forall i \in I. \tag{4.6e}$$

The objective function (4.5a) minimizes the sum of the facility location cost and the worst-case allocation cost. The *max* operator identifies the worst-case scenario and the *min* operator in problem (4.6) finds the least costly recourse solution $(\boldsymbol{x}, \boldsymbol{u})$ with a given first-stage location decision and a specific scenario. Constraints (4.6c) mean that customers can only be reassigned to opened and functional facilities (i.e., those with $y = 1$ and $z = 0$). We use CFLP-D and CFLP-R to denote the model with only uncertain demand and with only facility disruptions, respectively. The two models can be obtained directly by setting the parameter $\Gamma_h = 0$ or $k = 0$. From model (4.6), we can observe that uncertainties only affect the right-hand side of constraints and that the model has the property of *fixed recourse* (i.e., the coefficients of the recourse variables are not influenced by uncertainties).

Remark. *Random variables $\boldsymbol{z}$ can be relaxed to continuous variables, i.e., $0 \leq \boldsymbol{z} \leq 1$, to characterize facilities' partial disruptions, where a facility can still provide part of services to customers if it is opened and not completely disrupted. Under partial disruptions, the uncertainty budget k in set (4.3) can take any non-negative real value, which gives more flexibility to control the conservativeness of the robust solutions, compared to the case with k taking an integer value. More precisely, when $\boldsymbol{z}$ are binary variables, k can still be set to a non-negative real value, but the worst-case scenario is always reached when the uncertainty budget is an integer value, i.e., at $\lfloor k \rfloor$.*

4.3 Solution Method

In this section, we first introduce the solution method, and then extend our modeling and solution schemes to facility fortification models.

4.3.1 C&CG Algorithm

In the two-stage framework, x_{ij} and u_i are no longer a single variable but rather a mapping from the space of observations $\mathbb{R}_+^{|I|} \times \{0,1\}^{|J|}$ to $\mathbb{R}_+ \cup \{0\}$. This flexibility comes at the price of significant computational challenges. To solve the robust model, we may consider a vertex enumeration method, which reformulates the original robust model to a single mixed integer linear program (MILP) model by enumerating all the extreme points of the uncertainty set. However, in general the number of extreme vertices of a convex polyhedron is exponential in size with respect to the faces that describe it, which prevents us from using a full enumeration method. Therefore, we resort to the C&CG algorithm, which identifies a subset of the vertices of the uncertainty set and then applies a *reduced* vertex enumeration method. We use the CFLP-DR to describe the algorithm.

Our C&CG algorithm is based on the one developed by the authors of [14], which is implemented in a master-subproblem framework. The master problem (MP) is solved to generate the first-stage solution, and the subproblem is solved to identify the worst-case realization of the uncertain parameters under a given first-stage solution. Each time after a subproblem is solved, we compute the gap between the upper and lower bounds. If the optimality gap is reached, the algorithm terminates; otherwise, we add the identified worst-case scenario and its associated variables and constraints to the MP, and the algorithm iterates.

The master problem is written as

$$\phi = \min_{\boldsymbol{y},s,\{\boldsymbol{x}\}_{l=1}^{n},\{\boldsymbol{u}\}_{l=1}^{n}} s \tag{4.7a}$$

$$\text{s.t. } s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij}^l + \sum_{i\in I} p_i u_i^l \qquad \forall l \in \{1,\ldots,n\}, \tag{4.7b}$$

$$\sum_{j\in J} x_{ij}^l + u_i^l \geq h_i^l \qquad \forall l \in \{1,\ldots,n\}, i \in I, \tag{4.7c}$$

$$\sum_{i\in I} x_{ij}^l \leq C_j y_j (1 - z_j^l) \qquad \forall l \in \{1,\ldots,n\}, j \in J, \tag{4.7d}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{4.7e}$$

$$x_{ij}^l \geq 0 \qquad \forall l \in \{1,\ldots,n\}, i \in I, j \in J, \tag{4.7f}$$

$$u_i^l \geq 0 \qquad \forall l \in \{1,\ldots,n\}, i \in I. \tag{4.7g}$$

The MP seeks to find the best location decision in light of the set of worst-case scenarios identified in the subproblem. The allocation variables, x_{ij}^l and u_i^l, now feature an extra index l, which means that these variables are associated with the lth scenario (added after finishing the lth iteration). Similarly, parameters h_i^l and z_j^l are the worst-case realizations of random variables h_i and z_j identified in the lth iteration via solving the subproblem. Since constraints (4.7b) are based on a subset of the uncertainty set $\mathcal{W}$, model (4.7) naturally provides a valid relaxation (or a lower bound) to the original two-stage RO model. By adding significant scenarios gradually to model (4.7), stronger lower bounds can be expected [14].

To identify the significant scenarios, we solve the $\max_{(\boldsymbol{h},\boldsymbol{z})\in\mathcal{W}} g(\hat{\boldsymbol{y}},\boldsymbol{h},\boldsymbol{z})$ problem after getting a location decision $\hat{\boldsymbol{y}} \in \mathbb{R}^{|J|}$ from the MP. Since unmet demand is associated with a penalty cost, the second-stage problem is always feasible (a propriety termed as *relatively complete recourse*). Meanwhile, its optimal value is finite as the uncertainty set is bounded and nonempty. Thus, strong duality holds and we can use the Karush–Kuhn–Tucker (KKT) condition to derive the SP. Let $\boldsymbol{\alpha} \in \mathbb{R}^{|I|}$ and $\boldsymbol{\beta} \in \mathbb{R}^{|J|}$ be the dual variables associated with

constraints (4.6b) and (4.6c), respectively, then the SP is written as

$$
\begin{aligned}
\psi = \max_{\substack{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}^{\alpha},\\ \boldsymbol{w}^{\beta},\boldsymbol{w}^{x},\boldsymbol{w}^{u},\boldsymbol{z}}} & \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} + \sum_{i\in I} p_i u_i \\
\text{s.t. } & \sum_{j\in J} x_{ij} + u_i \geq \bar{h}_i + \theta_i h_i^{\Delta} && \forall i \in I, \\
& \sum_{i\in I} x_{ij} \leq C_j\hat{y}_j(1-z_j) && \forall j \in J, \\
& \alpha_i - \beta_j \leq d_{ij} && \forall i \in I, j \in J, \\
& \alpha_i \leq p_i && \forall i \in I, \\
& \sum_{j\in J} x_{ij} + u_i \leq \bar{h}_i + \theta_i h_i^{\Delta} + M_i^{\alpha}(1-w_i^{\alpha}) && \forall i \in I, \\
& \alpha_i \leq M_i^{\alpha} w_i^{\alpha} && \forall i \in I, \\
& \sum_{i\in I} x_{ij} \geq C_j\hat{y}_j(1-z_j) + M_j^{\beta}(w_j^{\beta}-1) && \forall j \in J, \\
& \beta_j \leq M_j^{\beta} w_j^{\beta} && \forall j \in J, \\
& \alpha_i - \beta_j \geq d_{ij} + M_{ij}^{x}(w_{ij}^{x}-1) && \forall i \in I, j \in J \\
& x_{ij} \leq M_{ij}^{x} w_{ij}^{x} && \forall i \in I, j \in J \\
& \alpha_i \geq p_i + M_i^{u}(w_i^{u}-1) && \forall i \in I, \\
& u_i \leq M_i^{u} w_i^{u} && \forall i \in I, \\
& \theta_i \leq 1 && \forall i \in I, \\
& \sum_{i\in I} \theta_i \leq \Gamma_h, \\
& \sum_{j\in J} z_j \leq k, \\
& x_{ij}, u_i, \alpha_i, \beta_j, \theta_i \geq 0 && \forall i \in I, j \in J, \\
& w_i^{\alpha}, w_j^{\beta}, w_{ij}^{x}, w_i^{u}, z_j \in \{0,1\} && \forall i \in I, j \in J.
\end{aligned}
$$

We set $M_i^{\alpha} = p_i, M_j^{\beta} = \max\{C_j, \max_i\{d_{ij}(\bar{h}_i + h_i^{\Delta}), p_i(\bar{h}_i + h_i^{\Delta})\}\}, M_{ij}^{x} = \max\{C_j, d_{ij}(\bar{h}_i + h_i^{\Delta}), p_i(\bar{h}_i + h_i^{\Delta})\} + d_{ij}, M_i^{u} = \max\{p_i, \bar{h}_i + h_i^{\Delta}\}$.

The detailed implementation of the C&CG algorithm is given in Algorithm 2. In the initialization step, we solve the deterministic model and get an initial location decision $\hat{\boldsymbol{y}}$. Note that the initial solution can be any feasible solution and not necessarily need to be that of the deterministic model. In the consequent steps, subproblem and MP are alternately solved to close the optimality gap. Specifically, in Step 1, we solve the subproblem with provided $\hat{\boldsymbol{y}}$ to identify the worst-case scenario and update the upper bound. If the termination con-

dition is satisfied (Step 2), the algorithm ends, else the identified worst-case scenario and its associated variables and constraints are added to the MP. In Step 3, the MP is solved and the iteration continues. The C&CG algorithm is guaranteed to converge in a finite number of iterations, which is upper bounded by the number of extreme points of the uncertainty set.

Algorithm 2: C&CG algorithm for the ARC model

Initialization: Let $LB = -\infty$, $UB = \infty, n = 0$. Solve the deterministic model with $h_i = \bar{h}_i$ to get an initial location decision $\hat{\boldsymbol{y}}$. Set LB as the objective value of the deterministic model.

Iterate until the algorithm terminates:

Step 1: Solve the subproblem based on $\hat{\boldsymbol{y}}$ to find the worst-case scenario $(\hat{\boldsymbol{h}}, \hat{\boldsymbol{z}})$. Let $\hat{\psi}$ be the subproblem's optimal value. Set $UB = \min\{UB, \hat{\psi}\}$ and $n = n + 1$.

Step 2: If $(UB - LB)/UB \leq \epsilon$, the algorithm terminates; else, add the identified worst-case scenario and its associated variables and constraints to the MP.

Step 3: Solve the MP to get a location decision $\hat{\boldsymbol{y}}$ and its optimal value $\hat{\phi}$. Set $LB = \hat{\phi}$ and go to Step 1.

The authors of [1] propose a new C&CG algorithm, which formulates the second-stage problem as a minimum cost flow problem. We provide the details of the C&CG algorithm developed in [1] in Appendix B. The main difference between our algorithm and theirs lies in the formulation of the subproblem (the master problems are the same). We use the KKT condition to derive the subproblem, which introduces big-M values to the model; whereas [1] reformulates the subproblem as a network flow problem, which introduces auxiliary binary variables to linearize the nonlinear terms in the objective function. Table 4.1 presents the number of variables and constraints in the subproblems for the CFLP-DR, where P is a constant dependent on the maximum flow cost (i.e., $p^{max} = \max_{i \in I, j \in J}\{d_{ij}, p_i\}$) in the system, in particular, $P = \lceil \log_2(p^{max} + 1) \rceil - 1$. Table 4.1 shows that the size of our subproblem is decided by the number of nodes whereas that in [1] is affected by both the number of nodes and the maximum flow cost.

Table 4.1 Comparison of subproblems for the CFLP-DR

Authors	#Continuous variables	#Integer variables	#Constraints
[1]	$3\|I\| + 2\|J\| + 2\|I\|P$	$2\|I\| + 2\|J\| + 3 + (\|I\| + 1)P$	$13\|I\| + 10\|J\| + 6 + \|I\|\|J\| + 8\|I\|P$
This work	$3\|I\| + \|J\| + \|I\|\|J\|$	$2\|I\| + 2\|J\| + \|I\|\|J\|$	$7\|I\| + 3\|J\| + 3\|I\|\|J\| + 2$

Another common solution method for adjustable RO models is the affine policy, or affine decision rule, which restricts adjustable variables to be affine functions of uncertain parameters [121]. This restriction is expected to produce a possibly conservative, yet tractable, robust

counterpart. We also implemented this method for our problems, but results showed that in most cases, especially for the CFLP-D, the affine policy actually consumed more computational time than the C&CG algorithm. Thus, we choose to omit this method here. We also note that some decomposition methods are available for the robust counterpart obtained from the affine policy, and interested readers can refer to [18] and [125].

4.3.2 Application to Facility Fortification Problems

In the previous sections, we suggest that we could improve the reliability of a system by considering disruption risks at the initial design phase. However, in reality, we may face situations where facilities are already existing and decision-makers are prone to improve the reliability of facilities by investing in protection and security measures, such as purchasing insurances or installing structural reinforcements, instead of constructing a system from scratch [63,119,126,127]. In this section, we discuss the extension of our modeling and solution schemes to facility fortification problems.

We build the model based on the notation used in Section 4.2, with the exception that J now denotes the set of already existing facilities. Correspondingly, f_j is the cost of fortifying facility $j \in J$. And $y_j = 1$ if facility $j \in J$ is chosen to be fortified, $y_j = 0$ otherwise. As in the four mentioned papers, we assume that a fortified facility becomes immune to disruptions. This assumption is widely used in the context of reliable facility location problems, where a reliable (or protected) facility is assumed to be nonfailable [38,126,128–131]. The decision-maker has a total budget B for implementing fortification strategies. We also incorporate demand uncertainty to the fortification model, as at the moment of making facility enhancement decisions, future customer demand is normally not known perfectly.

The two-stage robust facility fortification problem under uncertainties can be formulated as

$$\min_{\boldsymbol{y}} \max_{(\boldsymbol{h},\boldsymbol{z})\in\mathcal{W}} g'(\boldsymbol{y},\boldsymbol{h},\boldsymbol{z}) \tag{4.8a}$$

$$\text{s.t.} \sum_{j\in J} f_j y_j \le B, \tag{4.8b}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{4.8c}$$

where $g'(\boldsymbol{y},\boldsymbol{h},\boldsymbol{z})$ is defined by Equations (4.6a)–(4.6b), (4.6d)–(4.6e), and

$$\sum_{i\in I} x_{ij} \le C_j - C_j(1-y_j)z_j \qquad \forall j \in J. \tag{4.9}$$

Constraints (4.9) mean that a fortified facility (i.e., $y_j = 1$) is always available with capacity

C_j. If facility j is not fortified, it can still supply C_j units of product when $z_j = 0$, as the facility is already existing; however, it cannot serve any customer when $z_j = 1$. Note that besides imposing a budget constraint on the fortification cost, we can also optimize the cost term $\sum_{j\in J} f_j y_j$ as in Section 4.2, i.e., we include it in the objective function and omit the budget constraint.

Correspondingly, we re-write the master problem of the C&CG algorithm as

$$\phi = \min_{\boldsymbol{y},s,\{\boldsymbol{x}\}_{l=1}^{n},\{\boldsymbol{u}\}_{l=1}^{n}} s \tag{4.10a}$$

$$\text{s.t. } s \geq \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij}^{l} + \sum_{i\in I} p_i u_i^{l} \qquad \forall l \in \{1,\ldots,n\} \tag{4.10b}$$

$$\sum_{i\in I} x_{ij}^{l} \leq C_j - C_j(1-y_j)z_j^{l} \qquad \forall l \in \{1,\ldots,n\}, j \in J. \tag{4.10c}$$

$$\text{and (4.7c), (4.7f)–(4.7g), (4.8b)–(4.8c).} \tag{4.10d}$$

Constraints (4.10c) suggest that a fortified facility is always operational, even if it may be identified to be disrupted in some worst-case scenarios.

The subproblem is augmented through replacing the objective function and the two capacity-related constraints by

$$\psi = \max_{\substack{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}^{\alpha},\\ \boldsymbol{w}^{\beta},\boldsymbol{w}^{x},\boldsymbol{w}^{u},\boldsymbol{z}}} \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} + \sum_{i\in I} p_i u_i \tag{4.11a}$$

$$\sum_{i\in I} x_{ij} \leq C_j - C_j(1-\hat{y}_j)z_j \qquad \forall j \in J, \tag{4.11b}$$

$$\sum_{i\in I} x_{ij} \geq C_j - C_j(1-\hat{y}_j)z_j + M_j^{\beta}(\omega_j^{\beta} - 1) \qquad \forall j \in J. \tag{4.11c}$$

In terms of computational complexity, the master problem (4.10) has one more constraint, i.e., the budget constraint (4.8b), compared to the master problem in Section 4.3.1. And the number of constraints of the subproblem keeps unchanged. Meanwhile, no additional variables are introduced to the algorithm. Therefore, we can also expect the C&CG algorithm to solve the robust fortification model efficiently.

4.4 Computational Experiments

We adopt the instances generated in Chapter 3 with slight modifications, which are originally from [34]. These instances are derived from 1990 census data. The 49 nodes include the state capitals of the continental United States and Washington, D.C. There are 35 instances in

total. The nominal demand $\bar{h}_i = P_i \times 10^{-5}$, where P_i is the population at node i. We generate the maximal demand deviation h_i^{Δ} uniformly from the interval $[0.15\bar{h}, \bar{h}]$. The transportation cost d_{ij} is the great circle distance between nodes i and j in miles. For simplicity, we set the unit penalty cost p_i the same for all the customers, which is the greatest travel distance in the system. We denote instances as *Fac-X-Cus-Y*, which means that the considered instance has X candidate facilities and Y customers.

All the algorithms and models were coded in Python programming language, using Gurobi 8.1.1 as the solver. The calculations were run on a cluster of Lenovo SD350 servers with 2.4 GHz Intel Skylake cores and 202 GB of memory under Linux CentOS 7 system. Each experiment was conducted on a four-core processor of one node.

4.4.1 The Impact of Uncertainty on Optimal Solutions

We use instances Fac-10-Cus-10 and Fac-15-Cus-15 to conduct the experiments and set $\Gamma_h = 0.2|I|$ and $k = 2$. Results are presented in Table 4.2. We set the deterministic model's results as benchmarks and the other models' results are normalized by dividing those of the deterministic model. A ratio smaller (or larger) than 1 means that the robust models generate solutions of smaller (or larger) costs. Note that the nominal costs of the robust models are calculated by fixing the location decisions and solving the resulting minimum cost flow problems. The worst-case cost of the deterministic model is obtained by fixing the location decision and solving the subproblem of the C&CG algorithm with the same uncertainty parameters as the robust models.

Table 4.2 The impact of uncertainty on location decision and cost

Instance	Model	Opened facilities	Nominal cost ratio	Worst-case cost ratio
Fac-10-Cus-10	CFLP	[0, 2, 3, 6, 8]*	1.00	1.00
	CFLP-D	[0, 2, 3, 5, 6, 8]	1.04	0.94
	CFLP-R	[0, 2, 3, 4, 5, 6, 8]	1.08	0.68
	CFLP-DR	[0, 2, 3, 4, 5, 6, 7, 8]	1.13	0.73
Fac-15-Cus-15	CFLP	[0, 1, 2, 3, 4, 5, 7]	1.00	1.00
	CFLP-D	[0, 1, 2, 3, 4, 5, 7]	1.00	1.00
	CFLP-R	[0, 1, 2, 3, 4, 5, 7, 14]	1.09	0.94
	CFLP-DR	[0, 1, 2, 3, 4, 5, 7, 10, 14]	1.17	0.82

* Facilities are indexed from 0.

The first impact is the selection of opened facilities. As expected, when uncertainties are considered, more facilities are opened to mitigate potential risks. The CFLP-DR model generates solutions of the greatest number of opened facilities, due to the fact that two types of uncertainties are considered simultaneously. The second impact is cost. Generally, considering uncertainty increases the nominal cost and decreases the worst-case cost. Moreover,

the nominal cost of model CFLP-DR has the greatest increase in comparison with the deterministic model. A similar phenomenon is also observed by the authors of [57], where the system has the largest cost when two types of uncertainties are simultaneously considered, compared to one type of uncertainty.

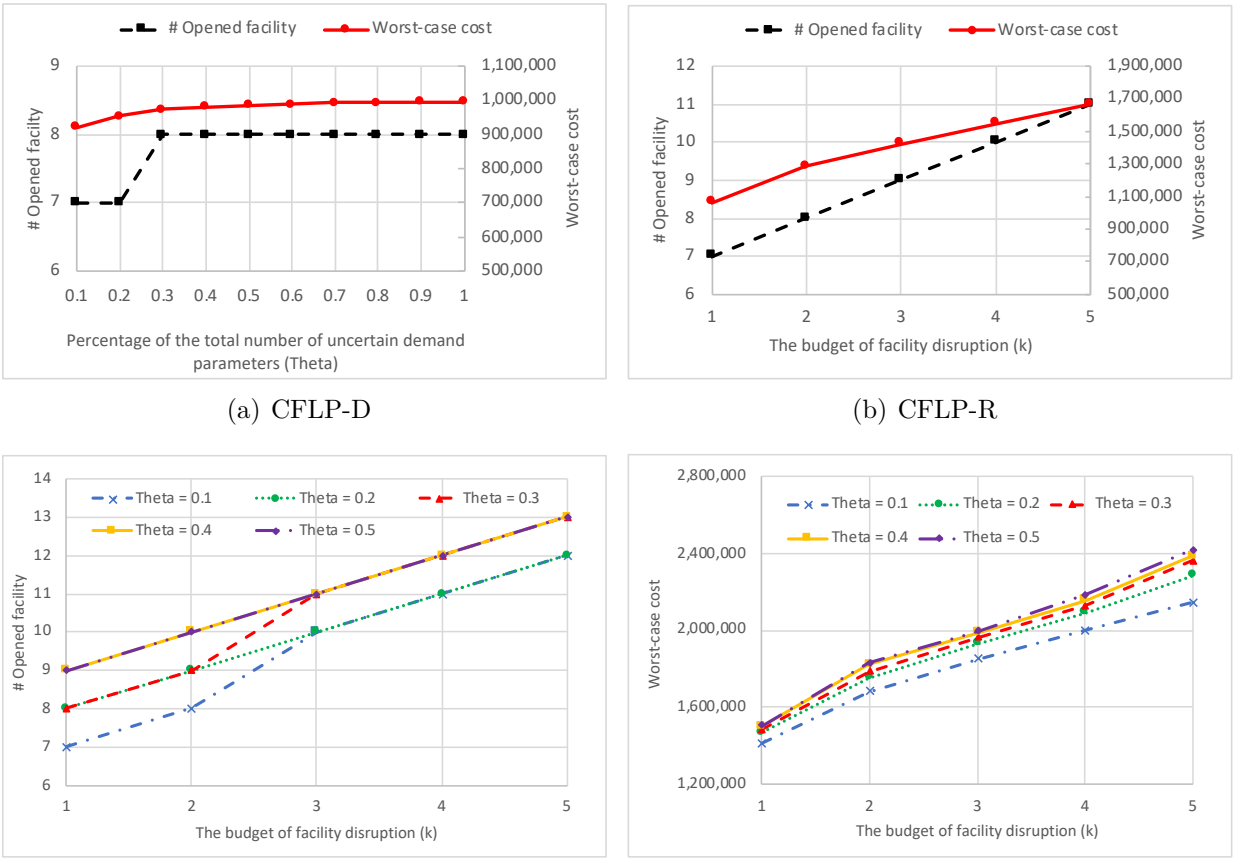
Table 4.2 also shows that facility disruption risk (or provider-side uncertainty) has a greater influence on the location decisions and the costs, compared to demand uncertainty. For instance Fac-10-Cus-10, the CFLP-D opens one more facility than the deterministic model. However, when disruption risk is further considered, the CFLP-DR locates two more facilities compared to the CFLP-D. For instance Fac-15-Cus-15, the location decision of the CFLP-D is the same as the deterministic problem. However, the CFLP-R and CFLP-DR generate different solutions with more opened facilities. The authors of [62] also indicate that the supplier's uncertainty is a more pressing issue than the effects of demand uncertainty. Their computational results show that 3% provider-side uncertainty has almost the same impact on the worst-case profit as 30% demand uncertainty. Table 4.2 further displays that sometimes a slight increase in the nominal cost can lead to a significant decrease in the worst-case cost. For example, for instance Fac-10-Cus-10, the nominal cost ratio of the CFLP-R (CFLP-DR) is 1.08 (1.13) whereas the worst-case cost ratio is 0.68 (0.73). This observation is consistent with other works that study facility location problems under disruptions [4,42].

4.4.2 The Impact of Uncertainty Budget on Optimal Solutions

We denote $\Gamma_h = \Theta|I|$, which means there are at most Θ of customers whose demand parameters are allowed to deviate from their nominal values. For space consideration, we only present the results of instance Fac-15-Cust-15 in Figure 4.1.

Figure 4.1(a) indicates that the budget of demand uncertainty has a slight impact on the location decision and the worst-case cost after a threshold. Specifically, when Θ increases from 0.2 to 0.3, one more facility is opened. When $\Theta \geq 0.3$, the number of opened facilities stays the same, and the rate of the cost growth is relatively small. The system does not open more facilities when the demand uncertainty increases, because it is more cost-efficient to simply penalize the unmet demand instead of paying the fixed expenses for locating new facilities. On the contrary, the budget of facility disruption has a significant influence on facility configuration and the worst-case cost. Figure 4.1(b) displays that the number of opened facilities and the worst-case cost increase quickly with respect to k for the CFLP-R. The authors of [132] also notice that the worst-case cost increases almost linearly over the uncertainty budget in a three-echelon network design problem under disruptions.

Figures 4.1(c)–4.1(d) are the summarized results of the CFLP-DR. The detailed results are



(a) CFLP-D

(b) CFLP-R

(c) CFLP-DR

(d) CFLP-DR

Figure 4.1 The impact of uncertainty budget on optimal solutions (Instance Fac-15-Cus-15)

given in Table 4.3, where the last column is obtained by setting the results in the first row of different Θ values as benchmarks. Both figures show that the uncertainty of facility disruptions plays a dominating role. Under the same value of k, the maximal difference in the number of opened facilities is 2, and the worst-case costs are also close. However, under the same value of Θ, the number of opened facilities and the worst-case cost ratio increase almost linearly with k. We note that sometimes even the number of opened facilities is the same under the same value of k for different Θ, there might be a difference in facility configuration. For example, according to Table 4.3, when $k = 4$, 11 sites are opened when $\Theta = 0.1$ and $\Theta = 0.2$. However, facilities 6 and 8 are opened in the former circumstance, and facilities 1 and 11 are opened in the latter. Thus, to improve supply chain flexibility via location decisions, we can either increase the number of opened facilities or change the facility configuration under a fixed number of opened facilities. For example, in the robust p-median problem [42], where exactly p facilities should be opened under all the situations, a selected set of p facilities to be opened in the first stage can differ when the uncertainty

budget changes.

Table 4.3 Detailed results of the CFLP-DR for Instance Fac-15-Cust-15

Θ	k	Opened facilities	# Opened facilities	Worst-case cost ratio
0.1	1	[0, 1, 2, 3, 4, 5, 7]	7	1.00
	2	[0, 1, 2, 4, 5, 7, 10, 14]	8	1.19
	3	[0, 1, 2, 4, 5, 7, 10, 11, 13, 14]	10	1.31
	4	[0, 2, 3, 4, 5, **6**, 7, **8**, 10, 13, 14]	11	1.41
	5	[0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 13, 14]	12	1.52
0.2	1	[0, 1, 2, 3, 4, 5, 7, 14]	8	1.00
	2	[0, 1, 2, 3, 4, 5, 7, 10, 14]	9	1.20
	3	[0, 1, 2, 4, 5, 7, 10, 11, 13, 14]	10	1.32
	4	[0, **1**, 2, 3, 4, 5, 7, 10, **11**, 13, 14]	11	1.42
	5	[0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 13, 14]	12	1.56
0.3	1	[0, 1, 2, 3, 4, 5, 7, 14]	8	1.00
	2	[0, 1, 2, 3, 4, 5, 7, 10, 14]	9	1.20
	3	[0, 2, 3, 4, 5, 7, 8, 10, **11**, 13, 14]	11	1.32
	4	[0, 2, 3, 4, 5, 6, 7, **8**, 10, 11, 13, 14]	12	1.43
	5	[0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14]	13	1.59
0.4	1	[0, 1, 2, 3, 4, 5, 7, 11, 14]	9	1.00
	2	[0, 1, 2, 3, 4, 5, 7, 10, 11, 14]	10	1.21
	3	[0, **1**, 2, 3, 4, 5, 7, 8, 10, 13, 14]	11	1.32
	4	[0, **1**, 2, 3, 4, 5, 6, 7, 10, 11, 13, 14]	12	1.44
	5	[0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14]	13	1.59
0.5	1	[0, 1, 2, 3, 4, 5, 7, 11, 14]	9	1.00
	2	[0, 1, 2, 3, 4, 5, 7, 10, 11, 14]	10	1.22
	3	[0, **1**, 2, 3, 4, 5, 7, 8, 10, 13, 14]	11	1.32
	4	[0, **1**, 2, 3, 4, 5, 6, 7, 10, 11, 13, 14]	12	1.45
	5	[0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14]	13	1.61

4.4.3 Analyses of Algorithm Performance

This section first compares our C&CG algorithm with that proposed in [1], and then explores the impact of uncertainty budget on algorithm performance. The optimality tolerance for algorithms is set to 0.01% and the CPU time limit is set to 7200 seconds. Note that, when this limit is reached, we still allow the current iteration to be finished unless the walltime limit–10800 seconds–is reached. In following tables, *#Opt* is the number of instances out of 35 that are solved to optimality. *#Iter* is the number of iterations. *Gap* is the optimality gap between the upper and lower bounds.

Performance Comparison

We compare the algorithms by setting $\Gamma_h = 0.2|I|$ and $k = 2$. Experiments are conducted on model CFLP-DR where both types of uncertainties are considered. In order to use the C&CG algorithm in [1] for this problem, the assumption that all the transportation costs $d_{ij}, i \in I, j \in J$ and penalty costs $p_i, i \in I$ are integer is required. Thus, we round them to

the nearest integers in the experiments in this subsection. Results are reported in Table 4.4, where $\#Fac$ and $\#Cust$ are the number of facilities and customers, respectively. We mark the results with less CPU time and a smaller optimality gap in bold.

Table 4.4 displays that our C&CG algorithm can solve 33 out of 35 instances to optimality within the time limit. However, the C&CG algorithm proposed in [1] can only solve 18 instances and the maximum walltime limit is reached in 14 instances. For the 18 instances that are solved to optimality by both algorithms, our C&CG algorithm generally consumes less CPU time. We further observe that the time difference of the two algorithms mainly results from the resolution time used in the subproblem. As indicated by Table 4.1, the size of the subproblem in [1] is related to the value of parameter P. Thus, for realistic size problems with relatively high flow costs, the subproblem in [1] has a greater computational complexity in comparison with the subproblem derived from the KKT condition. Based on the results here, our C&CG algorithm is used for further experiments in the subsequent sections.

The Impact of Uncertainty Budget on Algorithm Performance

This section studies the impact of uncertainty budget on algorithm performance. Experiments are conducted for different types of uncertainties as presented in Section 4.2.2 to provide computational insights.

The CFLP-D. Table 4.5 and Figure 4.2 present the results of the CFLP-D. Table 4.5 shows that the C&CG algorithm can generate optimal solutions for all the instances with different budgets in a short time, and the iteration number of the algorithm is also small. Figure 4.2(a) shows that the #Iter first increases and then decreases with the increasing uncertainty budget. When $\Theta = 1$, the C&CG algorithm finds optimal solutions in only 1 iteration and identifies that all the uncertain parameters would take value 1. Figure 4.2(b) shows that for the CFLP-D, most time is consumed to solve the subproblem and the computing time of the master problem is relatively shorter.

The CFLP-R. Table 4.6 and Figure 4.3 present the results of the CFLP-R. Table 4.6 displays that the C&CG algorithm can solve all the instances to optimality for a small budget $k = 1$ and 2 in a short time. Although the #Opt decreases to 25 and 18 for $k = 3$ and 4, the average optimality gap is not significant, i.e., 2.32% and 6.83% respectively. Figure 4.3 shows that the number of iterations and the total computing time increase over k. In contrast to the results of the CFLP-D, the master problem consumes more CPU time than the subproblem. This can be explained by the fact that as the number of iterations increases, the size of the master problem gradually increases as more variables and constraints are added to it.

Table 4.4 Algorithm comparison on model CFLP-DR

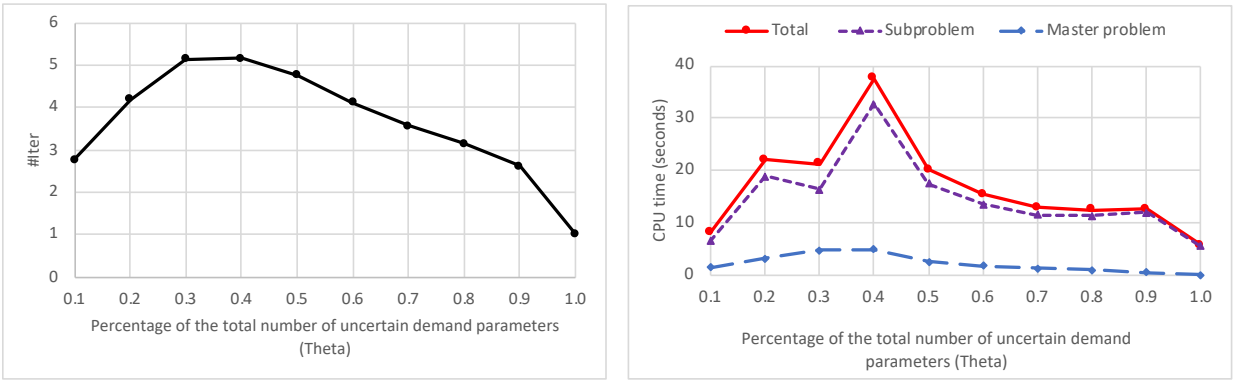
		Our C&CG				C&CG in [1]			
		CPU time (seconds)				CPU time (seconds)			
#Fac	#Cust	Total	MP	SP	Gap	Total	MP	SP	Gap
10	10	**2.91**	0.23	2.68	0.00	5.57	0.23	5.34	0.00
	15	**5.42**	0.24	5.18	0.00	24.75	0.28	24.47	0.00
	20	**28.69**	0.51	28.18	0.00	42.48	0.53	41.95	0.00
	25	**19.47**	0.34	19.13	0.00	279.67	0.35	279.32	0.00
	30	**59.99**	0.49	59.50	0.00	116.03	0.57	115.46	0.00
	35	**218.09**	0.32	217.77	0.00	307.52	0.34	307.18	0.00
	40	189.74	0.12	189.62	0.00	**79.70**	0.13	79.57	0.00
	45	291.14	0.40	290.74	0.00	**135.88**	0.46	135.42	0.00
	49	**355.35**	0.50	354.85	0.00	431.46	0.58	430.88	0.00
15	15	**23.86**	2.23	21.63	0.00	27.62	2.21	25.41	0.00
	20	**21.39**	0.85	20.54	0.00	351.66	0.88	350.78	0.00
	25	**79.17**	5.86	73.31	0.00	1242.28	6.22	1236.06	0.00
	30	**196.99**	6.15	190.84	0.00	4211.47	6.41	4205.06	0.00
	35	**84.79**	1.39	83.40	0.00	2825.44	1.48	2823.96	0.00
	40	**298.99**	7.99	291.00	0.00	7981.79	4.07	7977.72	1.08
	45	**435.26**	3.29	431.97	0.00	T			
	49	**560.31**	5.22	555.09	0.00	T			
20	20	**38.34**	2.85	35.49	0.00	619.23	3.54	615.69	0.00
	25	**290.60**	31.07	259.53	0.00	2274.41	33.64	2240.77	0.00
	30	**195.57**	18.70	176.87	0.00	8895.59	20.13	8875.46	0.00
	35	**299.15**	18.51	280.64	0.00	8197.95	0.54	8197.41	30.83
	40	**412.84**	17.29	395.55	0.00	T			
	45	**546.90**	11.23	535.67	0.00	8550.16	0.78	8549.38	19.51
	49	**645.02**	33.20	611.82	0.00	T			
25	25	**242.27**	46.25	196.02	0.00	6409.23	47.87	6361.36	0.00
	30	**404.90**	80.88	324.02	0.00	T			
	35	**456.49**	131.57	324.92	0.00	T			
	40	**709.49**	132.95	576.54	0.00	T			
	45	**2566.22**	261.17	2305.05	0.00	T			
	49	**3694.82**	120.48	3574.34	0.00	T			
30	30	**892.16**	292.67	599.49	0.00	T			
	35	**2075.79**	912.55	1163.24	0.00	T			
	40	**2083.45**	977.17	1106.28	0.00	T			
	45	7309.41	3961.57	3347.84	**1.54**	T			
	49	7327.06	5432.46	1894.60	**0.14**	T			

T: the walltime limit is reached.

The CFLP-DR. Table 4.7 summarizes the results of the CFLP-DR. It shows that when both uncertainties are simultaneously considered, fewer instances (33, 33, 34 out of 35) are solved to optimality for the case of $k = 2$, whereas all the instances are optimally solved for the CFLP-R when $k = 2$. When $k = 3$ and 4, the optimality gaps are close to those of the CFLP-R. However, we can observe that the gaps increase over Θ under the same value of k in general. Figure 4.4(a) further shows that when $k = 4$, the #Opt has a relatively larger variation under different values of Θ, varying between 22 and 18. We also notice that under the same value of Θ, the #Opt has an even larger variation with respect to k, especially for the case of $\Theta = 0.6$ (#Opt varies between 35 and 18). Figure 4.4(b) displays that the case of $\Theta = 0.2$ has the maximal number of iterations in general, leading to a longer computing time

Table 4.5 Algorithm performance for the CFLP-D over different budgets (average results)

Θ	#Opt	#Iter	Gap	Computing time		
				Total	MP	Subproblem
0.1	35/35	2.77	0.00	7.99	1.48	6.49
0.2	35/35	4.20	0.00	22.02	3.13	18.85
0.3	35/35	5.14	0.00	21.32	4.82	16.45
0.4	35/35	5.17	0.00	37.55	4.87	32.63
0.5	35/35	4.77	0.00	20.08	2.60	17.43
0.6	35/35	4.11	0.00	15.43	1.86	13.54
0.7	35/35	3.57	0.00	12.95	1.37	11.56
0.8	35/35	3.14	0.00	12.38	0.95	11.41
0.9	35/35	2.63	0.00	12.62	0.56	12.04
1.0	35/35	1.00	0.00	5.65	0.08	5.57



(a) #Iter (b) CPU time

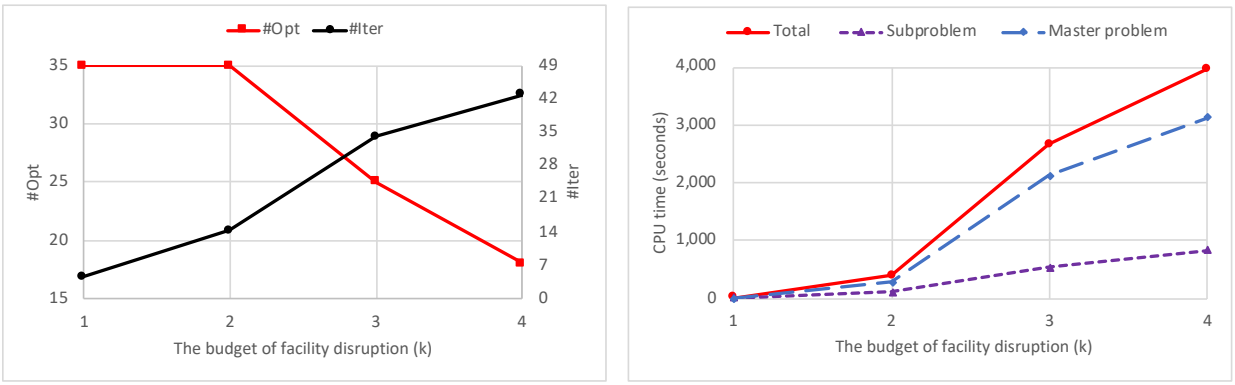
Figure 4.2 Average results of algorithm performance for the CFLP-D

for the master problem (as shown in Figure 4.4(d)); however, for the other two cases, their subproblems consume much more time, resulting in a longer total CPU time as reported in Figure 4.4(c). In addition, we can observe that the disruption risk has a greater impact on computational efficiency than that of the demand uncertainty. In particular, under a fixed value of k, the variations of the #Opt and the total CPU time over Θ are not as obvious as those under a fixed value of Θ but with a varying k.

Summaries. (1) Among the three robust models, the CFLP-D is the easiest one in terms of computational complexity, for which all the instances can be solved to optimality in a short time framework. (2) For the CFLP-R, when the uncertainty budget is small, all the instances can be optimally solved; for a large budget, the average optimality gap is promising within the time limit. Meanwhile, the master problem consumes more time than the subproblem because of the great number of iterations. (3) The CFLP-DR is the most difficult one among the three robust models, which requires the most computational time. The reference [57] also finds that the robust counterpart under two types of uncertainties requires the longest

Table 4.6 Algorithm performance for the CFLP-R over different budgets (average results)

k	#Opt	#Iter	Gap	Computing time		
				Total	MP	Subproblem
1	35/35	4.57	0.00	16.12	5.84	10.25
2	35/35	14.37	0.00	390.27	281.74	108.25
3	25/35	34.11	2.32	2660.07	2124.10	534.74
4	18/35	42.91	6.83	3970.05	3126.92	841.44



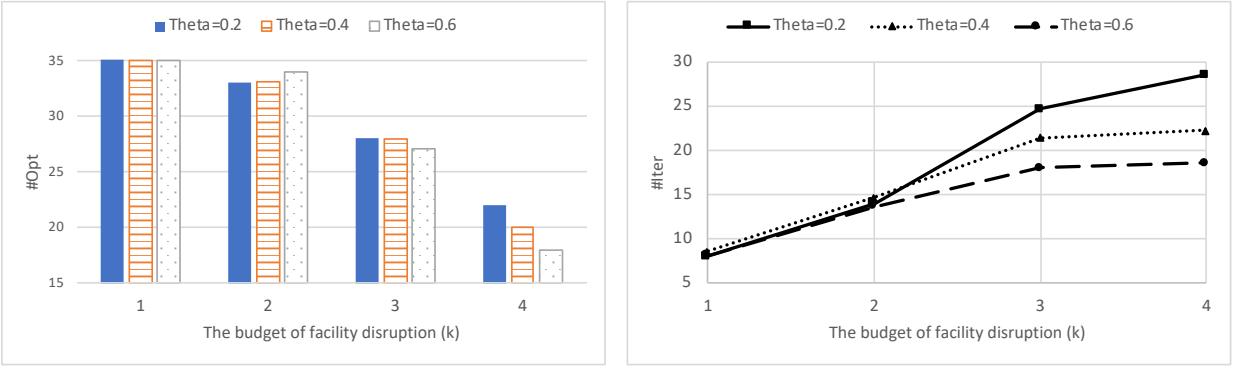
(a) #Opt and #Iter

(b) CPU time

Figure 4.3 Average results of algorithm performance for the CFLP-R

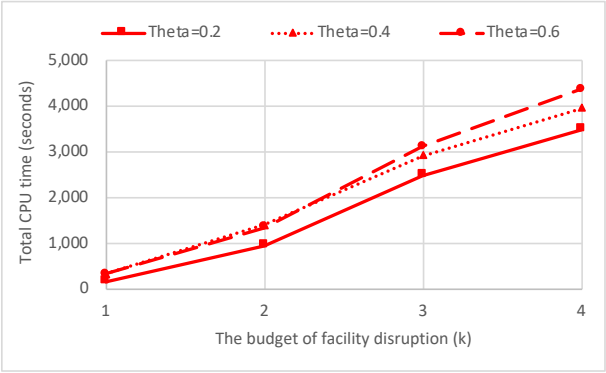
Table 4.7 Algorithm performance for the CFLP-DR over different budgets (average results)

Θ	k	#Opt	#Iter	Gap	Computing time		
					Total	MP	Subproblem
0.2	1	35/35	8.00	0.00	163.50	27.31	136.06
	2	33/35	14.00	0.05	944.05	348.08	595.64
	3	28/35	24.63	1.38	2500.03	641.84	1857.46
	4	22/35	28.49	5.15	3494.25	940.69	2552.62
0.4	1	35/35	8.57	0.00	331.03	28.37	302.53
	2	33/35	14.69	0.04	1391.08	319.11	1071.62
	3	28/35	21.40	2.02	2915.23	387.11	2527.53
	4	20/35	22.29	7.24	3960.69	513.61	3446.44
0.6	1	35/35	8.14	0.00	325.37	12.51	312.75
	2	34/35	13.69	0.07	1344.43	170.79	1173.34
	3	27/35	18.00	2.74	3114.85	285.34	2829.04
	4	18/35	18.60	7.87	4372.40	316.42	4055.53

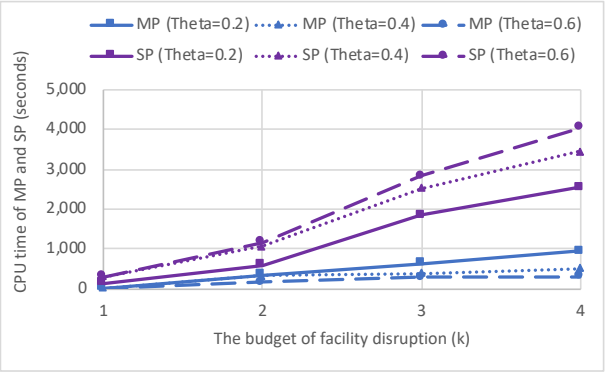


(a) #Opt

(b) #Iter



(c) Total CPU time



(d) CPU time of master and sub problems

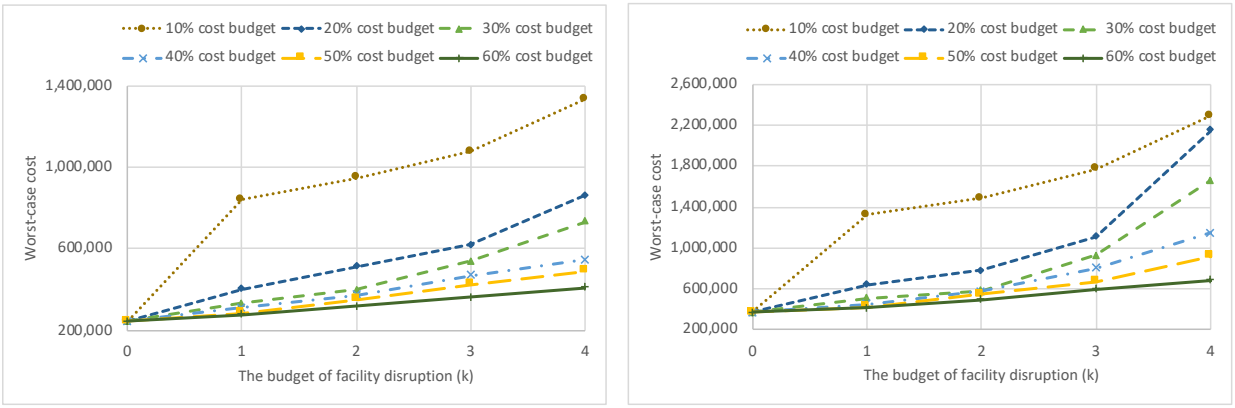
Figure 4.4 Average results of different algorithms for the CFLP-DR

computing time, compared to that under a single source of uncertainty. (4) The disruption risk has a larger impact on algorithm performance than the demand uncertainty.

4.4.4 Insights from the Robust Fortification Model

This section provides results for the robust fortification models. Experiments are conducted using a randomly selected instance Fac-10-Cus-30. For the models with both disruptions and uncertain demand, we set the budget of demand uncertainty $\Gamma_h = 0.3|I|$.

In the first group of experiments, we study the impact of budget B on the system's worst-case cost by setting $B = \sigma \sum_{j \in J} f_j$, where $\sigma \in \{10\%, \cdots, 60\%\}$. Results are reported in Figure 4.5. It shows that the worst-case cost decreases as the cost budget increases because more investments lead to more flexibility in facility fortification decisions, such as increasing the number of fortified facilities or fortifying an expensive facility with a large capacity. When σ increases from 10% to 30%, the worst-case cost has a quick decrease under the same value of k, especially for the case with both types of uncertainties (as shown in Figure 4.5(b)).



(a) Only facility disruptions (b) Both facility disruptions and demand uncertainty

Figure 4.5 Worst-case cost under different cost budgets and uncertainty budgets (Instance Fac-10-Cus-30)

However, if we further increase the cost budget from 30% to 60%, the worst-case cost begins to decrease slowly, in particular when $k \leq 3$. Thus, the robust fortification models can be used as decision tools for managers to see the trade-off between the investment in protection measures and the resulting robustness of a system.

Next, we compare the robust fortification models with the robust CFLP models. To perform the experiments, we first solve the robust CFLP models and get the location decisions and costs. For the robust fortification models, we then set the budget B as the location cost of the corresponding robust CFLP model. This experimental setting is to explore the impacts of different strategies (obtained from different models) under the same cost budget on system robustness. Results are reported in Table 4.8. It shows that the location or fortification costs of the two types of models are very close; however, there is a significant gap in their worst-case flow costs. This is because in the fortification models, if a facility is chosen to be protected, it becomes immune to disruptions; whereas in the robust CFLP models, a new located facility can be disrupted later and thus loses its capacity completely. Decision-makers can then leverage these models based on their cost budget to make an optimal trade-off among different strategies to improve overall supply chain system's reliability and resilience.

4.5 Conclusions

This chapter solves a fixed-charge location problem where two types of parameters are subject to uncertainties simultaneously: demand and facility availability. We apply a two-stage RO framework for the problem, which allows the allocation decisions to be made after the

Table 4.8 Comparison between robust CFLP models and robust fortification models

	CFLP-R			Fortification model with only facility disruptions		
k	Opened facilities	Facility cost	Worst-case flow Cost	Fortified facilities	Facility cost	Worst-case flow cost
1	[0, 2, 3, 4, 5, 7, 8]	478100	966320.64	[0, 1, 2, 3, 5]	421800	276697.39
2	[0, 2, 3, 4, 5, 6, 7, 8]	544100	1203400.07	[0, 1, 2, 3, 4, 5, 6]	526200	295728.79
3	[0, 2, 3, 4, 5, 6, 7, 8, 9]	640700	1284592.17	[0, 1, 2, 3, 4, 5, 6, 9]	622800	270611.69
4	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	742500	1510399.81	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	742500	242932.87
	CFLP-DR			Fortification model with both types of uncertainties		
k	Opened facilities	Facility cost	Worst-case flow cost	Fortified facilities	Facility cost	Worst-case flow cost
1	[0, 2, 3, 4, 5, 6, 7, 8]	544100	1405946.47	[0, 1, 2, 3, 5, 9]	518400	407327.56
2	[0, 2, 3, 4, 5, 6, 7, 8, 9]	640700	1728833.74	[0, 1, 2, 3, 4, 5, 6, 9]	622800	403133.55
3	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	742500	1982132.83	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	742500	371110.91
4	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	742500	2546018.16	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	742500	371110.91

uncertainties are realized. We implement the C&CG method proposed in [14] to solve the robust models exactly, and further extend the modeling and solution schemes to facility fortification problems under uncertainties. We benchmark our C&CG algorithm with the one developed in [1], where the second-stage problem is reformulated to a minimum cost flow problem. Numerical tests show that our algorithm can solve more instances to optimality and generally outperform the benchmark approach. Results also indicate that, among the three robust models, the CFLP-D is the easiest one in terms of computational complexity, for which all the instances can be optimally solved under different budgets. The CFLP-DR is the most difficult one, as two types of uncertainties are simultaneously considered. Our tests further demonstrate that disruption risk (or provider-side uncertainty) has a greater effect on solution configuration and cost, compared to demand uncertainty. Our solution optimization framework allows decision-makers to determine an optimal and robust facility location decision to improve overall reliability and resilience of the supply chain facing complex uncertainties.

CHAPTER 5 A TWO-STAGE ROBUST APPROACH FOR THE RELIABLE LOGISTICS NETWORK DESIGN PROBLEM

This chapter is based on the following article.

- Cheng, C., Qi, M., Zhang, Y., Rousseau, L.-M., 2018. A Two-Stage Robust Approach for the Reliable Logistics Network Design Problem. *Transportation Research Part B: Methodological*, 111, 185-202.

5.1 Introduction

The logistics network design problem (LNDP) is key to achieving efficient operations among suppliers, manufacturers, and customers [133]. Compared to the classical facility location problem, it considers multiple echelons and decides the number of suppliers and warehouses, their locations and capacities, and the product flow throughout the network [134]. One importance aspect in LNDP is to deal with uncertainty, like uncertain set-up costs of facilities, uncertain transportation costs and customer demand [53,135]. Facility disruption is another type of uncertainty. In this chapter, we use a two-stage RO scheme for a LNDP under disruptions. Our study makes the following contributions:

1. To the best of our knowledge, this study is the first to solve the reliable LNDP using a two-stage RO approach, which can produce less conservative solutions.

2. The adjustable robust model can be extended to include multiple uncertainty sets and impose upper bounds on the worst-case performance of these sets. It can also be extended to problems with partial disruptions.

3. We introduce a variable fixing technique to enhance the C&CG algorithm, which outperforms the basic C&CG algorithm and the BD method. We present managerial insights based on the numerical results.

We note that our work differs from those on the reliable PMP and the reliable UFLP by considering multiple echelons and facility capacities. Specifically, our work differs from two related papers ([42] and [124]) in the following aspects: (i) The authors of [42] focus on analyzing the structural properties of the robust PMP and exploring the modeling capability of the two-stage RO by considering capacitated PMP and demand changes. However, we extend the basic RO scheme to include multiple uncertainty sets to characterize decision

makers' conservative levels. And the model in [42] can be recognized as a special case of this modeling scheme. We also introduce upper bounds for the worst-case performance, and this modeling framework can be used as a decision support tool for system expansion with reliability considerations. Our numerical tests demonstrate that the new models can generate less conservative solutions. (ii) The authors of [124] present robust unit commitment models, where the load is subject to interval uncertainty. They focus on building the connection between the robust models and the stochastic models. However, we apply the modeling scheme to a reliable LNDP and explain the connections and differences among various models. Extensive numerical tests are conducted to study the conservativeness of different models and the price of robustness. Values of key parameters are also analyzed in our study to provide insights for supply chain decision-makers.

The rest of this chapter is organized as follows. Section 5.2 describes our problem and presents three two-stage RO models. Section 5.3 introduces the C&CG algorithm, and Section 5.4 presents the numerical results. Section 5.5 concludes our work.

5.2 Mathematical Models

In this section, we first introduce our notation. We then present a basic two-stage RO model for the reliable LNDP and explore the modeling capability of two-stage RO by describing two variants of the basic model: the *expanded robust* LNDP model and the *risk-constrained robust* LNDP model.

5.2.1 Notation

We consider a general network $(\mathcal{V}, \mathcal{A})$. Let $\mathcal{V}_\mathcal{S}$, $\mathcal{V}_\mathcal{T}$, and $\mathcal{V}_\mathcal{D}$ be the sets of supply, transshipment, and demand nodes. Define $\mathcal{V}_0 = \mathcal{V}_\mathcal{S} \cup \mathcal{V}_\mathcal{T}$ to be the set of facilities for which open/close decisions are required, and $\mathcal{V} = \mathcal{V}_0 \cup \mathcal{V}_\mathcal{D}$. f_j is the fixed cost to open facility $j \in \mathcal{V}_0$. c_{ij} is the unit transportation cost on arc $(i, j) \in \mathcal{A}$. Q_j is the capacity of facility $j \in \mathcal{V}_0$. b_j is the supply of node $j \in \mathcal{V}$. $b_j \geq 0$ if $j \in \mathcal{V}_\mathcal{S}$, $b_j = 0$ if $j \in \mathcal{V}_\mathcal{T}$, and $b_j \leq 0$ if $j \in \mathcal{V}_\mathcal{D}$. θ_i is the unit penalty cost for unsatisfied demand at node $i \in \mathcal{V}_\mathcal{D}$. Variable $y_j = 1$ if facility $j \in \mathcal{V}_0$ is opened in the first stage, $y_j = 0$ otherwise. Variable x_{ij} is the product flow on arc $(i, j) \in \mathcal{A}$ in a specific disruptive scenario. u_i is the unsatisfied demand at node $i \in \mathcal{V}_\mathcal{D}$ in a specific disruptive scenario.

Our two-stage robust optimization model uses a budgeted uncertainty set to describe possible disruptive scenarios without requiring any probabilistic information. We assume that at most

m facilities fail simultaneously:

$$A = \left\{ \mathbf{z} \in \{0,1\}^{|\mathcal{V}_0|} : \sum_{j \in \mathcal{V}_0} z_j \leq m \right\}, \tag{5.1}$$

where $z_j = 1$ if facility j is disrupted and $z_j = 0$ otherwise.

5.2.2 Formulations

We first present the basic two-stage RO model, in which only one uncertainty set is considered. Then we introduce two models with multiple uncertainty sets.

Basic Two-stage RO Model

We formulate the basic two-stage RO LNDP model as follows, where $\mathcal{V}_i^+ = \{j \in \mathcal{V} | (i,j) \in \mathcal{A}\}$ and $\mathcal{V}_i^- = \{j \in \mathcal{V} | (j,i) \in \mathcal{A}\}$.

RO-LNDP$_0$:

$$\min_{\mathbf{y}} \sum_{j \in \mathcal{V}_0} f_j y_j + \max_{\mathbf{z} \in A} \min_{\mathbf{x},\mathbf{u} \in S(\mathbf{y},\mathbf{z})} \Big(\sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} + \sum_{j \in \mathcal{V}_\mathcal{D}} \theta_j u_j \Big) \tag{5.2a}$$

$$\text{s.t. } y_j \in \{0,1\} \quad \forall j \in \mathcal{V}_0, \tag{5.2b}$$

$$\text{where } S(\mathbf{y},\mathbf{z}) = \Big\{ \sum_{i \in \mathcal{V}_j^+} x_{ji} \leq b_j \qquad \forall j \in \mathcal{V}_\mathcal{S}, \tag{5.2c}$$

$$\sum_{i \in \mathcal{V}_j^+} x_{ji} = \sum_{i \in \mathcal{V}_j^-} x_{ij} \qquad \forall j \in \mathcal{V}_\mathcal{T}, \tag{5.2d}$$

$$\sum_{i \in \mathcal{V}_j^-} x_{ij} + u_j = -b_j \qquad \forall j \in \mathcal{V}_\mathcal{D}, \tag{5.2e}$$

$$\sum_{i \in \mathcal{V}_j^+} x_{ji} \leq (1 - z_j) Q_j y_j \qquad \forall j \in \mathcal{V}_0, \tag{5.2f}$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in \mathcal{A}, \tag{5.2g}$$

$$u_j \geq 0 \qquad \forall j \in \mathcal{V}_\mathcal{D} \Big\}. \tag{5.2h}$$

The objective function in (5.2a) minimizes the cost of the worst-case scenario. The *max* operator represents the disruptive scenario in A that generates the largest recourse cost, given the facility locations $\mathbf{y}$. The *min* operator identifies the least costly solution, and the set $S(\mathbf{y},\mathbf{z})$ represents possible recourse operations. Constraints (5.2c)–(5.2e) are the product flow conservation equations for all nodes. Constraints (5.2f) ensure that when a facility is

opened and functional the flow does not exceed its capacity, and they prohibit any flow when it is closed or destroyed. Constraints (5.2b), (5.2g), and (5.2h) define the variable types.

Expanded Two-stage RO Model

Two key factors influence the solution of the two-stage RO: the uncertainty set and the worst-case performance. A small uncertainty set cannot adequately capture the random factor; a large set leads to solutions that are costly and overly conservative. To deal with this, the authors of [124] suggest using multiple uncertainty sets and assigning different weights to the worst-case performances of these sets. Our RO-LNDP$_0$ model can be extended to multiple uncertainty sets as follows:

RO-LNDP$_1$:

$$\min_{\mathbf{y}} \sum_{j\in\mathcal{V}_0} f_j y_j + \sum_{k=1}^{K} \rho_k \left(\max_{\mathbf{z}\in A_k} \min_{\mathbf{x},\mathbf{u}\in S_k(\mathbf{y},\mathbf{z})} \Big(\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + \sum_{j\in\mathcal{V}_\mathcal{D}} \theta_j u_j \Big) \right) \tag{5.3}$$

where $\mathbf{y}$ and $S_k(\mathbf{y},\mathbf{z})$ are defined by constraints (5.2b) and (5.2c)–(5.2h), respectively. In the objective function (5.3), A_k denotes the kth uncertainty set, with weight ρ_k ($0 \le \rho_k \le 1$ and $\sum_k \rho_k = 1$).

Risk-constrained Two-stage RO Model

Another way to guarantee the quality of the solutions is to impose upper bounds on the worst-case performance of the uncertainty sets. Then any solution that is feasible with respect to these uncertainty sets provides a performance guarantee. We extend our RO-LNDP$_0$ model to introduce the *risk-constrained robust* LNDP model, where ξ_k is the performance restriction on uncertainty set $A_k, \forall k \in \{1,\ldots,|K|\}$, and x_{ij0} and u_{j0} are the normal disruption-free decisions.

RO-LNDP$_2$:

$$\min \sum_{j\in\mathcal{V}_0} f_j y_j + \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij0} + \sum_{j\in\mathcal{V}_\mathcal{D}} \theta_j u_{j0} \tag{5.4a}$$

$$\text{s.t.} \sum_{i\in\mathcal{V}_j^+} x_{ji0} \le b_j \qquad \forall j \in \mathcal{V}_\mathcal{S}, \tag{5.4b}$$

$$\sum_{i\in\mathcal{V}_j^+} x_{ji0} = \sum_{i\in\mathcal{V}_j^-} x_{ij0} \qquad \forall j \in \mathcal{V}_\mathcal{T}, \tag{5.4c}$$

$$\sum_{i\in\mathcal{V}_j^-} x_{ij0} + u_{j0} = -b_j \qquad \forall j \in \mathcal{V}_\mathcal{D}, \tag{5.4d}$$

$$\sum_{i \in \mathcal{V}_j^+} x_{ji0} \leq Q_j y_j \qquad \forall j \in \mathcal{V}_0, \tag{5.4e}$$

$$\max_{\mathbf{z} \in A_k} \min_{\mathbf{x},\mathbf{u} \in S_k(\mathbf{y},\mathbf{z})} \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} + \sum_{j \in \mathcal{V}_\mathcal{D}} \theta_j u_j \leq \xi_k, \qquad \forall k \in \{1, \ldots, |K|\}, \tag{5.4f}$$

$$y_j \in \{0, 1\} \qquad \forall j \in \mathcal{V}_0, \tag{5.4g}$$

$$x_{ij0} \geq 0 \qquad \forall (i,j) \in \mathcal{A}, \tag{5.4h}$$

$$u_{j0} \geq 0 \qquad \forall j \in \mathcal{V}_\mathcal{D}, \tag{5.4i}$$

where $S_k(\mathbf{y}, \mathbf{z})$ is defined by (5.2c)–(5.2h).

For comparison purposes, in Appendix C we formulate the generic (or deterministic) LNDP (G-LNDP) that ignores disruptions. In Appendix C, we give the formulation of the two-stage stochastic programming (SP) model.

Summary of Models and Another Extension

The connections and differences among models are shown in Figure 5.1 and Table 5.1. The G-LNDP model is a special case of the RO-LNDP$_0$ model with $m = 0$, and also a special case of the RO-LNDP$_1$ model with $K = 1$ and $m = 0$. The RO-LNDP$_1$ model reduces to the RO-LNDP$_0$ model when $K = 1$ and $m > 0$. Furthermore, the authors of [124] have proved that the RO-LNDP$_1$ model is equivalent to the SP model when the uncertainty sets are individual scenarios. The RO-LNDP$_2$ model reduces to the G-LNDP model when the performance bound ξ_k is sufficiently large.

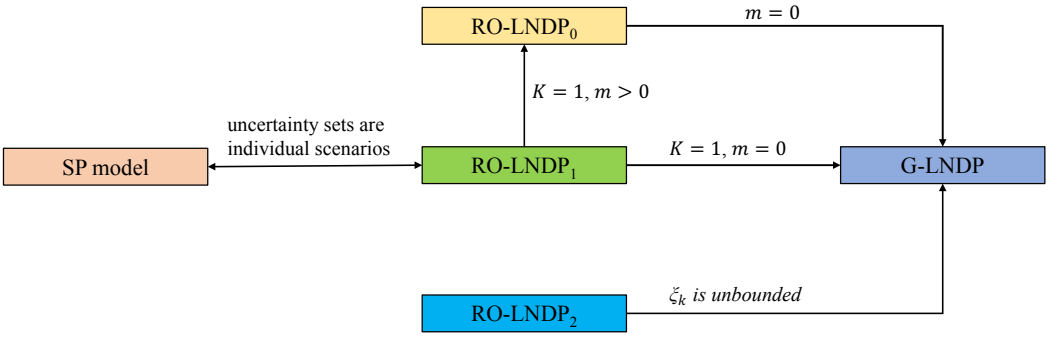


Figure 5.1 Connections among models

From Table 5.1, our three two-stage robust models differ in the number of uncertainty sets and the objective function. The basic RO model has one uncertainty set, and the other models have multiple uncertainty sets. The objective of the basic RO model is to minimize the cost of the worst-case scenario in the uncertainty set. The expanded RO model identifies

the worst-case scenario in each uncertainty set and minimizes the weighted sum. The risk-constrained RO model minimizes the cost of the normal disruption-free situation.

Table 5.1 Model comparison

Model	Possibility information required?	Uncertainty set/scenarios	Objective (minimize cost)
Deterministic LNDP	Not applicable	Not applicable	Normal disruption-free case
Basic RO	No	One	Worst-case
Expanded RO	No	Multiple	Weighted sum of multiple worst-cases
Risk-constrained RO	No	Multiple	Normal disruption-free case
Stochastic programming	Yes	Multiple	Weighted sum of multiple scenarios

Another extension of our work allows partial disruption in which a damaged facility can still satisfy part of the demand. We introduce a parameter δ_j $(0 < \delta_j \leq 1)$ to represent the change in facility j's capacity in a disruptive scenario, and constraints (5.2f) become

$$\sum_{i \in \mathcal{V}_j^+} x_{ji} \leq (1 - \delta_j z_j) Q_j y_j \quad \forall j \in \mathcal{V}_0 \tag{5.5}$$

5.3 Solution Method

Two-stage RO models are usually difficult to solve [121]. Although BD can be used to find optimal solutions for the two-stage problem (if it is linear), it is not efficient for large problems. Recently, the C&CG algorithm has been developed to solve two-stage RO models. It has performed well on unit commitment problems [124,136] and p-median facility location problems [42]. We use this algorithm, and we introduce an enhancement strategy to further improve its computational efficiency.

5.3.1 C&CG Algorithm

The C&CG algorithm is implemented in a master-subproblem framework. In the subproblem, the solution for the master problem (i.e., the location decision) is known and we solve the remaining max–min problem. Since unmet demand will be penalized in disruptive scenarios, the second-stage problem is always feasible. Therefore, we find its dual and obtain a max–max problem, which can be merged into a maximization problem. We describe our algorithm for RO-LNDP$_0$; the other RO models can be solved with minor modifications.

Dual of the Second-stage Problem

We introduce the dual variables $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$, and $\boldsymbol{\pi}$ for constraints (5.2c), (5.2d), (5.2e), and (5.2f), respectively. The resulting dual problem is as follows:

$$
\begin{aligned}
\max \ & \sum_{j \in \mathcal{V}_\mathcal{S}} b_j \alpha_j - \sum_{j \in \mathcal{V}_\mathcal{D}} b_j \gamma_j + \sum_{j \in \mathcal{V}_0} (1 - z_j) Q_j \hat{y}_j \pi_j \\
\text{s.t. } & \alpha_i + \beta_j + \pi_i \leq c_{ij} && \forall i \in \mathcal{V}_\mathcal{S}, j \in \mathcal{V}_\mathcal{T} \cap \mathcal{V}_i^+, \\
& \alpha_i + \gamma_j + \pi_i \leq c_{ij} && \forall i \in \mathcal{V}_\mathcal{S}, j \in \mathcal{V}_\mathcal{D} \cap \mathcal{V}_i^+, \\
& -\beta_i + \gamma_j + \pi_i \leq c_{ij} && \forall i \in \mathcal{V}_\mathcal{T}, j \in \mathcal{V}_\mathcal{D} \cap \mathcal{V}_i^+, \\
& \gamma_j \leq \theta_j && \forall j \in \mathcal{V}_\mathcal{D}, \\
& \alpha_j \leq 0 && \forall j \in \mathcal{V}_\mathcal{S}, \\
& \pi_j \leq 0 && \forall j \in \mathcal{V}_0.
\end{aligned}
$$

Since $\pi_j \leq 0$ and $z_j \in \{0, 1\}$, the nonlinear term $z_j \pi_j$ in the objective function is the product of a binary variable and a continuous variable. We can linearize it by introducing a new continuous variable $w_j = z_j \pi_j$ and using a big-M method, with the following constraints:

$$
\begin{aligned}
& w_j \geq \pi_j, \\
& w_j \geq -M z_j, \\
& w_j \leq \pi_j + M(1 - z_j), \\
& w_j \leq 0.
\end{aligned}
$$

The value of M can be set as follows: For each facility $j \in \mathcal{V}_0$ and demand node $i \in \mathcal{V}_\mathcal{D}$, define c'_{ji} as the minimal cost from j to i, i.e., $c'_{ji} = \min\{c_{ji}, c_{jk} + c_{ki}\}$, where k is a transshipment node. If there is no arc between facility j and demand node i, then $c_{ji} = +\infty$. Define $M'_j = \max_{i \in \mathcal{V}_\mathcal{D}} \{\theta_i - c'_{ji}\}$ and $M''_j = \max_{i \in \mathcal{V}_j^+} \{c_{ji}\}$. Then we set

$$
-\pi_j \leq \max\{M'_j, M''_j\} = M_j \quad j \in \mathcal{V}_0. \tag{5.6}
$$

Therefore, the linearized objective function of the subproblem is

$$
\chi = \max \sum_{j \in \mathcal{V}_\mathcal{S}} b_j \alpha_j - \sum_{j \in \mathcal{V}_\mathcal{D}} b_j \gamma_j + \sum_{j \in \mathcal{V}_0} Q_j \hat{y}_j (\pi_j - w_j). \tag{5.7}
$$

Framework of the C&CG Algorithm

We now describe the framework of the C&CG algorithm and present the formulation of the MP, which will be solved iteratively. At each iteration r, we identify a worst-case scenario $\hat{\mathbf{z}}^r$ by solving the linearized subproblem. Then we create the recourse variables $(\mathbf{x}^r, \mathbf{u}^r)$ and the corresponding constraints, as well as this specific scenario, and add them to the MP. Let LB and UB be the lower and upper bounds, $Gap = (UB - LB)/UB$, and let the optimality tolerance be ϵ. The C&CG algorithm is as follows:

(1) Set LB = -∞, UB = +∞, and $r = 0$.

(2) Take any arbitrary $\hat{\mathbf{y}} \in \{0,1\}^{|\mathcal{V}_0|}$ as initial solution.

(3) Solve the linearized subproblem with regards to $\hat{\mathbf{y}}$ to identify the worst-case scenario $\hat{\mathbf{z}}$. Update $r = r + 1$. Create recourse variables $(\mathbf{x}^r, \mathbf{u}^r)$ and corresponding constraints, and add them to the following MP.

$$
\begin{aligned}
&\min \sum_{j\in\mathcal{V}_0} f_j y_j + \phi \\
&\text{s.t. } \phi \geq \sum_{(i,j)\in\mathcal{A}} c_{ij} x_{ij}^l + \sum_{j\in\mathcal{V}_\mathcal{D}} \theta_j u_j^l && \forall l \in \{1,\dots,r\}, \\
&\quad \sum_{i\in\mathcal{V}_j^+} x_{ji}^l \leq b_j && \forall j \in \mathcal{V}_\mathcal{S}, l \in \{1,\dots,r\}, \\
&\quad \sum_{i\in\mathcal{V}_j^+} x_{ji}^l = \sum_{i\in V_j^-} x_{ij}^l && \forall j \in \mathcal{V}_\mathcal{T}, l \in \{1,\dots,r\}, \\
&\quad \sum_{i\in\mathcal{V}_j^-} x_{ij}^l + u_j^l = -b_j && \forall j \in \mathcal{V}_\mathcal{D}, l \in \{1,\dots,r\}, \\
&\quad \sum_{i\in\mathcal{V}_j^+} x_{ji}^l \leq (1 - z_j^l) Q_j y_j && \forall j \in \mathcal{V}_0, l \in \{1,\dots,r\}, \\
&\quad y_j \in \{0,1\} && \forall j \in \mathcal{V}_0, \\
&\quad x_{ij}^l \geq 0 && \forall (i,j) \in \mathcal{A}, l \in \{1,\dots,r\}, \\
&\quad u_j^l \geq 0 && \forall j \in \mathcal{V}_\mathcal{D}, l \in \{1,\dots,r\}.
\end{aligned}
$$

(4) Iterate until the algorithm terminates:

(i) Solve the MP to find an optimal solution $(\hat{\mathbf{y}}, \phi)$; set LB to the optimal value of the MP.

(ii) Solve the subproblem with regards to $\hat{\mathbf{y}}$ to identify the worst-case scenario $\hat{\mathbf{z}}$. Update $UB = \min\left\{UB, \sum_{j\in\mathcal{V}_0} f_j y_j^r + \chi^r\right\}$.

(iii) If $Gap \leq \epsilon$, terminate; otherwise, update $r = r + 1$ and create the recourse variables

and corresponding constraints. Add them to the MP and go to step (i).

We also implement a Benders decomposition algorithm, where a single cutting plane

$$\phi \geq \sum_{j \in \mathcal{V}_S} b_j \alpha_j^r - \sum_{j \in \mathcal{V}_D} b_j \gamma_j^r + \sum_{j \in \mathcal{V}_0} (1 - z_j^r) Q_j y_j \pi_j^r \tag{5.8}$$

is iteratively added to the MP, which carries only the first-stage decision variable $\mathbf{y}$.

Remarks: (1) For the RO-LNDP$_1$ model, the objective function of the MP becomes

$$\min \sum_{j \in \mathcal{V}_0} f_j y_j + \sum_{k=1}^{K} \rho_k \phi_k \tag{5.9}$$

where ϕ_k corresponds to the kth uncertainty set. For each uncertainty set, we need to solve a subproblem to identify the worst-case scenario, and we add its recourse variables and corresponding constraints to the MP.

(2) For the RO-LNDP$_2$ model, at each iteration we solve a subproblem for each uncertainty set, to check whether its worst-case performance violates the bound. Once there exists such scenarios, we add all the identified worst-case scenarios to the MP until all the performance bounds are respected.

5.3.2 Algorithm Enhancement

In this section we introduce a *variable fixing technique* to improve the algorithm's performance. This technique has been shown to be efficient in reducing the computational burden for facility location problems [4,137,138]. We generalize it to solve the reliable LNDP. The idea is as follows:

Let $\hat{\mathbf{y}}$ be the incumbent solution with corresponding upper bound UB'.

- If $\hat{y}_j = 0$: We add an additional constraint $y_j = 1$ to the MP and solve it to optimality; if its optimal value is larger than UB', then we fix y_j to 0 in the MP.
- If $\hat{y}_j = 1$: We add an additional constraint $y_j = 0$ to the MP and solve it to optimality; if its optimal value is larger than UB', then we fix y_j to 1 in the MP.

Fixing some y_j reduces the feasible space of the MP, which can help improve computational efficiency. On the other hand, each time a new constraint with $y_j = 1$ or 0 is added, and this might increase the computational time, especially when the number of facilities is large. Therefore, (1) after adding the new constraint, we solve the corresponding linear relaxation

and get a lower bound, and we compare this bound with UB' ("C&CG LP"); (2) we solve the MP with the new constraint to optimality and compare the optimal value with UB' ("C&CG Optimal"). The variable fixing technique is performed each time after solving the subproblem. When this operation is finished, we add the identified scenario (via solving the SP) and its associated variables and constraints to the MP.

5.4 Numerical Experiments and Analyses

In this section, we present the instances, discuss our numerical tests, analyze the influence of the parameters, and present some insights. The algorithm is coded in the C# programming language and run on a PC with a 2.53 GHz Intel Core Dual Processor and 3 GB of memory. The MP and subproblem are solved using Gurobi 6.0.

5.4.1 Instances

We randomly generate instances of different sizes. The method is based on that of [64], with a few modifications. The instances are labeled "$d - |\mathcal{V}_\mathcal{S}| - |\mathcal{V}_\mathcal{T}| - |\mathcal{V}_\mathcal{D}|$," where d is the edge density (20%, 30%, or 50%) and $|\mathcal{V}_\mathcal{S}|$, $|\mathcal{V}_\mathcal{T}|$, and $|\mathcal{V}_\mathcal{D}|$ are the number of supply, transshipment, and demand nodes. The number of nodes ranges from 60 to 100.

For each demand node $j \in \mathcal{V}_\mathcal{D}$, the unmet-demand penalty is 1500, and the demand b_j is drawn uniformly from $[-110, -50]$. Let $S_b = -\sum_{j\in\mathcal{V}_\mathcal{D}} b_j$ be the sum of all the demands (the negative sign appears because $b_j \leq 0$); define $\bar{s} = \frac{S_b}{|\mathcal{V}_\mathcal{S}|}$ and $\bar{c} = \frac{S_b}{|\mathcal{V}_\mathcal{T}|}$.

For each facility node $j \in \mathcal{V}_0$, the fixed cost is drawn uniformly from $[5000, 15000]$; if $j \in \mathcal{V}_\mathcal{S}$, then its capacity Q_j is drawn uniformly from $[1.5\bar{s}, 2.5\bar{s}]$ and its supply b_j is the same as its capacity; if $j \in \mathcal{V}_\mathcal{T}$, then its capacity Q_j is drawn uniformly from $[1.5\bar{c}, 2.5\bar{c}]$ and its supply b_j is 0.

The arcs are constructed based on the probability specified by the edge density. In detail, for two nodes i, j ($i \in \mathcal{V}_\mathcal{S}, j \in \mathcal{V}_\mathcal{T}$ or $i \in \mathcal{V}_\mathcal{S}, j \in \mathcal{V}_\mathcal{D}$ or $i \in \mathcal{V}_\mathcal{T}, j \in \mathcal{V}_\mathcal{D}$) and edge density d, we generate a random number $r \in [0, 1]$. If $r \leq d$, then we construct an arc between i and j. The unit transportation cost of each arc $(i, j) \in \mathcal{A}$ is drawn uniformly from $[1, 500]$.

5.4.2 Algorithm Performance

We now evaluate the performance of the algorithms. We set the maximal number of facilities that can fail simultaneously to 5, i.e., $m = 5$ (we arbitrarily choose a large value for m to test the efficiency of the C&CG algorithm). The facilities are destroyed completely when

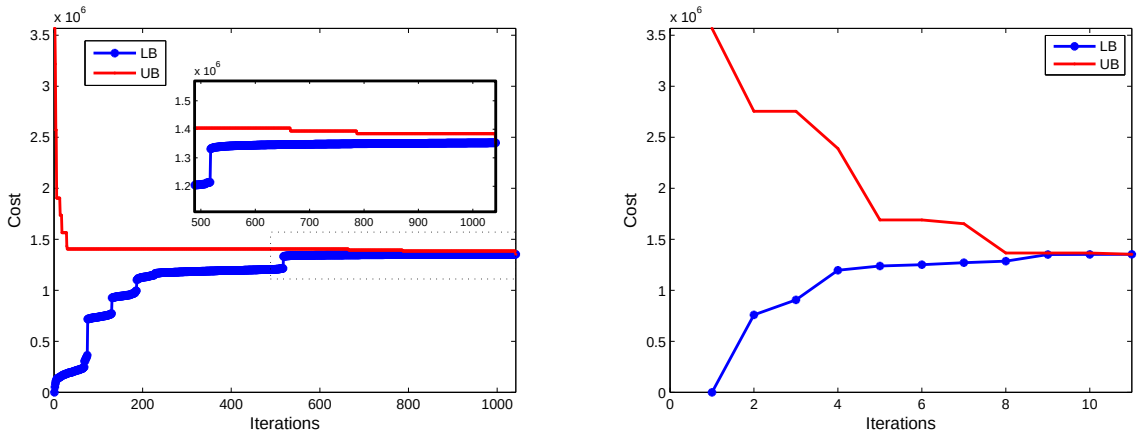
disruptions occur (i.e., $\delta_j = 1, \forall j \in \mathcal{V}_0$). The model is RO-LNDP$_0$, and the results are shown in Table 5.2. The optimality tolerance ϵ is set to 10^{-6} and the time limit is 10,800 seconds (when this limit is reached, the current iteration will be completed). In Table 5.2, the results in the "C&CG No Fix" columns are obtained by using the C&CG algorithm without enhancement. The column Iter gives the number of iterations; Time indicates the computational time in seconds; and Gap is the relative percentage gap between the upper and lower bounds.

Table 5.2 Performance of different algorithms

	Benders				C&CG LP				C&CG No Fix		C&CG Optimal	
Instance	Iter	Time	Gap	UB	Iter	Time	Gap	UB	Time	Gap	Time	Gap
20%-10-20-30	1042	696.84	0.00	1352471	11	5.66	0.00	1352471	5.19	0.00	72.71	0.00
30%-10-20-30	3710	5126.89	0.00	1532355	6	0.94	0.00	1532355	0.71	0.00	11.32	0.00
50%-10-20-30	919	10831.45	78.59	767158	9	5.37	0.00	751133	4.75	0.00	48.14	0.00
50%-**20**-20-30	553	10836.64	74.65	663328	41	4005.28	0.00	627139	9662.08	0.00	10839.47	4.91
50%-**30**-20-30	387	10851.49	65.83	577131	25	6930.84	0.00	532691	12303.52	4.31	12794.40	25.40
50%-**40**-20-30	391	10812.02	57.25	502841	22	8177.58	0.00	421706	11763.53	6.24	18315.67	13.03
30%-10-**30**-30	809	10813.28	35.98	1591183	9	2.86	0.00	1531845	2.76	0.00	46.52	0.00
30%-10-**40**-30	227	10914.11	78.85	1123941	7	4.24	0.00	1095014	3.77	0.00	78.95	0.00
30%-10-**50**-30	105	10910.24	77.63	1317280	8	3.61	0.00	1249269	3.41	0.00	100.41	0.00
20%-10-20-**40**	3504	10804.99	13.06	1960874	19	15.09	0.00	1923711	24.96	0.00	211.40	0.00
20%-10-20-**50**	1546	10808.08	17.94	2265886	14	14.68	0.00	2249769	20.85	0.00	120.54	0.00
20%-10-20-**60**	642	494.06	0.00	2698076	17	16.04	0.00	2698076	19.71	0.00	219.11	0.00
50%-20-30-50	592	10814.97	81.88	951850	32	7440.17	0.00	709762	11930.12	2.75	11707.76	6.08
Average	1110	8824.23	44.74	1331106	17	2047.87	0.00	1282688	3518.87	1.02	4197.42	3.80

Table 5.2 shows that for most instances (9 out of 13), the C&CG algorithm is hundreds of times faster than the BD, and the number of iterations is much lower. Within the time limit, BD can find optimal solutions for just 3 instances. In contrast, the C&CG algorithm is able to generate optimal solutions for most of the instances, and C&CG LP finds the optimal solution for all the instances. Therefore, the C&CG algorithm outperforms BD in both computational time and solution quality.

To further demonstrate the superiority of C&CG LP, Figure 5.2 shows the convergence curves of the two algorithms for the instance 20%-10-20-30. In Figure 5.2(a), the gap between the lower and upper bounds reduces slowly and does not reach zero even after 1000 iterations; the C&CG LP algorithm finds the optimal solution after 11 iterations. The instances 50%-20-20-30, 50%-30-20-30, and 50%-40-20-30 show that the number of supply nodes has a significant impact on the computational efficiency. For these instances, C&CG LP finds optimal solutions within the time limit and performs better than C&CG No Fix and C&CG Optimal. Therefore, for our model comparison and parameter sensitivity analysis, we use the C&CG LP. On the other hand, we note that the results in Table 5.2 are based on a vanilla implementation of the BD and that improvement might be achieved using Benders Branch & Cut approaches.



(a) Benders decomposition

(b) C&CG LP

Figure 5.2 Convergence curves for instance 20%-10-20-30

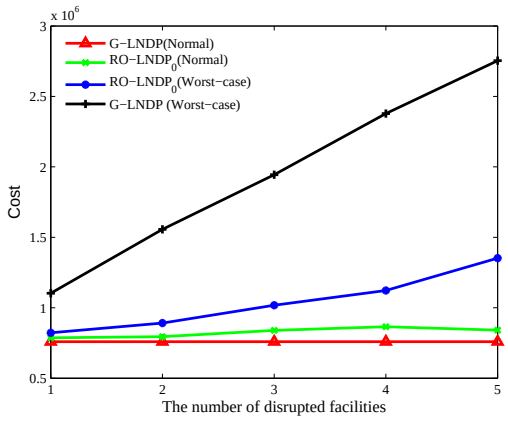
5.4.3 Impact of Reliability

Obviously, if disruptions are not considered (i.e., G-LNDP), the system's normal cost will be lower, but it may be more expensive to implement mitigation/recourse operations when they become necessary. To investigate the impact of reliability on the system cost, we conduct the following experiments:

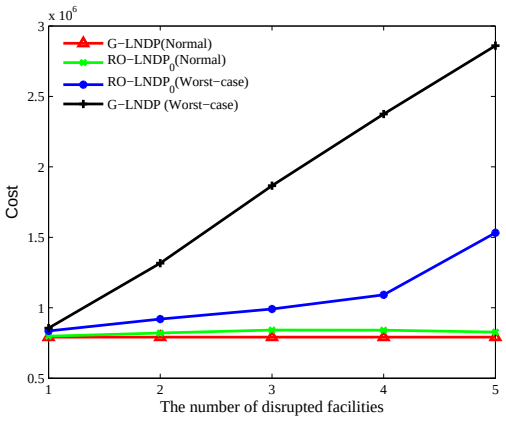
(1) We solve RO-LNDP$_0$ to find the location decision. We then fix this decision and solve a MCFP to find the system's normal cost under RO-LNDP$_0$. This indicates the impact of disruptions on the system's normal cost.

(2) We solve G-LNDP and fix the location decision. We then solve the slave problem of the C&CG algorithm to identify the worst-case cost. This indicates the cost of not considering disruptions in advance and handling them as they occur.

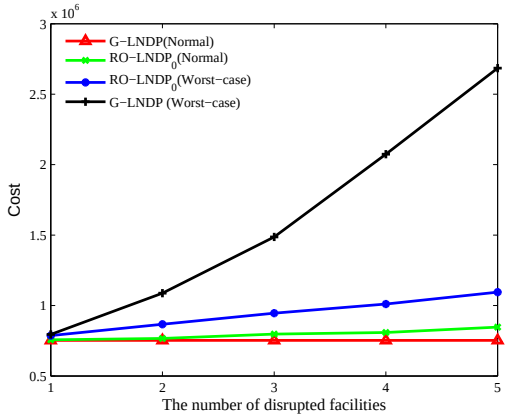
For space considerations, Figure 5.3 shows the curves of just 6 instances. The results for the other instances are similar. Figure 5.3 confirms that considering disruptions will increase the normal cost. However, ignoring disruptions during the design phase leads to higher costs when they do occur. As m increases, the deviation of the worst-case cost for RO-LNDP$_0$ and G-LNDP also increases. However, the normal cost of these two models is similar. We conclude that the two-stage RO model gives a considerable decrease of the recourse cost in the worst disruptive situation with only a small increase in the normal cost.



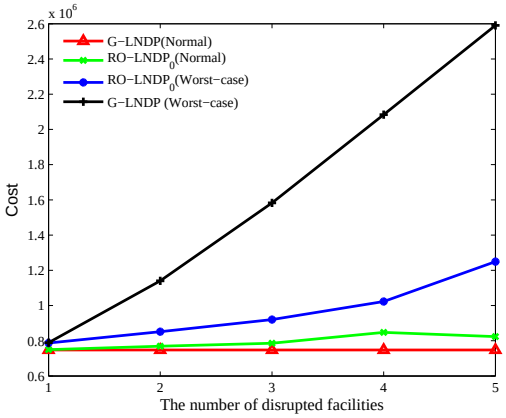
(a) Instance 20%-10-20-30



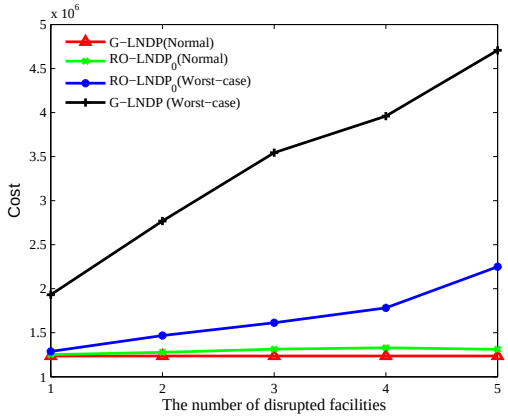
(b) Instance 30%-10-20-30



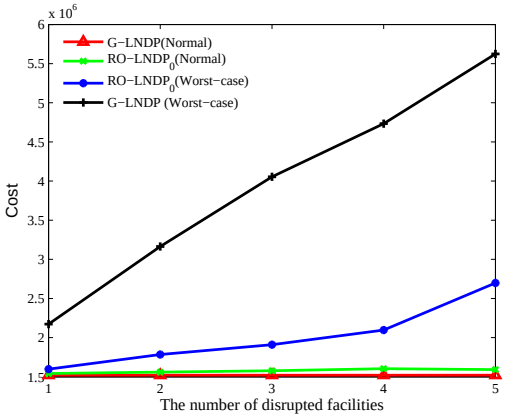
(c) Instance 30%-10-40-30



(d) Instance 30%-10-50-30



(e) Instance 20%-10-20-50



(f) Instance 20%-10-20-60

Figure 5.3 Impact of reliability

5.4.4 Model Comparison and Analyses

In this section, we compare the results of different models.

Basic RO Model and Expanded RO Model

The experimental setup for each model is as follows. We have one uncertainty set with $m = 2$ for the basic two-stage RO model. For the expanded two-stage RO model, we have three uncertainty sets A_0, A_1, and A_2. Set A_0 is the normal disruption-free situation. In sets A_1 and A_2, *exactly* 1 and 2 facilities will fail simultaneously, respectively. For the weight coefficients, we consider two cases: (1) $\rho_0 = 0.8$, $\rho_1 = 0.15$, and $\rho_2 = 0.05$, where the decision-maker believes that in most cases the system will not experience disruption; (2) $\rho_0 = 0.65$, $\rho_1 = 0.25$, and $\rho_2 = 0.10$, where disruption is more likely.

In the following tables, the column Iter gives the number of iterations; FacN. gives the number of opened facilities in the optimal solution; and Time gives the computational time in seconds.

Table 5.3 Comparison of basic two-stage RO model and expanded two-stage RO model

Instance	Basic two-stage RO model				Expanded two-stage RO model							
					$\rho_0 = 0.8$, $\rho_1 = 0.15$, $\rho_2 = 0.05$				$\rho_0 = 0.65$, $\rho_1 = 0.25$, $\rho_2 = 0.10$			
	Iter	Cost	FacN.	Time	Iter	Cost	FacN.	Time	Iter	Cost	FacN.	Time
20%-10-20-30	15	891019	13	2.55	8	798609	10	21.33	11	813431	13	41.22
30%-10-20-30	5	919641	11	0.53	7	816046	9	10.12	7	827148	9	10.23
50%-10-20-30	6	588292	15	2.71	3	471472	15	1.23	3	483600	15	1.13
30%-10-30-30	6	901721	11	1.83	4	815471	8	4.12	4	827332	9	3.82
30%-10-40-30	6	867244	9	2.28	5	773600	8	8.53	4	783327	9	4.35
30%-10-50-30	6	851158	11	3.64	4	767809	8	9.38	4	778780	9	8.34
20%-10-20-40	8	1323446	14	4.70	6	1072862	10	5.75	6	1104976	13	6.41
20%-10-20-50	5	1467915	14	1.36	6	1280748	13	13.83	5	1295309	13	9.65
20%-10-20-60	8	1783269	14	6.25	13	1572904	13	156.05	9	1592708	13	65.98
Average	7.2	1065967	12.4	2.87	6.2	929947	10.4	25.59	5.9	945179	11.4	16.79

Table 5.3 shows that the solution under the basic RO model is more conservative than that of the expanded RO model. To hedge against the worst disruptive scenario in the uncertainty set, the basic RO model tends to open more facilities, increasing the cost. In the expanded RO model, where more weight is put on the disruptive scenarios, the cost increases as expected; however, it is still lower than that of the basic RO model. The computational time is slightly higher for the expanded RO model than the basic RO model. However, with CC&G LP, the expanded model can still be solved to optimality in a short time.

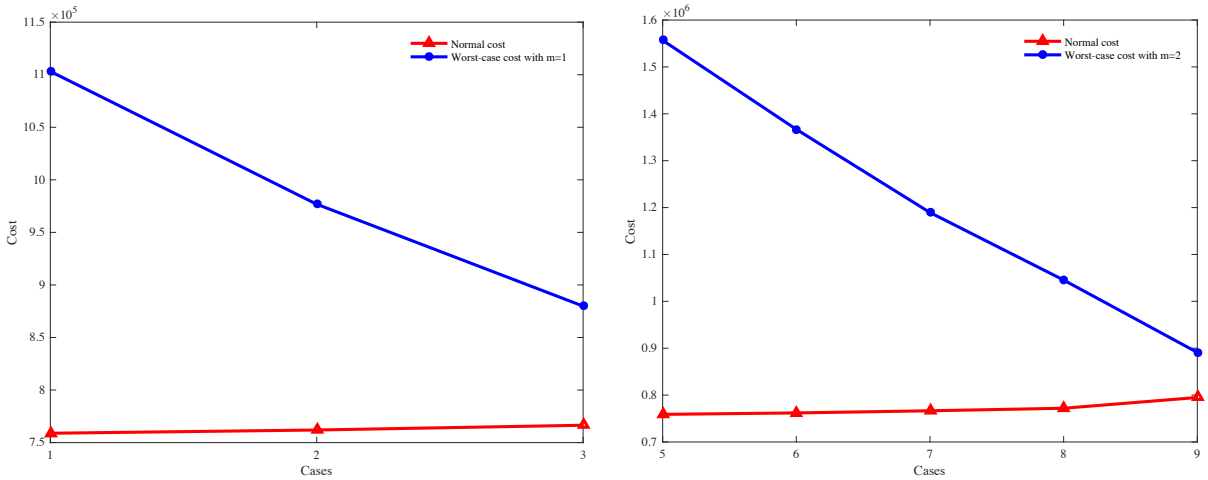
Risk-constrained Robust Model and Deterministic LNDP model

For the risk-constrained RO model, we use two uncertainty sets, i.e., A_k ($k = 1, 2$), and for set A_k at most k facilities will fail simultaneously. We change the performance bound on each uncertainty set and investigate the results. The detailed experimental setup is as follows: for Cases 1–4 (see Table 5.4), the performance bound on A_2 is always satisfied, and we change the bound on A_1 gradually; after we have obtained the location decision, we assume that one facility will be disrupted and compute the worst-case cost. Similarly, for Cases 5–10, the performance bound on A_1 is always satisfied, and we change the bound on A_2 and compute the normal and worst-case costs. The numerical analysis is conducted on instance 20%-10-20-30, and Table 5.4 and Figure 5.4 give the results.

Table 5.4 Results for risk-constrained RO model for instance 20%-10-20-30

Case	ξ_1	ξ_2	Normal cost	FacN.	Worst-case cost	Time
1	**1100000**	1600000	758897	7	1103177	0.20
2	**1000000**	1600000	761983	8	976794	0.39
3	**800000**	1600000	766515	9	879850	0.78
4	**700000**	1600000	Infeasible[a]	N/A[b]	N/A	0.21
5	1100000	**1600000**	758897	7	1556769	0.36
6	1100000	**1400000**	761983	8	1366854	0.55
7	1100000	**1200000**	766515	9	1189425	1.26
8	1100000	**1000000**	771935	10	1045396	2.15
9	1100000	**800000**	794881	13	891019	3.24
10	1100000	**600000**	Infeasible	N/A	N/A	0.22

[a] The constraint is too tight and no feasible solution is found.
[b] Not applicable.



(a) Cases 1–3 (b) Cases 5–9

Figure 5.4 Results of risk-constrained two-stage RO model

Table 5.4 shows that when the bounds on the worst-case performance are loose (i.e., Cases 1

and 5), the problem reduces to a deterministic LNDP. As the bounds become more restrictive, solutions with higher normal costs and lower worst-case costs are obtained, and more facilities are opened to improve reliability. In Cases 4 and 10, the model becomes infeasible, which suggests that if greater reliability is required, the system needs to obtain extra facilities or increase the capacity of some facilities. Therefore, the risk-constrained two-stage RO model can be employed as a decision support tool for system expansion with reliability considerations.

Figure 5.4 indicates that a slight increase in the normal cost can lead to a significant decrease in the worst-case cost. In particular, from Case 1 to 3, the normal cost increases by only 1.00%; however, the worst-case cost reduces by 20.24%. From Case 5 to 9, the worst-case cost reduces by 42.76% with only a 4.74% increase in the normal cost. We conclude that compared with G-LNDP, the risk-constrained RO model is capable of improving reliability substantially with only a slight increase in the normal cost.

Summary of Model Comparison

Based on the numerical experiments in this section, we draw the following conclusions: (1) The proposed CC&G algorithm is able to solve the three two-stage RO models to optimality in a reasonable time. (2) The two-stage RO models can improve system reliability with only a slight increase in the normal cost. Thus, all the three robust models can be applied to situations where the decision-makers want to design a reliable supply chain network but without precise probability information about risks. (3) We can use the risk-constrained model when we care more about the system's normal cost while still want to control the worst-case cost to some extent.

5.4.5 Parameter Analysis

This section analyzes the effects of parameters.

Budget of Uncertainty and Partial Disruption

As mentioned earlier, the scope of the uncertainty set and partial disruptions will affect the system design and operation. However, to what extent they will affect the cost remains unknown. To investigate this, we explore changing the value of m and δ simultaneously for instance 20%-10-20-30 and the RO-LNDP$_0$ model. Figure 5.5 presents the results.

It can be seen that with both partial and complete disruption, the worst-case cost increases in general as the budget of the uncertainty set increases. For all values of δ, as m varies from 10

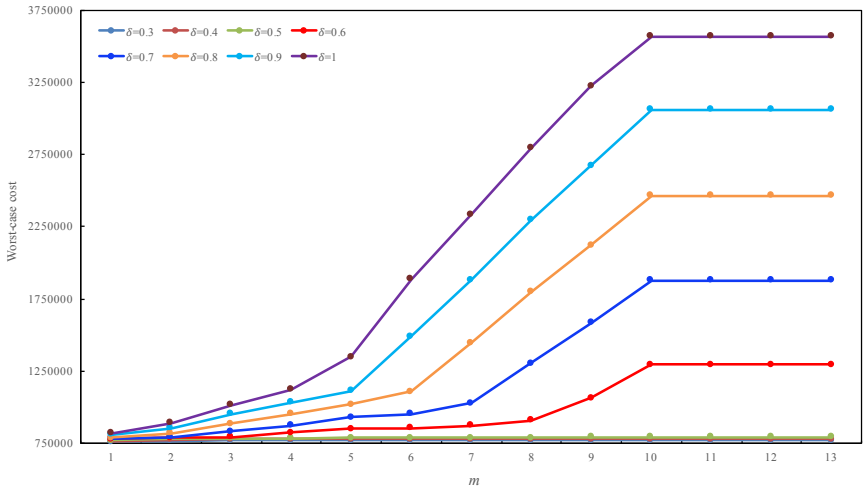


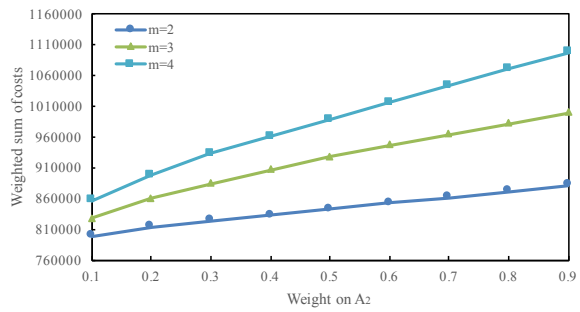
Figure 5.5 Impact of m and δ on cost for instance 20%-10-20-30

to 13, the cost remains stable. We explore the details of the solutions and find that in these cases the system does not open any facilities and all the demands are penalized. We also find that when partial disruption is considered, the cost for different values of m may be only slightly different. In particular, when δ is between 0.3 and 0.5, the variation is small although the budget of the uncertainty set increases gradually. This indicates that for regions where relatively minor disasters are likely, decision-makers can consider more disruptive scenarios with little increase in the worst-case cost. On the other hand, when δ is larger than 0.5, the system is much more sensitive to the budget of the uncertainty sets.

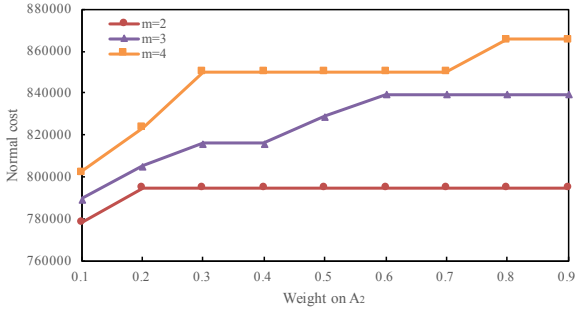
Weight of the Uncertainty Set

For the expanded RO model, the weights put on the uncertainty sets characterize decision-makers' protective level, which may influence the location decision. We conduct our analyses on four instances, where the number of facilities ranges from 30 to 60. For each instance, we consider two uncertainty sets A_1 and A_2: A_1 with $m_1 = 0$ (i.e., disruption-free case) and A_2 with $m_2 = 2$ or 3 or 4. We change the weight gradually and observe its influence. Results are presented in Figure 5.6, where the left side is the objective value of the RO-LNDP$_1$ model. The right side is the normal cost of the system, where we fix the location decision and solve a MCFP.

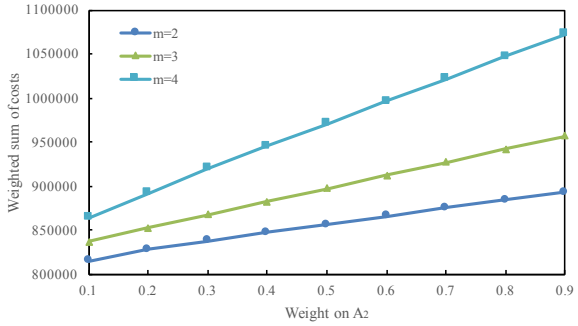
It shows that the objective value of the RO-LNDP$_1$ model increases almost linearly with ρ_2, which suggests that the worst-case performance of set A_2 accounts for a large portion of the objective value. When $m_2 = 2$, the normal cost is less sensitive to the value of ρ_2, especially for instance 30%-10-40-30. Normally there exists some regions where the normal



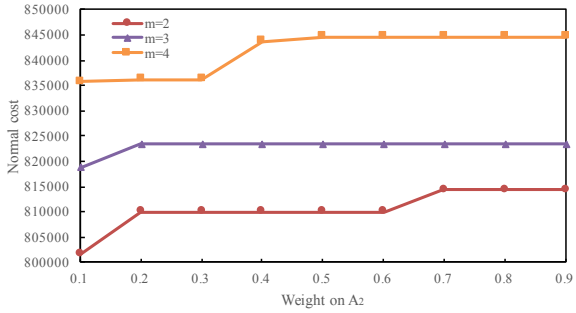
(a) Instance 20%-10-20-30



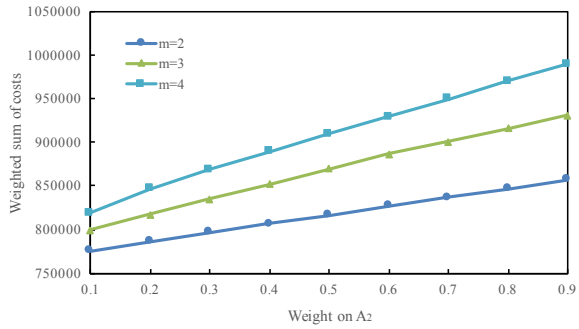
(b) Instance 20%-10-20-30



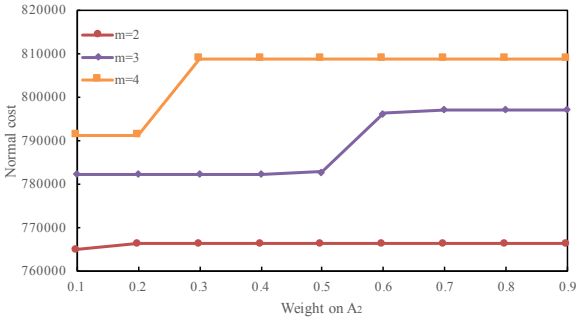
(c) Instance 30%-10-30-30



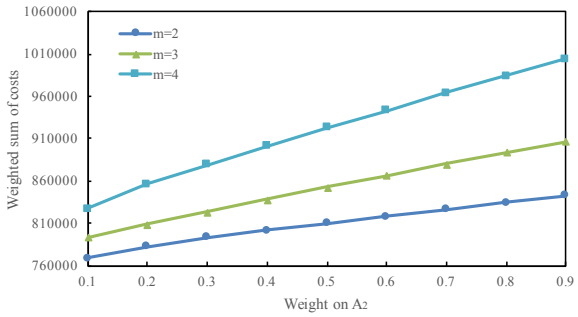
(d) Instance 30%-10-30-30



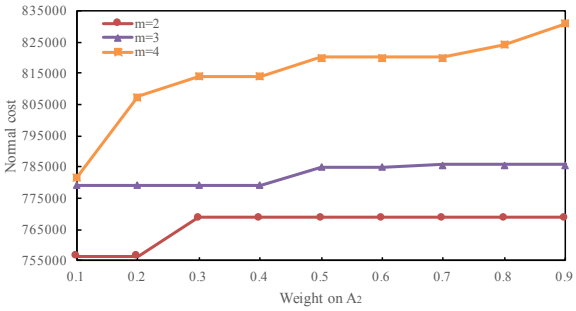
(e) Instance 30%-10-40-30



(f) Instance 30%-10-40-30



(g) Instance 30%-10-50-30



(h) Instance 30%-10-50-30

Figure 5.6 Impact of weight on the uncertainty set

cost keeps stable for each budget m_2. In these situations, the location decisions are the same. Therefore, it is possible that sometimes the estimating errors in the weight will not significantly influence the system's configuration. However, for the decision-makers, they should carefully determine the weight of larger uncertainty sets.

5.5 Conclusions

In this chapter, we consider a reliable LNDP, where both supply and transshipment nodes are subject to disruptions. Three two-stage RO models are constructed, which are solved to optimality by the C&CG algorithm. Our numerical tests show that (i) the C&CG algorithm, especially the C&CG LP method, outperforms BD in both solution quality and computational time; (ii) the two-stage RO models give a considerable decrease in the cost of the worst disruptive scenario for only a small increase in the normal cost; (iii) when partial disruptions are considered, sometimes the system experiences small increases in the worst-case cost even when the uncertainty budget increases dramatically.

CHAPTER 6 DRONE ROUTING WITH ENERGY FUNCTION: FORMULATION AND EXACT ALGORITHM

This chapter is based on the following article.

- Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2020. Drone Routing with Energy Function: Formulation and Exact Algorithm. To appear at *Transportation Research Part B: Methodological*.

Before starting this chapter, we first clarify the definition of drone routing problem (DRP) and multi-trip drone routing problem (MTDRP). DRP refers to the problem where a fleet of drones visit a set of customer locations and each drone can visit multiple customers in a trip. In this case, drones can only be dispatched once from the depot. When drones can perform multiple trips (each trip starts and ends at the depot), this problem is referred to as the MTDRP.

6.1 Introduction

Based on the reviewed papers in Chapter 2, we find that only a few papers explicitly consider energy constraints, and many use an approximation that is linear in the payload. In addition, to the best of our knowledge, no benchmark instance is available for algorithm evaluation, and no efficient exact algorithm has been developed for the DRP. To fill some gaps in this area, this chapter solves a MTDRP with time windows, where a fleet of homogeneous multirotor drones are dispatched to deliver packages to customers within stipulated time slots. The main contributions of our study is as follows:

1. We explicitly represent drone's energy consumption as a nonlinear function of payload and travel time, instead of assuming that flight range (maximum distance or time) is a fixed number. To tackle the nonlinear energy function, instead of relying on a linear approximation (e.g., as in reference [2]), we propose adding two types of cuts in the solution process. Our results show that using a linear energy approximation can lead to routes that are energy infeasible under the nonlinear energy consumption model.
2. A 2-index formulation scheme is presented, which is solved by a B&C algorithm. To the best of our knowledge, this work is the first to formulate a MTDRP and use an exact algorithm for drone routing problems.

3. We generate several benchmark instance sets based on the realistic parameters and known instance sets in the literature, which will be available to the research community and allow for a better comparison of algorithms.

4. We provide extensive computational results of the formulation and the algorithm.

The rest of this chapter is organized as follows. Section 6.2 describes our problem, presents the mathematical model, and introduces valid inequalities to strengthen it. Section 6.3 presents techniques for the calculation of energy consumption and provides details of our exact algorithm. Numerical tests and analyses are presented in Section 6.4. This is followed by the conclusions in Section 6.5.

6.2 Formulation

This section presents the problem, constructs the mathematical model, and introduces valid inequalities to strengthen the model.

6.2.1 Problem Definition

The problem is defined on a directed graph $G = (N, A)$, where $N = \{0, \ldots, n+1\}$ is the set of nodes. Node 0 represents the starting depot, and node $n+1$ is a copy of node 0 and it represents the returning depot. $N' = \{1, \ldots, n\}$ is the set of customers. For notational convenience, we denote $N^+ = \{0, \ldots, n\}$ and $N^- = \{1, \ldots, n+1\}$. $A = \{(i,j) : i \in \{0\}, j \in N' \text{ and } i \in N', j \in N^-, i \neq j\}$ is the set of arcs. Sets $\delta^-(i)$ and $\delta^+(i)$ represent node i's predecessor and successor nodes, respectively.

Each customer is associated with a non-negative demand d_i, and a hard time window $[a_i, b_i]$. For the depots, $[a_0, b_0] = [a_{n+1}, b_{n+1}]$, where a_0 and b_0 are the earliest possible departure time and the latest possible arrival time, respectively. A fleet of K homogeneous multirotor drones are based at the depot. Q is the maximum payload of a drone and we assume that $d_i \leq Q, \forall i \in N'$. Each drone can perform several trips and during a trip it can visit several customers. Drone speed is assumed to be a constant number, and with each arc (i,j) is associated a travel time t_{ij} and a travel cost c_{ij}. Further, it is assumed that the triangle inequality is satisfied for t_{ij}. Without loss of generality, here we assume the service time at each customer is 0, because we can set t_{ij} to be the sum of travel time on arc (i,j) and the service time at node i. We consider multirotor drones in the study as these have been often used in drone delivery analyses and we use data from [2]. Hybrid drones may have different performance characteristics and require a different energy model.

The problem consists in designing a set of drone routes, such that the objective function is optimized and the following constraints are satisfied: (1) Each route starts at depot 0 and ends at depot $n+1$. (2) Every customer is visited exactly once. (3) The sum of duration of trips assigned to the same drone does not exceed b_{n+1}. (4) The drone weight capacity constraint, battery energy constraint, and customers' time windows must be respected.

6.2.2 Mathematical Model

For our problem, as there is no limit on the number of trips that each drone can perform, we do not consider the 3-index formulation with a *trip* index. Further, our preliminary tests also indicate that the 3-index formulation with a *drone* index provides worse results than the 2-index formulation. Therefore, we present a 2-index formulation for our MTDRP.

Decision Variables. There are two sets of binary variables: $x_{ij} = 1$ if arc (i, j) is traversed by a drone, 0 otherwise. $z_{ij} = 1$ if a trip finishing with customer i is followed by another trip visiting j as the first customer (performed by the same drone), 0 otherwise. There are four sets of continuous variables: q_{ij} is the product weight carried through arc (i, j) (kg). τ_i is the start of service time at node $i \in N^-$ $(second)$. f_i is the accumulated energy consumption of a drone upon arrival at node i (kWh). e_{ij} is the energy consumption on arc (i, j) (kWh).

Constraints. We organize the constraints into five groups:

(i) *Route feasibility:*

$$\sum_{j \in \delta^+(i)} x_{ij} = 1 \qquad \forall i \in N', \tag{6.1}$$

$$\sum_{j \in \delta^-(i)} x_{ji} = 1 \qquad \forall i \in N', \tag{6.2}$$

$$\sum_{j \in \delta^+(0)} x_{0j} = \sum_{j \in \delta^-(n+1)} x_{j,n+1}. \tag{6.3}$$

Constraints (6.1) and (6.2) guarantee that each customer is visited exactly once. Constraints (6.3) indicate that the number of trips leaving the starting depot is equal to the number arriving at the ending depot.

(ii) *Weight related constraints:*

$$\sum_{i \in \delta^-(j)} q_{ij} - \sum_{i \in \delta^+(j)} q_{ji} = d_j \qquad \forall j \in N', \tag{6.4}$$

$$q_{ij} \leq Q x_{ij} \qquad \forall (i, j) \in A, \tag{6.5}$$

$$q_{i,n+1} = 0 \qquad \forall i \in N'. \tag{6.6}$$

Equations (6.4) impose that each customer's demand must be satisfied, and also eliminate subtours. Constraints (6.5) guarantee that drone weight capacity is respected. Equations (6.6) indicate that drones cannot carry any product from a customer to the ending depot.

(iii) *Drone energy constraints:*

We only consider drones' energy consumption during level flight in this study. The authors of [2] suggest that the average power during hover is an upper bound on the average power during flight. Since there are no available field tests of small drones making multiple deliveries or of actual delivery drones in *production mode*, in this study, we use the theoretical power consumption during hovering to approximate the horizontal power consumption for a delivery drone making multiple-stop trips. The work in [139] describes the energy consumption, $P(q)$, of a single rotor helicopter in hover as a convex function of payload q. Based on the assumption that each rotor shares the total weight of a drone equally, the authors of [2] derive the power consumption equation for a h-rotor drone as

$$P(q_{ij}) = (W + m + q_{ij})^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}}, \tag{6.7}$$

where W is the frame weight (kg), m is the battery weight (kg), q_{ij} is the payload (kg), g is the force due to gravity (N), ρ is the fluid density of air (kg/m^3), ς is the area of spinning blade disc (m^2), h is the number of rotors, and the unit of P is *Watt*. In the experiments of [74], the power consumption in hover also takes a similar form, i.e., $P(q_{ij}) = c_p[(W + m + q_{ij})g]^{\frac{3}{2}}$, where c_p is a parameter. We rewrite Equation (6.7) as

$$P(q_{ij}) = k(W + m + q_{ij})^{\frac{3}{2}}, \tag{6.8}$$

where k depends on the details of the drone and the environmental parameters and it is a constant in our model. Based on field tests, the authors of [2] propose to approximate power consumption as

$$P(q_{ij}) = \alpha(m + q_{ij}) + \beta, \tag{6.9}$$

where $\alpha(kW/kg)$ and $\beta(kW)$ are two constant numbers obtained by a linear approximation. As shown in Figure 6.1, when the sum of the battery weight and payload is smaller than A, the linear approximation function overestimates the energy consumption from the nonlinear model, and therefore drone routes calculated with the linear approximation will be "energy feasible" if the nonlinear model is used to calculate energy consumption. However, when the battery and payload weight is larger than A, then the linear approximation function underestimates the energy consumption from the nonlinear model. In this case, drone routes

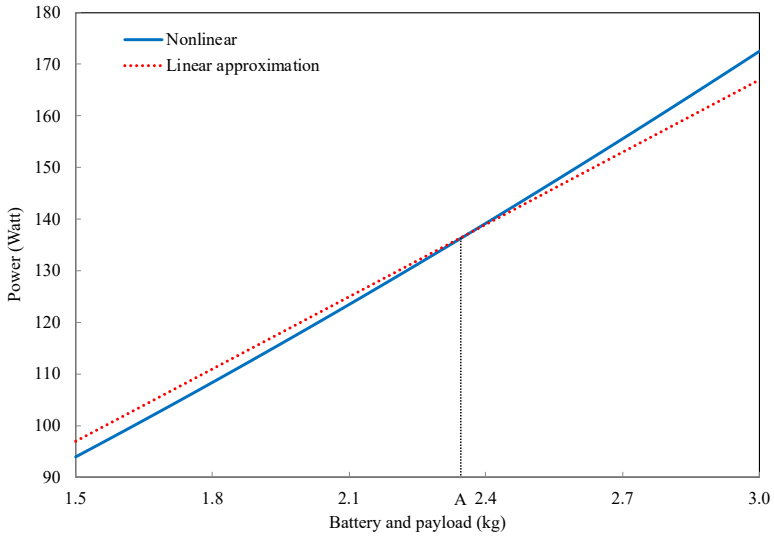


Figure 6.1 Energy calculation from linear and nonlinear functions (Figure 1 in reference [2])

calculated with the linear approximation may be "energy infeasible" (i.e., exceed the battery's energy capacity) if the nonlinear model is used to calculate energy consumption. We use Equation (6.8) to compute power consumption in this study, and drones' energy consumption constraints are written as

$$f_0 = 0, \tag{6.10}$$

$$f_i + k'(W + m + q_{ij})^{\frac{3}{2}} t_{ij} \leq M_{ij}(1 - x_{ij}) + f_j \quad \forall (i,j) \in A, \tag{6.11}$$

$$f_{n+1} \leq \sigma. \tag{6.12}$$

Equations (6.10) indicate that at the beginning of each trip the accumulated energy consumption is 0, that is, every time a drone begins a new trip we swap it with a fully charged battery. This assumption is common in the literature [24,30,83]. Equations (6.11) establish the energy relationship between node i and its immediate successor j, where k' is a constant that includes k from earlier and the conversion from *Watt-second* to kWh and M_{ij} is an arbitrary large constant. We can observe that, when $x_{ij} = 0$, according to Equations (6.5), q_{ij} also equals 0, then we can set $M_{ij} = k'(W+m)^{\frac{3}{2}} t_{ij} + \sigma$ (σ is the battery energy available for a drone trip (kWh)). When $x_{ij} = 1$, the second term of the left-hand side of Equations (6.11) is the energy consumption on arc (i,j). Constraints (6.12) mean that battery's energy capacity constraint must be respected. Since constraints (6.11) are nonlinear, the model cannot be solved directly by a mixed-integer linear programming (MILP) solver. In Section 6.3.1, we introduce different types of cuts to tackle this group of constraints implicitly.

We also give the linear approximation version of constraints (6.11):

$$f_i + [\alpha(m + q_{ij}) + \beta]t_{ij}/3600 \leq M'_{ij}(1 - x_{ij}) + f_j \qquad \forall (i,j) \in A, \tag{6.13}$$

where $M'_{ij} = (\alpha m + \beta)t_{ij}/3600 + \sigma$. In numerical tests, we will compare the difference in solution construction when using these two versions of the energy expressions.

(iv) *Time and trip related constraints:*

$$\tau_i + t_{ij} - M''_{ij}(1 - x_{ij}) \leq \tau_j \qquad \forall i \in N', j \in N^-, \tag{6.14}$$

$$a_i \leq \tau_i \leq b_i \qquad \forall i \in N^-, \tag{6.15}$$

$$\tau_i + (t_{i,n+1} + t_{0j}) \leq \tau_j + (1 - z_{ij})M'''_{ij} \qquad \forall i, j \in N', i \neq j, \tag{6.16}$$

$$\sum_{\substack{i \in N' \\ i \neq j}} z_{ij} \leq x_{0j} \qquad \forall j \in N', \tag{6.17}$$

$$\sum_{\substack{j \in N' \\ j \neq i}} z_{ij} \leq x_{i,n+1} \qquad \forall i \in N', \tag{6.18}$$

$$\sum_{j \in N'} x_{0j} - \sum_{i \in N'} \sum_{\substack{j \in N' \\ j \neq i}} z_{ij} \leq K. \tag{6.19}$$

Constraints (6.14) establish the time relationship between customer i and its immediate successor j. We set the large constants $M''_{ij} = \max\{b_i + t_{ij} - a_j, 0\}$ [140]. Constraints (6.15) denote that the time window constraint must be respected. Here we impose the time window constraint instead of the deadline constraint, because the latter is a special case of the former with $a_i = 0, \forall i \in N$. This model fits best when drones land at customer sites for delivery, as we assume that drones can wait at customer locations until the opening of the time window and the energy consumption during this period is negligible. Note that in the case where the energy consumption during that period must be taken into account (e.g., in case when drones are equipped with cameras and sensors on to actively detect dangerous situations such as package or drone theft, or for a hovering while waiting), we can also incorporate the energy consumption of performing these activities in our model and our solution scheme can still be directly used. The detailed description on the modifications is presented in D. Equations (6.16) establish the time relationship between consecutive trips performed by the same drone, where $M'''_{ij} = t_{i,n+1} + t_{0j} + b_i$. These constraints take into account the time to return to the depot and replace the battery. Constraints (6.17)–(6.18) connect variables $\mathbf{x}$ and $\mathbf{z}$ [104]. Constraints (6.19) limit the number of drones that can be used in the system.

(v) *Variable domains:*

$$x_{ij} \in \{0,1\},\ q_{ij}, e_{ij} \geq 0 \qquad \forall (i,j) \in A, \tag{6.20}$$

$$f_i \geq 0 \qquad \forall i \in N, \tag{6.21}$$

$$\tau_i \geq 0 \qquad \forall i \in N^-, \tag{6.22}$$

$$z_{ij} \in \{0,1\} \qquad \forall i,j \in N'. \tag{6.23}$$

Objective Function. We consider the applications of logistics companies who use drones for last-mile delivery, in order to reduce an overall transportation cost. Therefore, we consider a general form of the objective function which also incorporates the energy consumption

$$\min \sum_{(i,j)\in A} (c_{ij}x_{ij} + \delta e_{ij}), \tag{6.24}$$

where δ is the battery-related cost ($\$/kWh$) which includes the cost of electricity and the amortization of lithium-ion battery. We will show how variables $e_{ij}, \forall (i,j) \in A$ are incorporated in the constraints and linked to variables f_i and f_j in following sections. Note that the energy cost could be negligible in realistic applications, and we add it here for two reasons: First, to keep consistent with some existing works, which also include the energy cost in the objective function to incorporate the depreciation and operating cost of battery as a function of energy usage [2,90]; Second, to demonstrate that our objective function is quite flexible. The model and approach can be used to solve a traditional VRP objective which minimizes the travel cost by dropping the second term, or a green supply chain related objective that minimizes the energy consumption/cost by dropping the first term. We analyze the impact of different objectives on computational efficiency and solution configurations in Section 6.4.3. For notational convenience, in the following sections we use R, E, and $R+E$ to represent the model that minimizes travel cost ($\delta = 0$), energy cost ($c_{ij} = 0, \forall (i,j) \in A$), and both travel and energy costs (as in the objective function (6.24)), respectively. For the energy calculation, we use a subscript e if the nonlinear energy function is used, and a subscript a if the linear approximation method is used.

We note that constraints in group (i), (ii), (iv), and (v) are adapted from studies on VRP and MTVRP [103,104,140]. However, the nonlinear energy constraints and the objective function are newly introduced. Moreover, the time-window constraints, which are not considered in [104] and [2], are also considered in our study. Thus, our model generalizes the other models in the literature, such that it can capture important practical constraints. We further emphasize that our modeling and solution schemes (introduced in next section) simultaneously optimize

multi-trip drone routing operations and energy consumption under time windows constraints. We also include a more complex nonlinear energy function.

6.2.3 Valid Inequalities

We use constraints (6.25) to indicate the least number of trips needed to visit all the customers [141,142].

$$\sum_{j \in N'} x_{0j} \geq \left\lceil \frac{\sum_{i \in N'} d_i}{Q} \right\rceil. \tag{6.25}$$

Constraints (6.26) are derived from Equations (6.8) using the constant d_j to replace the variable q_{ij}, which yields linear equations and a lower bound of $P(q_{ij})$ since $q_{ij} \geq d_j$ when $x_{ij} = 1$. Constraints (6.26) mean that if arc (i, j) is traversed by a drone, the energy consumption is at least equal to the value of the right-hand side.

$$e_{ij} \geq k'(W + m + d_j)^{\frac{3}{2}} t_{ij} x_{ij} \qquad \forall (i, j) \in A. \tag{6.26}$$

6.3 Solution Method

In this section, we introduce the techniques to handle the nonlinear energy consumption, and develop a B&C algorithm for our model. We note that our solution method can also be applied to other applications with nonlinear energy functions.

6.3.1 Cuts for Nonlinear Energy Function

Logical Cut (Infeasibility Cut). We first solve the model without constraints (6.10)–(6.12). When a feasible solution is generated, we check whether it satisfies the energy capacity constraint for each trip. For any violated trip $\{0, i_1, \ldots, i_l, n+1\}$, we add the logical cut

$$x_{i_1 i_2} + x_{i_2 i_3} + \ldots + x_{i_{l-1} i_l} \leq l - 2, \tag{6.27}$$

where i_{l-1} is the $(l-1)$th customer in the trip, and there are l customers in total in the trip. Equation (6.27) means that the customer sequence is not allowed to be performed.

Subgradient Cut. In Equation (6.8), $P(q_{ij})$ is a convex function in q_{ij}. Thus, the tangent line at point $(\bar{q}_{ij}, \bar{P}(\bar{q}_{ij}))$ (we use a bar '-' to represent known values) is

$$P(q_{ij}) = \bar{P}(\bar{q}_{ij}) + \bar{\beta}_{ij}(q_{ij} - \bar{q}_{ij}), \tag{6.28}$$

where $\bar{\beta}_{ij} = \frac{3}{2}k(W+m+\bar{q}_{ij})^{\frac{1}{2}}$, and it is the derivative of function $P(q_{ij})$ at point $(\bar{q}_{ij}, \bar{P}(\bar{q}_{ij}))$. Figure 6.2 is an illustration of the tangent line. Therefore, the subgradient cut derived for constraints (6.11) can be added using a conditional form as follows:

$$e_{ij} \geq [\bar{P}(\bar{q}_{ij})x_{ij} + \bar{\beta}_{ij}(q_{ij} - \bar{q}_{ij})]/1000 \times (t_{ij}/3600) \quad \forall (i,j) \in A. \tag{6.29}$$

When $x_{ij} = 0$, the right-hand side of Equation (6.29) is a negative number ($q_{ij} = 0$ because of constraints (6.5)) and the cut is inactive. When $x_{ij} = 1$, the cut is added and the right-hand side of (6.29) underestimates the energy from Equation (6.8).

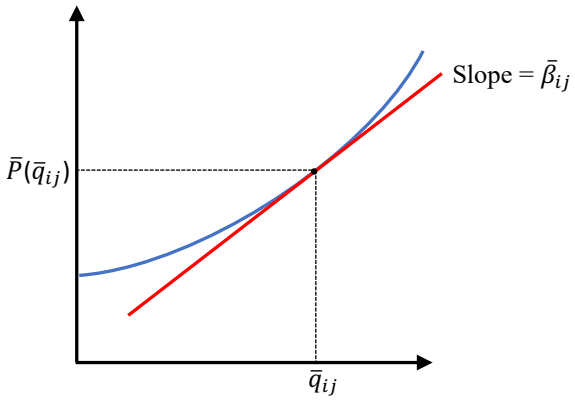


Figure 6.2 The tangent line of the power function

Remarks: (i) Being different from the logical cuts, constraints (6.10) and (6.12) are necessary when applying the subgradient cuts, and constraints (6.11) become $f_i + e_{ij} \leq M_{ij}(1 - x_{ij}) + f_j, \forall (i,j) \in A$. (ii) For the models *with* energy costs in the objective, i.e., the E and $R+E$ models, we must apply the subgradient cuts to ensure the involvement of energy cost. However, logical cuts are optional because the subgradient cuts can also guarantee that the energy capacity constraints are respected. (iii) For the models *without* energy costs, i.e., the R model, we can implement the cuts in three ways: only add logical cuts, only add subgradient cuts, or add both together. If there is only one customer in a trip, we do not add either logical or subgradient cuts for the R model, because we can guarantee that each customer is eligible to be serviced by a drone when generating the instance sets. Moreover, when only the logical cuts are used for model R, we do not need valid inequalities (6.26).

Our techniques can be applied for any energy function that is convex or piecewise convex in payload. If it is not a convex function, then our logical cut can be used. In other words, our method generalizes the ones presented in the literature.

6.3.2 Branch-and-Cut Algorithm

The B&C algorithm has been extensively used to solve MILP problems, and it is a combination of a cutting plane method with a B&B algorithm [143]. In our B&C scheme, we first add valid inequalities to the formulations at the root node of the search tree. We then solve the LP relaxation problem at each node of the tree. Each time a fractional solution is obtained, we detect and generate violated cuts in a cutting-plane fashion and the LP relaxation at the current B&B node is re-optimized. If all the cuts are respected and the solution still has fractional-valued integer variables, the branching process continues. If an integer solution is obtained and no cuts are generated, we consider updating the incumbent solution and pruning some nodes. This process continues until all nodes of the tree are evaluated.

Separation of Subtour Inequalities

Although constraints (6.4) can eliminate subtours, we introduce another group of subtour elimination constraints (SECs) which can help improve computational efficiency for the B&C scheme. The SECs are as follows [144]:

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \leq |S| - q(S) \qquad \forall S \subseteq N', |S| \geq 2, \tag{6.30}$$

where $q(S) = \left\lceil \frac{\sum_{i\in S} d_i}{Q} \right\rceil$ is the minimum number of trips needed to visit customers in set S. The separation algorithm is performed by using the CVRPSEP package in [145].

Implementation of Cuts and SECs

For the logical and subgradient cuts, they are applied when an integer solution is obtained. For the SECs, we only generate them at the root node since they are redundant for our models due to the fact that subtours are eliminated by constraints (6.4) and it is time consuming to solve the separation problems at all nodes of the B&B tree.

6.4 Numerical Experiment

In this section, we present the instances and discuss our numerical tests for the MTDRP with the energy function presented in this study. The B&C algorithms are coded in Python on Pycharm 2.7 using Gurobi 7.5.1 as the solver. All the parameters are set to their default values in the solver. The experiments are performed on a cluster of Intel Xeon X5650 CPUs with 2.67 GHz and 24 GB RAM under Linux 6.3. Each experiment is conducted on a single

core of one node unless specified. The computing time limit is set to four hours.

6.4.1 Instance Sets

We introduce two sets of benchmark instances. The first set, named *Set A*, is created based on the instance generation frameworks presented in [96] and [2]. The second set, named *Set B*, is an extension of Solomon's instances, taking into account drones' specific characteristics. For Set A instances, we further consider two types of instances and each has 10–50 customers. For type 1 instances, named *Set* A_1, the depots are located at the lower left corner of the region. For type 2 instances, named *Set* A_2, the depots are in the middle of the region. We use Set A instances for preliminary tests and performance comparisons. We conduct experiments on Set B instances. The detailed instance generation procedures are presented in D. All the instances and solutions are also available at the following URL: `https://sites.google.com/view/chengchun/instances`.

We assume that 4-cell 14.8V lithium polymer batteries are used for drones. According to the field tests in [2], we set $\alpha = 0.217\ kW/kg$, $\beta = 0.185\ kW$, $m = 1.5\ kg$, $W = 1.5\ kg$, $Q = 1.5\ kg$, $g = 9.81\ N/kg$, $\rho = 1.204\ kg/m^3$, $\varsigma = 0.0064\ m^2$, $h = 6$, $\delta = 360\ \$/kWh$. For Set A instances, we set the battery energy capacity $\sigma = 0.27\ kWh$; For Set B instances, we set $\sigma = 0.027\ kWh$.

6.4.2 Enhancement Strategy Evaluation

This section analyzes the effect of valid inequalities and SECs. We conduct all the tests on instances with 10–30 customers in Set A. First, we only apply subgradient cuts to the model to evaluate the valid inequalities and SECs. After knowing the performances, we further compare different implementations of cuts. Results are provided in Table 6.1. For each model, we present detailed results of the largest instances (i.e., those with 30 customers) in the first six rows, and the results of all instances in the last two rows. The column *None* gives the results without any enhancement strategy. The remaining columns indicate that one (or all) valid inequalities or SECs are added to the model. *Opt* is the number of instances solved to optimality. *UP*, *LB*, and *RLB* are the best upper bound, the best lower bound, and the lower bound at the root node, respectively. *Gap* is the percentage difference between the best upper and lower bounds. *CPU* is the time in seconds consumed to solve the instance.

Table 6.1 shows that different implementations of cuts yield different performances. In general, the simultaneous application of logical cuts, subgradient cuts, valid inequalities (6.25)–(6.26), and the SECs, gives the best performance for the three models. Specifically, a few

Table 6.1 Average results with different valid inequalities and SECs for Set A instances

	Cust		Only subgradient					Only logical	Subgradient +logical
			None	(6.25)	(6.26)	(6.25)+(6.26)	(6.25)+(6.26) +SECs	(6.25)+SECs	(6.25)+(6.26) +SECs
R_e	30	Opt	7/10⋆	7/10	5/10	7/10	7/10	7/10	**8/10**
		UB	11604.97	11608.23	11604.97	11616.74	11604.97	11604.97	11611.46
		LB	11539.12	11544.82	11520.78	11541.28	11553.25	11558.07	11575.70
		Gap	0.56	0.53	0.72	0.63	0.44	0.39	0.31
		CPU	6810.79	6038.29	10041.20	7436.01	6501.20	6033.06	6643.34
		RLB	11013.32	11041.65	11013.31	11033.22	11037.09	11055.78	11039.42
	All	Opt	46/50⋆	45/50	43/50	45/50	46/50	46/50	**47/50**
		Gap	0.14	0.16	0.22	0.19	0.13	0.11	0.10
E_e	30	Opt	0/10	0/10	3/10	3/10	4/10	Not Applicable	**6/10**
		UB	833.62	836.92	828.35	828.63	828.25		828.63
		LB	597.37	611.69	810.58	812.01	819.55		819.65
		Gap	28.24	26.82	2.24	2.07	1.05		1.07
		CPU	14400.00	14400.00	11515.46	12046.49	11437.83		9440.20
		RLB	123.92	80.82	705.74	704.07	708.00		708.00
	All	Opt	31/50	31/50	41/50	41/50	43/50		**46/50**
		Gap	8.83	8.06	0.54	0.50	0.22		0.22
$(R+E)_e$	30	Opt	6/10	5/10	6/10	6/10	**7/10**	Not Applicable	**7/10**
		UB	12451.89	12471.68	12437.36	12437.36	12437.36		12450.29
		LB	12321.26	12293.59	12343.51	12365.55	12369.68		12374.19
		Gap	1.03	1.40	0.75	0.57	0.54		0.60
		CPU	11631.34	9808.45	8577.97	8623.99	8512.23		8270.53
		RLB	11015.17	11181.01	11709.88	11742.26	11766.60		11766.70
	All	Opt	44/50	43/50	45/50	45/50	**46/50**		**46/50**
		Gap	0.30	0.38	0.19	0.18	0.16		0.18

⋆ indicates the number of instances (out of 10 and 50) that are solved to optimality.

more instances can be solved to optimality for the R_e and E_e models. For the $(R+E)_e$ model, the number of optimally solved instances is the same when only using the subgradient cut or using both cuts together; however, the average optimality gap is relatively close. We can also observe that, for instances with 30 customers, the average RLB is improved from 123.92 to 705.74 for model E_e when the valid inequalities based on the energy function (i.e., constraints (6.26)) are used. In addition, the E_e model consumes the most computation time on average, because its average RLB is not as tight as those of the other two models. In particular, RLB/LB= 0.86 for the E_e model whereas RLB/LB= 0.95 for the other two models. In the following sections, we use the 2-index formulation constructed in Section 6.2.2, together with constraints (6.25)–(6.26), SECs, and both logical and subgradient cuts, to perform our tests for each model.

6.4.3 Details of Solutions for Set A Instances With Size 10–30

Tables 6.2–6.3 give a summary details of results. *Cust* is the number of customers. *Log*, *Sub*, and *SECs* are the number of generated logical cuts, subgradient cuts, and SECs, respectively. In Table 6.3, *UAVs* is the number of drones used, and *Swap* represents the average number of battery swaps. When calculating *Swap*, we do not count the first trip performed by a drone.

For example, if a drone has conducted 3 trips, then the value of *Swap* would be 2. *T/d* indicates the average number of trips performed by each drone. The last column in Table 6.3 is the proportion of energy cost to total cost. More detailed results for each instance are presented in Appendix D.

Table 6.2 Average results on cuts for Set A instances with size 10–30

		R_e						E_e						$(R+E)_e$					
	Cust	Opt	Gap	CPU	Log	Sub	SECs	Opt	Gap	CPU	Log	Sub	SECs	Opt	Gap	CPU	Log	Sub	SECs
Set A_1	10	5/5	0.0	0.5	0.2	92.2	21.4	5/5	0.0	0.9	0.4	138.2	31.4	5/5	0.0	0.6	0.0	134.8	25.2
	15	5/5	0.0	176.6	3.4	400.6	27.6	5/5	0.0	82.6	2.6	595.4	33.2	5/5	0.0	306.2	2.8	518.6	32.6
	20	5/5	0.0	162.2	1.6	349.8	35.4	5/5	0.0	72.3	1.6	558.2	39.4	5/5	0.0	224.0	0.6	615.6	35.6
	25	4/5	0.4	4666.1	3.0	552.4	34.2	5/5	0.0	4508.6	4.8	1282.0	39.8	4/5	0.6	4945.6	4.4	1512.0	38.2
	30	3/5	0.6	6860.3	3.8	1172.8	42.4	3/5	1.0	8989.8	3.4	1884.6	47.8	3/5	0.7	8865.7	2.8	1759.0	47.0
Set A_2	10	5/5	0.0	0.4	0.0	40.8	16.6	5/5	0.0	0.5	0.4	142.6	13.8	5/5	0.0	0.3	0.0	71.4	9.8
	15	5/5	0.0	5.0	2.4	152.2	34.0	5/5	0.0	4.1	1.4	274.4	27.6	5/5	0.0	5.7	1.8	297.8	29.4
	20	5/5	0.0	27.0	2.0	259.0	34.6	5/5	0.0	37.3	1.6	487.6	40.0	5/5	0.0	33.1	1.4	500.8	36.4
	25	5/5	0.0	202.3	2.2	469.4	36.6	5/5	0.0	301.0	4.0	921.0	37.2	5/5	0.0	313.1	2.4	853.0	39.4
	30	5/5	0.0	6426.4	3.2	927.2	41.2	3/5	1.2	9890.6	5.0	1815.8	35.4	4/5	0.5	7675.3	4.0	1437.4	39.6

Table 6.3 Average results on drones for Set A instances with size 10–30

		R_e			E_e			$(R+E)_e$			
	Cust	UAVs	Swap	T/d	UAVs	Swap	T/d	UAVs	Swap	T/d	Cost_e (%)
Set A_1	10	2.0	3.2	2.6	2.0	3.2	2.6	2.0	3.2	2.6	6.6
	15	2.2	4.0	2.9	2.2	4.2	3.0	2.2	4.0	2.9	6.6
	20	3.6	6.2	2.8	3.6	6.2	2.8	3.6	6.2	2.8	6.8
	25	3.8	7.4	3.0	3.8	7.6	3.0	3.8	7.6	3.0	6.8
	30	5.0	8.4	2.7	5.0	8.4	2.7	5.0	8.2	2.6	6.8
Set A_2	10	2.0	3.4	2.7	2.0	3.6	2.8	2.0	3.4	2.7	6.4
	15	2.8	5.6	3.1	2.8	6.2	3.3	2.8	5.6	3.1	6.4
	20	3.4	7.2	3.2	3.4	7.4	3.2	3.4	7.2	3.2	6.6
	25	4.0	8.6	3.2	4.0	9.2	3.3	4.0	8.6	3.2	6.7
	30	4.4	9.6	3.2	4.4	10.0	3.3	4.4	9.6	3.2	6.6
	Average	3.3	6.4	2.9	3.3	6.6	3.0	3.3	6.4	2.9	6.6

Table 6.2 shows that the number of subgradient cuts is much larger than that of the logical cuts. This is because the subgradient cuts are produced for edges while the logical cuts are generated for trips. For the largest problems (30 customers), the R_e model consumes the least computing time and the E_e model consumes the most computing time. Further, instances in Set A_1 require more time than those in Set A_2, because the locations of the depot and customers are more geographically dispersed in Set A_1.

Table 6.3 indicates that more drones are used with an increasing number of customers and that, in most cases, each drone performs 2 or 3 trips. For the $(R+E)_e$ model, the energy cost only accounts for a small portion (around 6.6%) of the total cost. The average results seem similar for the three models; however, with different objectives, different schedules are indeed generated for some instances. An example is given in Table 6.4. It shows that two

more trips are performed for the E_e model, leading to a greater travel distance and a lower energy consumption. Moreover, we find that for this instance the schedules generated by the R_e and $(R+E)_e$ models are quite similar, except that the travel direction of the second and fifth trips are opposite. Since we perform our tests on an undirected network, travel direction influences energy consumption because of different payloads on arcs. However, as the $(R+E)_e$ model includes the energy cost in the objective, it can always guarantee that drones travel in directions with minimal energy consumption. Thus, in realistic applications, for undirected networks, even though decision makers favor a VRP objective which minimizes the travel cost, they can still add energy cost in the objective and set a small value for energy price to save battery energy consumption and further reduce the recharging time.

Table 6.4 Schedules generated by different objectives for instance *Set_A2_Cust_15_2*

R_e		E_e		$(R+E)_e$	
Trips	Energy (kWh)	Trips	Energy (kWh)	Trips	Energy (kWh)
[0, 3, 1, 16]	0.1585	[0, 3, 1, 16]	0.1585	[0, 3, 1, 16]	0.1585
[0, 4, 2, 16]	**0.2389**	[0, 4, 2, 16]	0.2389	[0, 2, 4, 16]	**0.2344**
[0, 5, 10, 15, 16]	0.2099	[0, 10, 5, 16]	0.0937	[0, 5, 10, 15, 16]	0.2099
[0, 6, 12, 16]	0.2530	[0, 12, 16]	0.1645	[0, 6, 12, 16]	0.2530
[0, 7, 8, 16]	**0.1733**	[0, 8, 7, 16]	0.1637	[0, 8, 7, 16]	**0.1637**
[0, 9, 11, 16]	0.2418	[0, 9, 16]	0.1354	[0, 11, 9, 16]	0.1835
[0, 13, 16]	0.1690	[0, 13, 16]	0.1690	[0, 13, 16]	0.1690
[0, 14, 16]	0.1341	[0, 14, 16]	0.1341	[0, 14, 16]	0.1341
		[0, 11, 16]	0.0462		
		[0, 15, 6, 16]	0.1794		
Total energy (kWh)	1.5785		1.4834		1.5061
Total travel distance	7995.39		8153.26		7995.39

6.4.4 Performance Comparison Between Models with Nonlinear and Linear Energy Functions

In this section, we compare the model performance with our nonlinear and linear energy models. Table 6.5 presents a summary of results, with some detailed results in Appendix D. For solutions generated by the linear approximation models, after obtaining the trips, we calculate the energy consumption using the nonlinear model (6.8) for each trip and report the average results in the last two rows of Table 6.5. *Infeasible* is the number of instances for which the linear approximation models yield trips that when energy is calculated with (6.8) exceed the energy capacity. *Energy gap* is the percentage difference in energy calculation, which is computed as *(energy from (6.8) − energy from (6.9))/energy from (6.9)*.

From Table 6.5, we get two observations: (1) ***Computational efficiency***. For both models, the computational efficiency with the linear approximation (Equation (6.9)) is better than that of the nonlinear method (Equation (6.8)). By using the approximation method, more

Table 6.5 Statistics of solutions generated by models R and $R+E$ with nonlinear and linear energy functions

	model R		model $R+E$	
Energy function	Nonlinear	Linear	Nonlinear	Linear
Opt	47/50	49/50	46/50	50/50
Optimality gap	0.10	0.04	0.18	0.00
CPU	1852.69	911.59	2236.96	743.95
Travel distance	8276.33	8227.72	8278.71	8227.68
Infeasible	0/50	**20/50**	0/50	**18/50**
Energy gap (%)	0.00	**9.45**	0.00	**9.32**

instances can be solved in a shorter time frame. For the R and $R+E$ models with the nonlinear energy function, the average computation times are 1852.69 and 2236.96 seconds, respectively. Thus, we can conclude that, even though our original models are nonlinear, the use of logical and subgradient cuts can help solve large problems to optimality. (2) ***Feasibility and solution quality***. In multiple instances, the approximation models yield "energy infeasible" trips when energy is calculated based on the nonlinear model (6.8). For the R and $R+E$ models, the approximation method produces infeasible trips for 20 and 18 instances respectively. In addition, the energy gap is around 9% on average between the two methods. Note that one may argue that to guarantee the feasibility of trips, we can use an over-estimated linear energy function. However, our preliminary tests demonstrate that this strategy might produce solutions with more trips and energy consumption. We provide an example in Appendix D to explain the results of this new strategy.

To further display the importance of how energy is calculated, we give an example in Table 6.6 to show the different schedules generated by the two methods. It demonstrates that the first trip given by the two approximation models consumes 0.2925 kWh energy, which violates the battery's energy capacity (0.27 kWh). However, if the linear approximation method is used, it will consider these trips as feasible ones. Therefore, care is needed when modeling energy consumption to ensure energy feasibility of routes.

6.4.5 Impact of Time Windows

Here, we first consider new instances with tighter time windows at customers. We generate the width of customers' time windows according to a new normal distribution whose mean is $0.15(b_{n+1}-t_{j,n+1}-t_{0j})$, and keep other data unchanged. Next, we remove the time constraints (6.14)–(6.16) and solve a multi-trip drone routing problem. Summary results are reported in Table 6.7 and detailed results are presented in Appendix D.

From Table 6.2 and Table 6.7, when the time windows are tighter, one more instance can

Table 6.6 Detailed solutions of models with nonlinear and linear energy functions for instance *Set_A1_Cust_25_2*

	Nonlinear energy function		Linear energy function			
				Energy consumption		
	Trips	Energy	Trips	Linear	Nonlinear	Energy gap(%)
R	[0, 1, 14, 2, 6, 26]	0.2685	**[0, 1, 14, 2, 6, 16, 26]**	0.2656	0.2925	10.13
	[0, 4, 7, 8, 26]	0.1348	[0, 7, 8, 26]	0.1147	0.1240	8.11
	[0, 5, 20, 26]	0.2056	[0, 5, 20, 26]	0.1857	0.2056	10.72
	[0, 13, 11, 26]	0.1501	[0, 13, 11, 26]	0.1378	0.1501	8.93
	[0, 15, 3, 26]	0.1471	[0, 15, 3, 26]	0.1346	0.1471	9.29
	[0, 17, 10, 24, 12, 25, 26]	0.2528	[0, 12, 24, 10, 17, 4, 26]	0.1965	0.2175	10.69
	[0, 18, 9, 16, 26]	0.2138	[0, 25, 18, 9, 26]	0.1944	0.2146	10.39
	[0, 19, 21, 26]	0.1551	[0, 21, 19, 26]	0.1320	0.1447	9.62
	[0, 22, 26]	0.0134	[0, 22, 26]	0.0127	0.0134	5.51
	[0, 23, 26]	0.0684	[0, 23, 26]	0.0622	0.0684	9.97
$R+E$	[0, 1, 14, 2, 6, 26]	0.2685	**[0, 1, 14, 2, 6, 16, 26]**	0.2656	0.2925	10.13
	[0, 4, 7, 8, 26]	0.1348	[0, 8, 7, 26]	0.1099	0.1177	7.10
	[0, 20, 5, 26]	0.2039	[0, 20, 5, 26]	0.1845	0.2039	10.51
	[0, 13, 11, 26]	0.1501	[0, 13, 11, 26]	0.1378	0.1501	8.93
	[0, 15, 3, 26]	0.1471	[0, 15, 3, 26]	0.1346	0.1471	9.29
	[0, 17, 10, 24, 12, 25, 26]	0.2528	[0, 4, 17, 10, 24, 12, 26]	0.1903	0.2091	9.88
	[0, 18, 9, 16, 26]	0.2138	[0, 25, 18, 9, 26]	0.1944	0.2146	10.39
	[0, 21, 19, 26]	0.1447	[0, 21, 19, 26]	0.1320	0.1447	9.62
	[0, 22, 26]	0.0134	[0, 22, 26]	0.0127	0.0134	5.51
	[0, 23, 26]	0.0684	[0, 23, 26]	0.0622	0.0684	9.97

Table 6.7 Average results for models with tighter time windows and without time windows

		model R_e						model E_e						model $(R+E)_e$					
		Tighter time windows			Without time windows			Tighter time windows			Without time windows			Tighter time windows			Without time windows		
	Cust	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU
Set A_1	10	5/5	0.0	0.2	5/5	0.0	0.5	5/5	0.0	0.3	5/5	0.0	1.0	5/5	0.0	0.3	5/5	0.0	0.4
	15	5/5	0.0	32.2	5/5	0.0	527.4	5/5	0.0	47.8	5/5	0.0	44.6	5/5	0.0	41.7	5/5	0.0	129.5
	20	5/5	0.0	82.1	5/5	0.0	387.1	5/5	0.0	106.8	5/5	0.0	200.7	5/5	0.0	87.8	5/5	0.0	287.5
	25	5/5	0.0	2377.8	4/5	0.7	5948.0	3/5	1.6	6319.1	4/5	0.4	5641.6	5/5	0.0	3179.9	4/5	0.8	5061.6
	30	3/5	0.6	6565.7	1/5	0.5	12339.4	3/5	1.1	8552.8	2/5	1.3	12211.0	3/5	0.7	6425.4	1/5	0.8	12468.2
Set A_2	10	5/5	0.0	0.1	5/5	0.0	1.5	5/5	0.0	0.2	5/5	0.0	0.9	5/5	0.0	0.2	5/5	0.0	1.0
	15	5/5	0.0	1.0	5/5	0.0	8.8	5/5	0.0	1.3	5/5	0.0	4.4	5/5	0.0	1.2	5/5	0.0	9.6
	20	5/5	0.0	10.1	5/5	0.0	97.4	5/5	0.0	19.3	5/5	0.0	47.2	5/5	0.0	13.4	5/5	0.0	73.4
	25	5/5	0.0	140.2	5/5	0.0	1183.1	5/5	0.0	240.7	5/5	0.0	119.2	5/5	0.0	121.8	5/5	0.0	1006.6
	30	5/5	0.0	646.5	3/5	0.9	8849.3	5/5	0.0	2458.6	3/5	0.5	7969.1	5/5	0.0	507.5	4/5	0.5	6988.8

be solved to optimality for the R_e model, and two more instances for the $(R+E)_e$ model. Moreover, for problems of the same size, the average computation time is generally reduced for the three models. However, when the time constraints are absent, instances become much more difficult to handle. Fewer instances can be solved to optimality within the time limit and the average computation time also increases. These observations are consistent with the results provided by reference [95], where a B&P algorithm is used for the MTVRPTW.

6.4.6 Algorithm Performance on Extended Solomon's Instances

In this section we test our algorithm on Set B instances based on the well-known Solomon's instances. All the experiments are performed on 4 core processors with a 12-hour (43200

seconds) time limit. Summarized results are shown in Table 6.8 and detailed results on each instance are provided in Appendix D. In Table 6.8, column *Inst* is the instance label.

Table 6.8 Algorithm performance on Solomon's instances of type 2

	25 customers						40 customers					
	model R_e		model E_e		model $(R+E)_e$		model R_e		model E_e		model $(R+E)_e$	
Inst	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU
c201	0.00	1.4	0.00	11.5	0.00	2.5	0.00	48.3	0.00	229.9	0.00	43.8
c202	0.00	13.4	0.00	48.5	0.00	19.7	0.00	372.9	0.00	14073.8	0.00	955.4
c203	0.00	56.4	0.00	86.5	0.00	74.8	0.00	3881.6	0.95	43200.0	0.00	10685.1
c204	0.00	48.2	0.00	240.8	0.00	78.6	0.20	43200.0	1.55	43200.0	0.00	16940.6
c205	0.00	13.0	0.00	29.5	0.00	9.0	0.00	871.2	0.00	23630.3	0.00	881.7
c206	0.00	21.1	0.00	40.8	0.00	21.7	0.00	2011.9	1.83	43200.0	0.00	8368.8
c207	0.00	32.0	0.00	55.3	0.00	43.4	0.00	5757.3	0.00	5567.9	0.00	8358.9
c208	0.00	23.2	0.00	34.7	0.00	34.3	0.00	1936.7	0.63	43200.0	0.00	3547.3
r201	0.00	4.0	0.00	8.7	0.00	5.6	0.00	437.7	0.00	2730.5	0.00	348.1
r202	0.00	32.9	0.00	26.9	0.00	29.2	0.00	11666.6	1.22	43200.0	0.00	8183.1
r203	0.00	132.1	0.00	40.2	0.00	58.0	0.00	33302.4	0.00	17013.0	0.00	40920.9
r204	0.00	134.2	0.00	125.4	0.00	120.3	0.58	43200.0	0.00	37471.4	1.22	43200.0
r205	0.00	38.5	0.00	21.8	0.00	27.1	0.00	5921.7	0.00	7086.4	0.00	8098.5
r206	0.00	54.8	0.00	29.8	0.00	84.6	0.96	43200.0	0.00	5167.2	0.42	43200.0
r207	0.00	83.5	0.00	37.4	0.00	87.8	1.18	43200.0	0.00	22529.7	0.00	39388.0
r208	0.00	75.7	0.00	47.9	0.00	106.7	0.96	43200.0	0.36	43200.0	0.67	43200.0
r209	0.00	42.8	0.00	41.5	0.00	50.0	0.00	42044.0	5.91	43200.0	1.40	43200.0
r210	0.00	46.0	0.00	24.8	0.00	60.2	0.00	14821.3	0.00	2832.5	0.68	43200.0
r211	0.00	136.1	0.00	53.4	0.00	102.5	1.44	43200.0	0.62	43200.0	1.16	43200.0
rc201	0.00	28.0	0.00	31.7	0.00	60.0	0.00	959.5	7.42	43200.0	0.00	2540.3
rc202	0.00	366.9	0.00	315.0	0.00	125.1	0.79	43200.0	2.86	43200.0	0.62	43200.0
rc203	0.00	15.7	0.00	29.9	0.00	56.8	0.00	1542.6	0.00	509.2	0.00	5064.3
rc204	0.00	5.4	0.00	64.3	0.00	952.8	0.00	110.3	0.00	2663.9	0.00	9900.8
rc205	0.00	63.4	0.00	269.0	0.00	100.6	0.00	27642.5	13.59	43200.0	1.58	43200.0
rc206	0.00	65.4	0.00	59.5	0.00	755.0	0.00	4719.6	10.42	43200.0	0.00	39338.2
rc207	0.00	1253.2	0.00	58.3	0.00	7306.9	0.39	43200.0	1.10	43200.0	0.91	43200.0
rc208	0.00	207.4	0.00	23.0	0.00	157.5	0.00	1684.8	0.00	237.6	0.00	7961.0
Average	$0.00^{(0)}$	110.9	$0.00^{(0)}$	68.7	$0.00^{(0)}$	390.0	$0.24^{(8)}$	18716.0	$1.80^{(13)}$	26049.8	$0.32^{(9)}$	22234.3

$^{(-)}$ indicates the number of instances (out of 27) that are not solved to optimality.

We can observe that all instances with 25 customers are solved to optimality within the time limit. When the number of customers increases to 40, 19 out of 27 instances are optimally solved for model R_e, and this number decreases to 14 and 18 for model E_e and model $(R+E)_e$, respectively. The CPU time also varies widely, ranging from a few minutes to many hours. In terms of computational performance as opposed to the MTVRPTW which is relatively similar to the MTDRP considered in this work, our algorithms could generally solve larger instance sizes compared to those considered in exact algorithms for the MTVRPTW despite the fact that our original models are nonlinear and more complex.

6.4.7 Results for Large Instances of Set A

Here, we report the results of Set A instances with 35–50 customers in Table 6.9. More detailed results are given in Appendix D. All the experiments are performed on 4 core processors with a 12-hour time limit. The instances with 10–30 customers that were not optimally solved in previous experiments are also solved again with the longer time limit. Our results show

that all the previous instances, except *Set_A*$_1$*_Cust_30_5* for model E_e, are solved to optimality under the new experiment setting. The optimality gap of this instance for model E_e is 1.77%. For some instances, when we directly solve the E_e model or the $(R+E)_e$ model, we find that the optimality gap is over 5% within the time limit, mainly resulting from the poor lower bound. Considering the R_e model is relatively easier than the other two models, for these specific instances, we first solve the R_e model to get a feasible solution and then use this solution as a start for the other two models. The results of these instances are marked by a square in Table 6.9.

Table 6.9 Results using multicore processors for Set A instances with 35–50 customers

			model R_e		model E_e		model $(R+E)_e$	
	Cust	Inst	Gap	CPU	Gap	CPU	Gap	CPU
Set A$_1$	35	1	4.11	43200.0	4.55$^{\square}$	43200.0	3.24$^{\square}$	43200.0
		2	1.80	43200.0	3.58$^{\square}$	43200.0	2.78	43200.0
		3	0.00	20642.0	0.00$^{\square}$	21866.9	0.98	43200.0
		4	0.00	30214.8	0.00$^{\square}$	36895.7	0.21	43200.0
		5	0.00	29126.0	0.00	15121.7	0.00	20446.6
	40	1	3.38	43200.0	2.92	43200.0	3.95	43200.0
		2	0.00	13947.5	0.90	43200.0	0.59	43200.0
		3	3.74	43200.0	4.74	43200.0	3.78$^{\square}$	43200.0
		4	0.44	43200.0	0.00$^{\square}$	23208.1	0.39	43200.0
		5	0.73	43200.0	1.30$^{\square}$	43200.0	2.32	43200.0
	45	1	4.24	43200.0	3.65$^{\square}$	43200.0	3.96	43200.0
		2	2.15	43200.0	2.66$^{\square}$	43200.0	2.06$^{\square}$	43200.0
		3	1.51	43200.0	3.76	43200.0	2.46$^{\square}$	43200.0
		4	3.05	43200.0	4.42$^{\square}$	43200.0	3.15$^{\square}$	43200.0
		5	1.95	43200.0	1.79$^{\square}$	43200.0	2.29	43200.0
	Average		1.81	37942.0	2.28	38152.8	2.14	41683.1
Set A$_2$	35	1	2.53	43200.0	0.00$^{\square}$	31873.4	2.65	43200.0
		2	0.00	1755.4	0.00	18306.0	0.00	3397.5
		3	0.00	8732.7	2.83	43200.0	0.00	11645.0
		4	0.00	9765.1	0.00	38648.4	0.00	25076.3
		5	0.00	9491.5	0.00	42292.8	0.00	18041.8
	40	1	0.00	32162.3	0.00	6219.5	0.00	21628.1
		2	1.12	43200.0	1.44	43200.0	0.00	41897.4
		3	0.00	3298.6	0.00	30554.3	0.00	5495.7
		4	2.13	43200.0	0.00$^{\square}$	7308.4	2.16	43200.0
		5	5.05	43200.0	6.39$^{\square}$	43200.0	4.81	43200.0
	45	1	0.00	6142.2	0.00	1802.5	0.00	8093.6
		2	0.00	41018.0	1.30	43200.0	1.32	43200.0
		3	1.20	43200.0	2.15	43200.0	0.00	7452.0
		4	2.02	43200.0	1.85$^{\square}$	43200.0	2.68	43200.0
		5	0.00	37956.5	2.11	43200.0	1.00	43200.0
	50	1	1.80	43200.0	0.74$^{\square}$	43200.0	2.36	43200.0
		2	3.81	43200.0	2.24$^{\square}$	43200.0	3.80	43200.0
		3	2.92	43200.0	4.20$^{\square}$	43200.0	2.72	43200.0
		4	1.32	43200.0	3.29	43200.0	0.72$^{\square}$	43200.0
		5	1.80	43200.0	1.55$^{\square}$	43200.0	2.01$^{\square}$	43200.0
	Average		1.29	31276.1	1.50	34770.27	1.31	30896.4

$^{\square}$ We use the first feasible solution of the R model as an initial solution for this model.

Table 6.9 shows that the average gap ranges from 1.81% to 2.28% for instances in Set A_1 and from 1.29% to 1.50% for instances in Set A_2, which further confirms our previous observation that generally instances in Set A_2 are easier than those in Set A_1. For the R_e model, 13 out of 35 instances are solved to optimality. For the E_e and $(R + E)_e$ models, the number of optimally solved instances are 12 and 10 respectively. We also note that it is effective to use the first solution of the R_e model as a start for the other two models. In particular, for the E_e model, 5 instances can be solved to optimality by using this method. We further use this idea to model E_e for Solomon's r209, rc201, rc205, and rc206 instances with 40 customers (i.e., instances whose optimality gap is over 5% in Table 6.8). The results show that all these instances can be solved to optimality now.

6.5 Conclusions

This chapter solves a MTDRP with time windows. A 2-index formulation is introduced and a B&C algorithm is developed. We propose two types of cuts to tackle the nonlinear energy function. We demonstrate the differences between using a complex nonlinear energy consumption function and a linear approximation, which can result in higher energy use and energy infeasible drone routes. We generate benchmark instances for the drone routing problem and conduct extensive numerical experiments to evaluate the effects of valid inequalities and user cuts. The effectiveness of our modeling scheme and the B&C algorithm is confirmed by solving generated instances and Solomon's type 2 instances.

CHAPTER 7 ROBUST DRONE DELIVERY WITH WEATHER INFORMATION

This chapter is based on the following article.

- Cheng, C., Adulyasak, Y., Rousseau, L.-M., Sim, M., 2020. Available at `http://www.optimization-online.org/DB_FILE/2020/07/7897.pdf`. Robust Drone Delivery with Weather Information. Submitted to *Operations Research*.

7.1 Introduction

In this section, we first introduce the backgrounds of our study and then present the contributions.

7.1.1 Backgrounds

A well-designed drone delivery system mitigates the risks of weather uncertainty. Uncertain wind conditions, i.e., speeds and directions of the wind, can impact the transit times of the drones to their destinations, leading to late deliveries or even cancellations of service [26]. In order to tackle this important issue, we explore how the available weather data, such as wind observations, can be used to improve scheduling decisions in a drone delivery system, where a fleet of identical drones is dispatched to visit a set of N geographically dispersed customers. Each drone can make multiple round trips, where each round trip is a flight from the depot to the customer and back to the depot for preparation for the next delivery, which may include tasks such as mounting a new payload and/or swapping out a depleted battery. The transit times of the drones between the depot and their assigned customers are affected by wind conditions, i.e., the wind may increase or decrease their flight times depending on its speed and direction. Figure 7.1 illustrates the relationship between wind condition and flight times. The depot and customers are distributed within a two-dimensional space and, without any loss of generality, the coordinates of the depot are $(0, 0)$. The wind vector is represented by the polar coordinates (r, θ), where r denotes the speed of the wind, and θ is the angle between the direction of the wind and the x-axis. The location of customer i, $i \in [N]$ is represented by the polar coordinates (d_i, ϕ_i), where d_i is the distance from the depot.

The launch speed of drones in the forward direction (i.e., from the depot to customers), which depends on the weight and size of the customer's payload, is $\bar{r}_i$. In the absence of a payload,

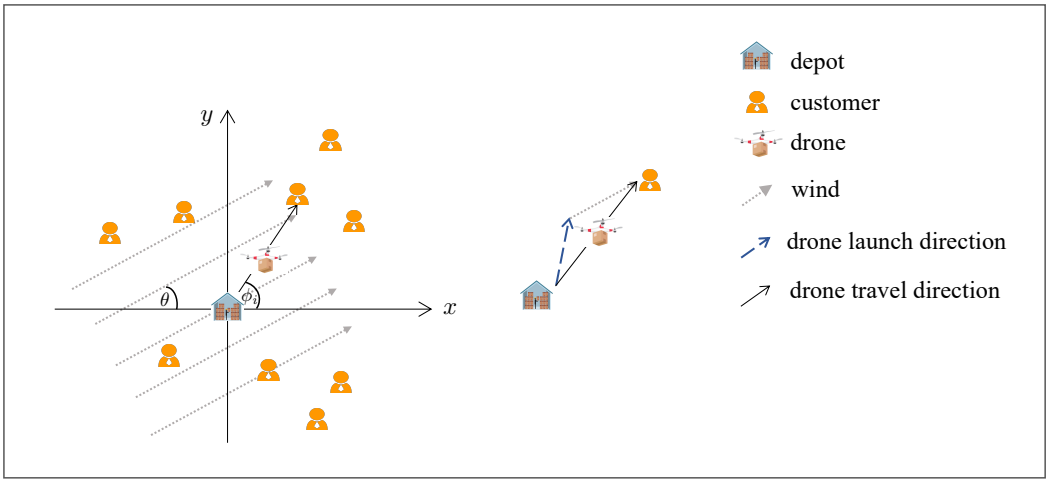


Figure 7.1 Illustration for the calculation of flight times. The right part is based on vector addition in Physics.

the launch speed of drones in the backward direction is $\bar{r}_0$, $\bar{r}_0 \geq \bar{r}_i$, $i \in [N]$. We assume $r < \bar{r}_i$, $i \in [N]$ to forbid delivery under very windy conditions. The automatic navigation system within a delivery drone would adapt according to the wind condition, to offset the wind's influence on flight direction. We can derive the nominal forward flight time u_i and the backward flight time v_i as

$$\begin{cases} u_i(r,\theta) \triangleq \dfrac{d_i}{\sqrt{(\bar{r}_i)^2 - r^2\sin^2(\theta-\phi_i)} + r\cos(\theta-\phi_i)}, \\ v_i(r,\theta) \triangleq \dfrac{d_i}{\sqrt{(\bar{r}_0)^2 - r^2\sin^2(\theta-\phi_i)} - r\cos(\theta-\phi_i)}. \end{cases} \tag{7.1}$$

Note that since $r < \min_{i\in[N]} \bar{r}_i$, the launch speed of the drone has a positive influence on flight duration. While a drone may not increase its launch speed, it could reduce its speed or even delay its flight, which may be necessary to avoid an early arrival at the customer's location. For instance, if a drone departs from the depot at τ and the customer can only be served after $\bar{\tau}^1$, then the arrival time of the drone at the customer's location would be $\max\{\tau + u_i, \bar{\tau}^1\}$.

In practice, the wind condition cannot be perfectly predicted and the actual flight times may deviate from the nominal values determined solely by the wind speed. To account for the uncertainty of wind vector (r,θ), we denote the uncertain forward and backward flight times as

$$\tilde{u}_i = u_i(\tilde{r},\tilde{\theta}) \qquad \tilde{v}_i = v_i(\tilde{r},\tilde{\theta}).$$

Weather data, such as historical wind observations, are commonly available from the national or local meteorological information center, and can be used as predictors of flight times. Figure 7.2 is a wind rose diagram of a province in China during a particular time interval. Wind information is reported in terms of speed and direction. Each piece of information can be recognized as a wind observation sample, based on which we can calculate the flight times between the depot and customers, thus generating samples of flight times. Specifically, given H samples of the observed wind speed and direction, (r_h, θ_h), $h \in [H]$, the corresponding forward and backward flight times of a drone to the ith customer, $i \in [N]$ are given by $u_i(r_h, \theta_h)$ and $v_i(r_h, \theta_h)$, respectively.

7.1.2 Contributions

We propose a drone delivery system to fulfill customers' requests for their delivery to be made in either the morning or in the afternoon. Given the weather data, our goal is to robustly optimize the schedules of the drone delivery system to mitigate the risks of delivery delays due to uncertainty in wind conditions. The specific challenges and our contributions are summarized as follows.

1. We propose a novel two-period data-driven adaptive distributionally robust optimization (DRO) model that permits the modeler to use wind observation data to improve scheduling decisions in a drone delivery system. The scheduling decisions for the fleet of drones are made in the morning, with provision for different delivery schedules in the afternoon that adapt to updated weather information available by midday.

2. We show how to construct the ambiguity set characterizing the uncertain flight times of the drones using wind observation data. We propose a cluster-wise ambiguity set, which has the benefits of tractability while avoiding overfitting to the empirical distribution. We incorporate other weather information in the form of discrete scenarios to enhance the ambiguity set and improve the drone schedule adaptation.

3. Our approach minimizes a decision criterion that is based on the essential riskiness index [146], which has performed well in vehicle routing problems in terms of meeting delivery deadlines by limiting the probability of tardy delivery and the magnitude of lateness. We exploit structures in our proposed decision criterion to reduce the complexity of our model.

4. For computational scalability, we propose a branch-and-cut (B&C) approach to solve the adaptive DRO model. Results show that the B&C algorithm can generally solve

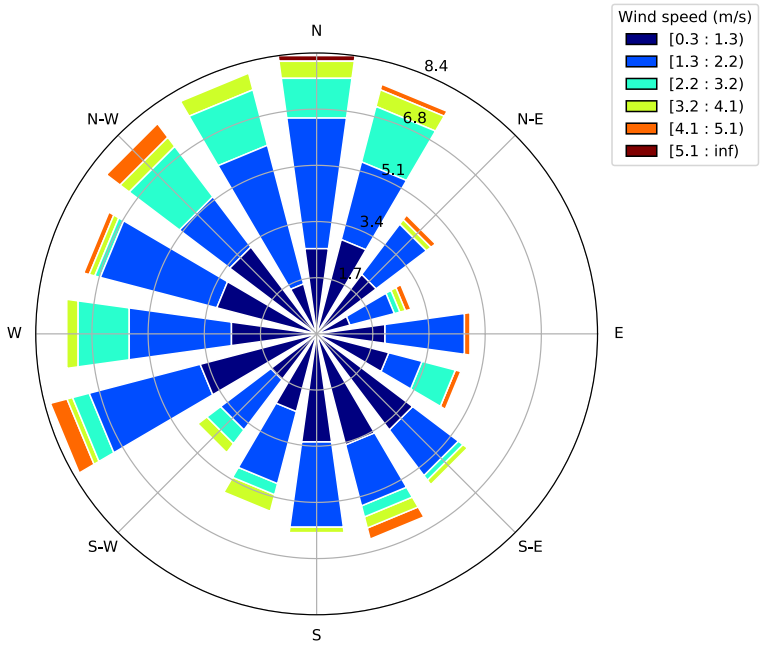


Figure 7.2 Wind information collected from 145 subregions of Sichuan Province, China, ranging from time interval 00:00 to 04:00 on September 14$^{\text{th}}$, 2019. Data are downloaded from China National Meteorological Information Center (`http://data.cma.cn/site/index.html`). The color-coded bands represent wind speed ranges and the circles denote different frequencies (from 0 to 8.4%).

more instances to optimality in a shorter time, compared to the MATLAB based toolbox RSOME. To the best of our knowledge, this is the fist algorithmic improvement developed for the recently proposed DRO approach with an event-wise ambiguity set in [116].

5. We show in our computational studies that a small number of clusters obtained via K-means can achieve high-quality robust drone delivery schedules. We validate that the adaptive DRO model can effectively reduce lateness in out-of-sample tests in comparison with other classical models.

The rest of this chapter is organized as follows. At the end of this section, we summarize the notation used throughout this chapter. Section 7.2 constructs the ambiguity set from weather data. Section 7.3 presents the decision criterion, builds the DRO model, and introduces the adaptive policy. Section 7.4 reformulates the distributionally robust model and presents the solution method. Section 7.5 reports numerical results. Section 7.6 concludes this chapter.

Notation. We denote by $[N] \triangleq \{1, \ldots, N\}$ the set of positive running indices up to N.

Boldface lowercase and uppercase characters represent vectors and matrices with appropriate dimensions, respectively. $\boldsymbol{a}'$ is the transpose of $\boldsymbol{a}$. We use $\mathcal{P}_0(\mathbb{R}^I)$ to represent the set of all distributions on $\mathbb{R}^I$. A random variable, $\tilde{\boldsymbol{z}}$ is denoted with a tilde sign and we use $\tilde{\boldsymbol{z}} \sim \mathbb{P}$, $\mathbb{P} \in \mathcal{P}_0(\mathcal{W})$, $\mathcal{W} \subseteq \mathbb{R}^I$ to define $\tilde{\boldsymbol{z}}$ as an I-dimensional random variable with distribution $\mathbb{P}$ over the support $\mathcal{W}$. We use ξ^+ to represent $\max\{\xi, 0\}$. For a vector $\boldsymbol{u} \in \mathbb{R}^I$, the expression $|\boldsymbol{u}|$ denotes the vector of absolute values of the components of $\boldsymbol{u}$. $\boldsymbol{e}$ corresponds to the vector of 1s with an appropriate dimension. $\boldsymbol{e}_i$ is the ith standard basis vector.

7.2 Weather Uncertainty and Influence on Flight

We model a drone delivery system comprising a fleet of D identical drones that are dispatched to visit a set of N, $N > D$ geographically dispersed customers. Each drone can make multiple round trips, where each round trip is a flight from the depot to the customer and back to the depot for preparation for the next delivery. We assume that the drone has sufficient energy to perform a round trip, since a depleted battery can quickly be swapped with a fully charged one. Hence, with sufficient supply of charged batteries, we do not need to incorporate charging decisions into our model. If a fully charged battery is not sufficient for a round trip between the depot and a customer site, then that customer would be deemed a drone-ineligible customer. We characterize the uncertainty of flight times as the result of wind conditions. Without loss of generality, we assume that the time for mounting new payloads and swapping batteries at the depot is negligible, as it is not influenced by wind conditions and can be incorporated into the model easily [83]. The service time at customer locations is also neglected here because, whenever necessary, we can add it to the forward or backward flight time as a constant number.

We consider a two-period model, where Period 1 and Period 2 represent the morning and the afternoon, respectively. Period 1 starts at time 0 and ends at the midday time, $\bar{\tau}^1$, while Period 2 starts at $\bar{\tau}^1$ and ends at $\bar{\tau}^2$. At the beginning of Period 1, the model optimizes the scheduling policy, which determines the drones' delivery schedules in the morning, while the schedules in the afternoon can flexibly adapt to the observed wind vector, (r^1, θ^1), in the morning, and/or other weather information that would be known by midday. We assume that the modeler has H samples of wind observation data at the beginning of the time period. Under sample h, $h \in [H]$, we define (r_h^1, θ_h^1) and (r_h^2, θ_h^2) to be the observed wind vectors in Period 1 and Period 2, respectively.

Ambiguity Set for Uncertain Flight Times

The forward and backward flight times for serving customer i, $i \in [N]$ in Period 1 are uncertain and we denote them by the random variables $\tilde{u}_i^1$ and $\tilde{v}_i^1$, which are functions of the wind vectors from Equation (7.1). Likewise, we also denote by the random variables $\tilde{u}_i^2$ and $\tilde{v}_i^2$ the forward and backward flight times for serving customer i, $i \in [N]$ in Period 2. For convenience, we define the vectorial notation $\tilde{\boldsymbol{u}}^t \triangleq (\tilde{u}_1^t, \ldots, \tilde{u}_N^t)'$ and $\tilde{\boldsymbol{v}}^t \triangleq (\tilde{v}_1^t, \ldots, \tilde{v}_N^t)'$, $t \in [2]$. Based on Equation (7.1), we also define

$$\boldsymbol{u}(r, \theta) \triangleq (u_1(r, \theta), \ldots, u_N(r, \theta))'$$

and

$$\boldsymbol{v}(r, \theta) \triangleq (v_1(r, \theta), \ldots, v_N(r, \theta))'.$$

We first characterize the ambiguity set in the first period using a cluster-wise ambiguity set introduced by [116]. We partition the wind vector chart into K^1 non-overlapping clusters $\mathcal{U}_k^1$, $k \in [K^1]$ so that the index set of the subsamples

$$\mathcal{L}_k = \{h \in [H] \mid (r_h^1, \theta_h^1) \in \mathcal{U}_k^1\}, \tag{7.2}$$

are each associated with a region to which the morning wind vectors belong (see Figure 7.3). There are various ways to partition the regions on the wind vector chart, and we will explore this further in our computational studies.

We introduce the random variable $\tilde{\kappa}^1$ taking discrete values in $[K^1]$ to represent the scenario $\tilde{\kappa}^1 = k$ associated with the cluster $\mathcal{U}_k^1$ that would contain the realized wind vector observed in Period 1. Accordingly, using wind observational data, we construct the cluster-wise ambiguity set $\mathcal{G}^1 \subseteq \mathcal{P}(\mathbb{R}^{2N} \times [K^1])$ associated with the random variable $(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\kappa}^1)$ as follows:

$$\mathcal{G}^1 = \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^{2N} \times [K^1]) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\kappa}^1) \sim \mathbb{P} & \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{u}}^1 \mid \tilde{\kappa}^1 = k] = \boldsymbol{\mu}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{v}}^1 \mid \tilde{\kappa}^1 = k] = \boldsymbol{\nu}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}[|\tilde{\boldsymbol{u}}^1 - \boldsymbol{\mu}_k^1| \mid \tilde{\kappa}^1 = k] \leq \boldsymbol{\sigma}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}[|\tilde{\boldsymbol{v}}^1 - \boldsymbol{\nu}_k^1| \mid \tilde{\kappa}^1 = k] \leq \boldsymbol{\varsigma}_k^1 & \forall k \in [K^1] \\ \mathbb{P}\left[\begin{array}{l} \underline{\boldsymbol{u}}_k^1 \leq \tilde{\boldsymbol{u}}^1 \leq \bar{\boldsymbol{u}}_k^1 \\ \underline{\boldsymbol{v}}_k^1 \leq \tilde{\boldsymbol{v}}^1 \leq \bar{\boldsymbol{v}}_k^1 \end{array} \;\middle|\; \tilde{\kappa}^1 = k \right] = 1 & \forall k \in [K^1] \\ \mathbb{P}[\tilde{\kappa}^1 = k] = |\mathcal{L}_k|/H & \forall k \in [K^1] \end{array} \right\}, \tag{7.3}$$

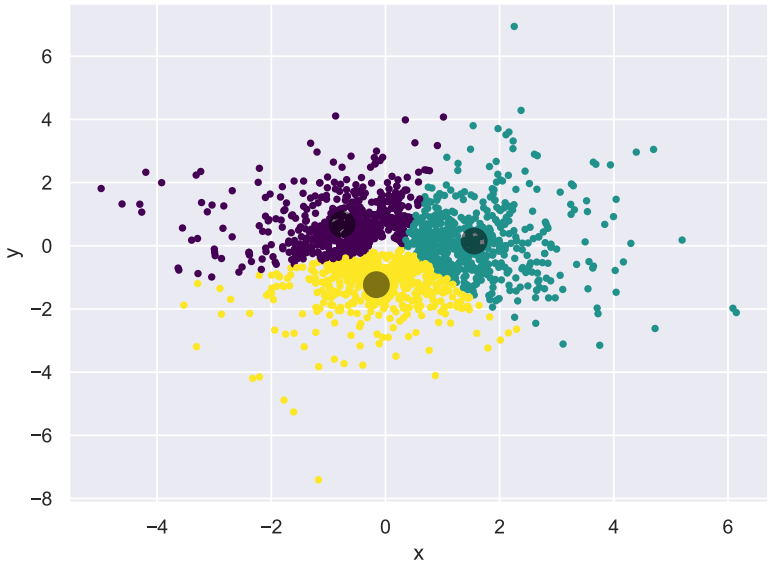


Figure 7.3 Partitioning the wind vector chart into clusters using K-means clustering algorithm. Note that wind vectors are converted from polar coordinates to Cartesian coordinates when performing the clustering operation. The dark points in each cluster are the centroids.

where, the mean flight times associated with the cluster k, $k \in [K^1]$ are

$$\boldsymbol{\mu}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \boldsymbol{u}(r_h^1, \theta_h^1), \quad \boldsymbol{\nu}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \boldsymbol{v}(r_h^1, \theta_h^1),$$

the mean absolute deviations are

$$\boldsymbol{\sigma}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \mathbb{E}_{\mathbb{P}} \left[\left| \boldsymbol{u}(r_h^1, \theta_h^1) - \boldsymbol{\mu}_k^1 \right| \right], \quad \boldsymbol{\varsigma}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \mathbb{E}_{\mathbb{P}} \left[\left| \boldsymbol{v}(r_h^1, \theta_h^1) - \boldsymbol{\nu}_k^1 \right| \right],$$

and the parameters of the supports are

$$\begin{aligned} [\underline{\boldsymbol{u}}_k^1]_i &= \min_{h \in [\mathcal{L}_k]} u_i(r_h^1, \theta_h^1), \quad [\underline{\boldsymbol{v}}_k^1]_i = \min_{h \in [\mathcal{L}_k]} v_i(r_h^1, \theta_h^1), \\ [\bar{\boldsymbol{u}}_k^1]_i &= \max_{h \in [\mathcal{L}_k]} u_i(r_h^1, \theta_h^1), \quad [\bar{\boldsymbol{v}}_k^1]_i = \max_{h \in [\mathcal{L}_k]} v_i(r_h^1, \theta_h^1). \end{aligned} \qquad \forall i \in [N].$$

In the spirit of RO, the ambiguity set is often designed to encompass distributions that may deviate from the empirical distribution, which has the benefit of mitigating the risks of overfitting that may result in poor out-of-sample performance. Observe that when $K^1 = 1$, $\mathcal{G}^1$ is reduced to a marginal moment ambiguity set which, analogous to the *box uncertainty set*, can be quite conservative when applied to solving DRO problems. As we increase the number

of clusters, the level of conservativeness reduces, though the computational complexity of the model would increase. In the extreme, each cluster can contain exactly one sample, in which case, due to the support and deviations constraint, the ambiguity set would only contain the empirical distribution. In practice, the number of clusters is determined empirically in validation tests. If the number is small, this approach would benefit from greater scalability.

By midday, in preparation for Period 2, the random variable $(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\kappa}^1)$ would be realized. Apart from the wind data, we assume that there is other weather information such as atmospheric pressures, weather forecasts, and other covariates, which could be used to predict the wind condition in the afternoon. To incorporate this information into our ambiguity set, we have to discretize the weather information into S finite scenarios, which can be done using standard classification approaches used in machine learning. Accordingly, we use $\tilde{s}$ to denote the random weather scenarios, which would be realized by midday. We assume that the weather data contains historical information of the weather scenarios, which we denote by s_h, $h \in [H]$. We define the index set of the subsamples associated with the different weather scenarios,

$$\mathcal{V}_s = \{h \in [H] \mid s_h = s\}. \tag{7.4}$$

In the absence of other weather information beyond wind observation data, we can use the random scenario $\tilde{\kappa}^1$ defined for the ambiguity set $\mathcal{G}^1$ as the random scenario for predicting wind condition in the afternoon, in which case, $S = K^1$ and $\mathcal{V}_s = \mathcal{L}_s$ for all $s \in [S]$.

Given a realized scenario $\tilde{s} = s$, we can characterize Period 2 flight times $(\tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2)$, via a cluster-wise ambiguity set as before. We partition the wind vectors (r_h^2, θ_h^2), $h \in \mathcal{V}_s$ into K^2 non-overlapping clusters $\mathcal{U}_g^2$, $g \in [K^2]$. Each cluster $g \in [K^2]$ is associated with the subset of samples

$$\mathcal{L}_{sg}^2 = \left\{h \in \mathcal{V}_s \mid (r_h^2, \theta_h^2) \in \mathcal{U}_g^2\right\}.$$

We also introduce the random variable $\tilde{\kappa}^2$, taking discrete values in $[K^2]$ to represent the scenario $\tilde{\kappa}^2 = g$ associated with the cluster $\mathcal{U}_g^2$ that would contain the realized wind vector observed in Period 2. For each scenario s, $s \in [S]$, we construct the ambiguity set $\mathcal{G}_s^2 \subseteq$

$\mathcal{P}(\mathbb{R}^{2N} \times [K^2])$ as follows:

$$\mathcal{G}_s^2 = \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^{2N} \times [K^2]) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{\kappa}^2) \sim \mathbb{P} & \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^2 \mid \tilde{\kappa}^2 = g\right] = \boldsymbol{\mu}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^2 \mid \tilde{\kappa}^2 = g\right] = \boldsymbol{\nu}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^2 - \boldsymbol{\mu}_{sg}^2| \mid \tilde{\kappa}^2 = g\right] \leq \boldsymbol{\sigma}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^2 - \boldsymbol{\nu}_{sg}^2| \mid \tilde{\kappa}^2 = g\right] \leq \boldsymbol{\varsigma}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{P}\left[\begin{array}{l} \underline{\boldsymbol{u}}_{sg}^2 \leq \tilde{\boldsymbol{u}}^2 \leq \bar{\boldsymbol{u}}_{sg}^2 \\ \underline{\boldsymbol{v}}_{sg}^2 \leq \tilde{\boldsymbol{v}}^2 \leq \bar{\boldsymbol{v}}_{sg}^2 \end{array} \middle| \tilde{\kappa}^2 = g \right] = 1 & \forall g \in [K^2] \\ \mathbb{P}\left[\tilde{\kappa}^2 = g\right] = |\mathcal{L}_{sg}^2|/|\mathcal{V}_s| & \forall g \in [K^2] \end{array} \right\}, \tag{7.5}$$

where the mean flight times associated with cluster g, $g \in [K^2]$ under scenario s, $s \in \mathcal{S}$ are

$$\boldsymbol{\mu}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}^2|} \sum_{h \in \mathcal{L}_{sg}^2} \boldsymbol{u}(r_h^2, \theta_h^2), \quad \boldsymbol{\nu}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}^2|} \sum_{h \in \mathcal{L}_{sg}^2} \boldsymbol{v}(r_h^2, \theta_h^2),$$

the mean absolute deviations are

$$\boldsymbol{\sigma}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}^2|} \sum_{h \in \mathcal{L}_{sg}^2} \mathbb{E}_{\mathbb{P}}\left[|\boldsymbol{u}(r_h^2, \theta_h^2) - \boldsymbol{\mu}_{sg}^2|\right], \quad \boldsymbol{\varsigma}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}^2|} \sum_{h \in \mathcal{L}_{sg}^2} \mathbb{E}_{\mathbb{P}}\left[|\boldsymbol{v}(r_h^2, \theta_h^2) - \boldsymbol{\nu}_{sg}^2|\right],$$

and the parameters of the supports are

$$\begin{array}{ll} \left[\underline{\boldsymbol{u}}_{sg}^2\right]_i = \min\limits_{h \in [\mathcal{L}_{sg}^2]} u_i(r_h^2, \theta_h^2), & \left[\underline{\boldsymbol{v}}_{sg}^2\right]_i = \min\limits_{h \in [\mathcal{L}_{sg}^2]} v_i(r_h^2, \theta_h^2), \\ \left[\bar{\boldsymbol{u}}_{sg}^2\right]_i = \max\limits_{h \in [\mathcal{L}_{sg}^2]} u_i(r_h^2, \theta_h^t), & \left[\bar{\boldsymbol{v}}_{sg}^2\right]_i = \max\limits_{h \in [\mathcal{L}_{sg}^2]} v_i(r_h^2, \theta_h^2). \end{array} \qquad \forall i \in [N].$$

In general, the discrete random variable $\tilde{\kappa}^1$, $\tilde{\kappa}^2$ and $\tilde{s}$ are dependent and hence we denote $\mathcal{W}$ as the support of the joint discrete random variable $(\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s})$. We are now ready to characterize the full ambiguity set. The actual joint distribution of $(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \sim \mathbb{P}$ is unknown but belongs to an ambiguity set, $\mathcal{F}$, as follows:

$$\mathcal{F} = \left\{ \mathbb{Q} \in \mathcal{P}(\mathbb{R}^{4N} \times [S]) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \sim \mathbb{Q} & \\ \exists \mathbb{P} \in \mathcal{P}(\mathbb{R}^{4N} \times \mathcal{W}) : & \\ \quad (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, (\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s})) \sim \mathbb{P} & \\ \exists \mathbb{Q}^1 \in \mathcal{G}^1 : (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\kappa}^1) \sim \mathbb{Q}^1 & \\ \exists \mathbb{Q}_s^2 \in \mathcal{G}_s^2 : (\tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{\kappa}^2)|_{\tilde{s}=s} \sim \mathbb{Q}_s^2 & \forall s \in [S] \\ \mathbb{P}\left[(\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s}) = (k, g, s)\right] = q_{kgs} & \forall (k, g, s) \in \mathcal{W} \end{array} \right\}, \tag{7.6}$$

where

$$q_{kgs} = \frac{|\mathcal{V}_s \cap \mathcal{L}_k \cap \mathcal{L}^2_{sg}|}{H},$$

or explicitly as

$$\mathcal{F} = \left\{ \mathbb{Q} \in \mathcal{P}(\mathbb{R}^{4N} \times [S]) \;\middle|\; \begin{array}{ll}
(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \sim \mathbb{Q} & \\
\exists \mathbb{P} \in \mathcal{P}(\mathbb{R}^{4N} \times \mathcal{W}) : & \\
\quad (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, (\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s})) \sim \mathbb{P} & \\
\mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^1 \mid \tilde{\kappa}^1 = k\right] = \boldsymbol{\mu}^1_k & \forall k \in [K^1] \\
\mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^1 \mid \tilde{\kappa}^1 = k\right] = \boldsymbol{\nu}^1_k & \forall k \in [K^1] \\
\mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^1 - \boldsymbol{\mu}^1_k| \mid \tilde{\kappa}^1 = k\right] \leq \boldsymbol{\sigma}^1_k & \forall k \in [K^1] \\
\mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^1 - \boldsymbol{\nu}^1_k| \mid \tilde{\kappa}^1 = k\right] \leq \boldsymbol{\varsigma}^1_k & \forall k \in [K^1] \\
\mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^2 \mid \tilde{s} = s, \tilde{\kappa}^2 = g\right] = \boldsymbol{\mu}^2_{sg} & \forall s \in [S], g \in [K^2] \\
\mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^2 \mid \tilde{s} = s, \tilde{\kappa}^2 = g\right] = \boldsymbol{\nu}^2_{sg} & \forall s \in [S], g \in [K^2] \\
\mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^2 - \boldsymbol{\mu}^2_{sg}| \mid \tilde{s} = s, \tilde{\kappa}^2 = g\right] \leq \boldsymbol{\sigma}^2_{sg} & \forall s \in [S], g \in [K^2] \\
\mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^2 - \boldsymbol{\nu}^2_{sg}| \mid \tilde{s} = s, \tilde{\kappa}^2 = g\right] \leq \boldsymbol{\varsigma}^2_{sg} & \forall s \in [S], g \in [K^2] \\
\mathbb{P}\left[\begin{array}{l|l} \underline{\boldsymbol{u}}^1_k \leq \tilde{\boldsymbol{u}}^1 \leq \bar{\boldsymbol{u}}^1_k & \tilde{\kappa}^1 = k, \\ \underline{\boldsymbol{v}}^1_k \leq \tilde{\boldsymbol{v}}^1 \leq \bar{\boldsymbol{v}}^1_k & \tilde{\kappa}^2 = g, \\ \underline{\boldsymbol{u}}^2_{sg} \leq \tilde{\boldsymbol{u}}^2 \leq \bar{\boldsymbol{u}}^2_{sg} & \tilde{s} = s \\ \underline{\boldsymbol{v}}^2_{sg} \leq \tilde{\boldsymbol{v}}^2 \leq \bar{\boldsymbol{v}}^2_{sg} & \end{array} \right] = 1 & \forall (k,g,s) \in \mathcal{W} \\
\mathbb{P}\left[(\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s}) = (k,g,s)\right] = q_{kgs} & \forall (k,g,s) \in \mathcal{W}
\end{array} \right\}. \tag{7.7}$$

7.3 The Drone Delivery Model

Our goal is to robustly optimize the schedules of the drone delivery system to mitigate the risks of delivery delays caused by wind uncertainty. Customers specify their preference for delivery in either the morning or in the afternoon. Hence, we partition the customers into three groups, $\mathcal{C}^1$, $\mathcal{C}^2$, and $\mathcal{C}^3$. The first two groups, $\mathcal{C}^1$ and $\mathcal{C}^2$ are the sets of customers that expect to have their delivery made in Period 1 and Period 2 respectively. The third group $\mathcal{C}^3$ is the set of unconstrained customers who can be served in either period. Customers in $\mathcal{C}^3$ are notified to expect their delivery in either Period 1 or Period 2. Hence, customers assigned to be served in Period 1 should be served within $[0, \bar{\tau}^1]$, while customers assigned to be served in Period 2 should be served within $[\bar{\tau}^1, \bar{\tau}^2]$. Poor service occurs whenever the scheduled delivery time windows could not be met.

Decision Criterion

For a given delivery policy, the arriving time of a drone is a function of the random variable $(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \sim \mathbb{P}$, $\mathbb{P} \in \mathcal{F}$, which we denote by $\xi(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s})$ for some function $\xi : \mathbb{R}^{4N} \times [S] \mapsto \mathbb{R}$. For convenience, we use $\tilde{\xi} \triangleq \xi(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s})$ to represent the random arriving time. A service is delayed whenever $\tilde{\xi} > \tau$, where $\tau \in \{\bar{\tau}^1, \bar{\tau}^2\}$ is the delivery time expected by the customer.

Our goal is to mitigate the risks of service delays under distributional ambiguity, and we evaluate such risks using the *essential riskiness index* (ERI) proposed by [146]. The index is endowed with good computational properties and it performs well as an optimization criterion in a vehicle routing problem, to help meet delivery deadlines by limiting the probability of tardy delivery and the magnitude of lateness [147].

Definition 7.3.1 *Given an arriving time function, $\xi : \mathbb{R}^{4N} \times [S] \mapsto \mathbb{R}$ and the expected delivery time, $\tau \in \mathbb{R}_+$. The essential riskiness index is defined as*

$$\rho_\tau(\tilde{\xi}) = \min \left\{ \gamma \geq 0 \;\middle|\; \sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} \left[\max \left\{ \tilde{\xi} - \tau, -\gamma \right\} \right] \leq 0 \right\}, \tag{7.8}$$

where $\min \emptyset = \infty$ *by convention.*

We briefly present some of the important properties as ERI.

i) **Monotonicity**: If $\mathbb{P}\left[\tilde{\xi}_1 \geq \tilde{\xi}_2\right] = 1$, $\forall \mathbb{P} \in \mathcal{F}$, then $\rho_\tau(\tilde{\xi}_1) \geq \rho_\tau(\tilde{\xi}_2)$;

ii) **Satisficing**: $\rho_\tau(\tilde{\xi}) = 0$ if and only if $\mathbb{P}\left[\tilde{\xi} \leq \tau\right] = 1$, $\forall \mathbb{P} \in \mathcal{F}$;

iii) **Infeasibility**: If $\sup\limits_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\tilde{\xi}\right] > \tau$, then $\rho_\tau(\tilde{\xi}) = \infty$;

iv) **Convexity**: For all $\lambda \in [0, 1]$, $\rho_\tau(\lambda\tilde{\xi}_1 + (1-\lambda)\tilde{\xi}_2) \leq \lambda\rho_\tau(\tilde{\xi}_1) + (1-\lambda)\rho_\tau(\tilde{\xi}_2)$;

v) **Delay bounds**:

$$\mathbb{P}\left[\tilde{\xi} - \tau > \rho_\tau(\tilde{\xi})\vartheta\right] \leq \frac{1}{1+\vartheta}, \quad \forall \vartheta > 0, \mathbb{P} \in \mathcal{F}.$$

Monotonicity ensures that an uncertain arriving time that dominates another would not be better off under ERI. Satisficing ensures that if the delivery time can always be met, then the risk of poor service under ERI is zero. The infeasibility property ensures that the average arriving time, even under distributional ambiguity, should not exceed the expected delivery time. The property of convexity provides a more tractable formulation compared to non-convex ones, such as the criterion based on the probability of service delays. Finally, as

shown in the property of delay bounds, the ERI ensures that a bound on the probability of lateness reduces reciprocally as the magnitude of lateness increases in multiples of the index value. Thus, unlike the probability of service delay, the ERI accounts for both the probability of delay and its magnitude. ERI has other useful and interesting properties and we refer interested readers to [146] and [147].

The drone delivery problem can be modeled as a multi-trip VRPTW proposed in [147], where the objective of the model is to minimize the sum of the ERIs associated with service delay risks for all customers. Accordingly, to ensure high quality of services across all customers, the drone delivery system could also minimize the following joint decision criterion

$$\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) \triangleq \sum_{i\in[N]} \rho_{\tau_i}(\tilde{\xi}_i),$$

where $\tilde{\xi}_i$ and τ_i are, respectively, the uncertain arriving time and expected delivery time for the ith customer. The benefit of this criterion in achieving high-quality solutions with reasonable computational effort have been demonstrated in [147]. Moreover, the joint decision criterion satisfies the following salient properties:

i) **Monotonicity**: If $\mathbb{P}\left[\tilde{\boldsymbol{\xi}}_1 \geq \tilde{\boldsymbol{\xi}}_2\right] = 1$, $\forall \mathbb{P} \in \mathcal{F}$, then $\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}_1) \geq \varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}_2)$;

ii) **Satisficing**: $\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) = 0$ if and only if $\mathbb{P}\left[\tilde{\boldsymbol{\xi}} \leq \boldsymbol{\tau}\right] = 1$, $\forall \mathbb{P} \in \mathcal{F}$;

iii) **Non-abandonment**: If there exists $i \in [N]$ such that $\sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\tilde{\xi}_i\right] > \tau_i$, then $\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) = \infty$;

iv) **Convexity**: For all $\lambda \in [0,1]$, $\varrho_{\boldsymbol{\tau}}(\lambda\tilde{\boldsymbol{\xi}}_1 + (1-\lambda)\tilde{\boldsymbol{\xi}}_2) \leq \lambda\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}_1) + (1-\lambda)\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}_2)$.

Because we are using a nonstandard decision criterion, these properties serve as an axiomatic framework for which our decision criterion would be justified. In our opinion, these properties are reasonable for any joint decision criterion that evaluates the overall customer service. In particular, the non-abandonment property ensures that any feasible solution for which the objective value is finite implies that the average arriving time at every customer's location would not exceed the expected delivery time. We are cognizant of other joint decision criteria that would also preserve the same properties, such as the one that takes the value of the highest ERI among all customers. However, from our numerical experience, minimizing the maximum ERI results in having an excess of multiple optimal solutions, which has an impact on the quality of the solutions as well as the computational time to solve the model. We will illustrate this in our computational studies.

While it may be possible to cast our problem as a static robust VPRTW proposed in [147], our goal is to extend to an adaptive model, where it is possible to improve the drone delivery

schedule in response to the new information available by midday. The static robust VPRTW of [147] would be far too complex for our purpose. One simplifying assumption that enables us to take this approach is that customers can only choose between two delivery deadlines, $\{\bar{\tau}^1, \bar{\tau}^2\}$. Alongside this, we propose the following joint decision criterion:

$$\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) \triangleq \sum_{d\in[D]:\mathcal{K}_d^1\neq\emptyset} \left(\max_{i\in\mathcal{K}_d^1}\left\{\rho_{\bar{\tau}^1}\left(\tilde{\xi}_i\right)\right\}\right) + \sum_{d\in[D]:\mathcal{K}_d^2\neq\emptyset} \left(\max_{i\in\mathcal{K}_d^2}\left\{\rho_{\bar{\tau}^2}\left(\tilde{\xi}_i\right)\right\}\right), \tag{7.9}$$

where $\mathcal{K}_d^t \subseteq [N]$ denotes the set of customers assigned to be served by drone d, $d \in [D]$ in period t, $t \in [2]$. Observe that the joint decision criterion also satisfies the properties of monotonicity, satisficing, non-abandonment, and convexity.

The main reason for our proposed joint decision criterion is the potential to have a simpler formulation that is more scalable computationally, as we will demonstrate in the following proposition.

Proposition 7.3.1 *The joint decision criterion $\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}})$ is equivalent to*

$$\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) = \sum_{d\in[D]:\mathcal{K}_d^1\neq\emptyset} \rho_{\bar{\tau}^1}\left(\tilde{\xi}_{\ell_d^1}\right) + \sum_{d\in[D]:K_d^2\neq\emptyset} \rho_{\bar{\tau}^2}\left(\tilde{\xi}_{\ell_d^2}\right),$$

where ℓ_d^t denotes the last customer served by drone d, $d \in D$ in Period t, $t \in [2]$.

Proof. For each drone d, $d \in [D]$, in each period the arriving time at the last assigned customer is always larger than the arrival times at other assigned customers, i.e.,

$$\tilde{\xi}_{\ell_d^1} \geq \tilde{\xi}_i \qquad \forall i \in \mathcal{K}_d^1, \qquad\qquad \tilde{\xi}_{\ell_d^2} \geq \tilde{\xi}_i \qquad \forall i \in \mathcal{K}_d^2.$$

Based on the monotonicity property of $\varrho_\tau(\tilde{\xi})$, we have

$$\rho_{\bar{\tau}^1}\left(\tilde{\xi}_{\ell_d^1}\right) \geq \rho_{\bar{\tau}^1}\left(\tilde{\xi}_i\right) \qquad \forall i \in \mathcal{K}_d^1, \qquad \rho_{\bar{\tau}^2}\left(\tilde{\xi}_{\ell_d^2}\right) \geq \rho_{\bar{\tau}^2}\left(\tilde{\xi}_i\right) \qquad \forall i \in \mathcal{K}_d^2,$$

which indicate that

$$\rho_{\bar{\tau}^1}\left(\tilde{\xi}_{\ell_d^1}\right) = \max_{i\in\mathcal{K}_d^1}\left\{\rho_{\tau_1}(\tilde{\xi}_i)\right\}, \qquad \rho_{\bar{\tau}^2}\left(\tilde{\xi}_{\ell_d^2}\right) = \max_{i\in\mathcal{K}_d^2}\left\{\rho_{\tau_2}(\tilde{\xi}_i)\right\}.$$

Observe that the random arrival time at the last customer by drone d, $d \in [D]$ in Period 1 is

$$\tilde{\xi}_{\ell_d^1} = \sum_{i\in\mathcal{K}_d^1\setminus\{\ell_d^1\}} (\tilde{u}_i^1 + \tilde{v}_i^1) + \tilde{u}_{\ell_d^1}^1. \tag{7.10}$$

Note that the arrival time in Period 2 is not an additive function of the flight times because customers in Period 2 can only be served after $\bar{\tau}^1$. We assume that whenever the anticipated arrival is before $\bar{\tau}^1$, the drone is able to reduce its speed or delay its departure from the depot so that the arrival would be punctual at $\bar{\tau}^1$. We also need to account for the first customer to be served in Period 2 by drone d, $d \in [D]$, which we denote by $\underline{\ell}_d^2$. Hence, the random arrival time at the last customer by drone d, $d \in [D]$ in Period 2 is

$$\tilde{\xi}_{\ell_d^2} = \max\left\{\sum_{i\in\mathcal{K}_d^1}(\tilde{u}_i^1 + \tilde{v}_i^1) + \tilde{u}_{\underline{\ell}_d^2}^2, \bar{\tau}^1\right\} + \tilde{v}_{\underline{\ell}_d^2}^2 + \sum_{i\in\mathcal{K}_d^2\setminus\left\{\underline{\ell}_d^2,\ell_d^2\right\}}\left(\tilde{u}_i^2 + \tilde{v}_i^2\right) + \tilde{u}_{\ell_d^2}^2 \qquad \forall d \in [D]. \quad (7.11)$$

Under our proposed joint decision criterion, Proposition 7.3.1 implies that we only need to keep track of, for every drone, the last customer to be served in the morning, and the first and last customers served in the afternoon by the drone. In particular, the sequence of customers being served before the last customer in Period 1, and the sequence of customers being served between the first and the last customer in Period 2, would not affect the joint decision criterion. This greatly reduces the complexity of the model compared to a VRPTW formulation, in terms of the number of binary variables needed to model the problem.

Scheduling Decisions and Event-wise Adaptations

To obtain an explicit mathematical optimization model, we first denote $\bar{\mathcal{C}}^1 = \mathcal{C}^1 \cup \mathcal{C}^3$ and $\bar{\mathcal{C}}^2 = \mathcal{C}^2 \cup \mathcal{C}^3$ as the sets of customers who can expect their deliveries in the morning and afternoon, respectively. For scheduling decisions (y_{id}^1, x_{id}^1), $i \in \bar{\mathcal{C}}^1$, $d \in [D]$ associated with Period 1, we define the binary variable $y_{id}^1 = 1$ if and only if customer i is the last customer visited by drone d in Period 1, i.e., $i \in \mathcal{K}_d^1 \cap \{\ell_d^1\}$, and the binary variable $x_{id}^1 = 1$ if and only if $i \in \mathcal{K}_d^1 \setminus \{\ell_d^1\}$.

As we have mentioned, the drone schedule in the afternoon can flexibly adapt to wind information, which is available by midday when the random scenario $\tilde{s}$ is realized. Hence, the decisions associated with Period 2 are functions of $s \in [S]$, which we denote by $y_{id}^2(s), x_{id}^2(s), z_{id}^2(s)$, $i \in \bar{\mathcal{C}}^2$, $d \in [D]$. For a given scenario $s \in [S]$, the binary variable $y_{id}^2(s) = 1$ if and only if $i \in \mathcal{K}_d^2 \cap \{\ell_d^2\}$, the binary variable $x_{id}^2(s) = 1$ if and only if $i \in \mathcal{K}_d^2 \setminus \{\underline{\ell}_d^2, \ell_d^2\}$ and the binary variable $z_{id}^2(s) = 1$ if and only if $i \in \mathcal{K}_d^2 \cap \{\underline{\ell}_d^2\}$.

The adaptive scheduling decisions in Period 2 can be defined within a family of binary function maps. For flexibility in adaptation, we adopt the event-wise adaptation introduced in [116]. We first define an event $\mathcal{E} \subseteq [S]$ by a subset of scenarios. A partition of scenarios then induces a collection $\mathcal{S}$ of mutually exclusive and collectively exhaustive (MECE) events.

Correspondingly, we define a mapping $f_{\mathcal{S}} : [S] \mapsto \mathcal{S}$ such that $f_{\mathcal{S}}(s) = \mathcal{E}$, for which $\mathcal{E}$ is the only event in $\mathcal{S}$ that contains the scenario s. Given a collection $\mathcal{S}$ of MECE events, we define the event-wise adaptation to characterize the scheduling decisions in Period 2 as follows:

$$\mathcal{A}(\mathcal{S}) \triangleq \left\{ \boldsymbol{x} : [S] \mapsto \{0,1\}^{\bar{\mathcal{C}}^2 \times D} \;\middle|\; \begin{array}{l} \boldsymbol{x}(s) = \boldsymbol{x}^{\mathcal{E}},\ \mathcal{E} = f_{\mathcal{S}}(s) \\ \text{for some } \boldsymbol{x}^{\mathcal{E}} \in \{0,1\}^{\bar{\mathcal{C}}^2 \times D} \end{array} \right\}.$$

In the case of a non-adaptive (or static) policy, we have $\mathcal{S} = \{[S]\}$, so that the scheduling decisions in Period 2 would not change its solutions in response to the outcomes of the scenario $\tilde{s}$. For the case of full adaption, the collection $\mathcal{S}$ would have S different events, each containing an element of $[S]$.

The feasible drone delivery schedule is as follows:

$$\mathcal{X} = \left\{ (\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2) \;\middle|\; \begin{array}{ll} \displaystyle\sum_{d\in[D]} (x^1_{id} + y^1_{id}) = 1 & \forall i \in \mathcal{C}^1, \quad (7.12a) \\ \displaystyle\sum_{d\in[D]} \left(x^2_{id}(s) + y^2_{id}(s) + z^2_{id}(s)\right) = 1 & \forall i \in \mathcal{C}^2, s \in [S], \quad (7.12b) \\ \displaystyle\sum_{d\in[D]} \left(x^1_{id} + y^1_{id} + x^2_{id}(s) + y^2_{id}(s) + z^2_{id}(s)\right) = 1 & \forall i \in \mathcal{C}^3, s \in [S], \quad (7.12c) \\ \displaystyle\sum_{i\in\bar{\mathcal{C}}^1} y^1_{id} = 1 & \forall d \in [D], \quad (7.12d) \\ \displaystyle\sum_{i\in\bar{\mathcal{C}}^2} y^2_{id}(s) = 1 & \forall d \in [D], s \in [S], \quad (7.12e) \\ \displaystyle\sum_{i\in\bar{\mathcal{C}}^2} z^2_{id}(s) = 1 & \forall d \in [D], s \in [S], \quad (7.12f) \\ \boldsymbol{x}^1, \boldsymbol{y}^1 \in \{0,1\}^{\bar{\mathcal{C}}^1 \times D} & \\ \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2 \in \mathcal{A}(\mathcal{S}). & \end{array} \right\}$$

Constraints (7.12a)–(7.12c) mean that customers in each set are respectively scheduled to be visited exactly once in their allowable delivery time slots. Constraints (7.12d)–(7.12e) impose that each drone at each period can only have one customer as the last served one. Constraints (7.12f) mean that for each drone in Period 2, it can only have one customer as the first visited customer. The index s in the constraints suggests that the afternoon schedule should be feasible under any scenario s, $s \in [S]$.

The schedule decision can be enhanced by the following lexicographic ordering constraints,

which break the symmetry between drones [148]

$$\sum_{i=1,i\in\mathcal{C}^1}^{j} 2^{j-i}x^1_{id} \geq \sum_{i=1,i\in\mathcal{C}^1}^{j} 2^{j-i}x^1_{id+1} \qquad \forall j \in \mathcal{C}^1, d \in [D-1]. \tag{7.13}$$

Now, for a given d, $d \in [D]$, the arrival time at the last customer in Period 1 can be written as the function

$$\zeta^1_d(\boldsymbol{x}^1, \boldsymbol{y}^1, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1) \triangleq \sum_{i\in\bar{\mathcal{C}}^1} \left(x^1_{id}(\tilde{u}^1_i + \tilde{v}^1_i) + y^1_{id}\tilde{u}^1_i\right), \tag{7.14}$$

and likewise the arrival time at the last customer in Period 2 can be expressed as

$$\begin{aligned} & \zeta^2_d(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \\ \triangleq \quad & \max\left\{\sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id} + y^1_{id})(\tilde{u}^1_i + \tilde{v}^1_i) + \sum_{i\in\bar{\mathcal{C}}^2} z^2_{id}(\tilde{s})\tilde{u}^2_i, \bar{\tau}^1\right\} \\ & \quad + \sum_{i\in\bar{\mathcal{C}}^2}\left(z^2_{id}(\tilde{s})\tilde{v}^2_i + x^2_{id}(\tilde{s})(\tilde{u}^2_i + \tilde{v}^2_i) + y^2_{id}(\tilde{s})\tilde{u}^2_i\right). \end{aligned} \tag{7.15}$$

Thus, under our proposed decision criterion, we solve the following DRO problem:

$$\begin{aligned} \inf \ & \sum_{d\in[D]} (\gamma^1_d + \gamma^2_d) \\ \text{s.t.} \ & \sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\max\left\{\zeta^1_d(\boldsymbol{x}^1, \boldsymbol{y}^1, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1) - \bar{\tau}^1, -\gamma^1_d\right\}\right] \leq 0 && \forall d \in [D] \quad (7.16a) \\ & \sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\max\left\{\zeta^2_d(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) - \bar{\tau}^2, -\gamma^2_d\right\}\right] \leq 0 && \forall d \in [D] \quad (7.16b) \\ & \gamma^1_d, \gamma^2_d \geq 0 && \forall d \in [D], \\ & (\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2) \in \mathcal{X}. \end{aligned}$$

The DRO model (7.16) with cluster-wise moment-based ambiguity set and event-wise adaption fits into the *robust stochastic optimization* framework recently introduced by [116]. They also provide an algebraic modeling toolbox, *RSOME - Robust Stochastic Optimization Made Easy*, which facilitates rapid prototyping of our model. Specifically, by using affine recourse adaptation on a lifted ambiguity set, we can model the problem intuitively via RSOME. However, while it is convenient to do so, the MATLAB-based toolbox generates numerous auxiliary variables and results in slow computational performance. Our goal is to eradicate all of the auxiliary variables and propose a B&C approach to solve our problem. We use RSOME as a reference to check the correctness of our approach.

7.4 Solving via Branch-and-Cut

For greater scalability, we present a classical RO reformulation, which would enable us to develop a B&C method to solve the model. We first address the supremum over the ambiguity set in the constraints (7.16a).

Note that $\sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\max\left\{\zeta_d^1(\boldsymbol{x}^1,\boldsymbol{y}^1,\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1)-\bar{\tau}^1,-\gamma_d^1\right\}\right]$ is equivalent to

$$\sup_{\mathbb{P}\in\mathcal{G}^1} \mathbb{E}_{\mathbb{P}}\left[\max\left\{\zeta_d^1(\boldsymbol{x}^1,\boldsymbol{y}^1,\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1)-\bar{\tau}^1,-\gamma_d^1\right\}\right].$$

Under the law of total probability, constraints (7.16a) can be reformulated as

$$\sum_{k\in[K^1]} q_k \sup_{\mathbb{P}\in\mathcal{G}_k^1} \mathbb{E}_{\mathbb{P}}\left[\max\left\{\sum_{i\in\bar{\mathcal{C}}^1}\left(x_{id}^1(\tilde{u}_i^1+\tilde{v}_i^1)+y_{id}^1\tilde{u}_i^1\right)-\bar{\tau}^1,-\gamma_d^1\right\}\right]\leq 0 \qquad \forall d\in[D]$$

where $q_k=|\mathcal{L}_k|/H$,

$$\mathcal{G}_k^1=\left\{\mathbb{Q}\in\mathcal{P}(\mathbb{R}^{4N}) \;\middle|\; \begin{array}{l}(\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{a}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{b}}^1)\sim\mathbb{P}\\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{u}}^1]=\boldsymbol{\mu}_k^1\\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{v}}^1]=\boldsymbol{\nu}_k^1\\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{a}}^1]\leq\boldsymbol{\sigma}_k^1\\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{b}}^1\right]\leq\boldsymbol{\varsigma}_k^1\\ \mathbb{P}\left[(\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{a}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{b}}^1)\in\Xi_k^1\right]=1\end{array}\right\},$$

and

$$\Xi_k^1\triangleq\left\{(\boldsymbol{u}^1,\boldsymbol{a}^1,\boldsymbol{v}^1,\boldsymbol{b}^1)\in\mathbb{R}^{4N} \;\middle|\; \begin{array}{l}\underline{\boldsymbol{u}}_k^1\leq\boldsymbol{u}^1\leq\bar{\boldsymbol{u}}_k^1\\ \boldsymbol{a}^1\geq|\boldsymbol{u}^1-\boldsymbol{\mu}_k^1|\\ \underline{\boldsymbol{v}}_k^1\leq\boldsymbol{v}^1\leq\bar{\boldsymbol{v}}_k^1\\ \boldsymbol{b}^1\geq|\boldsymbol{v}^1-\boldsymbol{\nu}_k^1|\end{array}\right\}.$$

Note here that we adopt the "lifted ambiguity set" first proposed by [149] to keep the expectation constraints within the ambiguity set linear.

Theorem 7.4.1 *The optimization problem*

$$\sup_{\mathbb{P}\in\mathcal{G}_k^1} \mathbb{E}_{\mathbb{P}}\left[\max\left\{\sum_{i\in\bar{\mathcal{C}}^1}\left(x_{id}^1(\tilde{u}_i^1+\tilde{v}_i^1)+y_{id}^1\tilde{u}_i^1\right)-\bar{\tau}^1,-\gamma_d^1\right\}\right] \tag{7.17}$$

is equivalent to

$$\max_{(\boldsymbol{u}_1^1,\boldsymbol{v}_1^1,p_1,p_2)\in\mathcal{Y}_{kd}^1}\left\{\left(\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1+y_{id}^1)\boldsymbol{e}_i\right)'\boldsymbol{u}_1^1+\left(\sum_{i\in\bar{\mathcal{C}}^1}x_{id}^1\boldsymbol{e}_i\right)'\boldsymbol{v}_1^1-\bar{\tau}^1p_1-\gamma_d^1p_2\right\},\tag{7.18}$$

where

$$\mathcal{Y}_{kd}^1=\left\{(\boldsymbol{u}_1^1,\boldsymbol{v}_1^1,p_1,p_2)\;\middle|\;\begin{array}{ll}\exists\boldsymbol{a}_j^1,\boldsymbol{b}_j^1,j\in[2]\;: & \\ \boldsymbol{u}_1^1+\boldsymbol{u}_2^1=\boldsymbol{\mu}_k^1 & \\ \boldsymbol{v}_1^1+\boldsymbol{v}_2^1=\boldsymbol{\nu}_k^1 & \\ \boldsymbol{a}_1^1+\boldsymbol{a}_2^1\leq\boldsymbol{\sigma}_k^1 & \\ \boldsymbol{b}_1^1+\boldsymbol{b}_2^1\leq\boldsymbol{\varsigma}_k^1 & \\ p_1+p_2=1 & \\ \underline{\boldsymbol{u}}_k^1p_j\leq\boldsymbol{u}_j^1\leq\bar{\boldsymbol{u}}_k^1p_j & \forall j\in[2]\\ \boldsymbol{a}_j^1\geq|\boldsymbol{u}_j^1-\boldsymbol{\mu}_k^1p_j| & \forall j\in[2]\\ \underline{\boldsymbol{v}}_k^1p_j\leq\boldsymbol{v}_j^1\leq\bar{\boldsymbol{v}}_k^1p_j & \forall j\in[2]\\ \boldsymbol{b}_j^1\geq|\boldsymbol{v}_j^1-\boldsymbol{\nu}_k^1p_j| & \forall j\in[2]\\ p_1,p_2\geq0 & \\ \boldsymbol{u}_j^1,\boldsymbol{a}_j^1,\boldsymbol{v}_j^1,\boldsymbol{b}_j^1\in\mathbb{R}^N & \forall j\in[2],\end{array}\right\}.$$

Proof. See Appendix E.

To address the supremum over the ambiguity set in the constraints (7.16b), we also note that under the law of total probability, constraints (7.16b) is equivalent to

$$\begin{aligned}\sum_{(k,g,s)\in\mathcal{W}}q_{kgs}\sup_{\mathbb{P}\in\mathcal{F}_{kgs}}\mathbb{E}_{\mathbb{P}}\Bigg[\max\Bigg\{&\max\Bigg\{\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1+y_{id}^1)(\tilde{u}_i^1+\tilde{v}_i^1)+\sum_{i\in\bar{\mathcal{C}}^2}z_{id}^2(s)\tilde{u}_i^2,\bar{\tau}^1\Bigg\}\\&+\sum_{i\in\bar{\mathcal{C}}^2}\left(z_{id}^2(s)\tilde{v}_i^2+x_{id}^2(s)(\tilde{u}_i^2+\tilde{v}_i^2)+y_{id}^2(s)\tilde{u}_i^2\right)-\bar{\tau}^2,-\gamma_d^2\Bigg\}\Bigg]\leq0\qquad\forall d\in[D]\end{aligned}\tag{7.19}$$

where

$$\mathcal{F}_{kgs} = \left\{ \mathbb{Q} \in \mathcal{P}(\mathbb{R}^{8N}) \;\middle|\; \begin{array}{l} (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{a}}^1, \tilde{\boldsymbol{b}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{\boldsymbol{a}}^2, \tilde{\boldsymbol{b}}^2) \sim \mathbb{P} \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^1\right] = \boldsymbol{\mu}_k^1 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^1\right] = \boldsymbol{\nu}_k^1 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{a}}^1\right] \leq \boldsymbol{\sigma}_k^1 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{b}}^1\right] \leq \boldsymbol{\varsigma}_k^1 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^2\right] = \boldsymbol{\mu}_{sg}^2 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^2\right] = \boldsymbol{\nu}_{sg}^2 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{a}}^2\right] \leq \boldsymbol{\sigma}_{sg}^2 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{b}}^2\right] \leq \boldsymbol{\varsigma}_{sg}^2 \\ \mathbb{P}\left[(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{a}}^1, \tilde{\boldsymbol{b}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{\boldsymbol{a}}^2, \tilde{\boldsymbol{b}}^2) \in \Xi_{kgs}^2\right] = 1 \end{array} \right\},$$

and

$$\Xi_{kgs}^2 = \left\{ (\boldsymbol{u}^1, \boldsymbol{v}^1, \boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{u}^2, \boldsymbol{v}^2, \boldsymbol{a}^2, \boldsymbol{b}^2) \in \mathbb{R}^{8N} \;\middle|\; \begin{array}{ll} \underline{\boldsymbol{u}}_k^1 \leq \boldsymbol{u}^1 \leq \bar{\boldsymbol{u}}_k^1, & \underline{\boldsymbol{v}}_k^1 \leq \boldsymbol{v}^1 \leq \bar{\boldsymbol{v}}_k^1 \\ \boldsymbol{a}^1 \geq |\boldsymbol{u}^1 - \boldsymbol{\mu}_k^1|, & \boldsymbol{b}^1 \geq |\boldsymbol{v}^1 - \boldsymbol{\nu}_k^1| \\ \underline{\boldsymbol{u}}_{sg}^2 \leq \boldsymbol{u}^2 \leq \bar{\boldsymbol{u}}_{sg}^2, & \underline{\boldsymbol{v}}_{sg}^2 \leq \boldsymbol{v}^2 \leq \bar{\boldsymbol{v}}_{sg}^2 \\ \boldsymbol{a}^2 \geq |\boldsymbol{u}^2 - \boldsymbol{\mu}_{sg}^2|, & \boldsymbol{b}^2 \geq |\boldsymbol{v}^2 - \boldsymbol{\nu}_{sg}^2| \end{array} \right\}.$$

Theorem 7.4.2 *The optimization problem,*

$$\begin{aligned} \sup_{\mathbb{P} \in \mathcal{F}_{kgs}} \mathbb{E}_{\mathbb{P}} \Bigg[\max \Bigg\{ & \max \left\{ \sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)(\tilde{u}_i^1 + \tilde{v}_i^1) + \sum_{i \in \bar{\mathcal{C}}^2} z_{id}^2(s) \tilde{u}_i^2, \bar{\tau}^1 \right\} \\ & + \sum_{i \in \bar{\mathcal{C}}^2} (z_{id}^2(s)\tilde{v}_i^2 + x_{id}^2(s)(\tilde{u}_i^2 + \tilde{v}_i^2) + y_{id}^2(s)\tilde{u}_i^2) - \bar{\tau}^2, -\gamma_d^2 \Bigg\} \Bigg], \end{aligned} \tag{7.20}$$

is equivalent to

$$\begin{aligned} \max_{(\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, \boldsymbol{u}_1^2, \boldsymbol{v}_1^2, \boldsymbol{u}_2^2, \boldsymbol{v}_2^2, p_1, p_2, p_3) \in \mathcal{Y}_{kgsd}^2} & \left\{ \left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \right)' (\boldsymbol{u}_1^1 + \boldsymbol{v}_1^1) + \left(\sum_{i \in \bar{\mathcal{C}}^2} (z_{id}^2(s) + x_{id}^2(s)) \boldsymbol{e}_i \right)' (\boldsymbol{u}_1^2 + \boldsymbol{v}_1^2) \right. \\ & + \left(\sum_{i \in \bar{\mathcal{C}}^2} y_{id}^2(s) \boldsymbol{e}_i \right)' \boldsymbol{u}_1^2 - \bar{\tau}^2 p_1 + \bar{\tau}^1 p_2 \\ & + \left(\sum_{i \in \bar{\mathcal{C}}^2} z_{id}^2(s) \boldsymbol{e}_i \right)' \boldsymbol{v}_2^2 + \left(\sum_{i \in \bar{\mathcal{C}}^2} x_{id}^2(s) \boldsymbol{e}_i \right)' (\boldsymbol{u}_2^2 + \boldsymbol{v}_2^2) \\ & \left. + \left(\sum_{i \in \bar{\mathcal{C}}^2} y_{id}^2(s) \boldsymbol{e}_i \right)' \boldsymbol{u}_2^2 - \bar{\tau}^2 p_2 - \gamma_d^2 p_3 \right\} \end{aligned}$$

where

$$\mathcal{Y}^2_{kgsd} = \left\{ (\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, \boldsymbol{u}_1^2, \boldsymbol{v}_1^2, \boldsymbol{u}_2^2, \boldsymbol{v}_2^2, p_1, p_2, p_3) \;\middle|\; \begin{array}{ll} \exists \boldsymbol{u}_2^1, \boldsymbol{v}_2^1, \boldsymbol{u}_3^1, \boldsymbol{v}_3^1, \boldsymbol{a}_j^1, \boldsymbol{b}_j^1, j \in [3] \;: & \\ \boldsymbol{u}_1^1 + \boldsymbol{u}_2^1 + \boldsymbol{u}_3^1 = \boldsymbol{\mu}_k^1 & \\ \boldsymbol{v}_1^1 + \boldsymbol{v}_2^1 + \boldsymbol{v}_3^1 = \boldsymbol{\nu}_k^1 & \\ \boldsymbol{a}_1^1 + \boldsymbol{a}_2^1 + \boldsymbol{a}_3^1 \leq \boldsymbol{\sigma}_k^1 & \\ \boldsymbol{b}_1^1 + \boldsymbol{b}_2^1 + \boldsymbol{b}_3^1 \leq \boldsymbol{\varsigma}_k^1 & \\ \boldsymbol{u}_1^2 + \boldsymbol{u}_2^2 + \boldsymbol{u}_3^2 = \boldsymbol{\mu}_{sg}^2 & \\ \boldsymbol{v}_1^2 + \boldsymbol{v}_2^2 + \boldsymbol{v}_3^2 = \boldsymbol{\nu}_{sg}^2 & \\ \boldsymbol{a}_1^2 + \boldsymbol{a}_2^2 + \boldsymbol{a}_3^2 \leq \boldsymbol{\sigma}_{sg}^2 & \\ \boldsymbol{b}_1^2 + \boldsymbol{b}_2^2 + \boldsymbol{b}_3^2 \leq \boldsymbol{\varsigma}_{sg}^2 & \\ p_1 + p_2 + p_3 = 1 & \\ \underline{\boldsymbol{u}}_k^1 p_j \leq \boldsymbol{u}_j^1 \leq \bar{\boldsymbol{u}}_k^1 p_j & \forall j \in [3] \\ \underline{\boldsymbol{v}}_k^1 p_j \leq \boldsymbol{v}_j^1 \leq \bar{\boldsymbol{v}}_k^1 p_j & \forall j \in [3] \\ \boldsymbol{a}_j^1 \geq |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1 p_j| & \forall j \in [3] \\ \boldsymbol{b}_j^1 \geq |\boldsymbol{v}_j^1 - \boldsymbol{\nu}_k^1 p_j| & \forall j \in [3] \\ \underline{\boldsymbol{u}}_{sg}^2 p_j \leq \boldsymbol{u}_j^2 \leq \bar{\boldsymbol{u}}_{sg}^2 p_j & \forall j \in [3] \\ \underline{\boldsymbol{v}}_{sg}^2 p_j \leq \boldsymbol{v}_j^2 \leq \bar{\boldsymbol{v}}_{sg}^2 p_j & \forall j \in [3] \\ \boldsymbol{a}_j^2 \geq |\boldsymbol{u}_j^2 - \boldsymbol{\mu}_{sg}^2 p_j| & \forall j \in [3] \\ \boldsymbol{b}_j^2 \geq |\boldsymbol{v}_j^2 - \boldsymbol{\nu}_{sg}^2 p_j| & \forall j \in [3] \\ p_j, p_2, p_3 \geq 0 & \\ \boldsymbol{u}_j^1, \boldsymbol{a}_j^1, \boldsymbol{v}_j^1, \boldsymbol{b}_j^1, \boldsymbol{u}_j^2, \boldsymbol{a}_j^2, \boldsymbol{v}_j^2, \boldsymbol{b}_j^2 \in \mathbb{R}^N & \forall j \in [3] \end{array} \right\}.$$

Proof. See Appendix E.

With Theorems 7.4.1 and 7.4.2, we can now transform the DRO problem to a classical linear RO problem [150] as follows:

$$\begin{aligned} \inf \; & \sum_{d \in [D]} (\gamma_d^1 + \gamma_d^2) \\ \text{s.t.} \; & \sum_{k \in [K^1]} q_k \left(\left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \right)' \boldsymbol{u}_1^{1k} + \left(\sum_{i \in \bar{\mathcal{C}}^1} x_{id}^1 \boldsymbol{e}_i \right)' \boldsymbol{v}_1^{1k} - \bar{\tau}^1 p_1^k - \gamma_d^1 p_2^k \right) \\ & \qquad \leq 0 \qquad \forall (\boldsymbol{u}_1^{1k}, \boldsymbol{v}_1^{1k}, p_1^k, p_2^k) \in \mathcal{Y}_{kd}^1, k \in [K^1], \qquad \forall d \in [D] \qquad (7.21a) \\ & \sum_{(k,g,s) \in \mathcal{W}} q_{kgs} \left(\left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \right)' (\boldsymbol{u}_1^{1kgs} + \boldsymbol{v}_1^{1kgs}) + \left(\sum_{i \in \bar{\mathcal{C}}^2} (z_{id}^2(s) + x_{id}^2(s)) \boldsymbol{e}_i \right)' (\boldsymbol{u}_1^{2kgs} + \boldsymbol{v}_1^{2kgs}) \right. \end{aligned}$$

$$+\left(\sum_{i\in\bar{\mathcal{C}}^2} y^2_{id}(s)\boldsymbol{e}_i\right)' \boldsymbol{u}_1^{2kgs} - \bar{\tau}^2 p_1^{kgs} + \bar{\tau}^1 p_2^{kgs} + \left(\sum_{i\in\bar{\mathcal{C}}^2} z^2_{id}(s)\boldsymbol{e}_i\right)' \boldsymbol{v}_2^{2kgs}$$
$$+\left(\sum_{i\in\bar{\mathcal{C}}^2} x^2_{id}(s)\boldsymbol{e}_i\right)' (\boldsymbol{u}_2^{2kgs} + \boldsymbol{v}_2^{2kgs}) + \left(\sum_{i\in\bar{\mathcal{C}}^2} y^2_{id}(s)\boldsymbol{e}_i\right)' \boldsymbol{u}_2^{2kgs} - \bar{\tau}^2 p_2^{kgs} - \gamma_d^2 p_3^{kgs}\Bigg) \leq 0$$
$$\forall(\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, \boldsymbol{u}_1^2, \boldsymbol{v}_1^2, \boldsymbol{u}_2^2, \boldsymbol{v}_2^2, p_1, p_2, p_3) \in \mathcal{Y}^2_{kgsd}, (k,g,s) \in \mathcal{W} \qquad \forall d \in [D] \tag{7.21b}$$
$$\gamma_d^1, \gamma_d^2 \geq 0 \qquad \forall d \in [D]$$
$$(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2) \in \mathcal{X},$$

where constraints (7.21a) and (7.21b) are dynamically added, which has the advantage of keeping the number of decision variables small compared to the explicit formulation via RSOME. Specifically, we solve model (7.21) by ignoring these two groups of constraints. Whenever an integer solution is generated, we solve the equivalent maximum models presented in Theorem 7.4.1 and Theorem 7.4.2. If the resulting solutions violate the constraints, cuts are generated and added to the model, and the model is solved again. This process continues until the optimality gap is satisfied.

7.5 Numerical Experiments

In this section, we first introduce the instance sets and benchmark approaches, and then conduct numerical tests to evaluate the proposed B&C algorithm and the cluster-wise DRO framework.

Instance Sets

We set drones' launch speed in both directions to 20 m/s, i.e., $\bar{r}_0 = \bar{r}_i = 20$ m/s, $i \in [N]$, based on Amazon Prime Air and Workhorse HorseFly drones (used by UPS to deliver packages), which can fly at speeds up to 50 mph (i.e., 22.35 m/s) [151,152]. For each customer i, $i \in [N]$, we generate a random number from the interval $[6000, 18000]$ to denote the distance d_i between the depot and the customer. This indicates that if wind is neglected, it generally takes 10 to 30 minutes for a drone to perform a round trip. The distance data is based on the following reports: The Amazon Prime Air drone can fly 15 miles (in about 20 minutes at a speed of 20 m/s) to deliver packages [153] and the HorseFly drone has a 30-minute flight time [152]. The angle $\phi_i, i \in [N]$ between customer i and the x-axis takes a random integer value between 0 and 360. We set $\mathcal{C}^1 = \{1, \ldots, 0.4N\}$, $\mathcal{C}^2 = \{0.4N + 1, \ldots, 0.8N\}$, and $\mathcal{C}^3 = \{0.8N + 1, \ldots, N\}$. Let $\bar{\tau}^1 = 4000$ and $\bar{\tau}^2 = 8000$. We randomly generate instances of 15 and 20 customers, where 2 and 3 drones are used in the delivery system, respectively.

For a fixed number of customers, we generate 5 instances.

With respect to weather information, we use the wind data downloaded from the China National Meteorological Information Center (`http://data.cma.cn/site/index.html`). For each geographic region, the realized wind of the last 7 days (including the current day) are available. Each region is divided into several subregions, where the average wind speed and degree are reported hourly, i.e., 24 records per day. We utilize the wind data of Sichuan Province, a landlocked province in Southwest China, ranging from September 14^{th} to 19^{th}, 2019. The wind information is reported for 145 subregions; therefore, we have 3480 records per day. We consider every two hours as a time horizon, and the first hour is Period 1 and the second hour is Period 2. Thus, for each day, we can get 1740 wind samples. Note that for some missing data, we randomly generate continuous values based on the lower and upper bounds of the available data. We use the data of September 14^{th} as the (in-sample) training data, i.e., $H = 1740$, and the data from September 15^{th} to 19^{th} as the (out-of-sample) test data, producing 8700 samples.

To construct the cluster-wise ambiguity set, we use the K-means clustering algorithm to partition the wind samples. We note that we also tried other clustering algorithms such as the hierarchical clustering [154], and our preliminary tests showed that the results were similar to those under the K-means clustering. Thus, we opt to use the K-means clustering as it is widely adopted clustering algorithm and easy to implement. For the adaptive scheduling decisions in Period 2, we define each first-period cluster as an event. All of the algorithms and models are implemented in Python programming language using Gurobi 7.5.1 as the solver. The computations are executed on an Intel Core i5 2.3 GHz processor with 8GB memory.

Benchmark Approaches

We evaluate our DRO framework by comparing its solutions with those generated by the four other benchmark models. The first one is the deterministic model that maximizes the slack of time. The second one maximizes the joint on-time service probability. The third one has the same objective function as our DRO model, but it is implemented in a stochastic programming framework. The last one minimizes the maximum ERI.

Deterministic Model (DM). Since the considered drone delivery system is service-oriented, a natural objective would be to construct a drone schedule that maximizes the

slack of time in a deterministic model.

$$
\begin{aligned}
\max \quad & \sum_{d\in[D]}\sum_{t\in[2]} l_{dt} \\
\text{s.t.} \quad & \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}(u^1_i+v^1_i)+y^1_{id}u^1_i) \le \bar{\tau}^1 - l_{d1} && \forall d\in[D], \\
& \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(u^1_i+v^1_i) + \sum_{i\in\bar{\mathcal{C}}^2}((z^2_{id}+x^2_{id})(u^2_i+v^2_i)+y^2_{id}u^2_i) \le \bar{\tau}^2 - l_{d2} && \forall d\in[D], \\
& \bar{\tau}^1 + \sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}v^2_i + x^2_{id}(u^2_i+v^2_i)+y^2_{id}u^2_i) \le \bar{\tau}^2 - l_{d2} && \forall d\in[D], \\
& l_{dt} \ge 0 && \forall d\in[D], t\in[2], \\
& (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X}.
\end{aligned}
$$

The first three constraints calculate the slack time l_{dt} for drone d, $d\in[D]$ in period t, $t\in[2]$. Since the objective function only captures the magnitude of slack and does not indicate its probability, if two solutions have the same magnitude of slack time, the one with a potential probability of 5% may have the same preference as the other one with a probability of 95%. In computational experiments, we set the flight times $\boldsymbol{u}^1$, $\boldsymbol{v}^1$, $\boldsymbol{u}^2$, and $\boldsymbol{v}^2$ as their sample means.

Maximizing On-time Probability (MOP). To improve service quality, decision-makers may prefer to penalize lateness to guarantee that drones can serve customers within stipulated time windows as well as possible. Thus, another benchmark method is to maximize the joint on-time service probability as follows:

$$
\begin{aligned}
\max \quad & \mathbb{P}\left[\boldsymbol{\zeta}-\bar{\boldsymbol{\tau}}\le 0\right] \\
\text{s.t.} \quad & (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X},
\end{aligned}
\tag{7.22}
$$

where $\boldsymbol{\zeta} = \begin{bmatrix}\zeta^1_d,\ldots,\zeta^1_D \\ \zeta^2_d,\ldots,\zeta^2_D\end{bmatrix}$ and $\bar{\boldsymbol{\tau}} = (\bar{\tau}^1,\bar{\tau}^2)'\boldsymbol{e}'$, $\boldsymbol{e}\in\mathbb{R}^D$. In contrast to the objective of the deterministic model, the objective of Model (7.22) only captures the lateness probability and ignores the magnitude of lateness completely. Thus, if the lateness probabilities of two solutions are the same, the one with a lateness magnitude of 40 minutes may have the same preference as the one with a lateness of 1 minute.

Moreover, as the objective to be maximized is not a concave function, we use an empirical

distribution approximation reformulation to solve Model (7.22) as follows:

$$
\begin{aligned}
\max \quad & \frac{1}{H}\sum_{h\in[H]} I_h \\
\text{s.t.} \quad & \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}(u^1_{ih}+v^1_{ih})+y^1_{id}u^1_{ih})-\bar{\tau}^1 \le (1-I_h)M^1_h && \forall d\in[D], h\in[H], \\
& \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(u^1_{ih}+v^1_{ih})+\sum_{i\in\bar{\mathcal{C}}^2}((z^2_{id}+x^2_{id})(u^2_{ih}+v^2_{ih})+y^2_{id}u^2_{ih}) \\
& -\bar{\tau}^2 \le (1-I_h)M^2_h && \forall d\in[D], h\in[H], \\
& \bar{\tau}^1+\sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}v^2_{ih}+x^2_{id}(u^2_{ih}+v^2_{ih})+y^2_{id}u^2_{ih})-\bar{\tau}^2 \le (1-I_h)M^2_h && \forall d\in[D], h\in[H], \\
& I_h\in\{0,1\} && \forall h\in[H], \\
& (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X},
\end{aligned}
$$

where $u^t_{ih}\in\mathbb{R}$ and $v^t_{ih}\in\mathbb{R}$ are the travel times in Period t, $t\in[2]$ with respect to sample h, $h\in[H]$. $I_h=1$ if and only if all drones at all periods serve customers on time under sample h, $h\in[H]$. M^t_h, $h\in[H]$, $t\in[2]$ is a sufficiently large number. We choose $M^1_h=\sum_{i\in\bar{\mathcal{C}}^1}(u^1_{ih}+v^1_{ih})$ and $M^2_h=\max\left\{\sum_{i\in\bar{\mathcal{C}}^1}(u^1_{ih}+v^1_{ih}),\bar{\tau}^1\right\}+\sum_{i\in\bar{\mathcal{C}}^2}(u^2_{ih}+v^2_{ih})$.

Minimizing ERI Using Empirical Distribution (ERI-E). For Model (7.16), we assume the travel times follow an empirical distribution, thus the expectations in constraints (7.16a) and (7.16b) are now evaluated over a known distribution. The resulting model is

$$
\begin{aligned}
\inf \quad & \sum_{d\in[D]}(\gamma^1_d+\gamma^2_d) \\
\text{s.t.} \quad & \tfrac{1}{H}\sum_{h\in[H]} w^t_{dh}\le 0 && \forall d\in[D], t\in[2], \\
& w^1_{dh}\ge \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}(u^1_{ih}+v^1_{ih})+y^1_{id}u^1_{ih})-\bar{\tau}^1 && \forall d\in[D], h\in[H], \\
& w^2_{dh}\ge \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(u^1_{ih}+v^1_{ih})+\sum_{i\in\bar{\mathcal{C}}^2}((z^2_{id}+x^2_{id})(u^2_{ih}+v^2_{ih})+y^2_{id}u^2_{ih})-\bar{\tau}^2 && \forall d\in[D], h\in[H], \\
& w^2_{dh}\ge \bar{\tau}^1+\sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}v^2_{ih}+x^2_{id}(u^2_{ih}+v^2_{ih})+y^2_{id}u^2_{ih})-\bar{\tau}^2 && \forall d\in[D], h\in[H], \\
& w^t_{dh}\ge -\gamma^t_d && \forall d\in[D], h\in[H], t\in[2], \\
& \gamma^t_d\ge 0 && \forall d\in[D], t\in[2], \\
& (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X}.
\end{aligned}
$$

Minimizing the Maximum ERI (M-ERI). We can also minimize the maximum ERI with empirical distribution. To do so, we replace the objective of the ERI-E model with

$$\min\gamma$$

and add the additional constraints

$$\gamma \geq \gamma_d^t \qquad \forall d \in [D], t \in [2].$$

For all methods, we first derive the delivery decisions from training data, then evaluate their out-of-sample performance using the following four indicators, which are relevant to decision-makers in a service-oriented system.

- *MaxPro:* The maximum lateness probability across all customers.
- *SumPro:* The sum of lateness probabilities of all customers.
- *MaxExp:* The maximum expected lateness duration across all customers.
- *SumExp:* The sum of expected lateness durations of all customers.

When calculating the expected lateness, we assume the travel times follow an empirical distribution. In addition, for the sequence of customers (except the last customer in Period 1 and the first and last customers in Period 2), we assume that drones will serve a customer first if the customer is nearer to the depot.

Algorithm Performance

We compare the performance of the B&C algorithm with RSOME by solving the adaptive DRO model. The optimality gap for both methods is set to 0.01%. Average results are reported in Table 7.1, where *time ratio* is obtained by using the computing time of the B&C algorithm to divide that of the RSOME. We conduct the comparison on instances with 15 customers, as RSOME cannot provide a solution within 6 hours for some instances with $N = 20$ when $K^1 \geq 2$.

Table 7.1 shows that the B&C algorithm can solve instances to optimality within a shorter time frame in most cases (10 out of 12). In particular, when $K^2 = 3$, the average CPU time of the B&C is equal to or less than half of the computing time of the RSOME. Therefore, we can conclude that the B&C algorithm can generally achieve a greater scalability for the adaptive DRO model with a cluster-wise ambiguity set, compared to RSOME.

Comparison of Decision Criteria

We first evaluate the performance of different decision criteria. For models MOP, M-ERI, and ERI-E, the optimality gap is set to 0.5% and the time limit for solving each model is set

Table 7.1 Performance comparison (CPU time in seconds) between B&C and RSOME

N	K^1	K^2	B&C	RSOME	Time ratio
15	1	1	4.98	3.36	1.48
		2	10.09	19.38	0.52
		3	18.57	37.42	0.50
	2	1	19.93	29.15	0.68
		2	50.53	120.05	0.42
		3	102.70	344.24	0.30
	3	1	88.89	201.41	0.44
		2	243.65	294.56	0.83
		3	471.87	1868.23	0.25
	4	1	880.01	466.13	1.89
		2	2353.60	5485.28	0.43
	5	1	3478.40	6615.62	0.53

to 3600 seconds. The average results of out-of-sample tests are reported in Table 7.2.

Table 7.2 Average results of out-of-sample tests under different decision criteria

N	Model	MaxPro	SumPro	MaxExp	SumExp	CPU time
15	DM	0.1835	0.2410	17.17	25.43	0.06
	MOP	0.0362	0.0743	7.93	16.11	2563.31
	M-ERI	0.0360	0.0732	7.56	15.65	530.04
	ERI-E	0.0339	0.0668	7.57	14.81	90.22
20	DM	0.1303	0.1809	17.38	29.01	0.06
	MOP	0.0309	0.0892	7.11	20.36	3580.08
	M-ERI	0.0319	0.1053	6.90	22.78	3600.00
	ERI-E	0.0298	0.0850	7.09	19.86	3599.33

Table 7.2 shows that while Method DM consumes the least CPU time, it generates very poor solutions, leading to significant out-of-sample lateness when uncertainty is present. Specifically, the lateness probability for the worst-case customer, *MaxPro*, is as much as 18.35% and 13.03% for $N = 15$ and 20, respectively; however, under other decision criteria, the value of *MaxPro* is around 3%.

In general, Method ERI-E provides a better out-of-sample performance than the three other methods. Specifically, it shows a superiority in indicators *MaxPro*, *SumPro*, and *SumExp*. It also consumes much less CPU time than Methods M-ERI and MOP when $N = 15$. Method M-ERI performs better with regard to *MaxExp*, which is consistent with its objective function; however, it performs worse in other indicators and requires more CPU time in comparison with ERI-E. This is because there exist multiple optimal solutions under this criterion. Based on the results here, we use Method ERI-E to evaluate our robust framework in the next section.

Comparison Between Robust Method and Empirical Distribution

In this section, we compare the out-of-sample performance of the static DRO model and Method ERI-E. The DRO model is solved by the B&C algorithm with the same optimality gap and time limit as those of Method ERI-E. The average results are reported in Table 7.3, where the first row under each N presents the results of Method ERI-E.

Table 7.3 Average results of out-of-sample tests generated by the static DRO model

N	K^1	K^2	MaxPro	SumPro	MaxExp	SumExp	CPU time
15	ERI-E		0.0339	0.0668	7.57	14.81	90.22
	1	1	0.0391	0.0677	8.61	15.23	5.15
		2	0.0344	0.0655	8.01	15.06	11.37
		3	0.0332	0.0647	7.70	14.71	22.49
	2	1	0.0349	0.0647	8.05	14.86	9.90
		2	0.0337	0.0652	7.77	14.65	25.68
		3	0.0364	0.0657	7.83	14.90	56.56
	3	1	0.0340	0.0650	7.93	14.83	26.41
		2	0.0330	0.0646	7.81	14.77	72.18
		3	0.0325	0.0644	7.83	14.79	93.55
	4	1	0.0334	0.0650	7.70	14.88	35.93
		2	0.0340	0.0650	7.71	14.82	89.10
	5	1	0.0329	0.0644	7.53	14.63	65.45
20	ERI-E		0.0298	0.0850	7.09	19.86	3599.33
	1	1	0.0406	0.0931	8.74	21.15	18.58
		2	0.0321	0.0865	7.34	20.34	49.37
		3	0.0323	0.0838	7.59	20.06	96.90
	2	1	0.0364	0.0889	8.14	20.66	68.16
		2	0.0300	0.0838	7.33	20.04	475.42
		3	0.0343	0.0849	7.69	20.00	595.31
	3	1	0.0350	0.0855	8.26	20.12	265.55
		2	0.0304	0.0830	7.23	19.82	891.86
		3	0.0303	0.0836	7.43	19.85	1734.09
	4	1	0.0297	0.0822	7.16	19.71	671.79
		2	0.0283	0.0821	6.81	19.73	2860.55
	5	1	0.0293	0.0839	6.98	19.87	1428.63

Table 7.3 shows that the static DRO model can find solutions with better out-of-sample performance in all indicators for $N = 15$ when $(K^1, K^2) = (5, 1)$ and for $N = 20$ when $(K^1, K^2) = (4, 2)$. Moreover, the static DRO model consumes less CPU time in these two cases compared to its empirical counterpart, ERI-E. When $N = 20$ and $(K^1, K^2) = (5, 1)$, the DRO model produces solutions with better performance in indicators *MaxPro*, *SumPro*, and *MaxExp*, and the value of *SumExp* is larger than that of Method ERI-E by only 0.01. However, the DRO model requires much less computing time (1428.63 seconds versus 3599.33 seconds) on average. Thus, we can conclude that the robust method is more effective in mitigating the service lateness than the empirical distribution for our drone delivery problem.

Comparison Between Adaptive and Static Robust Models

In this section, we compare the performance of the adaptive and static DRO models. The average results are reported in Table 7.4, with results of the static model shown in parentheses. Under the same cluster numbers, if the static model performs better in an indicator, we mark the results in bold.

Table 7.4 Average results of out-of-sample tests generated by the adaptive and static DRO models

N	K^1	K^2	MaxPro	SumPro	MaxExp	SumExp	CPU time
15	2	1	**0.0355 (0.0349)**	0.0612 (0.0647)	7.95 (8.05)	14.07 (14.86)	18.62 (9.90)
		2	0.0308 (0.0337)	0.0590 (0.0652)	7.41 (7.77)	13.67 (14.65)	57.03 (25.68)
		3	0.0312 (0.0364)	0.0587 (0.0657)	7.24 (7.83)	13.81 (14.90)	135.64 (56.56)
	3	1	0.0326 (0.0340)	0.0606 (0.0650)	7.62 (7.93)	13.89 (14.83)	76.69 (26.41)
		2	0.0318 (0.0330)	0.0602 (0.0646)	7.54 (7.81)	14.00 (14.77)	277.58 (72.18)
		3	0.0320 (0.0325)	0.0595 (0.0644)	7.52 (7.83)	13.74 (14.79)	537.31 (93.55)
	4	1	**0.0335 (0.0334)**	0.0612 (0.0650)	7.62 (7.70)	14.04 (14.88)	189.63 (35.93)
		2	0.0325 (0.0340)	0.0596 (0.0650)	7.27 (7.71)	13.68 (14.82)	578.85 (89.10)
	5	1	0.0321 (0.0329)	0.0607 (0.0644)	7.20 (7.53)	13.94 (14.63)	491.41 (65.45)
20	2	1	0.0306 (0.0364)	0.0814 (0.0889)	7.31 (8.14)	19.65 (20.66)	336.44 (68.16)
		2	**0.0342 (0.0300)**	0.0812 (0.0838)	**7.91 (7.33)**	19.14 (20.04)	2920.24 (475.42)
		3	0.0299 (0.0343)	0.0785 (0.0849)	7.14 (7.69)	19.00 (20.00)	3174.50 (595.31)
	3	1	0.0329 (0.0350)	0.0782 (0.0855)	7.70 (8.26)	18.23 (20.12)	3022.32 (265.55)
		2	0.0282 (0.0304)	0.0755 (0.0830)	7.13 (7.23)	18.33 (19.82)	3600.00 (891.86)
		3	0.0293 (0.0303)	0.0773 (0.0836)	6.98 (7.43)	18.26 (19.85)	3600.00 (1734.09)
	4	1	**0.0331 (0.0297)**	0.0792 (0.0822)	**7.40 (7.16)**	18.33 (19.71)	3600.00 (671.79)
		2	0.0269 (0.0283)	0.0783 (0.0821)	6.64 (6.81)	17.76 (19.73)	3600.00 (2860.55)
	5	1	**0.0338 (0.0293)**	0.0836 (0.0839)	**7.20 (6.98)**	18.51 (19.87)	3600.00 (1428.63)

Table 7.4 shows that under the same cluster numbers, in most cases the adaptive model can find solutions with better out-of-sample performance in all indicators compared to the static model. When $N = 20$, under some cluster numbers, even though the adaptive model is not solved to optimality, it can still generate solutions with better performance. The static model performs better in indicators *MaxPro* and *MaxExp* under some cluster numbers, while the adaptive model demonstrates superiority in the two other indicators. In the previous section, we observe that the static model performs better in all indicators when $(K^1, K^2) = (5, 1)$ for $N = 15$ compared to Method ERI-E, whereas the adaptive model can further improve performance under multiple cluster numbers besides $(K^1, K^2) = (5, 1)$, e.g., $(K^1, K^2) = (2, 2)$ and $(2, 3)$, etc. Likewise, for $N = 20$, the performance of the adaptive model also dominates that of the static model in all indicators when $(K^1, K^2) = (4, 2)$. Another observation is that with a small number of clusters, both robust models outperform the empirical distribution; however, the out-of-sample performance does not necessarily improve with a larger number of clusters, because the ambiguity set would converge to the empirical distribution with

increasing clusters, which may lead to over-fitting. In practice, we can determine the number of clusters via validation tests.

Table 7.5 gives the detailed solutions of one instance under $(K^1, K^2) = (3, 2)$ for $N = 15$. The centroids of the first-period clusters can be easily obtained after applying the K-means clustering algorithm, which is available in many open-source machine learning libraries (e.g., the scikit-learn library that we use). We can observe that under different clusters (or events), the adaptive DRO model generates different delivery schedules for Period 2. For any out-of-sample, when decision-makers have observed the wind realization in Period 1, they can calculate the distance between that realization and each cluster centroid to decide which cluster the observation belongs to, then adopt the corresponding delivery schedule in that cluster for Period 2.

Table 7.5 Detailed solutions of the DRO models for one instance

	Adaptive	Static
Delivery schedule in Period 1	[4, 1, 2, 3], [5, 15, 6]	[4, 1, 2, 6], [5, 15, 3]
	Delivery schedule in Period 2	
Centroid (cartesian coordinates)	Adaptive	Static
(-0.76, 0.68)	[9, 8, 7, 14], [11, 13, 10, 12]	[13, 8, 11, 12], [10, 9, 7, 14]
(1.55, 0.15)	[11, 9, 14, 10], [7, 8, 13, 12]	
(-0.16, -1.23)	[7, 9, 8, 14], [10, 11, 13, 12]	

7.6 Conclusions

In this chapter, we introduce a distributionally robust optimization model to solve a two-period drone scheduling problem with uncertain flight times, which can be implemented in a data-driven framework using historical weather information. We propose a cluster-wise moment-based ambiguity set by partitioning the wind vector chart into different clusters, which allows us to adapt the delivery schedule in the afternoon to updated weather information available by midday. For greater scalability, we develop a branch-and-cut algorithm for the adaptive robust model. To evaluate the proposed robust scheme, we benchmark our method against other classical models. Numerical results demonstrate that our robust framework, especially the adaptive robust model, can effectively reduce the service lateness at customers in out-of-sample tests.

CHAPTER 8 CONCLUSIONS AND RECOMMENDATIONS

In this chapter, we summarize all the works presented in this thesis and suggest future research directions.

8.1 Summary of Works

This thesis has solved two types of problems arising from supply chain management: the reliable facility location problem at the strategic level and the drone delivery problem at the operational level. For the facility location problem, we consider both supply-side and receiver-side uncertainties, which are common in real-world applications. Due to the unique character of disruption risk, i.e., being difficult to identify the distribution of disruptions, we adopt a two-stage RO framework to solve these problems. This framework does not require any probability information in comparison with stochastic models, and can also generate less conservative solutions compared to the static RO method. Specifically, location decisions are made in the first stage, and customer reassignment is decided in the second stage after the uncertainty information is observed. The work in Chapter 3 focuses on algorithm improvement, where an LP-based enumeration method is proposed for the C&CG algorithm. In Chapter 4, we consider disruptions and demand uncertainty simultaneously, which is seldom addressed in the literature. We focus on studying the impact of multiple uncertainties. In Chapter 5, we consider a reliable three-echelon network design problem. We build three two-stage RO models to see the trade-off between cost and reliability.

For the drone delivery problem, we have studied both the deterministic and stochastic versions. For the deterministic variant, we solve a MTDRP with time windows, where a nonlinear energy function is used to calculate drones' energy consumption, instead of using the linear approximation method as in the literature. We are the first to generate benchmark instances on the DRP and develop an exact algorithm to solve the problem. For the stochastic variant, we consider the impact of weather on drones' travel time. We use a DRO scheme to model the problem, where machine learning techniques are applied to historical weather data for the construction of the ambiguity set. The proposed modeling and solution schemes can effectively reduce the risk of lateness at customers.

8.2 Future Research

We consider future research can be conducted in following aspects.

Reliable Facility Location. One possible extension of our work would be to consider demand changes in disruption scenarios. Because when disruptions occur, customers' demand patterns may change considerably [32,42,155]. For example, after an earthquake or a flood, the demand for daily necessities (e.g., bottled water and medical supplies) may increase and the demand for luxuries may decrease or vanish. The second extension is to consider multiple recourse strategies in the second-stage problem, e.g., products are shared from a non-disrupted facility to a disrupted one, or part of the demand at a customer is satisfied by subcontracting facilities. It would also be interesting to combine reliable facility locations with other supply chain problems, such as vehicle routing and inventory management problems.

Drone Delivery Problem. The limitations of the energy function used in Chapter 6 include: (1) We did not consider the energy consumption of other flight status like taking off and landing. (2) Some other factors such as drone speed and wind speed were neglected in our power function. (3) The parameters associated with the drones considered in our study are small drones with limited payloads and a low travel speed; however, the recently developed drone models by Amazon and UPS can carry payloads of up to five pounds and fly at speeds up to 50 *mph*. Thus, we consider future research which can extend our work in Chapter 6 in the following aspects: (1) More complex power models with additional influence factors can be used for drone energy calculation. In this case, the energy consumption of other flight status should also be explicitly incorporated into the mathematical model. (2) Numerical tests can be conducted by using the parameters collected from production level delivery drones to provide operational insights for decision-makers. (3) Heuristic or metaheuristic algorithms can be designed to solve large-scale problems for the DRP.

For the drone delivery under uncertainty, we can include more covariate information (e.g., air pressure and temperature) to the ambiguity set when conducting the numerical tests. We can also extend the proposed partition-based framework to other applications, where uncertain parameters are correlated to some uncertain covariates. For example, the containership routing problems [156], where wind's speed and direction affect vessels' travel time.

REFERENCES

[1] D. Simchi-Levi, H. Wang, and Y. Wei, "Constraint generation for two-stage robust network flow problems," *INFORMS Journal on Optimization*, vol. 1, no. 1, pp. 49–70, 2019.

[2] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2017.

[3] S. H. Owen and M. S. Daskin, "Strategic facility location: A review," *European Journal of Operational Research*, vol. 111, no. 3, pp. 423–447, 1998.

[4] L. V. Snyder and M. S. Daskin, "Reliability models for facility location: The expected failure cost case," *Transportation Science*, vol. 39, no. 3, pp. 400–416, 2005.

[5] T. Cui, Y. Ouyang, and Z.-J. M. Shen, "Reliable facility location design under the risk of disruptions," *Operations Research*, vol. 58, no. 4-part-1, pp. 998–1011, 2010.

[6] Q. Chen, X. Li, and Y. Ouyang, "Joint inventory-location problem under the risk of probabilistic facility disruptions," *Transportation Research Part B: Methodological*, vol. 45, no. 7, pp. 991–1003, 2011.

[7] W. Xie, Y. Ouyang, and S. C. Wong, "Reliable location-routing design under probabilistic facility disruptions," *Transportation Science*, vol. 50, no. 3, pp. 1128–1138, 2015.

[8] V. Gabrel, C. Murat, and A. Thiele, "Recent advances in robust optimization: An overview," *European Journal of Operational Research*, vol. 235, no. 3, pp. 471–483, 2014.

[9] R. Jiang, M. Zhang, G. Li, and Y. Guan, "Benders' decomposition for the two-stage security constrained robust unit commitment problem," in *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 2012, pp. 1–10.

[10] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, "Adaptive robust optimization for the security constrained unit commitment problem," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 52–63, 2013.

[11] A. Billionnet, M.-C. Costa, and P.-L. Poirion, "2-Stage Robust MILP with continuous recourse variables," *Discrete Applied Mathematics*, vol. 170, pp. 21–32, 2014.

[12] D. Bertsimas and S. Shtern, "A scalable algorithm for two-stage adaptive linear optimization," 2018, working paper, https://arxiv.org/abs/1807.02812.

[13] D. Simchi-Levi, H. Wang, and Y. Wei, "Constraint generation for two-stage robust network flow problems," *INFORMS Journal on Optimization*, vol. 1, no. 1, pp. 49–70, 2019.

[14] B. Zeng and L. Zhao, "Solving two-stage robust optimization problems using a column-and-constraint generation method," *Operations Research Letters*, vol. 41, no. 5, pp. 457–461, 2013.

[15] J. Ayoub and M. Poss, "Decomposition for adjustable robust linear optimization subject to uncertainty polytope," *Computational Management Science*, vol. 13, no. 2, pp. 219–239, 2016.

[16] A. Ben-Tal, B. Do Chung, S. R. Mandala, and T. Yao, "Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains," *Transportation Research Part B: Methodological*, vol. 45, no. 8, pp. 1177–1189, 2011.

[17] D. Simchi-Levi, N. Trichakis, and P. Y. Zhang, "Designing response supply chain against bioattacks," *Operations Research*, vol. 67, no. 5, pp. 1246–1268, 2019.

[18] A. Ardestani-Jaafari and E. Delage, "The value of flexibility in robust location–transportation problems," *Transportation Science*, vol. 52, no. 1, pp. 189–209, 2017.

[19] C. Rose, "Amazon's Jeff Bezos Looks to the Future," 2013, december 6, 2019, https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/.

[20] P. Morgan, *Carbon fibers and their composites*. CRC press, Florida, 2005.

[21] T. B. Reddy, "Chapter 14: Lithium primary batteries," in *Linden's Handbook of Batteries, 4th edition*. McGraw-Hill Professional, New York, 2010, pp. 14.1–14.90.

[22] J. Doherty, "Alphabet's Wing begins first commercial drone delivery service in U.S., beating Amazon, Uber," 2019, accessed December 6, 2019, https://www.newsweek.com/wing-drone-first-commercial-delivery-1466471.

[23] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transportation Science*, vol. 52, no. 4, pp. 965–981, 2018.

[24] S. Chowdhury, A. Emelogu, M. Marufuzzaman, S. G. Nurre, and L. Bian, "Drones for disaster response and relief operations: A continuous approximation model," *International Journal of Production Economics*, vol. 188, pp. 167–184, 2017.

[25] M. Heutger and M. Kückelhaus, "Unmanned aerial vehicle in logistics: a DHL perspective on implications and use cases for the logistics industry." 2014, technical Report, DHL Trend Research, http://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/dhl_trend_report_uav.pdf.

[26] L. Walker, "Drone delivery for Amazon and Google slowed by headwinds," 2014, accessed February 15, 2020, https://www.newsweek.com/will-wind-be-end-commercial-drone-delivery-amazon-and-google-275999.

[27] G. Mehra, "8 obstacles to drone delivery, for ecommerce," 2015, accessed February 15, 2020, https://www.practicalecommerce.com/8-Obstacles-to-Drone-Delivery-for-Ecommerce.

[28] R. Enderle, "The 5 most pressing problems with drone delivery," 2019, accessed February 15, 2020, https://www.technewsworld.com/story/86060.html.

[29] T. Black, "The future of drone delivery depends on predicting the weather," 2017, accessed February 15, 2020, https://mashable.com/2017/06/22/drone-delivery-weather/.

[30] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[31] M. Stevenson and M. Spring, "Flexibility from a supply chain perspective: definition and review," *International journal of operations & production management*, 2007.

[32] L. V. Snyder, Z. Atan, P. Peng, Y. Rong, A. J. Schmitt, and B. Sinsoysal, "OR/MS models for supply chain disruptions: A review," *IIE Transactions*, vol. 48, no. 2, pp. 89–109, 2016.

[33] V. Rajagopal, S. P. Venkatesan, and M. Goh, "Decision-making models for supply chain risk mitigation: A review," *Computers & Industrial Engineering*, vol. 113, pp. 646–682, 2017.

[34] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*. John Wiley & Sons, 2011.

[35] Z.-J. M. Shen, R. L. Zhan, and J. Zhang, "The reliable facility location problem: Formulations, heuristics, and approximation algorithms," *INFORMS Journal on Computing*, vol. 23, no. 3, pp. 470–482, 2011.

[36] Z. Drezner, "Heuristic solution methods for two location problems with unreliable facilities," *Journal of the Operational Research Society*, vol. 38, no. 6, pp. 509–514, 1987.

[37] G. Yu, W. B. Haskell, and Y. Liu, "Resilient facility location against the risk of disruptions," *Transportation Research Part B: Methodological*, vol. 104, pp. 82–105, 2017.

[38] B. Afify, S. Ray, A. Soeanu, A. Awasthi, M. Debbabi, and M. Allouche, "Evolutionary learning algorithm for reliable facility location under disruption," *Expert Systems with Applications*, vol. 115, pp. 223–244, 2019.

[39] S. Xie and Y. Ouyang, "Reliable service systems design under the risk of network access failures," *Transportation Research Part E: Logistics and Transportation Review*, vol. 122, pp. 1–13, 2019.

[40] S. Xie, K. An, and Y. Ouyang, "Planning facility location under generally correlated facility disruptions: Use of supporting stations and quasi-probabilities," *Transportation Research Part B: Methodological*, vol. 122, pp. 115–139, 2019.

[41] N. Azad and E. Hassini, "A benders decomposition method for designing reliable supply chain networks accounting for multimitigation strategies and demand losses," *Transportation Science*, vol. 53, no. 5, pp. 1287–1312, 2019.

[42] Y. An, B. Zeng, Y. Zhang, and L. Zhao, "Reliable p-median facility location problem: Two-stage robust models and algorithms," *Transportation Research Part B: Methodological*, vol. 64, pp. 54–72, 2014.

[43] M. Lu, L. Ran, and Z.-J. M. Shen, "Reliable facility location design under uncertain correlated disruptions," *Manufacturing & Service Operations Management*, vol. 17, no. 4, pp. 445–455, 2015.

[44] T. Sawik *et al.*, *Supply Chain Disruption Management Using Stochastic Mixed Integer Programming*. Springer, 2018.

[45] L. V. Snyder, "Facility location under uncertainty: A review," *IIE Transactions*, vol. 38, no. 7, pp. 547–564, 2006.

[46] O. Baron, J. Milner, and H. Naseraldin, "Facility location: A robust optimization approach," *Production and Operations Management*, vol. 20, no. 5, pp. 772–785, 2011.

[47] A. Atamtürk and M. Zhang, "Two-stage robust network flow and design under demand uncertainty," *Operations Research*, vol. 55, no. 4, pp. 662–673, 2007.

[48] A. Marandi and G.-J. van Houtum, "Robust location-transportation problems with integer-valued demand," *Optimization Online*, 2020.

[49] B. Basciftci, S. Ahmed, and S. Shen, "Distributionally robust facility location problem under decision-dependent stochastic demand," 2019, working paper, https://arxiv.org/pdf/1912.05577.pdf.

[50] C. Wu, D. Du, and D. Xu, "An approximation algorithm for the two-stage distributionally robust facility location problem," in *Advances in Global Optimization*. Springer, 2015, pp. 99–107.

[51] A. Gourtani, T.-D. Nguyen, and H. Xu, "A distributionally robust optimization approach for two-stage facility location problems," *EURO Journal on Computational Optimization*, pp. 1–32, 2020.

[52] Y. Gao and Z. Qin, "A chance constrained programming approach for uncertain p-hub center location problem," *Computers & Industrial Engineering*, vol. 102, pp. 10–20, 2016.

[53] S. Mišković, Z. Stanimirović, and I. Grujičić, "Solving the robust two-stage capacitated facility location problem with uncertain transportation costs," *Optimization Letters*, vol. 11, no. 6, pp. 1169–1184, 2017.

[54] L. R. Matthews, C. E. Gounaris, and I. G. Kevrekidis, "Designing networks with resiliency to edge failures using two-stage robust optimization," *European Journal of Operational Research*, vol. 279, no. 3, pp. 704–720, 2019.

[55] M. S. Pishvaee, M. Rabbani, and S. A. Torabi, "A robust optimization approach to closed-loop supply chain network design under uncertainty," *Applied Mathematical Modelling*, vol. 35, no. 2, pp. 637–649, 2011.

[56] N. Noyan, B. Balcik, and S. Atakan, "A stochastic optimization model for designing last mile relief networks," *Transportation Science*, vol. 50, no. 3, pp. 1092–1113, 2016.

[57] C. A. Zetina, I. Contreras, J.-F. Cordeau, and E. Nikbakhsh, "Robust uncapacitated hub location," *Transportation Research Part B: Methodological*, vol. 106, pp. 393–410, 2017.

[58] S. Wang, Z. Chen, and T. Liu, "Distributionally robust hub location," *Transportation Science*, 2020, forthcoming.

[59] A. Baghalian, S. Rezapour, and R. Z. Farahani, "Robust supply chain network design with service level against disruptions and demand uncertainties: A real-life case," *European Journal of Operational Research*, vol. 227, no. 1, pp. 199–215, 2013.

[60] S. Zokaee, A. Bozorgi-Amiri, and S. J. Sadjadi, "A robust optimization model for humanitarian relief chain design under uncertainty," *Applied Mathematical Modelling*, vol. 40, no. 17-18, pp. 7996–8016, 2016.

[61] W. Ni, J. Shu, and M. Song, "Location and emergency inventory pre-positioning for disaster response operations: Min-max robust model and a case study of yushu earthquake," *Production and Operations Management*, vol. 27, no. 1, pp. 160–183, 2018.

[62] S. Mazahir and A. Ardestani-Jaafari, "Robust global sourcing under compliance legislation," *European Journal of Operational Research*, vol. 284, no. 1, pp. 152–163, 2020.

[63] L. V. Snyder, M. P. Scaparra, M. S. Daskin, and R. L. Church, "Planning for disruptions in supply chain networks," in *Models, methods, and applications for innovative decision making*. INFORMS, 2006, pp. 234–257.

[64] P. Peng, L. V. Snyder, A. Lim, and Z. Liu, "Reliable logistics networks design with facility disruptions," *Transportation Research Part B: Methodological*, vol. 45, no. 8, pp. 1190–1211, 2011.

[65] N. Azad, G. K. Saharidis, H. Davoudpour, H. Malekly, and S. A. Yektamaram, "Strategies for protecting supply chain networks against facility and transportation disruptions: an improved benders decomposition approach," *Annals of Operations Research*, vol. 210, no. 1, pp. 125–163, 2013.

[66] D. Shishebori, L. V. Snyder, and M. S. Jabalameli, "A reliable budget-constrained fl/nd problem with unreliable facilities," *Networks and Spatial Economics*, vol. 14, no. 3-4, pp. 549–580, 2014.

[67] S. Rezapour, R. Z. Farahani, and M. Pourakbar, "Resilient supply chain network design under competition: A case study," *European Journal of Operational Research*, vol. 259, no. 3, pp. 1017–1035, 2017.

[68] Y. Choi and P. M. Schonfeld, "Optimization of multi-package drone deliveries considering battery capacity," in *96th Annual Meeting of the Transportation Research Board, Washington, DC (Paper No. 17–05769)*, 2017.

[69] K. T. San, E. Y. Lee, and Y. S. Chang, "The delivery assignment solution for swarms of uavs dealing with multi-dimensional chromosome representation of genetic algorithm," in *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual*. IEEE, 2016, pp. 1–7.

[70] A. Troudi, S.-A. Addouche, S. Dellagi, and A. Mhamedi, "Sizing of the drone delivery fleet considering energy autonomy," *Sustainability*, vol. 10, no. 9, p. 3344, 2018.

[71] R. D'Andrea, "Guest editorial can drones deliver?" *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 647–648, 2014.

[72] M. A. Figliozzi, "Lifecycle modeling and assessment of unmanned aerial vehicles (Drones) CO_2e emissions," *Transportation Research Part D: Transport and Environment*, vol. 57, pp. 251–261, 2017.

[73] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nature communications*, vol. 9, no. 1, p. 409, 2018.

[74] Z. Liu, R. Sengupta, and A. Kurzhanskiy, "A power consumption model for multi-rotor small unmanned aircraft systems," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017, pp. 310–315.

[75] T. Kirschstein, "Comparison of energy demands of drone-based and ground-based parcel delivery services," *Transportation Research Part D: Transport and Environment*, vol. 78, p. 102209, 2020.

[76] J. Zhang, J. F. Campbell, D. Sweeney, II, and A. C. Hupman, "Energy consumption models for delivery drones: A comparison and assessment," 2020.

[77] A. Ponza, "Optimization of drone-assisted parcel delivery," Master's thesis, University of Padova, Italy, 2016.

[78] P. Bouman, N. Agatz, and M. Schmidt, "Dynamic programming approaches for the traveling salesman problem with drone," *Networks*, vol. 72, no. 4, pp. 528–542, 2018.

[79] S. Poikonen, B. Golden, and E. A. Wasil, "A branch-and-bound approach to the traveling salesman problem with a drone," *INFORMS Journal on Computing*, vol. 31, no. 2, pp. 335–346, 2019.

[80] M. Marinelli, L. Caggiani, M. Ottomanelli, and M. Dell'Orco, "En route truck–drone parcel delivery for optimal vehicle routing strategies," *IET Intelligent Transport Systems*, vol. 12, no. 4, pp. 253–261, 2017.

[81] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones," *International Journal of Production Economics*, vol. 214, pp. 220–233, 2019.

[82] M. Moshref-Javadi and S. Lee, "Using drones to minimize latency in distribution systems," in *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 2017, pp. 235–240.

[83] A. M. Ham, "Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 1–14, 2018.

[84] M. W. Ulmer and B. W. Thomas, "Same-day delivery with a heterogeneous fleet of drones and vehicles," *Networks*, vol. 72, no. 4, pp. 475–505, 2018.

[85] X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: Several worst-case results," *Optimization Letters*, vol. 11, no. 4, pp. 679–697, 2017.

[86] S. Poikonen, X. Wang, and B. Golden, "The vehicle routing problem with drones: Extended models and connections," *Networks*, vol. 70, no. 1, pp. 34–43, 2017.

[87] L. D. P. Pugliese and F. Guerriero, "Last-mile deliveries by using drones and classical vehicles," in *International Conference on Optimization and Decision Science*. Springer, 2017, pp. 557–565.

[88] Z. Wang and J.-B. Sheu, "Vehicle routing problem with drones," *Transportation Research Part B: Methodological*, vol. 122, pp. 350–364, 2019.

[89] R. Raj and C. C. Murray, "Fly slower, deliver faster: The multiple flying sidekicks traveling salesman problem with variable drone speeds." 2020, June 2, 2020, https://ssrn.com/abstract=3549622.

[90] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1298–1308, 2015.

[91] Z. Luo, Z. Liu, and J. Shi, "A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle," *Sensors*, vol. 17, no. 5, p. 1144, 2017.

[92] J. G. Carlsson and S. Song, "Coordinated logistics with a truck and a drone," *Management Science*, vol. 64, no. 9, pp. 4052–4069, 2017.

[93] J. F. Campbell, D. Sweeney, and Z. J. II, "Strategic design for delivery with trucks and drones," 2017, technical Report, https://www.villagereach.org/wp-content/uploads/2018/07/StrategicDesignforDeliverywithDronesandTrucks_4-17-17_SCMA-2017-0201.pdf.

[94] B. Fleischmann, "The vehicle routing problem with multiple use of the vehicles," 1990, working Paper, Facbereich Wirtschaftswissenschafte, Universität Hamburg.

[95] N. Azi, M. Gendreau, and J.-Y. Potvin, "An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles," *European Journal of Operational Research*, vol. 202, no. 3, pp. 756–763, 2010.

[96] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

[97] R. Macedo, C. Alves, J. V. de Carvalho, F. Clautiaux, and S. Hanafi, "Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model," *European Journal of Operational Research*, vol. 214, no. 3, pp. 536–545, 2011.

[98] A. Mingozzi, R. Roberti, and P. Toth, "An exact algorithm for the multitrip vehicle routing problem," *INFORMS Journal on Computing*, vol. 25, no. 2, pp. 193–207, 2013.

[99] F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud, "A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration," *4OR*, vol. 12, no. 3, pp. 235–259, 2014.

[100] N. Azi, M. Gendreau, and J.-Y. Potvin, "An adaptive large neighborhood search for a vehicle routing problem with multiple routes," *Computers & Operations Research*, vol. 41, pp. 167–173, 2014.

[101] Z. Wang, W. Liang, and X. Hu, "A metaheuristic based on a pool of routes for the vehicle routing problem with multiple trips and time windows," *Journal of the Operational Research Society*, vol. 65, no. 1, pp. 37–48, 2014.

[102] F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud, "Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 249, no. 2, pp. 551–559, 2016.

[103] D. Cattaruzza, N. Absi, and D. Feillet, "Vehicle routing problems with multiple trips," *4OR*, vol. 14, no. 3, pp. 223–259, 2016.

[104] İ. Karaoğlan, "A branch-and-cut algorithm for the vehicle routing problem with multiple use of vehicles," *International Journal of Lean Thinking*, vol. 6, no. 1, pp. 21–46, 2015.

[105] J. C. Rivera, H. M. Afsar, and C. Prins, "Multistart evolutionary local search for a disaster relief problem," in *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer, 2013, pp. 129–141.

[106] G. Radzki, A. Thibbotuwawa, and G. Bocewicz, "Uavs flight routes optimization in changing weather conditions—constraint programming approach," *Applied Computer Science*, vol. 15, no. 3, pp. 5–20, 2019.

[107] A. Thibbotuwawa, G. Bocewicz, B. Zbigniew, and P. Nielsen, "A solution approach for uav fleet mission planning in changing weather conditions," *Applied Sciences*, vol. 9, no. 19, p. 3972, 2019.

[108] S. J. Kim, G. J. Lim, J. Cho, and M. J. Côté, "Drone-aided healthcare services for patients with chronic diseases in rural areas," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 1, pp. 163–180, 2017.

[109] D. Vural, R. F. Dell, and E. Kose, "Locating unmanned aircraft systems for multiple missions under different weather conditions," *Operational Research*, pp. 1–20, 2019.

[110] D. Kim, K. Lee, and I. Moon, "Stochastic facility location model for drones considering uncertain flight distance," *Annals of Operations Research*, vol. 283, no. 1, pp. 1283–1302, 2019.

[111] S. J. Kim, G. J. Lim, and J. Cho, "Drone flight scheduling under uncertainty on battery duration and air temperature," *Computers & Industrial Engineering*, vol. 117, pp. 291–302, 2018.

[112] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations Research*, vol. 58, no. 3, pp. 595–612, 2010.

[113] D. Bertsimas, V. Gupta, and N. Kallus, "Data-driven robust optimization," *Mathematical Programming*, vol. 167, no. 2, pp. 235–292, 2018.

[114] C. Ning and F. You, "Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods," *Computers & Chemical Engineering*, vol. 112, pp. 190–210, 2018.

[115] A. Chassein, T. Dokka, and M. Goerigk, "Algorithms and uncertainty sets for data-driven robust shortest path problems," *European Journal of Operational Research*, vol. 274, no. 2, pp. 671–686, 2019.

[116] Z. Chen, M. Sim, and P. Xiong, "Robust stochastic optimization made easy with RSOME." *Management Science*, 2020, forthcoming.

[117] Z. Hao, L. He, Z. Hu, and J. Jiang, "Robust vehicle pre-allocation with uncertain covariates," *Production and Operations Management*, vol. 29, no. 4, pp. 955–972, 2020.

[118] G. Perakis, M. Sim, Q. Tang, and P. Xiong, "Robust pricing and production with information partitioning and adaptation." 2020, working paper, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3305039.

[119] F. Liberatore, M. P. Scaparra, and M. S. Daskin, "Hedging against disruptions with ripple effects in location analysis," *Omega*, vol. 40, no. 1, pp. 21–30, 2012.

[120] NetworkX, "NetworkX documentation," 2015, accessed September 17, 2018, https://networkx.github.io/documentation/networkx-1.10/index.html.

[121] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable robust solutions of uncertain linear programs," *Mathematical Programming*, vol. 99, no. 2, pp. 351–376, 2004.

[122] B. L. Gorissen, İ. Yanıkoğlu, and D. den Hertog, "A practical guide to robust optimization," *Omega*, vol. 53, pp. 124–137, 2015.

[123] D. Bertsimas and V. Goyal, "On the power and limitations of affine policies in two-stage adaptive optimization," *Mathematical programming*, vol. 134, no. 2, pp. 491–531, 2012.

[124] Y. An and B. Zeng, "Exploring the modeling capacity of two-stage robust optimization: Variants of robust unit commitment model," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 109–122, 2015.

[125] A. Ardestani-Jaafari and E. Delage, "Linearized robust counterparts of two-stage robust optimization problems with applications in operations management," *NFORMS Journal on Computing*, 2019, forthcoming.

[126] X. Qin, X. Liu, and L. Tang, "A two-stage stochastic mixed-integer program for the capacitated logistics fortification planning under accidental disruptions," *Computers & Industrial Engineering*, vol. 65, no. 4, pp. 614–623, 2013.

[127] L. Tang, C. Zhu, Z. Lin, J. Shi, and W. Zhang, "Reliable facility location problem with facility protection," *PloS one*, vol. 11, no. 9, 2016.

[128] M. Lim, M. S. Daskin, A. Bassamboo, and S. Chopra, "A facility reliability problem: Formulation, properties, and algorithm," *Naval Research Logistics (NRL)*, vol. 57, no. 1, pp. 58–70, 2010.

[129] F. Liberatore, M. P. Scaparra, and M. S. Daskin, "Analysis of facility protection strategies against an uncertain number of attacks: The stochastic r-interdiction median problem with fortification," *Computers & Operations Research*, vol. 38, no. 1, pp. 357–366, 2011.

[130] Q. Li, B. Zeng, and A. Savachkin, "Reliable facility location design under disruptions," *Computers & Operations Research*, vol. 40, no. 4, pp. 901–909, 2013.

[131] K. Zheng and L. A. Albert, "An exact algorithm for solving the bilevel facility interdiction and fortification problem," *Operations Research Letters*, vol. 46, no. 6, pp. 573–578, 2018.

[132] C. Cheng, M. Qi, Y. Zhang, and L.-M. Rousseau, "A two-stage robust approach for the reliable logistics network design problem," *Transportation Research Part B: Methodological*, vol. 111, pp. 185–202, 2018.

[133] H. Min and G. Zhou, "Supply chain modeling: past, present and future," *Computers & Industrial Engineering*, vol. 43, no. 1, pp. 231–249, 2002.

[134] M. S. Pishvaee, R. Z. Farahani, and W. Dullaert, "A memetic algorithm for bi-objective integrated forward/reverse logistics network design," *Computers & Operations Research*, vol. 37, no. 6, pp. 1100–1112, 2010.

[135] S. A. Alumur, S. Nickel, and F. Saldanha-da Gama, "Hub location under uncertainty," *Transportation Research Part B: Methodological*, vol. 46, no. 4, pp. 529–543, 2012.

[136] L. Zhao and B. Zeng, "Robust unit commitment problem with demand response and wind energy," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–8.

[137] I. Contreras, J.-F. Cordeau, and G. Laporte, "The dynamic uncapacitated hub location problem," *Transportation Science*, vol. 45, no. 1, pp. 18–32, 2011.

[138] Y. Zhang, L. V. Snyder, M. Qi, and L. Miao, "A heterogeneous reliable location model with risk pooling under supply disruptions," *Transportation Research Part B: Methodological*, vol. 83, pp. 151–178, 2016.

[139] G. J. Leishman, *Principles of helicopter aerodynamics, 2nd edition.* Cambridge university press, New York, 2006.

[140] G. Desaulniers, O. B. Madsen, and S. Ropke, "Chapter 5: The vehicle routing problem with time windows," in *Vehicle Routing: Problems, Methods, and Applications, 2nd edition.* SIAM, Philadelphia, 2014, pp. 119–159.

[141] F. Semet, P. Toth, and D. Vigo, "Chapter 2: Classical exact algorithms for the capacitated vehicle routing problem," in *Vehicle Routing: Problems, Methods, and Applications, Second Edition.* SIAM, Philadelphia, 2014, pp. 37–57.

[142] F. A. Santos, G. R. Mateus, and A. S. da Cunha, "A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem," *Transportation Science*, vol. 49, no. 2, pp. 355–368, 2014.

[143] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of applied optimization*, pp. 65–77, 2002.

[144] G. Laporte, "Generalized subtour elimination constraints and connectivity constraints," *Journal of the Operational Research Society*, vol. 37, no. 5, pp. 509–514, 1986.

[145] J. Lysgaard, A. N. Letchford, and R. W. Eglese, "A new branch-and-cut algorithm for the capacitated vehicle routing problem," *Mathematical Programming*, vol. 100, no. 2, pp. 423–445, 2004.

[146] Y. Zhang, R. Baldacci, M. Sim, and J. Tang, "Routing optimization with time windows under uncertainty," *Mathematical Programming*, vol. 175, no. 1-2, pp. 263–305, 2019.

[147] Y. Zhang, Z. Zhang, A. Lim, and M. Sim, "Robust data-driven vehicle routing with time windows," *Operations Research*, 2020, forthcoming.

[148] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems," *INFORMS Journal on Computing*, vol. 26, no. 1, pp. 103–120, 2014.

[149] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally robust convex optimization," *Operations Research*, vol. 62, no. 6, pp. 1358–1376, 2014.

[150] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, no. 4, pp. 769–805, 1998.

[151] N. Lavars, "Amazon to begin testing new delivery drones in the US." 2015, accessed February 15, 2020, https://newatlas.com/amazon-new-delivery-drones-us-faa-approval/36957/.

[152] S. Burns, "Drone meets delivery truck." 2017, accessed February 15, 2020, https://www.ups.com/us/es/services/knowledge-center/article.page?kid=cd18bdc2.

[153] D. Lee, "Amazon to deliver by drone 'within months'." 2019, accessed February 15, 2020, https://www.bbc.com/news/technology-48536319.

[154] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[155] O. Ergun, G. Karakus, P. Keskinocak, J. Swann, and M. Villarreal, "Operations research to improve disaster supply chain management," *Wiley encyclopedia of operations research and management science*, 2010.

[156] K. Kepaptsoglou, G. Fountas, and M. G. Karlaftis, "Weather impact on containership routing in closed seas: A chance-constraint optimization approach," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 139–155, 2015.

[157] A. Gupte, S. Ahmed, M. S. Cheon, and S. Dey, "Solving mixed integer bilinear problems using milp formulations," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 721–744, 2013.

[158] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems," *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.

[159] D. Bertsimas, M. Sim, and M. Zhang, "Adaptive distributionally robust optimization," *Management Science*, vol. 65, no. 2, pp. 604–618, 2019.

[160] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1-2, pp. 115–166, 2018.

[161] A. Beck and A. Ben-Tal, "Duality in robust optimization: primal worst equals dual best," *Operations Research Letters*, vol. 37, no. 1, pp. 1–6, 2009.

APPENDIX A SUPPLEMENT TO CHAPTER 3

Section A provides proofs of lemmas. Sections A and A present the duality-based C&CG algorithm and BD method for the robust UFLP and CFLP, respectively. Section A gives the reformulation of the AARC models. Section A presents the models for the robust uncapacitated and capacitated p-median problems. Section A gives the detailed results of the numerical tests discussed in Section 3.4. The results can also be downloaded from `https://sites.google.com/view/chengchun/instances`.

A.1 Proofs of Lemmas

A.1.1 Proof of Lemma 3.2.1

Suppose the number of open facilities is m, i.e., $\sum_{j\in J}\hat{y}_j = m$. Since the worst-case scenario always occurs at the extreme points, there will be exactly k disrupted facilities in a worst-case scenario. Figure A.1 gives an example where $m > k$. We index the facilities so that the first m are open and the rest are closed. Without loss of generality, we assume that the first k facilities are disrupted in scenario $\mathbf{z}^1$ and that facilities 2, ..., k and $m+1$ are disrupted in scenario $\mathbf{z}^2$. Since more open facilities are disrupted in scenario $\mathbf{z}^1$, the customers have more options in scenario $\mathbf{z}^2$, leading to $B_2 \leq B_1$. Therefore, we have $B_1 \geq B_2$ for $m > k$. When $m \leq k$, all the demand will be left unsatisfied and we have $B_1 = B_2$.

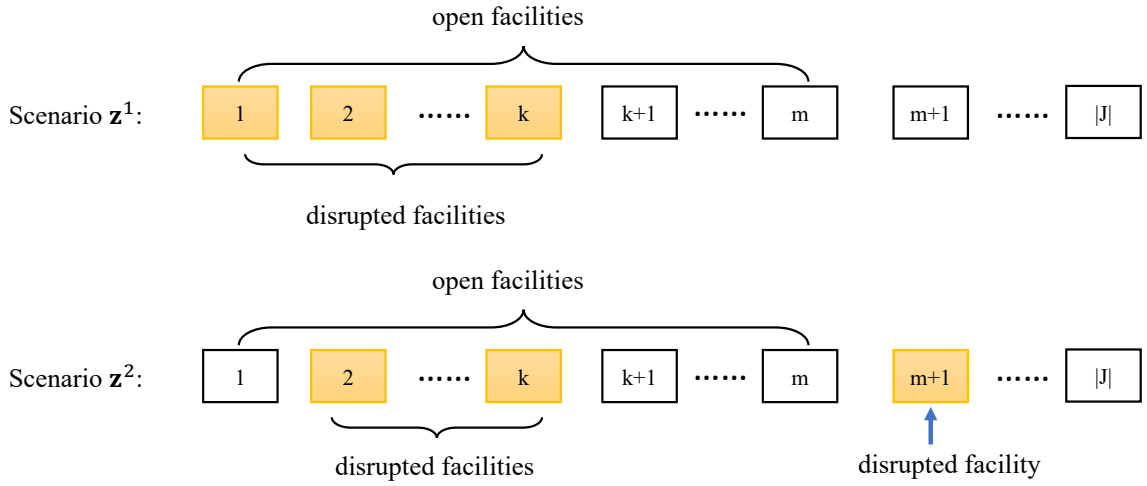


Figure A.1 Illustration for the proof of Lemma 3.2.1

A.1.2 Proof of Lemma 3.3.1

To proof $\mathbb{Z}'(k)$ is equivalent to $\mathbb{Z}(k)$, we need to show that even though we relax the variables $\mathbf{z}$ to be continuous, the optimal values of $\mathbf{z}$ in a worst-case scenario still take integer values (0 or 1). This can be proofed through the subproblems of the duality-based C&CG algorithms (refer to A and A). Specifically, for the UFLP, we can rewrite the objective function (A.3a) as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{j\in J}\left(\sum_{i\in I}\hat{y}_j\beta_{ij}(1-z_j)\right). \tag{A.1}$$

Constraints (A.3b)–(A.3g) indicate that variables $\mathbf{z}$ are not linked to other variables (i.e., $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$). Since the subproblem is a maximization problem, the third term subtracted from Equation (A.1) is expected to be as small as possible. To achieve this, we can rank $\sum_{i\in I}\hat{y}_j\beta_{ij}(\geq 0)$ for each j, and for the first k largest values, z_j would take the value of 1 (even though z_j is relaxed to be continuous) at optimum. For other values, z_j would be 0. Similarly, for the CFLP, we can rewrite the objective function (A.7) as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{j\in J}\left(\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j(1-z_j)\right). \tag{A.2}$$

If $\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j \leq 0$, z_j would be 0 at optimum. For those with $\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j > 0$, the same method describe above for the UFLP applies.

A.2 Duality-Based C&CG Algorithm and Benders Decomposition Method for Robust UFLP

A.2.1 Duality-Based C&CG Algorithm

Master Problem. The master problem is defined by Equations (3.8a)–(3.8g).

Subproblem. Let λ and β be the dual variables of constraints (3.3b) and (3.3c), respectively. The dual problem of the inner minimization problem can be written as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(1-z_j)\beta_{ij}, \tag{A.3a}$$

$$\text{s.t.}\quad \lambda_i - \beta_{ij} \leq h_i d_{ij} \qquad \forall i\in I, j\in J, \tag{A.3b}$$

$$\lambda_i \leq p_i h_i \qquad \forall i\in I, \tag{A.3c}$$

$$\sum_{j\in J} z_j \leq k, \tag{A.3d}$$

$$z_j \in \{0,1\} \qquad \forall i\in I, \tag{A.3e}$$

$$\lambda_i \geq 0 \qquad \forall i \in I, \tag{A.3f}$$

$$\beta_{ij} \geq 0 \qquad \forall i \in I, j \in J. \tag{A.3g}$$

Since the nonlinear term $z_j\beta_{ij}$ is the product of a binary variable z_j and a continuous variable β_{ij}, we can linearize it by introducing a new variable $\pi_{ij} = z_j\beta_{ij}$ and adding the following constraints:

$$\begin{aligned}
&\pi_{ij} \geq 0 && \forall i \in I, j \in J,\\
&\pi_{ij} \leq \beta_{ij} && \forall i \in I, j \in J,\\
&\pi_{ij} \leq \mathbb{M}_{ij} z_j && \forall i \in I, j \in J,\\
&\pi_{ij} \geq \beta_{ij} + \mathbb{M}_{ij}(z_j - 1) && \forall i \in I, j \in J,
\end{aligned} \tag{A.4}$$

where $\mathbb{M}_{ij} = \max\{h_i(p'_i - d_{ij}), 0\}$. Therefore, the full subproblem is

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\pi}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(\beta_{ij} - \pi_{ij}), \tag{A.5}$$

subject to constraints (A.3b)–(A.3g) and (A.4).

A.2.2 Benders Decomposition Method

Master Problem. The master problem of the Benders decomposition method is

$$\begin{aligned}
&\phi = \min_{\mathbf{y},s} s,\\
&\text{s.t.} \quad s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\hat{\lambda}^l_i - \sum_{i\in I}\sum_{j\in J}\hat{\beta}^l_{ij} y_j(1 - \hat{z}^l_j) && \forall l \in \{1, ..., n\},\\
&\qquad y_j \in \{0, 1\} && \forall j \in J,
\end{aligned} \tag{A.6}$$

where $\hat{\lambda}^l, \hat{\beta}^l$, and $\hat{z}^l$ are obtained at the lth iteration by solving the subproblem.

Subproblem. The subproblem is defined by Equations (A.3b)–(A.3g), (A.4), and (A.5).

A.3 Duality-Based C&CG Algorithm and Benders Decomposition Method for Robust CFLP

A.3.1 Duality-Based C&CG Algorithm

Master Problem. The master problem is defined by Equations (3.8a)–(3.8g) and (3.10).

Subproblem. Let γ be the dual variable of constraints (3.6). The resulting dual problem

can be written as follows:

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(1-z_j)\beta_{ij} - \sum_{j\in J} C_j\hat{y}_j\gamma_j(1-z_j), \tag{A.7}$$

$$\begin{aligned} \text{s.t.}\quad & \lambda_i - \beta_{ij} - h_i\gamma_j \le h_i d_{ij} && \forall i\in I, j\in J, \\ & \gamma_j \ge 0 && \forall j\in J, \\ & \text{and (A.3c)–(A.3g).} \end{aligned} \tag{A.8}$$

There are two nonlinear terms in the objective function (A.7), i.e., $z_j\beta_{ij}$ and $\gamma_j z_j$. We can use the technique in Section A to linearize the term $z_j\beta_{ij}$. For the term $\gamma_j z_j$, we introduce a new variable $\zeta_j = \gamma_j z_j$ and add the following constraints:

$$\begin{aligned} & \zeta_j \ge 0 && \forall j\in J, \\ & \zeta_j \le \gamma_j && \forall j\in J, \\ & \zeta_j \le \mathbb{M}_j z_j && \forall j\in J, \\ & \zeta_j \ge \gamma_j + \mathbb{M}_j(z_j - 1) && \forall j\in J, \end{aligned} \tag{A.9}$$

where $\mathbb{M}_j = \max\{\max_i(p_i - d_{ij}), 0\}$.

Therefore, the full subproblem is

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\zeta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(\beta_{ij} - \pi_{ij}) - \sum_{i\in J} C_j\hat{y}_j(\gamma_j - \zeta_j), \tag{A.10}$$

subject to constraints (A.4) and (A.8)–(A.9).

A.3.2 Benders Decomposition Method

Master Problem. The master problem of the Benders decomposition method is

$$\begin{aligned} & \phi = \min_{\mathbf{y},s} s, \\ & \text{s.t.}\quad s \ge \sum_{j\in J} f_j y_j + \sum_{i\in I}\hat{\lambda}_i^l - \sum_{i\in I}\sum_{j\in J} y_j(1-\hat{z}_j^l)\hat{\beta}_{ij}^l - \sum_{j\in J} C_j y_j\hat{\gamma}_j^l(1-\hat{z}_j^l) && \forall l\in\{1,...,n\}, \\ & \quad y_j\in\{0,1\} && \forall j\in J, \end{aligned}$$

where $\hat{\lambda}^l, \hat{\beta}^l$, and $\hat{z}^l$ are obtained at the lth iteration by solving the subproblem.

Subproblem. The subproblem is defined by Equations (A.4) and (A.8)–(A.10).

A.4 Reformulation of the AARC Models

The AARC model for the robust UFLP can be reformulated as

$$\min_{\substack{\mathbf{y},\mathbf{W},\mathbf{w},\mathbf{A},\mathbf{a},s,\delta,\\ \boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\theta},\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\varsigma},\boldsymbol{\pi},\boldsymbol{\nu}}} s, \tag{A.11a}$$

$$\text{s.t. } s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} w_{ij} + \sum_{i\in I} p_i h_i a_i + k\delta + \sum_{e\in J} \alpha_e, \tag{A.11b}$$

$$\delta + \alpha_e \geq \sum_{i\in I}\sum_{j\in J} h_i d_{ij} W_{ije} + \sum_{i\in I} p_i h_i A_{ie} \qquad \forall e \in J, \tag{A.11c}$$

$$\sum_{j\in J} w_{ij} + a_i - k\xi_i - \sum_{e\in J} \eta_{ie} \geq 1 \qquad \forall i \in I, \tag{A.11d}$$

$$\xi_i + \eta_{ie} \geq -\sum_{j\in J} W_{ije} - A_{ie} \qquad \forall i \in I, e \in J, \tag{A.11e}$$

$$-k\theta_{ij} - \sum_{e\in J} \mu_{ije} \geq -y_j + w_{ij} \qquad \forall i \in I, j \in J, \tag{A.11f}$$

$$\theta_{ij} + \mu_{ije} \geq W_{ije} + y_j \qquad \forall i \in I, j, e \in J, e = j, \tag{A.11g}$$

$$\theta_{ij} + \mu_{ije} \geq W_{ije} \qquad \forall i \in I, j, e \in J, e \neq j, \tag{A.11h}$$

$$w_{ij} - k\sigma_{ij} - \sum_{e\in J} \varsigma_{ije} \geq 0 \qquad \forall i \in I, j \in J, \tag{A.11i}$$

$$\sigma_{ij} + \varsigma_{ije} \geq -W_{ije} \qquad \forall i \in I, j, e \in J, \tag{A.11j}$$

$$-k\pi_i - \sum_{e\in J} \nu_{ie} + a_i \geq 0 \qquad \forall i \in I, \tag{A.11k}$$

$$\pi_i + \nu_{ie} \geq -A_{ie} \qquad \forall i \in I, e \in J, \tag{A.11l}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{A.11m}$$

$$\delta, \alpha_e, \xi_i, \eta_{ie}, \theta_{ij}, \mu_{ije}, \sigma_{ij}, \varsigma_{ije}, \pi_i, \nu_{ie} \geq 0 \qquad \forall i \in I, j, e \in J. \tag{A.11n}$$

The AARC model for the robust CFLP can be reformulated with (A.11) and the constraints

$$k\rho_j + \sum_{e\in J} \Gamma_{ej} + \sum_{i\in I} h_i w_{ij} \leq C_i y_j \qquad \forall j \in J,$$

$$\rho_j + \Gamma_{ej} \geq \sum_{i\in I} h_i W_{ije} + C_j y_j \qquad \forall e, j \in J, e = j$$

$$\rho_j + \Gamma_{ej} \geq \sum_{i\in I} h_i W_{ije} \qquad \forall e, j \in J, e \neq j$$

$$\rho_j, \Gamma_{ej} \geq 0 \qquad \forall e \in J, j \in J.$$

A.5 Robust Uncapacitated/Capacitated p-median Problem

The parameter ϱ is the weight of the worst-case cost. The variable b_{ij} is the fraction of demand from customer $i \in I$ that is satisfied by facility $j \in J$ in a disruptive scenario. The variable g_i is the unsatisfied portion of demand at customer $i \in I$ in a disruptive scenario. The definitions of the other parameters and variables are the same as in Section 3.2.1.

The uncapacitated p-median problem under disruptions can be formulated as

$$\min_{\mathbf{x},\mathbf{y}}(1-\varrho)\sum_{i\in I}\sum_{j\in J} d_{ij}h_i x_{ij} + \varrho \max_{\mathbf{z}\in\mathbb{Z}(k)} \min_{(\mathbf{b},\mathbf{g})\in S(\mathbf{y},\mathbf{z})} (\sum_{i\in I}\sum_{j\in J} d_{ij}h_i b_{ij} + \sum_{i\in I} p_i h_i g_i), \tag{A.12}$$

$$\begin{aligned}
\text{s.t. } & x_{ij} \le y_j && \forall i \in I, j \in J, \\
& \sum_{j\in J} x_{ij} = 1 && \forall i \in I, \\
& \sum_{j\in J} y_j = p, && \\
& x_{ij} \ge 0, && \forall i \in I, j \in J, \\
& y_j \in \{0,1\} && \forall j \in J,
\end{aligned} \tag{A.13}$$

$$\begin{aligned}
\text{where } S(\mathbf{y},\mathbf{z}) = \Big\{ & b_{ij} \le 1 - z_j && \forall i \in I, j \in J, \\
& b_{ij} \le y_j && \forall i \in I, j \in J, \\
& \sum_{j\in J} b_{ij} + g_i = 1 && \forall i \in I, \\
& b_{ij} \ge 0, && \forall i \in I, j \in J, \\
& g_i \ge 0, && \forall i \in I \Big\}.
\end{aligned} \tag{A.14}$$

The capacitated p-median problem under disruptions is

$$\min_{\mathbf{x},\mathbf{y}}(1-\varrho)\sum_{i\in I}\sum_{j\in J} d_{ij}h_i x_{ij} + \varrho \max_{\mathbf{z}\in\mathbb{Z}(k)} \min_{(\mathbf{b},\mathbf{g})\in S^C(\mathbf{y},\mathbf{z})} (\sum_{i\in I}\sum_{j\in J} d_{ij}h_i b_{ij} + \sum_{i\in I} p_i h_i g_i) \tag{A.15}$$

$$\text{s.t. (A.13) and } \sum_{i\in I} h_i x_{ij} \le C_j y_j, \forall j \in J, \tag{A.16}$$

$$\text{where } S^C(\mathbf{y},\mathbf{z}) = \Big\{\text{(A.14) and } \sum_{i\in I} h_i b_{ij} \le C_j y_j, \forall j \in J\Big\}. \tag{A.17}$$

A.6 Detailed Results of Numerical Tests in Section 3.4

Table A.1 Performance of exact algorithms for the reliable UFLP ($k = 2$)

		$p = p^{0.8}$										$p = p^{max}$									
		C&CG-E				C&CG-D			Benders decomposition			C&CG-E				C&CG-D			Benders decomposition		
$\|J\|$	$\|I\|$	UB	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU	UB	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU
10	10	498982	0.0	17	6.0	0.0	18	6.7	0.0	138	63.2	575257	0.0	13	4.9	0.0	16	6.0	0.0	247	216.3
	15	528912	0.0	15	9.5	0.0	17	9.1	0.0	159	97.2	612406	0.0	15	8.4	0.0	20	13.5	0.0	297	303.1
	20	603210	0.0	13	10.2	0.0	16	10.9	0.0	227	186.0	667035	0.0	9	7.1	0.0	15	9.9	0.0	180	186.6
	25	656478	0.0	18	28.6	0.0	23	27.5	0.0	316	540.8	716223	0.0	11	11.0	0.0	19	18.8	0.0	376	829.4
	30	725463	0.0	18	33.1	0.0	23	35.7	0.0	378	899.0	767841	0.0	10	11.0	0.0	16	17.1	0.0	412	1022.1
	35	743224	0.0	14	25.9	0.0	22	36.3	0.0	345	847.9	794982	0.0	11	18.2	0.0	16	23.7	0.0	413	1148.2
	40	765901	0.0	13	26.3	0.0	22	43.5	0.0	349	877.2	812675	0.0	10	15.0	0.0	16	24.7	0.0	450	1483.7
	45	782036	0.0	13	27.8	0.0	22	48.8	0.0	350	919.3	824046	0.0	10	19.0	0.0	14	22.4	0.0	452	1528.0
	49	785576	0.0	13	33.5	0.0	22	50.2	0.0	364	986.7	827587	0.0	9	16.1	0.0	12	20.0	0.0	449	1363.3
15	15	507674	0.0	33	73.1	0.0	34	68.0	0.0	1144	3254.7	612406	0.0	34	132.6	0.0	49	197.7	0.0	2931	17019.5
	20	560086	0.0	30	134.2	0.0	34	128.6	0.0	1332	4654.4	667035	0.0	27	131.1	0.0	37	148.6	0.0	3689	32234.0
	25	619200	0.0	31	239.4	0.0	37	203.0	0.0	2063	10909.5	716223	0.0	30	252.0	0.0	47	422.1	0.0	3965	30600.1
	30	692533	0.0	29	360.3	0.0	47	607.8	0.0	2644	18991.4	760569	0.0	28	299.5	0.0	43	427.7	0.0	4176	38070.5
	35	711992	0.0	29	465.2	0.0	44	654.2	0.0	2751	23261.9	774758	0.0	24	256.2	0.0	33	279.6	0.0	4708	51840.3
	40	749365	0.0	28	447.4	0.0	49	846.0	0.0	3278	35217.2	790384	0.0	18	166.4	0.0	29	243.2	0.0	5164	68536.5
	45	772850	0.0	30	713.2	0.0	50	1118.0	0.0	3693	49316.3	800817	0.0	17	173.5	0.0	28	250.6	0.1	4871	86416.6
	49	775470	0.0	30	866.0	0.0	51	1417.1	0.0	3917	60131.2	804014	0.0	17	209.1	0.0	28	291.5	0.0	4519	79727.6
20	20	582844	0.0	41	502.0	0.0	48	613.6	0.0	4464	49692.6	659613	0.0	42	688.4	0.0	71	1133.8	9.9	5680	86416.0
	25	636899	0.0	45	1301.3	0.0	63	1696.6	4.4	5363	86421.9	699041	0.0	42	1037.5	0.0	64	1338.7	11.2	6079	86422.8
	30	704875	0.0	44	2025.1	0.0	72	3484.1	8.8	5121	86413.2	738415	0.0	30	691.5	0.0	52	1184.6	11.6	5812	86400.4
	35	725853	0.0	44	3151.9	0.0	70	4303.2	9.5	4846	86417.5	753386	0.0	26	740.4	0.0	47	1249.1	13.1	4750	86416.5
	40	750490	0.0	42	3419.2	0.0	75	7522.3	11.8	4574	86437.5	769950	0.0	21	535.0	0.0	44	1275.4	16.5	4390	86428.5
	45	769946	0.0	43	5225.1	0.0	77	10202.0	12.2	4298	86421.3	782477	0.0	22	789.2	0.0	45	1648.3	15.8	4262	86426.9
	49	768482	0.0	43	6268.0	0.0	75	11054.3	12.8	4171	86417.9	787140	0.0	22	879.8	0.0	44	1804.4	16.1	4115	86423.7
25	25	644574	0.0	80	6779.1	0.0	111	9482.7	31.8	4564	86421.4	681276	0.0	41	1766.2	0.0	96	6867.7	28.8	4664	86423.8
	30	700445	0.0	70	9845.3	0.0	120	21201.6	32.2	4263	86400.2	713881	0.0	31	1259.2	0.0	65	3415.4	30.1	4403	86426.0
	35	719859	0.0	78	19168.8	0.0	118	24465.1	33.7	4033	86441.5	732067	0.0	32	1940.6	0.0	58	3537.1	39.2	4254	86408.9
	40	743654	0.0	72	23555.0	0.0	112	34208.0	33.6	4056	86404.1	747159	0.0	29	1729.1	0.0	57	3917.2	37.9	4055	86412.3
	45	755572	0.0	63	19870.3	0.0	101	33328.8	36.1	3208	86449.1	757796	0.0	30	2428.7	0.0	54	4381.3	42.0	3881	86433.8
	49	758595	0.0	63	24968.2	0.0	99	37129.2	35.6	3110	86424.2	761579	0.0	30	2457.7	0.0	53	5263.1	33.9	3632	86423.3
30	30	678930■	0.0	60	9823.6	0.0	110	21793.3	44.0	3261	86451.5	678930■	0.0	39	3700.4	0.0	88	13340.7	51.5	3470	86411.2
	35	692882	0.0	58	16109.7	0.0	113	33509.5	44.6	3183	86408.6	693170	0.0	39	5255.6	0.0	85	19314.8	48.7	3456	86417.3
	40	706009■	0.0	49	11674.5	0.0	96	24403.3	48.3	2980	86455.1	706009■	0.0	33	4652.8	0.0	87	24368.4	50.3	3290	86406.3
	45	715100■	0.0	39	8616.8	0.0	85	24132.3	46.7	3903	86434.8	715100■	0.0	34	7432.1	0.0	79	19160.9	51.4	3200	86418.8
	49	717993■	0.0	38	9656.3	0.0	77	23948.6	47.7	2738	86416.5	717993■	0.0	31	5780.0	0.0	71	17926.3	50.1	2978	86410.7
Average		692913	**0.0**$^{(0)}$	**38.5**	**5298.8**	**0.0**$^{(0)}$	61.5	9479.7	14.1$^{(17)}$	2731.0	49430.9	731978	**0.0**$^{(0)}$	**24.8**	**1299.9**	**0.0**$^{(0)}$	45.7	3815.5	15.9$^{(19)}$	3247.7	56230.1

$^{(-)}$ indicates the number of instances (out of 35) that are not solved to optimality.

Table A.2 Performance of C&CG algorithms for the reliable UFLP ($k = 3$)

		$p = p^{0.8}$										$p = p^{max}$									
		C&CG-E					C&CG-D					C&CG-E					C&CG-D				
$\|J\|$	$\|I\|$	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU
10	10	581419	581419	0.0	58	70.4	581419	581419	0.0	61	83.3	664644	664644	0.0	48	72.4	664644	664644	0.0	55	84.0
	15	606687	606687	0.0	49	72.6	606687	606687	0.0	51	91.4	708835	708835	0.0	38	68.2	708835	708835	0.0	49	80.1
	20	685150	685150	0.0	38	77.8	685150	685150	0.0	42	83.6	767025	767025	0.0	27	55.6	767025	767025	0.0	38	61.8
	25	727012	727012	0.0	37	83.0	727012	727012	0.0	42	88.1	801633	801633	0.0	23	48.0	801633	801633	0.0	31	52.3
	30	793175	793175	0.0	38	109.8	793175	793175	0.0	48	143.7	857585	857585	0.0	25	61.4	857585	857585	0.0	35	76.0
	35	824189	824189	0.0	35	132.4	824189	824189	0.0	44	159.7	881571	881571	0.0	21	50.5	881571	881571	0.0	26	50.5
	40	860196	860196	0.0	38	170.0	860196	860196	0.0	47	190.7	906841	906841	0.0	23	74.5	906841	906841	0.0	29	72.3
	45	877412	877412	0.0	37	217.2	877412	877412	0.0	50	274.8	922745	922745	0.0	22	72.9	922745	922745	0.0	33	124.3
	49	880912	880912	0.0	39	311.4	880912	880912	0.0	48	276.3	928582	928582	0.0	22	81.7	928582	928582	0.0	33	137.6
15	15	573674	573674	0.0	132	1357.0	573674	573674	0.0	132	1194.5	701244	701244	0.0	118	3543.0	701244	701244	0.0	174	4400.9
	20	626086	626086	0.0	97	1966.1	626086	626086	0.0	101	1937.9	758067	758067	0.0	93	2551.1	758067	758067	0.0	134	3472.1
	25	685200	685200	0.0	100	3282.7	685200	685200	0.0	109	3028.5	797761	797761	0.0	88	3655.6	797761	797761	0.0	124	4584.5
	30	765484	765484	0.0	104	6912.4	765484	765484	0.0	135	8803.8	845196	845196	0.0	82	5095.0	845196	845196	0.0	115	5179.9
	35	792141	792141	0.0	105	14979.4	792141	792141	0.0	138	12003.9	868055	868055	0.0	83	6906.9	868055	868055	0.0	108	5341.1
	40	831603	831603	0.0	99	11113.8	831603	831603	0.0	154	18134.2	891397	891397	0.0	81	7962.7	891397	891397	0.0	115	8712.9
	45	849287	849287	0.0	95	15875.8	849287	849287	0.0	149	20136.8	906789	906789	0.0	80	8499.3	906789	906789	0.0	114	13163.6
	49	854572	854572	0.0	99	18355.4	854572	854572	0.0	153	28532.6	912522	912522	0.0	79	10261.3	912522	912438	0.0	113	16003.2
20	20	648844	648844	0.0	149	16725.3	648844	648844	0.0	163	16450.0	758067	758067	0.0	211	72827.6	770378	750648	2.6	294	86521.7
	25	704705	704705	0.0	171	47133.7	704705	704705	0.0	204	55802.5	805544	791509	1.7	181	86957.6	803896	784845	2.4	251	86852.3
	30	810980	785113	3.2	167	87135.1	791956	776424	2.0	207	86832.0	858712	833995	2.9	158	86777.1	856296	825437	3.6	220	87275.0
	35	817898	791904	3.2	136	87257.7	806827	789331	2.2	184	87075.0	869673	854781	1.7	147	86491.7	880244	844644	4.0	196	87117.2
	40	845102	817182	3.3	125	87621.5	835486	808517	3.2	168	86704.8	905180	868811	4.0	128	88121.3	899316	863649	4.0	179	87287.3
	45	866326	832353	3.9	115	87235.1	858945	826816	3.7	148	86730.0	920423	882415	4.1	122	86526.1	913094	871764	4.5	164	86731.2
	49	866587	829993	4.2	106	87119.3	859647	825454	4.0	137	87334.6	916597	878310	4.2	110	88754.1	917465	874738	4.7	152	86576.5
25	25	722512	689040	4.6	187	86979.5	733612	689372	6.0	216	87123.9	792985	750907	5.3	149	87172.2	789422	739721	6.3	185	86990.8
	30	794483	740083	6.9	136	88124.1	800686	737852	7.9	170	86412.2	840771	794180	5.5	132	87491.8	840071	779216	7.2	165	87885.6
	35	816773	756471	7.4	119	88092.5	816073	752967	7.7	150	87588.9	862236	808366	6.3	112	88231.5	862236	796351	7.6	146	87823.3
	40	847408	775418	8.5	104	88103.2	845491	776465	8.2	137	86415.3	884444	821270	7.1	101	87814.8	886744	808443	8.8	139	86836.5
	45	871143	792717	9.0	95	88162.1	866575	780501	9.9	127	87966.3	897894	831530	7.4	92	88709.8	900894	812631	9.8	131	86658.4
	49	862157	792966	8.0	90	87301.8	871816	778058	10.8	118	86448.5	902493	835060	7.5	84	88990.3	901964	813645	9.8	120	88107.0
30	30	813609	718209	11.7	101	86513.8	799948	698980	12.6	148	87619.2	814791	735832	9.7	90	88312.6	814791	720955	11.5	134	88155.5
	35	846305	732658	13.4	91	86953.4	820083	702054	14.4	129	86807.5	828479	747594	9.8	81	87793.7	828479	727200	12.2	119	87997.9
	40	836242	750071	10.3	80	86758.7	841188	717048	14.8	119	88657.6	862206	762115	11.6	75	87077.0	846266	743155	12.2	108	88816.9
	45	858815	760575	11.4	72	87335.6	850912	733986	13.7	111	86737.8	875317	767546	12.3	65	89438.7	858815	748117	12.9	99	88136.6
	49	856048	761031	11.1	67	87930.4	886954	731535	17.5	106	87721.6	892334	769751	13.7	64	90669.3	863784	750988	13.1	94	88068.6
Average		785718	756958	**3.4**$^{(16)}$	**94.6**	**43930.6**	784398	751232	4.0$^{(16)}$	121.3	44616.9	845961	817501	**3.3**$^{(17)}$	**87.3**	**46206.2**	844418	810759	3.9$^{(18)}$	120.6	46726.7

Table A.3 Performance of C&CG algorithms for the reliable UFLP ($k = 4$)

		$p = p^{0.8}$										$p = p^{max}$									
		C&CG-E					C&CG-D					C&CG-E					C&CG-D				
$\|J\|$	$\|I\|$	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU
10	10	652719	652719	0.0	127	248.8	652719	652719	0.0	128	278.2	741207	741207	0.0	102	199.2	741207	741207	0.0	109	241.1
	15	686383	686383	0.0	113	320.7	686383	686383	0.0	116	340.5	786586	786586	0.0	68	140.8	786586	786586	0.0	79	212.0
	20	767135	767135	0.0	91	279.5	767135	767135	0.0	95	302.0	847063	847063	0.0	54	135.3	847063	847063	0.0	62	173.6
	25	823612	823612	0.0	90	434.7	823612	823612	0.0	96	425.9	891203	891118	0.0	45	118.5	891203	891203	0.0	57	186.8
	30	889222	889222	0.0	81	458.1	889222	889222	0.0	86	446.8	946661	946661	0.0	43	148.6	946661	946661	0.0	54	206.2
	35	908393	908393	0.0	64	337.1	908393	908393	0.0	71	368.1	972479	972479	0.0	38	132.9	972479	972479	0.0	47	197.4
	40	932796	932796	0.0	62	372.8	932796	932796	0.0	73	456.4	997343	997343	0.0	42	242.9	997343	997343	0.0	50	276.9
	45	950012	950012	0.0	64	457.7	950012	950012	0.0	66	444.6	1012693	1012693	0.0	44	300.1	1012693	1012693	0.0	49	307.7
	49	953512	953512	0.0	58	440.5	953512	953512	0.0	70	530.8	1018168	1018168	0.0	43	327.7	1018168	1018082	0.0	45	285.3
15	15	616640	616640	0.0	266	6555.3	616640	616640	0.0	266	6151.2	772112	772112	0.0	318	28154.9	772112	772112	0.0	367	32761.7
	20	692686	692686	0.0	284	18649.8	692686	692686	0.0	287	18272.5	827965	827965	0.0	226	19305.3	827965	827965	0.0	277	25858.1
	25	751800	751800	0.0	260	27426.2	751800	751800	0.0	284	32804.0	868503	868503	0.0	202	24562.7	868503	868503	0.0	266	35849.5
	30	831484	831484	0.0	233	43702.1	831484	831484	0.0	280	60800.5	916396	916396	0.0	165	20214.9	916396	916396	0.0	220	30363.1
	35	863341	863341	0.0	250	75780.7	876741	861509	1.7	298	86496.4	939255	939255	0.0	146	21851.5	939255	939255	0.0	214	34753.1
	40	905103	900615	0.5	245	86940.9	904203	894399	1.1	271	86593.8	962980	962980	0.0	150	29073.6	962980	962980	0.0	199	40101.1
	45	929682	916226	1.5	225	86724.7	939386	903999	3.8	244	86884.9	978367	978367	0.0	150	40801.2	978367	978367	0.0	196	47514.0
	49	967709	913510	5.6	212	87601.1	968654	902762	6.8	225	87297.0	984016	984016	0.0	147	62716.4	984016	984016	0.0	193	54714.0
20	20	790772	690311	12.7	306	86935.8	906692	690947	23.8	312	86680.0	839765	784557	6.6	249	86416.0	839317	775392	7.6	277	86638.9
	25	821252	731608	10.9	233	86427.5	841041	729516	13.3	236	87047.2	928818	818538	11.9	211	86942.0	975148	806483	17.3	245	86663.5
	30	875893	802006	8.4	193	86982.8	900666	798865	11.3	197	86518.6	995368	859164	13.7	179	86493.1	936196	843025	10.0	213	87571.4
	35	902098	818283	9.3	161	86861.5	883898	814024	7.9	168	86945.8	1000096	874381	12.6	160	86891.2	981979	861768	12.2	192	86770.4
	40	961494	841341	12.5	142	87047.2	966625	835487	13.6	158	86532.5	1010560	893316	11.6	143	88061.7	1053160	872199	17.2	175	86662.2
	45	978418	854260	12.7	130	87500.5	978418	853611	12.8	145	86605.9	1083634	903137	16.7	134	87641.9	1067505	881901	17.4	174	87697.0
	49	979877	852206	13.0	122	86412.6	999274	848137	15.1	136	86529.7	1089583	906469	16.8	122	87451.1	1139595	881141	22.7	138	86958.6
25	25	843426	697507	17.3	208	87165.6	843426	697556	17.3	212	86442.0	1041729	757541	27.3	158	86630.1	1041729	757894	27.3	189	87437.6
	30	896936	752883	16.1	167	86814.5	896936	753585	16.0	176	86589.5	1002241	806261	19.6	140	87452.6	1012063	792404	21.7	174	86827.7
	35	1107169	769964	30.5	144	86866.1	1124969	770771	31.5	151	86766.6	1059920	820717	22.6	114	86794.6	1094104	805658	26.4	145	86936.7
	40	1049406	793694	24.4	126	87446.4	1078383	792430	26.5	141	87173.8	1141068	831098	27.2	105	87041.2	1084829	820063	24.4	128	87679.4
	45	1029487	814015	20.9	117	88401.2	1155412	807665	30.1	123	87607.9	1092934	844462	22.7	96	86981.5	1098172	827539	24.6	117	86794.8
	49	1048787	811759	22.6	106	88528.7	1147299	810564	29.4	117	87094.0	1098883	852410	22.4	93	87759.6	1144324	830971	27.4	109	87028.6
30	30	962846	733919	23.8	124	87964.2	937165	729950	22.1	133	86577.4	1001449	755008	24.6	102	88148.6	949555	748844	21.1	120	86588.2
	35	1029750	747162	27.4	112	87357.8	1039650	745552	28.3	114	88320.6	1211945	763357	37.0	89	89523.0	1011051	759084	24.9	107	86633.9
	40	1015524	764160	24.8	93	86409.0	991704	758720	23.5	104	89294.4	1125585	777107	31.0	82	88355.5	968804	770979	20.4	93	88915.3
	45	1028772	777059	24.5	78	87528.6	993129	768889	22.6	103	86607.3	1126724	782416	30.6	71	91511.8	982929	773768	21.3	88	89086.5
	49	1035892	775372	25.2	75	88197.2	1003927	772468	23.1	93	88488.0	1132645	785563	30.6	69	90079.5	988863	775215	21.6	82	87424.2
Average		899429	802217	**9.8**$^{(21)}$	**153.2**	**57359.4**	909543	799937	10.9$^{(22)}$	165.0	58191.8	984170	865098	11.0$^{(18)}$	**122.9**	**52245.8**	966638	858493	**10.4**$^{(18)}$	150.0	53551.9

Table A.4 Performance of exact algorithms for the reliable CFLP ($k = 2$)

		$p = p^{0.8}$										$p = p^{max}$									
		C&CG-E				C&CG-D			Benders decomposition			C&CG-E				C&CG-D			Benders decomposition		
$\|J\|$	$\|I\|$	UB	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU	UB	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU
10	10	661398	0.0	15	12.7	0.0	22	13.7	0.0	581	1233.4	783884	0.0	8	3.8	0.0	15	8.2	0.0	804	1324.1
	15	645755	0.0	9	7.3	0.0	13	7.2	0.0	450	828.2	778554	0.0	5	2.2	0.0	9	5.4	0.0	668	1319.4
	20	772538	0.0	10	9.4	0.0	12	7.9	0.0	501	1404.9	869836	0.0	4	2.0	0.0	8	5.4	0.0	679	1722.4
	25	741589	0.0	6	5.4	0.0	11	9.3	0.0	431	1214.1	850409	0.0	4	2.6	0.0	8	8.6	0.0	689	1910.2
	30	835208	0.0	6	5.8	0.0	9	6.9	0.0	546	1854.9	944096	0.0	4	3.9	0.0	8	8.2	0.0	781	2699.2
	35	921959	0.0	6	7.2	0.0	12	13.9	0.0	531	1875.2	1040575	0.0	5	5.0	0.0	7	6.5	0.0	428	1592.0
	40	884624	0.0	4	3.1	0.0	5	4.0	0.0	418	1438.5	975424	0.0	4	3.7	0.0	5	4.4	0.0	416	1627.9
	45	912840	0.0	5	6.0	0.0	9	10.9	0.0	637	3050.4	1043720	0.0	5	6.0	0.0	7	8.0	0.0	437	1342.3
	49	901229	0.0	8	15.3	0.0	12	22.9	0.0	524	2550.8	1020038	0.0	6	11.2	0.0	9	15.7	0.0	777	3379.9
15	15	616640	0.0	15	34.8	0.0	22	38.8	0.0	5489	54527.7	774863	0.0	9	14.9	0.0	20	41.6	11.5	6464	86410.1
	20	642143	0.0	3	1.7	0.0	8	5.8	0.0	4076	35870.9	776148	0.0	5	5.3	0.0	7	5.5	6.3	6186	86403.1
	25	694645	0.0	12	75.8	0.0	20	94.6	1.3	5808	86425.6	794268	0.0	10	43.8	0.0	15	51.5	7.1	5733	86411.1
	30	769946	0.0	11	62.7	0.0	16	61.7	3.4	5463	86415.9	879594	0.0	9	37.6	0.0	10	24.2	11.2	5362	86426.5
	35	765382	0.0	11	87.4	0.0	17	99.4	3.1	5110	86414.2	873321	0.0	8	50.1	0.0	14	77.7	9.9	5152	86432.7
	40	814909	0.0	15	184.9	0.0	24	234.9	5.8	4857	86423.9	897956	0.0	7	39.5	0.0	17	113.6	10.6	4896	86428.4
	45	832349	0.0	14	238.3	0.0	20	217.1	6.5	4695	86411.7	908160	0.0	10	106.9	0.0	18	158.6	12.9	4680	86428.9
	49	841864	0.0	14	288.2	0.0	21	299.4	6.9	4604	86430.6	928131	0.0	10	126.8	0.0	20	268.0	14.1	4526	86431.5
20	20	678644	0.0	13	93.6	0.0	22	143.8	19.3	5219	86408.9	769183	0.0	13	127.1	0.0	20	140.4	40.4	5265	86424.5
	25	744972	0.0	17	310.8	0.0	31	493.0	22.3	5048	86417.8	841338	0.0	17	327.7	0.0	35	652.0	50.5	4999	86426.7
	30	795122	0.0	26	1758.9	0.0	45	2436.7	23.3	4781	86427.1	839647	0.0	20	773.8	0.0	41	1833.2	42.2	4699	86432.3
	35	781518	0.0	14	357.3	0.0	25	485.2	20.6	4500	86412.1	843204	0.0	16	549.1	0.0	30	835.7	41.3	4361	86425.4
	40	790731	0.0	14	495.5	0.0	26	786.3	25.1	4178	86443.4	859448	0.0	19	1003.2	0.0	30	1266.3	49.5	4197	86402.9
	45	821688	0.0	17	763.0	0.0	24	697.7	25.0	4099	86409.9	860618	0.0	14	482.9	0.0	24	792.6	42.4	3994	86420.1
	49	824165	0.0	12	537.6	0.0	23	846.8	24.3	3963	86420.5	865055	0.0	12	413.6	0.0	16	296.6	43.4	3958	86405.5
25	25	702635	0.0	26	1604.4	0.0	43	2071.1	33.1	4594	86407.3	761230	0.0	31	3646.7	0.0	61	5653.8	48.8	4487	86402.9
	30	762086	0.0	26	2845.0	0.0	43	3201.0	34.7	4253	86415.7	814792	0.0	26	3473.1	0.0	43	3924.4	57.6	4265	86420.6
	35	772713	0.0	28	4480.0	0.0	53	7280.6	43.7	3841	86446.1	791860	0.0	18	1377.2	0.0	33	2243.3	57.4	4026	86402.2
	40	798604	0.0	34	11002.4	0.0	52	9708.4	41.9	3604	86424.0	820394	0.0	27	6442.0	0.0	41	6167.8	63.5	3859	86438.2
	45	834526	0.0	29	9076.9	0.0	49	10727.4	35.3	3716	86412.1	872860	0.0	31	9765.9	0.0	50	17210.6	61.8	3670	86432.8
	49	845131	0.0	33	13195.9	0.0	62	25492.7	37.6	3551	86417.5	899402	0.0	31	13067.2	0.0	46	11649.9	63.1	3506	86416.4
30	30	739095	0.0	28	3474.5	0.0	47	5446.1	54.0	3387	86455.3	769123	0.0	30	6754.4	0.0	47	7908.0	71.9	3721	86434.9
	35	759074	0.0	32	10266.2	0.0	54	13073.9	54.3	3175	86403.4	768015	0.0	33	15433.0	0.0	50	15695.9	71.1	3483	86447.8
	40	758627	0.0	30	11124.8	0.0	44	9527.9	51.0	3050	86443.2	778126	0.0	30	14431.7	0.0	48	17432.8	66.7	3383	86424.1
	45	788212	0.0	39	29160.7	0.0	64	34066.7	58.1	2998	86427.2	809425	0.0	44	56750.6	0.0	76	74267.4	76.3	3264	86449.7
	49	762807	0.0	27	14199.2	0.0	44	14473.9	52.6	2948	86456.4	772213	0.0	33	27585.7	0.0	56	40500.3	71.2	3138	86430.6
Average		777582	**0.0**$^{(0)}$	**17.4**	**3308.4**	**0.0**$^{(0)}$	28.7	4060.2	19.5$^{(24)}$	3303.6	62286.2	853569	**0.0**$^{(0)}$	**15.9**	**4653.4**	**0.0**$^{(0)}$	27.0	5979.5	31.5$^{(26)}$	3455.8	64683.6

Table A.5 Performance of C&CG algorithms for the reliable CFLP ($k = 3$)

		$p = p^{0.8}$										$p = p^{max}$									
		C&CG-E					C&CG-D					C&CG-E					C&CG-D				
$\|J\|$	$\|I\|$	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU
10	10	674500	674500	0.0	9	3.4	674500	674500	0.0	11	2.9	887563	887563	0.0	11	6.1	887563	887563	0.0	17	7.7
	15	705200	705200	0.0	10	9.4	705200	705200	0.0	15	9.3	894354	894354	0.0	7	4.5	894354	894354	0.0	10	4.9
	20	888338	888338	0.0	16	27.7	888338	888338	0.0	21	25.7	1021977	1021977	0.0	9	9.1	1021977	1021977	0.0	9	5.2
	25	838189	838189	0.0	10	15.1	838189	838189	0.0	18	26.5	986737	986737	0.0	7	7.0	986737	986737	0.0	9	8.0
	30	951008	951008	0.0	8	11.0	951008	951008	0.0	10	11.7	1109140	1109140	0.0	6	6.8	1109140	1109140	0.0	12	14.8
	35	1031573	1031573	0.0	9	16.4	1031573	1031573	0.0	13	21.4	1211614	1211614	0.0	6	7.7	1211614	1211614	0.0	9	11.0
	40	1045425	1045425	0.0	6	9.3	1045425	1045425	0.0	9	10.8	1210166	1210166	0.0	4	5.4	1210166	1210166	0.0	6	7.5
	45	1028640	1028640	0.0	6	10.2	1028640	1028640	0.0	10	15.7	1218016	1218016	0.0	6	10.0	1218016	1218016	0.0	7	10.0
	49	1033264	1033264	0.0	11	28.4	1033264	1033264	0.0	16	42.8	1151594	1151594	0.0	8	15.9	1151594	1151594	0.0	8	13.0
15	15	616640	616640	0.0	7	5.4	616640	616640	0.0	12	12.1	867311	867311	0.0	16	58.3	867311	867311	0.0	28	94.1
	20	743978	743978	0.0	11	30.5	743978	743978	0.0	20	58.1	907315	907315	0.0	8	15.3	907315	907315	0.0	10	13.8
	25	777830	777830	0.0	26	584.4	777830	777830	0.0	45	961.9	897418	897418	0.0	22	291.9	897418	897418	0.0	25	179.5
	30	883012	883012	0.0	25	603.4	883012	883012	0.0	45	1005.1	995394	995394	0.0	12	83.5	995394	995394	0.0	20	115.5
	35	866226	866226	0.0	28	1329.0	866226	866226	0.0	42	1439.1	986094	986094	0.0	20	374.5	986094	986094	0.0	30	389.6
	40	904026	904026	0.0	24	745.4	904026	904026	0.0	43	1253.0	1013756	1013756	0.0	18	371.0	1013756	1013756	0.0	37	837.7
	45	941543	941543	0.0	29	1508.8	941543	941543	0.0	49	1879.5	1013742	1013742	0.0	16	300.6	1013742	1013742	0.0	25	383.6
	49	938294	938294	0.0	30	2265.5	938294	938294	0.0	47	2695.6	1040173	1040173	0.0	24	1186.3	1040173	1040173	0.0	38	1413.0
20	20	779753	779753	0.0	34	1950.5	779753	779753	0.0	61	2851.8	869765	869765	0.0	32	1878.2	869765	869765	0.0	59	3122.2
	25	836284	836284	0.0	44	4098.3	836284	836284	0.0	81	5926.3	945425	945425	0.0	45	3588.1	945425	945425	0.0	66	3811.4
	30	881840	881840	0.0	79	44387.7	881840	881840	0.0	125	46488.0	947137	947137	0.0	60	13576.0	947137	947137	0.0	106	21147.2
	35	891553	891553	0.0	50	11535.7	891553	891553	0.0	78	12350.1	947809	947809	0.0	37	4911.8	947809	947809	0.0	62	6495.1
	40	890806	890806	0.0	48	14352.1	890806	890806	0.0	78	19556.9	949312	949312	0.0	33	5618.6	949312	949312	0.0	72	13280.5
	45	925001	925001	0.0	43	15231.4	925001	925001	0.0	66	14181.0	970263	970263	0.0	30	4797.0	970263	970263	0.0	45	4551.9
	49	935123	935123	0.0	40	20120.0	935123	935067	0.0	66	21647.2	995549	995549	0.0	33	10088.0	995549	995549	0.0	57	15834.2
25	25	780754	780754	0.0	74	47616.8	780754	780754	0.0	140	83752.5	880305	868558	1.3	88	87748.5	905413	791863	12.5	132	87061.1
	30	890314	850502	4.5	76	87989.3	888598	844442	5.0	127	88189.4	932843	912978	2.1	78	87497.7	934721	905627	3.1	114	86830.5
	35	858813	855711	0.4	69	89420.9	872617	842934	3.4	111	86758.7	916247	916247	0.0	69	82643.3	917996	905371	1.4	111	88721.4
	40	925961	867658	6.3	60	88107.9	906924	852197	6.0	92	89264.4	977005	916964	6.2	57	86693.9	1044400	830406	20.5	82	88113.7
	45	940917	906049	3.7	55	87727.1	962540	892474	7.3	92	89701.3	989280	955383	3.4	53	88149.6	1126621	860095	23.7	69	89194.4
	49	970636	907606	6.5	57	87925.8	956572	889100	7.1	89	86735.5	1035978	965480	6.8	57	87771.5	1094801	911822	16.7	76	88165.1
30	30	886066	810901	8.5	57	88741.5	846902	799665	5.6	95	87282.0	880913	846701	3.9	54	89475.2	880913	840241	4.6	82	86622.9
	35	860799	828531	3.8	57	90009.0	872683	812351	6.9	88	89357.8	879918	842883	4.2	51	87005.0	946139	807529	14.7	70	87822.6
	40	865851	810320	6.4	45	87916.7	903404	803828	11.0	64	89110.1	1048410	785459	25.1	37	89911.6	915228	801871	12.4	57	87817.5
	45	931672	807548	13.3	47	88548.6	926719	794932	14.2	69	89367.3	980293	839718	14.3	45	86736.3	989309	825937	16.5	68	87356.2
	49	883712	826683	6.5	42	87397.2	900512	810434	10.0	65	91087.6	975109	819780	15.9	39	91347.6	968784	791825	18.3	53	88272.5
Average		880101	864580	**1.7**$^{(10)}$	**35.5**	**30008.0**	880751	860889	2.2$^{(10)}$	57.5	31516.5	986684	963079	**2.4**$^{(10)}$	**31.4**	**28919.8**	993199	951720	4.1$^{(11)}$	48.0	29649.4

Table A.6 Performance of C&CG algorithms for the reliable CFLP ($k = 4$)

		$p = p^{0.8}$										$p = p^{max}$									
		C&CG-E					C&CG-D					C&CG-E					C&CG-D				
$\|J\|$	$\|I\|$	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU	UB	LB	Gap	#Iter	CPU
10	10	674500	674500	0.0	5	0.7	674500	674500	0.0	7	1.1	1003363	1003363	0.0	15	10.8	1003363	1003363	0.0	16	6.5
	15	705200	705200	0.0	6	1.1	705200	705200	0.0	6	1.7	1026115	1026115	0.0	10	7.7	1026115	1026115	0.0	12	5.4
	20	937000	937000	0.0	11	8.9	937000	937000	0.0	13	7.6	1170836	1170836	0.0	6	3.9	1170836	1170836	0.0	8	4.2
	25	956838	956838	0.0	14	21.0	956838	956838	0.0	16	16.4	1140414	1140414	0.0	7	6.9	1140414	1140414	0.0	11	9.7
	30	1085102	1085102	0.0	13	21.3	1085102	1085102	0.0	11	10.4	1283306	1283306	0.0	7	9.8	1283306	1283306	0.0	9	8.4
	35	1147373	1147373	0.0	12	21.7	1147373	1147373	0.0	17	23.7	1366590	1366590	0.0	5	7.2	1366590	1366590	0.0	5	6.0
	40	1217119	1217119	0.0	10	17.7	1217119	1217119	0.0	12	15.5	1453446	1453446	0.0	6	10.4	1453446	1453446	0.0	7	7.2
	45	1175918	1175918	0.0	10	22.2	1175918	1175918	0.0	15	28.1	1408403	1408403	0.0	7	13.3	1408403	1408403	0.0	8	11.9
	49	1149064	1149064	0.0	10	22.1	1149064	1149064	0.0	13	22.2	1345055	1345055	0.0	8	19.9	1345055	1345055	0.0	8	12.1
15	15	616640	616640	0.0	4	0.7	616640	616640	0.0	4	1.6	958331	958331	0.0	22	129.7	958331	958331	0.0	37	280.9
	20	743978	743978	0.0	10	12.5	743978	743978	0.0	12	11.9	1047401	1047401	0.0	14	59.1	1047401	1047401	0.0	20	86.0
	25	871104	871104	0.0	71	7405.0	871104	871104	0.0	115	9955.2	993023	993023	0.0	25	399.5	993023	993023	0.0	43	898.0
	30	956560	956560	0.0	28	619.4	956560	956560	0.0	42	732.7	1108079	1108079	0.0	17	166.5	1108079	1108079	0.0	24	267.2
	35	943810	943810	0.0	41	3973.3	943810	943810	0.0	62	3318.4	1075563	1075563	0.0	22	495.4	1075563	1075563	0.0	41	1217.1
	40	1014921	1014921	0.0	57	6916.9	1014921	1014921	0.0	94	9049.5	1123652	1123652	0.0	37	1999.9	1123652	1123652	0.0	55	2039.7
	45	1024398	1024398	0.0	36	2072.4	1024398	1024398	0.0	59	3121.6	1129542	1129542	0.0	23	787.1	1129542	1129542	0.0	41	1050.2
	49	1038782	1038782	0.0	57	11737.4	1038782	1038782	0.0	91	13045.7	1147948	1147948	0.0	41	4400.4	1147948	1147948	0.0	64	4253.2
20	20	858292	858292	0.0	66	14963.5	858292	858292	0.0	110	22125.9	954378	954378	0.0	60	15935.7	954378	954378	0.0	115	28031.5
	25	916349	916349	0.0	65	10813.6	916349	916349	0.0	117	17866.0	1046138	1046138	0.0	84	28685.5	1046138	1046074	0.0	131	28583.9
	30	963299	934797	3.0	98	87620.9	993495	928867	6.5	143	87659.6	1038995	1026381	1.2	98	86529.9	1038995	1027101	1.1	154	87096.1
	35	985957	979072	0.7	85	88635.9	986512	974980	1.2	137	87685.9	1051449	1051449	0.0	84	58203.1	1051449	1051449	0.0	136	82687.6
	40	981238	971399	1.0	77	89697.7	995538	968796	2.7	118	86580.7	1081062	1054191	2.5	82	86609.0	1076877	1037931	3.6	115	86495.3
	45	1004936	1004936	0.0	61	49229.1	1004936	1004936	0.0	107	76496.6	1075922	1075922	0.0	51	28193.7	1075922	1075922	0.0	85	56421.8
	49	1019345	1012012	0.7	66	88772.2	1019345	1007506	1.2	98	87794.1	1094096	1086636	0.7	71	88811.7	1094096	1081691	1.1	94	86499.5
25	25	910365	821391	9.8	75	87246.2	893806	822443	8.0	119	86980.1	998167	895946	10.2	74	87744.0	1060031	828368	21.9	109	88406.8
	30	993270	891611	10.2	68	89024.8	987396	880025	10.9	107	87185.4	1119388	928505	17.1	71	87091.5	1161884	901793	22.4	106	87299.0
	35	1013066	887711	12.4	57	86901.0	1003039	871209	13.1	83	86531.5	1036120	947384	8.6	60	88666.1	1026306	943304	8.1	86	88564.5
	40	1020102	889234	12.8	50	88042.0	1028514	861182	16.3	77	86579.6	1084240	923085	14.9	57	86923.1	1135203	875479	22.9	66	88837.0
	45	1087776	944183	13.2	46	91477.8	1057821	923770	12.7	69	87499.5	1169098	990437	15.3	50	88196.7	1311976	912442	30.5	58	88027.0
	49	1108680	945436	14.7	54	90849.8	1062150	946661	10.9	79	86926.1	1242075	967731	22.1	47	89414.1	1279718	959153	25.1	67	87015.2
30	30	972545	841893	13.4	49	90487.5	994130	835840	15.9	76	89599.7	1147459	848236	26.1	43	89882.0	1152633	860958	25.3	69	88017.4
	35	960221	865837	9.8	51	87329.6	974677	858444	11.9	76	88832.3	1053868	881512	16.4	44	90432.4	1071524	861963	19.6	63	87958.9
	40	1013221	846683	16.4	40	89508.4	977676	817591	16.4	55	88137.8	1272081	832243	34.6	33	89883.8	1107581	818327	26.1	53	86460.7
	45	1038519	847268	18.4	39	87619.2	1026658	840228	18.2	59	91033.7	1212042	848634	30.0	36	89395.1	1191573	844568	29.1	53	87618.4
	49	1016254	850381	16.3	38	89939.1	1015705	824050	18.9	53	86558.4	1126318	845019	25.0	35	89246.3	1179802	823889	30.2	48	89638.7
Average		974907	930480	**4.4**$^{(15)}$	**42.3**	**41172.4**	972981	925699	4.7$^{(15)}$	64.9	42041.0	1130970	1056711	**6.4**$^{(14)}$	**38.6**	**39382.0**	1137062	1048167	7.6$^{(14)}$	57.6	40966.4

Table A.7 Results of linear decision rule for the reliable UFLP

		$k=2, p=p^{0.8}$			$k=2, p=p^{max}$			$k=3, p=p^{0.8}$			$k=3, p=p^{max}$			$k=4, p=p^{0.8}$			$k=4, p=p^{max}$		
$\|J\|$	$\|I\|$	UB	Gap	CPU	UB	Gap	CPU	UB	Gap	CPU	UB	Gap	CPU	UB	Gap	CPU	UB	Gap	CPU
10	10	570199	0.0	1.1	717029	0.0	1.9	664428	0.0	1.4	836103	0.0	2.6	674500	0.0	1.0	952107	0.0	2.4
	15	603056	0.0	2.0	797030	0.0	3.5	698079	0.0	2.9	931027	0.0	4.4	705200	0.0	3.4	1060930	0.0	3.9
	20	713670	0.0	3.5	880270	0.0	8.8	821742	0.0	6.4	1028371	0.0	7.6	926581	0.0	8.6	1165139	0.0	6.8
	25	773423	0.0	6.4	946045	0.0	10.2	881019	0.0	9.7	1103657	0.0	11.3	985105	0.0	13.3	1244257	0.0	11.2
	30	857858	0.0	14.9	1016762	0.0	15.0	973024	0.0	19.3	1181228	0.0	19.8	1083170	0.0	21.8	1324634	0.0	16.8
	35	887301	0.0	18.9	1055471	0.0	20.1	1007302	0.0	24.7	1223439	0.0	24.9	1116279	0.0	29.6	1366408	0.0	21.1
	40	932566	0.0	22.0	1093925	0.0	29.8	1056906	0.0	36.4	1264770	0.0	35.9	1169868	0.0	32.8	1410344	0.0	27.6
	45	952238	0.0	30.7	1112863	0.0	44.3	1077172	0.0	49.1	1284848	0.0	37.5	1191697	0.0	45.8	1430815	0.0	31.0
	49	955564	0.0	34.1	1120590	0.0	51.0	1079892	0.0	45.4	1293024	0.0	64.7	1194739	0.0	59.7	1439687	0.0	38.4
15	15	566085	0.0	11.9	793971	0.0	42.8	616640	0.0	11.5	913053	0.0	76.8	616640	0.0	7.5	1026543	0.0	111.1
	20	641895	0.0	21.2	880285	0.0	72.7	732654	0.0	53.9	1008935	0.0	142.7	743978	0.0	28.8	1121569	0.0	237.2
	25	720482	0.0	52.0	945555	0.0	139.0	818060	0.0	136.0	1072380	0.0	369.9	886660	0.0	214.1	1190732	0.0	338.5
	30	821913	0.0	96.9	1011814	0.0	224.8	921187	0.0	368.3	1141652	0.0	486.9	1017097	0.0	425.1	1265167	0.0	504.4
	35	851494	0.0	218.2	1042648	0.0	260.8	952239	0.0	379.5	1176723	0.0	400.4	1049812	0.0	648.8	1304135	0.0	924.1
	40	902476	0.0	289.1	1071550	0.0	655.6	1007567	0.0	687.2	1208118	0.0	568.0	1108766	0.0	740.7	1337794	0.0	1036.2
	45	931558	0.0	419.9	1089902	0.0	784.8	1039070	0.0	830.1	1227590	0.0	781.9	1142198	0.0	1249.2	1358536	0.0	1487.3
	49	932855	0.0	477.6	1097510	0.0	781.0	1040007	0.0	896.9	1235687	0.0	1288.9	1142561	0.0	1634.0	1367373	0.0	1794.4
20	20	685286	0.0	80.3	880285	0.0	480.3	784518	0.0	434.4	1008935	0.0	1001.4	858292	0.0	715.2	1121569	0.0	4238.6
	25	748990	0.0	250.0	945555	0.0	1325.8	845111	0.0	1135.3	1072380	0.0	5113.0	937676	0.0	1628.9	1190732	0.0	7145.5
	30	852570	0.0	974.6	1011814	0.0	2264.3	956708	0.0	1869.8	1141652	0.0	6734.4	1055956	0.0	4400.0	1265167	0.0	10544.2
	35	877921	0.0	1186.6	1042648	0.0	2789.5	981959	0.0	4757.4	1176723	0.0	12059.6	1082366	0.0	6318.2	1302771	0.0	20369.7
	40	915831	0.0	1773.7	1075639	0.0	4735.5	1022809	0.0	3949.7	1213404	0.0	16050.6	1125739	0.0	9758.0	1342796	0.0	25374.9
	45	945311	0.0	2945.4	1094088	0.0	5018.3	1054941	0.0	7117.0	1233046	0.0	19132.9	1159901	0.0	21282.0	1363695	0.0	34842.1
	49	943491	0.0	3951.1	1101731	0.0	9037.7	1052328	0.0	9513.5	1241242	0.0	25294.4	1156297	0.0	32723.4	1372587	0.0	44274.3
25	25	763776	0.0	1682.8	939198	0.0	10715.0	862661	0.0	4171.7	1063645	0.0	23294.9	953438	0.0	14803.6	1178237	11.1	68135.8
	30	841739	0.0	4266.0	1007068	0.0	21973.3	943461	0.0	12605.1	1133497	0.0	59829.7	1039154	0.0	29922.7	1252091	12.4	86402.3
	35	868988	0.0	8058.6	1039711	0.0	38336.7	970588	0.0	26553.9	1167519	4.6	86403.5	1068134	0.0	74862.4	1289658	15.0	86402.5
	40	905717	0.0	16857.3	1073319	0.0	80955.4	1010011	0.0	53336.0	1204340	7.2	86402.8	1109011	0.0	74033.8	1329510	15.5	86403.1
	45	932043	0.0	21609.6	1091657	0.0	58251.1	1038613	0.0	78674.3	1224220	11.1	86403.5	1138818	6.6	86403.3	1350251	16.0	86403.3
	49	932145	0.0	30198.8	1098928	0.0	76609.1	1038150	0.0	45553.4	1232528	11.3	86403.8	1137974	9.4	86404.3	1359042	18.9	86403.5
30	30	845385	0.0	22237.5	1005774	7.7	86403.2	949777	0.0	66973.4	1127288	13.6	86403.2	1043648	11.6	86403.3	1244733	20.3	86403.0
	35	871497	0.0	44567.7	1037272	10.7	86404.8	976149	9.0	86403.9	1161880	17.5	86405.2	1071961	13.2	86403.9	1282931	21.7	86403.6
	40	899207	0.0	53213.1	1070906	13.3	86404.5	1005269	10.0	86404.5	1198995	18.8	86405.6	1102247	15.5	86404.1	1323147	23.7	86407.5
	45	925720	0.0	77134.9	1089410	14.7	86404.9	1033957	10.6	86403.4	1224456	19.3	86406.0	1132014	18.4	86404.7	1343915	24.6	86404.6
	49	924368	5.0	86404.9	1097041	16.0	86405.2	1033749	12.1	86405.2	1227234	21.4	86405.6	1129870	19.3	86405.0	1352729	24.6	86404.9
Average		836989	0.1	10831.8	1010722	1.8	21361.7	941364	1.2	19023.4	1148103	3.6	27156.4	1030210	2.7	25155.5	1275192	5.8	31015.9

UB: Final solution generated by MILP reformulation.
Gap: Nonzero gap means that MILP model is not solved to optimality within time limit.

Table A.8 Results of linear decision rule for the reliable CFLP

		$k=2, p=p^{0.8}$			$k=2, p=p^{max}$			$k=3, p=p^{0.8}$			$k=3, p=p^{max}$			$k=4, p=p^{0.8}$			$k=4, p=p^{max}$		
$\|J\|$	$\|I\|$	UB	Gap	CPU	UB	Gap	CPU	UB	Gap	CPU	UB	Gap	CPU	UB	Gap	CPU	UB	Gap	CPU
10	10	674500	0.0	1.4	890072	0.0	2.6	674500	0.0	1.1	1011855	0.0	3.6	674500	0.0	0.7	1125862	0.0	2.7
	15	693415	0.0	3.2	915488	0.0	4.1	705200	0.0	1.8	1056270	0.0	4.5	705200	0.0	1.3	1191030	0.0	5.6
	20	833385	0.0	5.5	1063755	0.0	6.2	937000	0.0	5.9	1213711	0.0	6.6	937000	0.0	4.1	1355914	0.0	7.3
	25	839420	0.0	6.9	1061070	0.0	9.8	956559	0.0	12.5	1198966	0.0	11.3	1031000	0.0	9.7	1348747	0.0	12.1
	30	935122	0.0	9.7	1165880	0.0	10.5	1065362	0.0	13.0	1329070	0.0	14.2	1197607	0.0	19.0	1488548	0.0	18.3
	35	978102	0.0	14.5	1242756	0.0	14.7	1125291	0.0	17.8	1419324	0.0	22.3	1259460	0.0	21.4	1577851	0.0	25.1
	40	996275	0.0	17.1	1267090	0.0	27.5	1151708	0.0	24.6	1462030	0.0	36.4	1291741	0.0	24.8	1624182	0.0	26.9
	45	1007045	0.0	22.1	1268820	0.0	27.1	1168338	0.0	30.2	1431018	0.0	26.6	1290743	0.0	39.6	1585481	0.0	35.2
	49	1007722	0.0	23.7	1267710	0.0	24.8	1160178	0.0	39.4	1436090	0.0	35.2	1297542	0.0	49.4	1592186	0.0	44.1
15	15	616640	0.0	5.8	915260	0.0	24.8	616640	0.0	5.3	1024270	0.0	33.0	616640	0.0	2.5	1129854	0.0	40.7
	20	691177	0.0	13.6	971808	0.0	21.8	743978	0.0	11.7	1120582	0.0	51.4	743978	0.0	7.7	1255861	0.0	56.9
	25	769420	0.0	67.1	1017304	0.0	152.1	864194	0.0	255.3	1142099	0.0	124.4	886660	0.0	70.5	1247120	0.0	125.8
	30	887311	0.0	139.2	1091020	0.0	89.0	994627	0.0	195.7	1223114	0.0	113.7	1077686	0.0	222.0	1348498	0.0	142.0
	35	886525	0.0	210.4	1101966	0.0	290.9	988229	0.0	336.8	1232904	0.0	294.2	1085691	0.0	411.9	1352121	0.0	233.1
	40	949126	0.0	488.7	1148057	0.0	440.3	1060152	0.0	732.9	1282000	0.0	406.0	1158425	0.0	683.8	1400879	0.0	402.4
	45	959802	0.0	322.4	1149322	0.0	407.8	1071283	0.0	550.7	1292358	0.0	373.5	1175657	0.0	480.5	1417445	0.0	332.9
	49	978306	0.0	593.6	1171105	0.0	588.9	1086381	0.0	923.4	1307201	0.0	797.2	1187885	0.0	1115.6	1432267	0.0	757.2
20	20	745289	0.0	106.5	956845	0.0	332.0	839727	0.0	234.6	1073347	0.0	497.0	858292	0.0	124.9	1190589	0.0	922.6
	25	812879	0.0	276.2	1049567	0.0	813.7	920541	0.0	806.9	1176723	0.0	1080.6	962954	0.0	484.2	1296879	0.0	1074.9
	30	897004	0.0	1080.0	1069875	0.0	1826.9	1002328	0.0	2102.9	1189979	0.0	2577.1	1095914	0.0	5225.0	1304258	0.0	2809.2
	35	908835	0.0	832.7	1092221	0.0	1709.5	1015410	0.0	2119.8	1219803	0.0	2203.7	1116393	0.0	3525.9	1348414	0.0	3243.6
	40	935472	0.0	1925.8	1117000	0.0	2711.2	1048577	0.0	3125.9	1249686	0.0	4443.2	1144824	0.0	7814.9	1373793	0.0	4767.9
	45	970605	0.0	2963.8	1135531	0.0	2894.5	1079146	0.0	3484.6	1265803	0.0	2597.1	1178675	0.0	5280.7	1385367	0.0	2677.5
	49	959356	0.0	1886.3	1154443	0.0	5686.1	1070378	0.0	4600.0	1288184	0.0	10327.9	1185125	0.0	9426.8	1414308	0.0	6690.4
25	25	803128	0.0	1624.9	992177	0.0	5509.2	899755	0.0	3951.5	1112227	0.0	3954.9	990415	0.0	7128.2	1223621	0.0	20831.2
	30	880603	0.0	3501.7	1072390	0.0	8712.7	992164	0.0	8925.4	1193390	0.0	15831.2	1086647	0.0	35577.2	1301779	0.0	30889.2
	35	893560	0.0	5029.3	1070515	0.0	10283.4	995157	0.0	7593.5	1198921	0.0	24229.8	1094478	0.0	34906.2	1321183	0.0	28355.1
	40	927220	0.0	11840.0	1119287	0.0	24473.7	1027717	0.0	18066.8	1240035	0.0	35966.4	1123694	0.0	42541.6	1359203	0.0	45477.2
	45	956458	0.0	6691.0	1143306	0.0	15711.9	1080918	0.0	33867.5	1261493	0.0	15577.9	1171232	0.0	52022.7	1393041	0.0	40381.5
	49	959081	0.0	11494.6	1171296	0.0	37724.0	1085488	0.0	34587.0	1303546	4.3	86403.9	1180327	0.0	46741.0	1425057	0.0	66227.4
30	30	862877	0.0	5780.4	1053964	0.0	27293.9	976194	0.0	17230.7	1175561	0.0	61399.0	1068780	0.0	67094.6	1291995	0.0	61936.7
	35	888025	0.0	10323.6	1082463	0.0	27539.6	1016466	0.0	29953.2	1207302	0.0	35024.8	1113500	0.0	85251.9	1331542	0.0	56563.1
	40	912796	0.0	31269.6	1089676	8.9	86404.0	1027654	0.0	64229.2	1220840	4.2	86404.4	1126157	9.9	86404.4	1344724	7.4	86404.2
	45	925755	0.0	35246.0	1127557	5.7	86405.2	1061425	5.7	86405.1	1253117	7.1	86404.6	1152158	9.7	86404.7	1362643	6.2	86405.0
	49	929107	0.0	45069.1	1123753	10.6	86404.9	1062507	8.0	86405.4	1254651	6.8	86405.1	1157909	13.6	86405.2	1378679	10.1	86405.1
Average		882038	0.0	5111.0	1095153	0.7	12416.6	987748	0.4	11738.5	1230499	0.6	16105.2	1069283	0.9	19015.0	1357741	0.7	18123.7

APPENDIX B SUPPLEMENT TO CHAPTER 4

Section B discusses the static robust model for the CFLP under disruptions. Section B presents the C&CG algorithm developed in [1].

B.1 The Static Robust Optimization Model

The static RO model for the CFLP under facility disruptions is

$$
\begin{aligned}
&\min_{\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} + \sum_{i\in I} p_i u_i, && \\
&\text{s.t.} \sum_{j\in J} x_{ij} + u_i \geq \bar{h}_i && \forall i \in I, \\
&\quad \sum_{i\in I} x_{ij} \leq C_j y_j (1 - z_j) && \forall \boldsymbol{z} \in \mathcal{Z}_k, j \in J, \\
&\quad y_j \in \{0,1\} && \forall j \in J, \\
&\quad x_{ij} \geq 0 && \forall i \in I, j \in J, \\
&\quad u_i \geq 0 && \forall i \in I.
\end{aligned}
$$

Through duality theory, the second constraints can be reformulated as

$$\sum_{i\in I} x_{ij} \leq C_j y_j - kA_j - B_j \qquad \forall j \in J, \tag{B.1}$$

$$C_j y_j - A_j - B_j \leq 0 \qquad \forall j \in J, \tag{B.2}$$

$$A_j, B_j \geq 0 \qquad \forall j \in J. \tag{B.3}$$

Constraints (B.2)–(B.3) indicate that when $k \geq 1$, the equation $C_j y_j - kA_j - B_j \leq 0$ always holds. Therefore, when $k \geq 1$, we have $\sum_{i\in I} x_{ij} \leq 0, \forall j \in J$, which suggests that $x_{ij} = 0, \forall i \in I, j \in J$ and $u_i = \bar{h}_i, \forall i \in I$. Since the static RO model is a minimization problem, all y_j would be 0 at optimality. A logical explanation of this result is that when $k \geq 1$, the adversary can always select an opened facility to disrupt, thus rendering the problem infeasible in the absence of recourse.

B.2 C&CG Algorithm Developed in Reference [1]

The authors of [1] reformulate the subproblem of the C&CG algorithm as a minimum cost flow problem instead of using the KKT condition as we do. To transform the second-stage problem to a minimum cost flow problem, we first add a dummy facility node indexed by $|J|+1$ and a dummy demand node indexed by $|I|+1$ to the supply chain system. For notational convenience, we denote $I' = I \cup \{|I|+1\}$ and $J' = J \cup \{|J|+1\}$. We then add arcs from site $|J|+1$ to demand node $i \in I$ with transportation cost p_i, and arcs from $j \in J'$ to demand node $|I|+1$ with zero transportation cost. We assume the supply capacity at site $|J|+1$ is

$$C_{|J|+1} = \sum_{i \in I} h_i,$$

and the demand quantity at customer $|I|+1$ is

$$h_{|I|+1} = \sum_{j \in J} C_j y_j (1 - z_j).$$

Then, the net flow balance condition always holds for the considered system. Note that facility $|J|+1$ is always operational and its fixed cost is 0. Now we can equivalently formulate $g(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{z})$ as a minimum cost flow problem:

$$\min_{\boldsymbol{x}} \sum_{i \in I'} \sum_{j \in J'} d_{ij} x_{ij} \tag{B.4a}$$

$$\text{s.t.} \ -\sum_{j \in J'} x_{ij} = -h_i \qquad \forall i \in I', \tag{B.4b}$$

$$\sum_{i \in I'} x_{ij} = C_j y_j (1 - z_j) \qquad \forall j \in J, \tag{B.4c}$$

$$\sum_{i \in I'} x_{i,|J|+1} = C_{|J|+1} \tag{B.4d}$$

$$x_{ij} \geq 0 \qquad \forall i \in I', j \in J'. \tag{B.4e}$$

To use the method in [1] for our problem, the assumption that transportation cost d_{ij}, $i \in I', j \in J'$ are integer is required. We can derive the dual problem of model (B.4) as

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} -\sum_{i \in I'} h_i \alpha_i + \sum_{j \in J} C_j y_j (1 - z_j) \beta_j + C_{|J|+1} \beta_{|J|+1} \tag{B.5a}$$

$$\text{s.t.} \ -\alpha_i + \beta_j \leq d_{ij} \qquad \forall i \in I', j \in J', \tag{B.5b}$$

$$\boldsymbol{\alpha} \in \mathbb{R}^{|I'|}, \boldsymbol{\beta} \in \mathbb{R}^{|J'|}. \tag{B.5c}$$

Based on Proposition 2 in [1], model (B.5) is equivalent to

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} -\sum_{i\in I'} h_i\alpha_i + \sum_{j\in J} C_j y_j(1-z_j)\beta_j + C_{|J|+1}\beta_{|J|+1} \tag{B.6a}$$

$$\text{s.t. } -\alpha_i + \beta_j \le d_{ij} \qquad \forall i \in I', j \in J', \tag{B.6b}$$

$$\alpha_i \le \alpha^{max}, \beta_j \le \beta^{max} \qquad \forall i \in I', j \in J', \tag{B.6c}$$

$$\boldsymbol{\alpha} \in \mathbb{N}^{|I'|}, \boldsymbol{\beta} \in \mathbb{N}^{|J'|}, \tag{B.6d}$$

where $\alpha^{max} = \beta^{max} = \max_{i\in I', j\in J'} d_{ij}$ and $\mathbb{N}$ denotes nonnegative integers. The objective function (B.6a) can be explicitly written as

$$\max \sum_{i\in I} \bar{h}_i(\beta_{|J|+1} - \alpha_i) + \sum_{j\in J} C_j y_j(\beta_j - \alpha_{|I|+1}) + \sum_{i\in I} h_i^{\Delta}\theta_i(\beta_{|J|+1} - \alpha_i) - \sum_{j\in J} C_j y_j z_j(\beta_j - \alpha_{|I|+1}), \tag{B.7}$$

where the last two terms are nonlinear. The third term contains the product of continuous variables (i.e., θ_i) and integer variables (i.e., $\beta_{|J|+1}$ and α_i), whereas the fourth term involves the product of binary variables (i.e., z_j) and integer variables (i.e., β_j and $\alpha_{|I|+1}$). To linearize the third term, we replace integer variables $\beta_{|J|+1}$ and α_i, $i \in I$ with their binary representations [157]

$$\beta_{|J|+1} = \sum_{k=0}^{P} 2^k \beta_{|J|+1}^k, \qquad \alpha_i = \sum_{k=0}^{P} 2^k \alpha_i^k,$$

where $\beta_{|J|+1}^k$, α_i^k, $k \in \{0, \cdots, P\}$ are binary variables and $P = \left\lceil \log_2(\max_{i\in I', j\in J'} d_{ij} + 1) \right\rceil - 1$. Up to now, we can use the McCormick envelopes [158] to linearize the third term of objective (B.7). Let $W_{i,|J|+1}^k = \theta_i \beta_{|J|+1}^k$ and $Q_i^k = \theta_i \alpha_i^k$, then

$$\sum_{i\in I} h_i^{\Delta}\theta_i(\beta_{|J|+1} - \alpha_i) = \sum_{i\in I} h_i^{\Delta} \left(\sum_{k=0}^{P} 2^k \left(W_{i,|J|+1}^k - Q_i^k \right) \right) \tag{B.8}$$

and

$$W_{i,|J|+1}^k \le \beta_{|J|+1}^k, \quad W_{i,|J|+1}^k \le \theta_i, \quad W_{i,|J|+1}^k \ge \theta_i - (1 - \beta_{|J|+1}^k), \quad W_{i,|J|+1}^k \ge 0, \tag{B.9}$$

$$Q_i^k \le \alpha_i^k, \quad Q_i^k \le \theta_i, \quad Q_i^k \ge \theta_i - (1 - \alpha_i^k), \quad Q_i^k \ge 0. \tag{B.10}$$

We can directly use the McCormick envelopes [158] to linearize the fourth term of objective

(B.7). Let $T_j = z_j\beta_j$ and $U_j = z_j\alpha_{|I|+1}$, then

$$-\sum_{j\in J} C_j y_j z_j(\beta_j - \alpha_{|I|+1}) = -\sum_{j\in J} C_j y_j (T_j - U_j) \tag{B.11}$$

and

$$T_j \le \beta^{max} z_j, \quad T_j \le \beta_j, \quad T_j \ge \beta_j - (1-z_j)\beta^{max}, \quad T_j \ge 0, \tag{B.12}$$

$$U_j \le \alpha^{max} z_j, \quad U_j \le \alpha_{|I|+1}, \quad U_j \ge \alpha_{|I|+1} - (1-z_j)\alpha^{max}, \quad U_j \ge 0. \tag{B.13}$$

Finally, we get the subproblem of the C&CG algorithm in [1] as follows

$$\begin{aligned}
\max\ & \sum_{j\in J} f_j \hat{y}_j + \sum_{i\in I} \bar{h}_i(\beta_{|J|+1} - \alpha_i) + \sum_{j\in J} C_j y_j(\beta_j - \alpha_{|I|+1}) + \\
& \sum_{i\in I} h_i^{\Delta}\left(\sum_{k=0}^{P} 2^k \left(W^k_{i,|J|+1} - Q^k_i\right)\right) - \sum_{j\in J} C_j y_j (T_j - U_j) \\
\text{s.t. } & -\alpha_i + \beta_j \le d_{ij} && \forall i \in I', j \in J', \\
& \alpha_i \le \alpha^{max}, && \forall i \in I', \\
& \beta_j \le \beta^{max} && \forall j \in J', \\
& \boldsymbol{\alpha} \in \mathbb{N}^{|I'|}, \boldsymbol{\beta} \in \mathbb{N}^{|J'|}, \\
& \beta_{|J|+1} = \sum_{k=0}^{P} 2^k \beta^k_{|J|+1}, \\
& \alpha_i = \sum_{k=0}^{P} 2^k \alpha^k_i, && \forall i \in I, \\
& 0 \le \theta_i \le 1 && \forall i \in I, \\
& \sum_{i\in I} \theta_i \le \Gamma_h, \\
& \sum_{j\in J} z_j \le k, \\
& z_j \in \{0,1\}, && \forall j \in J, \\
& \beta^k_{|J|+1}, \alpha^k_i \in \{0,1\} && \forall i \in I, k \in \{0, \cdots, P\} \\
& \text{(B.9)–(B.10) and (B.12)–(B.13).}
\end{aligned}$$

APPENDIX C SUPPLEMENT TO CHAPTER 5

Section C introduces the deterministic LNDP model. Section C presents the two-stage stochastic programming model.

C.1 Deterministic LNDP model

G-LNDP:

$$
\begin{aligned}
&\min \sum_{j\in\mathcal{V}_0} f_j y_j + \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + \sum_{j\in\mathcal{V}_\mathcal{D}} \theta_j u_j && \\
&\text{s.t.} \sum_{i\in\mathcal{V}_j^+} x_{ji} \le b_j && \forall j \in \mathcal{V}_\mathcal{S} \\
&\quad \sum_{i\in\mathcal{V}_j^+} x_{ji} = \sum_{i\in V_j^-} x_{ij} && \forall j \in \mathcal{V}_\mathcal{T} \\
&\quad \sum_{i\in\mathcal{V}_j^-} x_{ij} + u_j = -b_j && \forall j \in \mathcal{V}_\mathcal{D} \\
&\quad \sum_{i\in\mathcal{V}_j^+} x_{ji} \le Q_j y_j && \forall j \in \mathcal{V}_0 \\
&\quad y_j \in \{0,1\} && \forall j \in \mathcal{V}_0 \\
&\quad x_{ij} \ge 0 && \forall (i,j) \in \mathcal{A} \\
&\quad u_j \ge 0 && \forall j \in \mathcal{V}_0
\end{aligned}
$$

C.2 Scenario-based Two-stage Stochastic Programming Model

Parameters and Variables. S is the set of disruptive scenarios. Parameter p_s is the occurrence probability of scenario $s \in S$. Parameter $a_{js} = 1$ if facility $j \in \mathcal{V}_0$ is disrupted in scenario $s \in S$, and $a_{js} = 0$ otherwise. Variable x_{ijs} is the product flow on arc (i,j) in scenario $s \in S$. Variable u_{js} is the unsatisfied demand at node $j \in \mathcal{V}_D$ in scenario $s \in S$.

$$
\begin{aligned}
&\min \sum_{j\in\mathcal{V}_0} f_j y_j + \sum_{s\in S} p_s \Big(\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ijs} + \sum_{j\in\mathcal{V}_\mathcal{D}} \theta_j u_{js} \Big) && \\
&\text{s.t.} \sum_{i\in V_j^+} x_{jis} \le b_j && \forall s \in S, j \in \mathcal{V}_\mathcal{S} \\
&\quad \sum_{i\in V_j^+} x_{jis} = \sum_{i\in V_j^-} x_{ijs} && \forall s \in S, j \in \mathcal{V}_\mathcal{T}
\end{aligned}
$$

$$\sum_{i \in V_j^-} x_{ijs} + u_{js} = -b_j \qquad \forall s \in S, j \in \mathcal{V}_\mathcal{D}$$

$$\sum_{i \in V_j^+} x_{jis} \leq (1 - a_{js}) Q_j y_j \qquad \forall s \in S, j \in \mathcal{V}_0$$

$$y_j \in \{0, 1\} \qquad \forall j \in \mathcal{V}_0$$

$$a_{js} \in \{0, 1\} \qquad \forall j \in \mathcal{V}_0, \forall s \in S$$

$$x_{ijs} \geq 0 \qquad \forall s \in S, (i, j) \in \mathcal{A}$$

$$u_{js} \geq 0 \qquad \forall s \in S, j \in \mathcal{V}_\mathcal{D}$$

Note that although a_{js} is a binary parameter in the model, we could allow it to be fractional, representing partial disruption.

APPENDIX D SUPPLEMENT TO CHAPTER 6

Section D introduces how to incorporate the drone energy consumption for waiting time into our mathematical model. Section D presents the detailed instance generation procedures. Section D provides the detailed solutions of the numerical tests discussed in Section 6.4.

D.1 Drone Energy Consumption for Waiting Time at Customer Locations

We can incorporate a non-zero energy consumption for waiting time at customer locations in our model. Specifically, we introduce new variables $w_i, \forall i \in N'$ to represent the waiting time for the opening of the time window at customer $i \in N'$. Variables $\tau_i, \forall i \in N^-$ now denote drones' arrival time at node $i \in N^-$. Correspondingly, constraints (6.14)–(6.16) are modified as follows to include w_i in the model

$$\tau_i + w_i + t_{ij} - M''_{ij}(1 - x_{ij}) \leq \tau_j \qquad \forall i \in N', j \in N^-, \tag{D.1}$$

$$a_i \leq \tau_i + w_i \leq b_i \qquad \forall i \in N', \tag{D.2}$$

$$a_{n+1} \leq \tau_{n+1} \leq b_{n+1}, \tag{D.3}$$

$$\tau_i + w_i + (t_{i,n+1} + t_{0j}) \leq \tau_j + (1 - z_{ij})M'''_{ij} \qquad \forall i, j \in N', i \neq j, \tag{D.4}$$

$$w_i \geq 0 \qquad \forall i \in N'. \tag{D.5}$$

Now we set $M''_{ij} = b_i + t_{ij}$, and M'''_{ij} take the same values as before. We assume the unit energy consumption for waiting (e.g., performing sensing activities, hovering, etc.) as γ (kWh/s). Then, constraints (6.11) are replaced by

$$f_0 + k'(W + m + q_{0j})^{\frac{3}{2}} t_{0j} \leq M_{0j}(1 - x_{0j}) + f_j \quad \forall j \in N', \tag{D.6}$$

$$f_i + \gamma w_i + k'(W + m + q_{ij})^{\frac{3}{2}} t_{ij} \leq M_{ij}(1 - x_{ij}) + f_j \quad \forall i \in N', j \in N^-, \tag{D.7}$$

where constraints (D.6) establish the energy relationship between the starting depot 0 and customer i, and constraints (D.7) are the energy relationship between customer i and node j (which can be a customer node or the ending depot $n + 1$). M_{ij} take the same values as before. The objective function becomes

$$\min \sum_{(i,j) \in A} (c_{ij} x_{ij} + \delta e_{ij}) + \sum_{i \in I} \delta \gamma w_i. \tag{D.8}$$

Then, our solution schemes can be directly applied for this extension.

D.2 Instance Generation Procedures

This section presents the detailed procedures for instance generation.

D.2.1 New Benchmark Instances (Set A)

In this set, we consider two types of instances and each has 10–50 customers. For type 1 instances, named *Set* A_1, the depots are located at the lower left corner of the region. For type 2 instances, named *Set* A_2, the depots are in the middle of the region. For a fixed number of customers in each type, we generate 5 instances. Our instances are labeled as *Set_*A_x*_Cust_*Y*_*Z, which represents that this is the Zth instance of Y customers in Set A_x.

Based on the size of drones, we consider the delivery of relatively lightweight items (including those like medicines). The demand of the first 40% of customers is drawn uniformly from $[0.1, 0.7]$ and the demand of the remaining customers is drawn uniformly from $[0.1, 1.5]$. We set $K = \lceil \frac{\sum_{i \in N'} d_i}{3Q} \rceil$, that is, we expect that each drone can perform 3 or more trips on average. For Set A_1, the coordinate of the depot is (0, 0). The $x-$coordinate and $y-$coordinate of each customer is drawn uniformly from $[0, 480]$. Since we assume travel distance and travel time are the same, if a customer is located at (0, 480), then the travel time from the depot to this customer would be 480 seconds. Meanwhile, we let $c_{ij} = t_{ij}\ \forall (i,j) \in A$. For the depots, we set $a_0 = a_{n+1} = 0$ and generate the right-hand side of the time window as follows: We first compute the travel time between the depot and each customer, i.e., t_{0j}, and rank them in a non-increasing order; we then sum up the first hth numbers in order, where $h = \lceil \frac{|N'|}{K} \rceil$ and the sum is denoted as s. Finally, we set $b_0 = b_{n+1} = \lceil 2s \rceil$. This generation scheme is based on the idea that, in an extreme situation, each drone trip only involves one customer and each drone performs at most h trips. And all the deliveries can be finished within $\lceil 2s \rceil$ time limit. As travel time satisfies triangle inequality, the earliest time that a customer j can be serviced is t_{0j}, and the latest time that a drone must leave j is $b_{n+1} - t_{j,n+1}$. To create customers' time windows, we refer to the method in [96]. We first randomly generate the center of the time window o_j from $[t_{0j}, b_{n+1} - t_{j,n+1}]$ using uniform distribution, then we generate the time window's width w_j as a normally distributed random number whose mean is $0.25(b_{n+1} - t_{j,n+1} - t_{0j})$ and standard deviation is $0.05(b_{n+1} - t_{j,n+1} - t_{0j})$. We set $a_j = \max(\lceil t_{0j} \rceil, \lfloor o_j - 0.5 w_j \rfloor)$, $b_j = \min\{\lfloor b_{n+1} - t_{j,n+1} \rfloor, \lfloor o_j + 0.5 w_j \rfloor\}$. For Set A_2, the coordinate of the depot is (480, 480). The x-coordinate and y-coordinate of each customer is drawn uniformly from $[0, 960]$. The method of generating the time windows is the same as that of Set A_1.

D.2.2 Instances Extended From Solomon's Instances (Set B)

We generate this set of instances based on the principle of minimal modifications to the original data. To fit Solomon's instances, we need to add a service time $s_i, \forall i \in N'$ to constraints (6.14) and (6.16) when conducting our numerical tests. We also make some modifications to customers' demands to fit the drone's payload and to allow multi-trip operations. In particular, for type C2 and RC2 instances with the first 25 and 40 customers, demands are multiplied by 0.03, because the minimal and the maximal demands are 10 and 40, respectively. For type R2 instances, demands are multiplied by 0.05 for those with the first 25 customers, because the minimal and the maximal demand are 2 and 29, respectively; demands are multiplied by 0.045 for those with the first 40 customers, because the maximal demand now becomes 31. We determine the number of drones as described in the former section.

D.3 Detailed Results

In this section, we first present the results of another energy strategy, which might be used to guarantee the feasibility of trips. We then provide the detailed results of our numerical tests in Section 6.4, which are also available at `https://sites.google.com/view/chengchun/instances`.

D.3.1 Results of Over-estimated Linear Energy Function

Based on the parameters of drones, the linear power function (6.9) with over-estimated parameters takes the form $P = 259.6m_t + 185.6$, where m_t is the sum of battery weight and payload. Note that the unit of power is *Watt* here. We calculate the gap in power between the nonlinear function and the over-estimated function as *(over-estimated − nonlinear)/nonlinear* $\times$ *100%*, by taking m_t from 0.0 to 3.0 with a step 0.1. It shows that the maximal gap is 11.6% and the average gap is 6.9%.

We further give an example to demonstrate that the linear model with an over-estimated energy function may generate solutions with more trips and energy consumption. The experiment was conducted on instance *Set_A$_2$_Cust_20_3* for model R, and results are reported in Table D.1. The unit of energy is kWh and the *energy gap* is calculated as *(over-estimated energy − nonlinear energy)/nonlinear energy* $\times$ *100%*.

Table D.1 shows that the linear model has one more trip than the nonlinear model. This is because fewer customers might be covered in a single trip when the energy consumption is overestimated. Take the trips marked in red for example. If the nonlinear model is used, we could have customer 6 in the trip; however, as energy is estimated higher than the actual value, the linear model cannot visit customer 6 after visiting customer 16. The total energy

Table D.1 Results of instance *Set_A$_2$_Cust_20_3* for model R with an over-estimated energy function

Nonlinear model		Linear approximation model with the over-estimated energy function			
Trips	Energy	Trips	Over-estimated energy	Nonlinear energy	Energy gap (%)
[0, 2, 5, 4, 20, 21]	0.2660	[0, 2, 5, 15, 21]	0.1617	0.1555	3.99
[0, 3, 10, 21]	0.2427	[0, 3, 10, 21]	0.2553	0.2427	5.19
[0, 9, 16, 6, 21]	0.2616	[0, 4, 20, 21]	0.2341	0.2193	6.75
[0, 7, 8, 11, 21]	0.2520	[0, 6, 21]	0.2225	0.2049	8.59
[0, 15, 21]	0.0906	[0, 7, 8, 11, 21]	0.2604	0.2520	3.33
[0, 12, 21]	0.0695	[0, 9, 16, 21]	0.1860	0.1753	6.10
[0, 13, 21]	0.1882	[0, 12, 21]	0.0737	0.0695	6.04
[0, 14, 21]	0.1561	[0, 13, 21]	0.1963	0.1882	4.30
[0, 17, 21]	0.0254	[0, 14, 21]	0.1636	0.1561	4.80
[0, 18, 1, 21]	0.1636	[0, 17, 21]	0.0265	0.0254	4.33
[0, 19, 21]	0.0980	[0, 18, 1, 21]	0.1727	0.1636	5.56
		[0, 19, 21]	0.1023	0.0980	4.39
Total energy	1.8137		2.0551	1.9505	

consumption of the nonlinear model and the linear model are 1.8137 kWh and 1.9505 kWh respectively, resulting in a gap $= (1.9505 - 1.8137)/1.8137 \times 100\% = 7.54\%$.

D.3.2 Detailed Results of Numerical Tests

Table D.2 Results for Set A instances with 10–30 customers

			R_e										E_e										$(R+E)_e$									
	Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	10	1	2930.4	2930.4	0.0	0.7	2816.4	0	130	17	2	5	203.3	203.3	0.0	0.8	187.0	0	152	21	2	5	3133.6	3133.6	0.0	0.9	3011.8	0	200	21	2	5
		2	4426.1	4426.1	0.0	0.1	4337.4	1	24	17	2	6	314.6	314.6	0.0	0.4	313.1	0	96	25	2	6	4740.7	4740.7	0.0	0.2	4695.6	0	80	16	2	6
		3	4252.3	4252.3	0.0	0.2	4205.5	0	68	13	2	5	305.5	305.5	0.0	0.5	303.9	0	125	59	2	5	4557.8	4557.8	0.0	0.5	4555.4	0	77	43	2	5
		4	4105.3	4105.3	0.0	1.2	3692.5	0	148	44	2	5	287.4	287.4	0.0	1.9	252.6	1	179	41	2	5	4392.7	4392.7	0.0	1.3	3886.2	0	180	22	2	5
		5	4225.0	4225.0	0.0	0.2	4070.9	0	91	16	2	5	301.0	301.0	0.0	1.1	288.8	1	139	11	2	5	4526.0	4526.0	0.0	0.4	4349.3	0	137	24	2	5
	15	1	6601.6	6601.6	0.0	14.7	6286.3	3	363	38	3	7	473.1	473.1	0.0	18.0	452.2	1	472	35	3	7	7074.7	7074.2	0.0	12.8	6742.0	0	316	38	3	7
		2	4114.7	4114.7	0.0	6.2	3831.2	2	155	20	2	4	284.5	284.5	0.0	18.5	255.0	6	787	35	2	4	4399.8	4399.8	0.0	22.5	4051.3	7	550	33	2	4
		3	5575.0	5574.7	0.0	761.0	4816.1	5	686	19	2	6	387.7	387.7	0.0	318.3	291.8	1	777	32	2	6	5970.1	5969.5	0.0	1391.5	5165.0	2	761	28	2	6
		4	5125.9	5125.4	0.0	31.3	4811.3	0	202	29	2	7	363.6	363.6	0.0	22.9	319.2	0	333	39	2	7	5493.1	5492.8	0.0	40.4	5124.8	0	376	29	2	7
		5	6900.0	6899.4	0.0	70.0	6210.1	7	597	32	2	7	484.3	484.3	0.0	35.6	402.7	5	608	25	2	8	7385.9	7385.5	0.0	63.8	6655.9	5	590	35	2	7
	20	1	7720.1	7720.1	0.0	2.3	7638.3	0	118	38	4	11	566.0	566.0	0.0	6.2	552.7	0	289	44	4	11	8287.7	8287.7	0.0	2.5	8247.6	0	384	44	4	11
		2	8912.6	8912.3	0.0	44.1	8574.7	2	404	33	4	10	638.3	638.2	0.0	35.4	554.8	1	678	36	4	10	9550.9	9550.8	0.0	75.7	9105.8	1	775	29	4	10
		3	8219.7	8219.7	0.0	18.2	8112.8	0	162	45	4	10	597.0	596.9	0.0	93.6	525.3	1	540	34	4	10	8819.3	8819.3	0.0	24.7	8636.7	0	541	37	4	10
		4	6229.5	6229.0	0.0	29.3	6073.8	0	454	34	3	9	459.2	459.2	0.0	141.4	379.5	2	479	35	3	9	6697.0	6696.5	0.0	34.0	6455.1	0	460	36	3	9
		5	7269.4	7268.7	0.0	717.3	6796.6	6	611	27	3	9	505.5	505.4	0.0	84.9	481.9	4	805	48	3	9	7785.3	7784.5	0.0	983.3	7207.8	2	918	32	3	9
	25	1	9962.5	9961.5	0.0	59.7	9814.6	0	136	37	4	12	712.3	712.2	0.0	76.9	636.3	1	891	33	4	12	10683.5	10682.5	0.0	217.0	10436.5	5	1454	31	4	12
		2	8064.6	8063.8	0.0	7817.4	7611.9	10	1002	22	3	10	575.1	575.1	0.0	10468.2	496.1	14	1495	50	3	10	8639.7	8638.9	0.0	8600.1	8105.8	13	2458	33	3	10
		3	9422.6	9421.7	0.0	609.4	9182.2	0	263	28	4	10	675.2	675.2	0.0	1244.2	562.4	5	1741	31	4	10	10097.8	10096.8	0.0	1186.0	9774.8	0	666	26	4	10
		4	9460.6	9459.8	0.0	443.8	9135.6	3	509	45	4	13	689.2	689.2	0.0	736.1	581.5	3	854	38	4	13	10149.8	10148.8	0.0	324.8	9702.3	2	1199	58	4	13
		5	10398.1	10213.6	1.8	14400.0	9786.8	2	852	39	4	11	747.8	747.7	0.0	10017.9	614.9	1	1429	47	4	12	11209.3	10899.6	2.8	14400.0	10428.0	2	1783	43	4	12
	30	1	9165.8	9164.9	0.0	2440.8	8841.7	7	1922	42	5	13	667.9	667.8	0.0	3225.0	554.2	3	1804	48	5	13	9836.3	9835.3	0.0	8584.4	9419.8	3	1947	46	5	13
		2	11811.4	11810.6	0.0	360.6	11564.9	1	484	40	5	14	858.0	857.9	0.0	942.4	734.0	0	1113	39	5	14	12669.3	12668.1	0.0	1732.8	12361.7	1	1476	45	5	14
		3	11530.4	11289.6	2.1	14400.0	10932.4	2	1243	34	5	14	831.8	807.8	2.9	14400.0	706.7	2	2226	33	5	14	12435.7	12077.9	2.9	14400.0	11681.0	2	1984	35	5	13
		4	11670.9	11669.7	0.0	2699.8	11281.7	0	949	45	5	13	845.5	845.4	0.0	11981.8	755.7	1	1829	55	5	13	12517.1	12515.9	0.0	5211.6	12073.9	0	1828	49	5	13
		5	11330.9	11222.6	1.0	14400.0	10924.6	9	1266	51	5	13	814.7	798.3	2.0	14400.0	729.8	11	2451	64	5	13	12090.4	12020.0	0.6	14400.0	11644.5	8	1560	60	5	13
Set A_2	10	1	4668.6	4668.6	0.0	0.2	4475.2	0	62	9	2	5	324.2	324.2	0.0	0.5	292.5	2	174	15	2	6	5000.2	5000.2	0.0	0.2	4772.6	0	75	2	2	5
		2	5447.2	5447.2	0.0	0.1	5385.9	0	44	2	2	6	379.9	379.9	0.0	0.1	378.7	0	128	8	2	6	5827.1	5827.1	0.0	0.1	5825.6	0	96	2	2	6
		3	4946.7	4946.7	0.0	0.4	4910.0	0	48	30	2	5	324.9	324.9	0.0	0.2	323.6	0	77	12	2	5	5271.5	5271.5	0.0	0.2	5267.8	0	45	15	2	5
		4	5775.2	5775.2	0.0	0.9	5046.2	0	26	18	2	5	382.9	382.9	0.0	1.3	311.9	0	222	20	2	5	6158.6	6158.6	0.0	0.8	5401.1	0	109	15	2	5
		5	5178.5	5178.5	0.0	0.6	4785.7	0	24	24	2	6	356.9	356.9	0.0	0.4	309.0	0	112	14	2	6	5535.3	5535.3	0.0	0.3	5112.1	0	32	15	2	6
	15	1	6427.6	6427.6	0.0	1.9	6295.3	0	88	44	3	8	444.2	444.2	0.0	1.2	428.6	0	140	25	3	8	6871.8	6871.8	0.0	1.0	6868.4	0	211	22	3	8
		2	7995.4	7995.4	0.0	10.9	7072.5	3	337	41	3	8	534.0	534.0	0.0	7.6	446.4	1	295	33	3	10	8537.6	8537.6	0.0	10.1	7542.8	2	351	43	3	8
		3	6173.9	6173.9	0.0	0.7	6165.3	2	40	33	3	9	436.3	436.3	0.0	3.1	398.6	0	246	29	3	9	6614.5	6614.5	0.0	1.5	6465.5	1	240	28	3	9
		4	8238.9	8238.9	0.0	5.5	7249.5	6	180	14	2	8	541.3	541.3	0.0	6.5	463.7	4	345	19	2	8	8780.2	8780.0	0.0	10.0	7878.0	5	322	25	2	8
		5	8114.1	8114.1	0.0	6.0	7565.1	1	116	38	3	9	544.0	544.0	0.0	2.1	508.1	2	346	32	3	10	8674.8	8674.5	0.0	5.8	7978.3	1	365	29	3	9
	20	1	10684.6	10684.6	0.0	46.5	9677.3	5	243	28	4	13	741.3	741.3	0.0	57.1	643.6	1	666	31	4	13	11425.9	11425.0	0.0	43.0	10524.4	0	323	34	4	13
		2	9093.2	9093.2	0.0	20.0	8408.8	1	181	26	3	10	631.2	631.2	0.0	36.4	540.8	2	428	42	3	10	9733.1	9733.1	0.0	27.1	9045.0	0	392	41	3	10
		3	9444.0	9444.0	0.0	14.5	9046.6	2	240	43	3	11	652.9	652.9	0.0	21.1	577.3	2	342	41	3	11	10096.9	10096.9	0.0	15.4	9590.1	2	313	34	3	11
		4	8858.5	8858.5	0.0	12.6	8573.3	0	244	42	4	10	619.5	619.5	0.0	4.6	595.0	1	438	44	4	11	9495.5	9495.5	0.0	22.0	9136.3	2	712	42	4	10
		5	7750.7	7750.4	0.0	41.6	7341.9	2	387	34	3	9	551.4	551.4	0.0	67.2	463.9	2	564	42	3	9	8302.2	8302.0	0.0	57.9	7805.1	3	764	31	3	9
	25	1	10667.2	10666.2	0.0	182.1	10044.2	1	394	32	4	12	752.2	752.1	0.0	570.9	646.2	5	1216	30	4	13	11440.1	11439.1	0.0	229.7	10733.9	2	408	50	4	12
		2	11623.1	11622.4	0.0	78.7	11094.8	0	52	35	4	13	805.3	805.2	0.0	186.4	707.5	6	1110	27	4	13	12430.0	12429.0	0.0	159.0	11747.4	4	798	34	4	13
		3	10243.9	10243.0	0.0	609.9	9584.1	5	769	39	4	12	717.8	717.7	0.0	668.6	601.7	3	1100	39	4	12	10977.3	10976.2	0.0	971.0	10216.3	3	1468	43	4	12
		4	11468.4	11468.4	0.0	44.4	11015.6	3	333	46	4	13	805.7	805.7	0.0	24.4	778.8	1	544	56	4	14	12279.3	12279.2	0.0	66.5	11705.7	2	736	39	4	13
		5	10986.4	10985.3	0.0	96.7	10415.8	2	799	31	4	13	784.1	784.0	0.0	54.8	668.3	5	635	34	4	14	11791.6	11790.5	0.0	139.2	11101.9	1	855	31	4	13
	30	1	14004.6	14003.2	0.0	6400.9	13082.9	4	525	43	5	16	983.4	983.3	0.0	1125.6	852.0	5	1232	39	5	17	15002.0	15000.5	0.0	5548.2	13939.4	3	1674	40	5	16
		2	11958.2	11957.0	0.0	5707.0	11162.0	0	572	49	4	14	840.2	814.1	3.1	14400.0	701.9	3	1256	40	4	14	12798.4	12797.1	0.0	10161.1	11913.3	2	573	55	4	14
		3	11432.8	11431.8	0.0	6124.3	10552.0	5	1332	33	5	14	802.6	802.5	0.0	12470.0	660.5	7	2277	33	5	14	12239.0	12237.8	0.0	7670.6	11183.5	4	1767	29	5	14
		4	10827.6	10826.6	0.0	373.0	10491.1	5	1174	38	4	13	763.6	763.5	0.0	7057.4	645.4	6	1438	30	4	13	11591.2	11590.0	0.0	596.6	11128.8	6	1788	35	4	13
		5	12382.2	12380.9	0.0	13527.0	11561.0	2	1033	43	4	13	878.7	855.7	2.6	14400.0	739.8	4	2876	35	4	14	13323.6	12999.4	2.4	14400.0	12321.1	5	1385	39	4	13

Trip: the number of trips performed by UAVs.

Table D.3 Results for linear approximation models

			R_a											$(R+E)_a$									
	Cust	Inst	UP	LB	Gap	CPU	RLB	SECs	UAVs	Trip	Viol▲	Linear energy	Nonlinear energy	UP	LB	Gap	CPU	RLB	SECs	UAVs	Trip	Viol	Linear energy	Nonlinear energy
Set A_1	10	1	2930.4	2930.4	0.0	0.8	2791.5	12	2	5	F▲	0.5204	0.5656	3117.4	3117.4	0.0	1.0	2996.2	23	2	5	F	0.5196	0.5645
		2	4426.1	4426.1	0.0	0.1	4385.3	10	2	6	F	0.7996	0.8740	4713.9	4713.9	0.0	0.1	4675.2	9	2	6	F	0.7996	0.8740
		3	4252.3	4252.3	0.0	0.2	4202.1	15	2	5	F	0.7789	0.8537	4531.3	4531.3	0.0	0.4	4528.0	20	2	5	F	0.7751	0.8486
		4	3948.8	3948.8	0.0	0.7	3655.7	16	2	4	T▲	0.7475	0.8268	4218.0	4218.0	0.0	1.3	3891.4	34	2	4	T	0.7475	0.8268
		5	4225.0	4225.0	0.0	0.3	4058.0	16	2	5	F	0.7638	0.8363	4499.9	4499.9	0.0	0.4	4368.2	19	2	5	F	0.7638	0.8363
	15	1	6529.0	6528.9	0.0	9.1	6276.7	22	3	7	T	1.2070	1.3284	6960.3	6960.3	0.0	10.4	6714.9	31	3	7	T	1.1981	1.3163
		2	4100.2	4100.2	0.0	9.6	3791.5	34	2	4	T	0.7629	0.8389	4373.4	4373.0	0.0	9.8	4018.0	22	2	4	T	0.7572	0.8314
		3	5315.6	5315.6	0.0	191.7	4861.4	31	2	5	T	0.9765	1.0718	5664.9	5664.4	0.0	196.8	5244.0	29	2	5	T	0.9705	1.0637
		4	5125.9	5125.9	0.0	66.0	4837.1	38	2	7	F	0.9368	1.0273	5461.2	5461.2	0.0	38.0	5145.9	30	2	7	F	0.9313	1.0199
		5	6722.6	6722.6	0.0	12.4	6196.3	25	2	6	T	1.2568	1.3845	7165.7	7165.5	0.0	17.1	6635.4	29	2	7	T	1.2131	1.3246
	20	1	7720.1	7720.1	0.0	1.6	7675.0	42	4	11	F	1.4327	1.5766	8235.7	8235.7	0.0	1.7	8178.0	37	4	11	F	1.4327	1.5766
		2	8773.9	8773.9	0.0	20.4	8546.9	29	4	10	T	1.6400	1.8091	9357.1	9357.1	0.0	26.8	9140.0	30	4	10	T	1.6202	1.7824
		3	8219.7	8219.7	0.0	24.6	8111.5	44	4	10	F	1.5268	1.6814	8765.1	8765.1	0.0	9.9	8650.2	42	4	10	F	1.5148	1.6653
		4	6229.5	6229.4	0.0	48.8	6075.1	33	3	9	F	1.1762	1.2989	6652.9	6652.8	0.0	25.6	6495.1	40	3	9	F	1.1762	1.2989
		5	7174.4	7173.7	0.0	159.8	6785.5	27	3	9	T	1.3128	1.4391	7640.7	7640.3	0.0	240.6	7257.9	45	3	9	T	1.2953	1.4156
	25	1	9962.5	9961.6	0.0	55.6	9799.6	31	4	12	F	1.8249	2.0027	10619.5	10619.5	0.0	63.3	10460.3	51	4	12	F	1.8249	2.0027
		2	7818.1	7817.4	0.0	71.8	7648.2	26	3	10	T	1.4362	1.5779	8330.7	8330.7	0.0	56.1	8165.8	35	3	10	T	1.4240	1.5615
		3	9422.6	9421.7	0.0	581.3	9191.1	41	4	10	F	1.7205	1.8866	10039.0	10039.0	0.0	522.4	9800.3	30	4	10	F	1.7123	1.8756
		4	9460.6	9459.6	0.0	321.9	9120.1	42	4	13	F	1.7414	1.9145	10087.5	10086.5	0.0	266.6	9738.7	42	4	13	F	1.7414	1.9145
		5	10398.1	10397.1	0.0	14287.6	9814.1	50	4	11	T	1.9580	2.1631	11097.5	11096.4	0.0	12119.9	10470.7	47	4	11	F	1.9427	2.1425
	30	1	9091.6	9091.0	0.0	544.5	8845.7	38	5	13	T	1.6714	1.8374	9691.7	9691.3	0.0	330.1	9479.7	54	5	13	T	1.6668	1.8311
		2	11811.4	11810.3	0.0	356.3	11601.2	40	5	14	F	2.1760	2.3937	12592.0	12591.0	0.0	403.7	12360.8	31	5	14	F	2.1682	2.3834
		3	11543.0	11358.3	1.6	14400.0	10938.3	45	5	14	T	2.1197	2.3291	12289.7	12288.5	0.0	11300.9	11695.5	33	5	14	F	2.1094	2.3154
		4	11670.9	11669.9	0.0	1537.7	11359.6	37	5	13	F	2.1578	2.3745	12441.4	12440.3	0.0	1834.2	12029.1	38	5	13	F	2.1395	2.3498
		5	11221.9	11220.7	0.0	2553.1	10897.0	39	5	13	T	2.0783	2.2881	11968.4	11967.3	0.0	1573.8	11683.8	56	5	13	T	2.0724	2.2805
Set A_2	10	1	4668.6	4668.6	0.0	0.2	4475.2	4	2	5	F	0.8428	0.9212	4972.0	4972.0	0.0	0.2	4770.1	7	2	5	F	0.8428	0.9212
		2	5447.2	5447.2	0.0	0.1	5447.2	1	2	6	F	0.9694	1.0552	5796.1	5796.1	0.0	0.1	5773.8	1	2	6	F	0.9694	1.0552
		3	4946.7	4946.7	0.0	0.3	4842.9	18	2	5	F	0.8483	0.9136	5249.1	5249.1	0.0	0.2	5155.9	7	2	5	F	0.8400	0.9024
		4	5775.2	5775.2	0.0	1.3	5053.4	27	2	5	F	0.9934	1.0714	6131.1	6131.1	0.0	0.8	5384.6	25	2	5	F	0.9888	1.0651
		5	5178.5	5178.5	0.0	0.3	4711.4	17	2	6	F	0.9139	0.9913	5507.5	5507.5	0.0	0.4	5073.5	31	2	6	F	0.9139	0.9913
	15	1	6427.6	6427.6	0.0	1.1	6424.2	47	3	8	F	1.1361	1.2340	6836.6	6836.6	0.0	0.7	6836.6	35	3	8	F	1.1361	1.2340
		2	7762.6	7762.6	0.0	7.4	7038.7	40	3	9	T	1.3402	1.4462	8244.7	8244.7	0.0	4.5	7679.7	31	3	9	T	1.3390	1.4447
		3	6173.9	6173.9	0.0	0.9	6132.3	61	3	9	F	1.1361	1.2453	6577.1	6577.1	0.0	0.7	6415.8	28	3	9	F	1.1202	1.2239
		4	8238.9	8238.9	0.0	8.0	7277.4	17	2	8	F	1.4012	1.5068	8742.4	8742.4	0.0	5.0	7898.3	23	2	8	F	1.3987	1.5038
		5	8114.1	8114.1	0.0	6.4	7525.2	36	3	9	F	1.4349	1.5575	8630.6	8630.6	0.0	7.1	7957.6	26	3	9	F	1.4349	1.5575
	20	1	10684.6	10683.7	0.0	34.4	9689.7	41	4	13	F	1.8992	2.0692	11365.6	11365.1	0.0	22.2	10454.5	50	4	13	F	1.8916	2.0590
		2	9093.2	9092.7	0.0	33.7	8492.2	37	3	10	F	1.6296	1.7775	9679.9	9679.0	0.0	31.7	8901.2	31	3	10	F	1.6296	1.7775
		3	9444.0	9444.0	0.0	23.1	9167.5	35	3	11	F	1.6753	1.8212	10045.1	10045.1	0.0	17.4	9728.2	41	3	11	F	1.6696	1.8137
		4	8858.5	8858.5	0.0	14.5	8539.0	37	4	10	F	1.6307	1.7899	9440.1	9440.1	0.0	8.1	9056.6	30	4	10	F	1.6154	1.7692
		5	7708.5	7708.0	0.0	24.8	7323.1	25	3	9	T	1.4241	1.5645	8216.4	8216.4	0.0	24.3	7832.3	32	3	9	T	1.4109	1.5466
	25	1	10667.2	10666.4	0.0	289.0	10062.3	40	4	12	F	1.9581	2.1469	11366.1	11365.4	0.0	161.4	10735.0	32	4	12	T	1.9338	2.1138
		2	11623.1	11623.1	0.0	136.7	11028.5	36	4	13	F	2.0624	2.2431	12365.5	12365.5	0.0	76.5	11793.3	42	4	13	F	2.0624	2.2431
		3	9984.1	9984.1	0.0	83.3	9571.5	37	4	12	T	1.8217	1.9937	10634.6	10634.6	0.0	39.5	10242.6	41	4	12	T	1.8068	1.9737
		4	11468.4	11468.4	0.0	58.2	10998.9	46	4	13	T	2.1005	2.3022	12211.3	12211.3	0.0	72.0	11725.9	40	4	13	F	2.0637	2.2527
		5	10854.2	10853.4	0.0	50.8	10465.3	28	4	14	T	2.0058	2.2046	11571.2	11571.2	0.0	29.0	11106.6	34	4	14	T	1.9916	2.1856
	30	1	13998.6	13997.3	0.0	1292.2	13068.9	42	5	16	T	2.5527	2.7955	14909.4	14908.0	0.0	1466.8	13960.3	39	5	16	T	2.5299	2.7653
		2	11958.2	11957.2	0.0	2828.7	11183.9	43	4	14	F	2.1636	2.3643	12728.9	12727.7	0.0	2617.0	11921.9	41	4	14	F	2.1410	2.3338
		3	11144.0	11144.0	0.0	1012.5	10558.9	29	5	13	T	2.0439	2.2413	11878.9	11878.2	0.0	1065.3	11232.4	32	5	13	T	2.0414	2.2380
		4	10827.6	10826.6	0.0	291.7	10481.7	38	4	13	F	1.9436	2.1211	11527.3	11526.5	0.0	179.7	11138.1	28	4	13	F	1.9435	2.1209
		5	12244.9	12243.8	0.0	4124.7	11640.1	44	4	12	T	2.2363	2.4497	13050.0	13048.7	0.0	2316.2	12319.9	37	4	12	T	2.2363	2.4497

▲: Whether there exist trips which violate the energy capacity constraint. If yes, a 'T' is used, 'F' otherwise.

Table D.4 Results for Set A instances with tighter time windows

			R_e										E_e										$(R+E)_e$									
	Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	10	1	3135.3	3135.3	0.0	0.2	2952.4	0	72	17	2	5	212.7	212.7	0.0	0.6	191.0	0	120	33	2	5	3351.5	3351.5	0.0	0.5	3169.0	0	152	24	2	5
		2	5499.8	5499.8	0.0	0.2	4679.7	1	26	12	2	7	364.6	364.6	0.0	0.2	317.7	1	103	8	2	7	5864.5	5864.5	0.0	0.4	4989.3	0	51	11	2	7
		3	4175.2	4175.2	0.0	0.3	4118.3	0	68	19	2	5	304.2	304.2	0.0	0.2	291.2	0	77	13	2	5	4479.4	4479.4	0.0	0.2	4425.9	0	77	23	2	5
		4	4315.4	4315.4	0.0	0.4	3956.9	0	36	34	2	5	296.6	296.6	0.0	0.6	291.5	1	120	55	2	5	4615.9	4615.9	0.0	0.3	4155.8	1	210	14	2	5
		5	4184.3	4184.3	0.0	0.1	4177.9	0	18	6	2	5	300.8	300.8	0.0	0.1	298.6	0	128	4	2	5	4486.2	4486.2	0.0	0.1	4433.1	1	45	4	2	5
	15	1	6636.2	6636.2	0.0	6.9	6335.3	3	393	30	3	7	469.8	469.8	0.0	5.2	426.1	2	540	26	3	7	7106.1	7105.8	0.0	7.1	6756.4	2	291	42	3	7
		2	3994.4	3994.4	0.0	1.8	3777.1	3	215	22	2	4	292.1	292.1	0.0	9.7	262.1	6	397	23	2	4	4286.5	4286.5	0.0	1.5	4001.6	0	120	20	2	4
		3	5583.5	5583.0	0.0	109.7	4955.7	3	353	30	2	6	398.3	398.2	0.0	186.7	298.6	2	758	35	2	6	5981.9	5981.6	0.0	146.9	5292.6	2	441	32	2	6
		4	5237.8	5237.6	0.0	13.1	4816.9	0	230	22	2	7	383.1	383.0	0.0	25.4	303.3	0	508	35	2	7	5620.9	5620.7	0.0	13.7	5248.2	0	485	44	2	7
		5	6956.9	6956.5	0.0	29.5	6248.5	4	300	29	2	7	481.7	481.7	0.0	11.9	402.9	4	361	35	2	8	7467.7	7467.0	0.0	39.5	6643.6	4	446	30	2	7
	20	1	7729.5	7729.5	0.0	0.7	7717.7	0	42	24	4	11	559.0	559.0	0.0	3.5	536.5	0	219	36	4	11	8288.5	8288.5	0.0	0.8	8277.3	0	281	27	4	11
		2	9253.7	9252.9	0.0	62.6	8741.4	0	565	37	4	10	666.4	666.3	0.0	333.0	549.8	2	744	23	4	11	9932.2	9931.4	0.0	175.3	9308.3	2	707	37	4	10
		3	8363.5	8363.5	0.0	16.7	8166.9	0	230	33	4	10	603.7	603.7	0.0	27.4	533.7	2	552	42	4	10	8976.2	8976.2	0.0	11.6	8726.1	0	240	47	4	10
		4	6429.8	6429.4	0.0	199.5	6049.3	1	728	20	3	9	446.2	446.1	0.0	68.8	377.3	0	810	28	3	9	6876.0	6876.0	0.0	116.4	6433.9	0	725	24	3	9
		5	7284.8	7284.1	0.0	131.1	6868.7	3	404	31	3	9	506.4	506.4	0.0	101.5	424.3	2	855	30	3	9	7791.2	7790.6	0.0	134.8	7304.7	4	646	36	3	9
	25	1	9992.0	9991.1	0.0	44.1	9824.5	4	465	25	4	12	714.5	714.4	0.0	37.5	690.5	3	603	30	4	12	10714.3	10713.7	0.0	49.7	10470.6	1	226	38	4	12
		2	8311.8	8310.9	0.0	2566.6	7616.0	6	1276	28	3	10	601.9	568.2	5.6	14400.0	460.0	11	1328	31	3	10	8918.8	8917.9	0.0	2700.7	8110.6	5	1206	53	3	10
		3	9644.4	9643.4	0.0	1034.5	9246.8	5	797	27	4	10	695.2	695.2	0.0	1121.3	566.7	1	979	27	4	10	10339.6	10338.6	0.0	1684.2	9840.3	1	816	33	4	10
		4	9699.3	9698.3	0.0	2054.5	9129.2	5	1157	49	4	13	698.5	698.5	0.0	1636.5	591.9	4	1606	45	4	13	10397.8	10396.8	0.0	1866.6	9732.4	2	1638	49	4	13
		5	10502.7	10501.6	0.0	6189.6	9848.1	3	1075	49	4	12	744.0	726.6	2.3	14400.0	614.2	4	1969	39	4	12	11255.9	11254.8	0.0	9598.6	10455.0	2	1113	45	4	12
	30	1	9314.1	9313.2	0.0	1912.9	8936.0	1	1684	40	5	13	680.2	680.1	0.0	8464.1	603.5	1	2688	44	5	13	9996.3	9995.3	0.0	1503.5	9534.8	1	1496	44	5	13
		2	12099.3	12098.1	0.0	1022.8	11784.0	1	1145	38	5	14	871.2	871.1	0.0	3270.8	755.2	0	2205	39	5	14	12970.5	12969.3	0.0	710.3	12559.7	0	1835	39	5	14
		3	11560.9	11403.8	1.4	14400.0	10922.5	1	697	38	5	14	831.8	803.8	3.4	14400.0	691.7	1	1588	33	5	14	12452.0	12225.3	1.8	14400.0	11667.0	3	1419	28	5	14
		4	11679.0	11677.9	0.0	1092.6	11337.4	2	1191	47	5	13	852.2	852.1	0.0	2229.4	710.3	0	1756	37	5	13	12533.7	12532.5	0.0	1113.3	12060.5	0	2149	40	5	13
		5	11508.9	11330.3	1.6	14400.0	11019.1	9	2249	56	5	13	824.8	808.4	2.0	14400.0	734.4	6	1863	57	5	13	12333.8	12123.0	1.7	14400.0	11687.5	2	2029	45	5	13
Set A_2	10	1	5287.2	5287.2	0.0	0.1	5239.2	0	20	10	2	6	355.7	355.7	0.0	0.4	355.2	0	32	25	2	6	5642.9	5642.9	0.0	0.2	5527.3	0	32	5	2	6
		2	6180.3	6180.3	0.0	0.1	6161.0	1	21	9	2	7	415.1	415.1	0.0	0.1	415.1	1	51	8	2	7	6595.4	6595.4	0.0	0.1	6592.6	1	51	5	2	7
		3	5675.9	5675.9	0.0	0.2	5596.2	0	20	21	2	6	358.4	358.4	0.0	0.1	354.2	0	64	5	2	6	6034.4	6034.4	0.0	0.2	5997.5	0	64	27	2	6
		4	5943.9	5943.9	0.0	0.2	5775.4	0	66	13	2	6	385.1	385.1	0.0	0.3	381.3	0	64	17	2	6	6329.0	6329.0	0.0	0.2	6133.7	0	32	15	2	6
		5	5656.0	5656.0	0.0	0.1	5473.3	1	84	10	2	7	370.1	370.1	0.0	0.2	351.7	1	84	8	2	7	6026.1	6026.1	0.0	0.3	5824.9	2	84	10	2	7
	15	1	6198.2	6198.2	0.0	0.7	6022.9	1	48	33	3	8	429.1	429.1	0.0	0.8	424.9	0	186	34	3	8	6627.3	6627.3	0.0	0.8	6420.4	0	115	21	3	8
		2	7759.7	7759.7	0.0	1.0	7415.4	0	132	32	3	8	533.1	533.1	0.0	1.1	506.5	0	166	27	3	9	8301.2	8301.2	0.0	0.9	7921.5	0	190	27	3	8
		3	6825.2	6825.2	0.0	1.0	6524.9	0	128	31	3	9	470.2	470.2	0.0	1.1	422.7	1	171	21	3	9	7295.4	7295.4	0.0	1.2	6920.2	0	178	25	3	9
		4	8627.0	8627.0	0.0	1.7	7999.3	3	148	34	2	8	565.3	565.3	0.0	2.7	499.5	3	360	21	2	9	9207.1	9207.1	0.0	2.2	8604.5	3	256	37	2	8
		5	8187.3	8187.3	0.0	0.7	8035.0	1	140	33	3	9	544.7	544.7	0.0	0.7	544.0	1	178	31	3	10	8743.9	8743.9	0.0	0.9	8599.1	0	246	44	3	9
	20	1	10542.8	10542.8	0.0	2.9	10058.5	3	378	22	4	12	732.4	732.4	0.0	2.8	672.8	2	461	34	4	13	11283.8	11282.8	0.0	2.5	10794.8	0	163	35	4	12
		2	9184.3	9184.3	0.0	15.4	8489.8	0	112	39	3	10	648.7	648.7	0.0	54.7	545.0	3	565	36	3	11	9834.2	9834.2	0.0	32.3	9072.8	2	277	35	3	10
		3	10565.5	10565.5	0.0	13.5	9821.2	4	256	23	3	11	714.0	713.9	0.0	11.4	654.4	4	380	30	3	11	11279.4	11279.4	0.0	14.6	10542.4	3	732	42	3	11
		4	9278.4	9278.4	0.0	9.1	8655.4	1	384	40	4	11	644.7	644.7	0.0	9.7	606.5	0	379	38	4	12	9925.9	9924.9	0.0	7.6	9234.8	1	314	32	4	11
		5	7929.8	7929.8	0.0	9.7	7610.9	1	614	38	3	9	562.7	562.7	0.0	17.7	492.5	1	365	29	3	10	8502.9	8502.9	0.0	9.8	8105.9	0	438	34	3	9
	25	1	10860.4	10859.6	0.0	432.8	10101.1	5	760	34	4	12	768.3	768.2	0.0	865.1	658.0	6	1445	38	4	12	11640.6	11639.5	0.0	223.9	10881.6	6	819	43	4	12
		2	11957.9	11956.9	0.0	52.5	11302.5	6	553	40	4	12	833.9	833.8	0.0	132.7	722.4	9	882	30	4	12	12794.5	12793.6	0.0	84.4	11962.1	2	374	25	4	12
		3	11106.8	11106.8	0.0	163.4	10176.9	3	225	25	4	13	754.9	754.9	0.0	158.0	648.4	3	494	37	4	13	11875.6	11874.6	0.0	262.0	10825.7	3	609	33	4	13
		4	11529.3	11529.3	0.0	16.8	11161.8	0	468	38	4	13	812.7	812.6	0.0	26.5	759.8	0	703	34	4	13	12342.3	12341.0	0.0	11.8	12037.1	0	382	54	4	13
		5	11174.2	11173.3	0.0	35.8	10622.9	0	719	34	4	14	801.5	801.5	0.0	21.2	758.0	1	718	47	4	15	11980.5	11980.5	0.0	26.9	11250.6	1	358	33	4	14
	30	1	14270.9	14269.5	0.0	1636.5	13189.4	5	741	47	5	17	1004.9	1004.8	0.0	2835.6	859.6	2	856	58	5	17	15283.9	15282.5	0.0	880.1	14064.2	2	1089	46	5	17
		2	11651.3	11650.3	0.0	132.5	11115.2	2	687	40	4	13	834.5	834.4	0.0	345.3	771.2	3	1035	58	4	14	12493.1	12493.1	0.0	130.2	11842.9	1	523	37	4	13
		3	11548.4	11547.2	0.0	697.8	10869.4	10	1074	46	5	14	818.3	818.2	0.0	2257.1	691.1	2	1742	33	5	14	12370.3	12369.2	0.0	678.8	11548.6	9	842	39	5	14
		4	11357.9	11356.8	0.0	526.9	10839.6	2	508	38	4	13	810.3	810.2	0.0	3828.6	679.4	8	1665	46	4	13	12178.5	12178.2	0.0	335.4	11488.5	4	303	35	4	13
		5	11998.2	11997.4	0.0	239.0	11581.4	3	822	34	4	13	843.1	843.0	0.0	3026.2	731.8	1	1395	32	4	13	12844.9	12844.1	0.0	513.2	12246.8	2	775	34	4	13

Table D.5 Results for Set A instances without time windows

			R_e										E_e										$(R+E)_e$									
	Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	10	1	2816.4	2816.4	0.0	0.4	2815.8	0	118	38	2	5	193.7	193.7	0.0	0.6	191.6	0	181	26	1	5	3011.9	3011.9	0.0	0.4	3008.9	0	158	27	2	5
		2	4426.1	4426.1	0.0	0.9	4228.5	0	36	50	2	6	310.2	310.2	0.0	0.9	284.3	0	221	14	2	6	4736.3	4736.3	0.0	0.5	4509.0	1	96	36	2	6
		3	4002.2	4002.2	0.0	0.2	4001.6	0	22	25	2	5	284.0	284.0	0.0	1.0	276.6	0	355	21	2	5	4288.9	4288.9	0.0	0.4	4288.2	0	155	34	2	5
		4	3547.3	3547.3	0.0	0.4	3491.5	0	46	14	2	4	258.0	258.0	0.0	1.9	246.9	1	288	32	2	4	3806.0	3806.0	0.0	0.5	3749.1	0	145	32	2	4
		5	4164.7	4164.7	0.0	0.6	4162.6	1	89	31	1	5	298.4	298.4	0.0	0.7	267.2	1	169	15	2	5	4463.1	4463.1	0.0	0.3	4449.2	1	76	30	1	5
	15	1	6417.5	6417.0	0.0	30.3	6282.6	2	260	36	3	7	458.8	458.8	0.0	20.0	440.2	2	621	39	3	7	6877.1	6876.4	0.0	35.0	6649.8	4	619	39	3	7
		2	3805.2	3805.2	0.0	1.1	3803.2	3	138	41	2	4	268.8	268.8	0.0	69.6	243.7	6	751	33	2	4	4076.3	4076.0	0.0	55.8	3730.7	4	558	32	2	4
		3	4925.5	4925.5	0.0	15.5	4839.1	1	534	33	2	5	356.2	356.1	0.0	31.4	337.8	3	999	34	2	5	5284.2	5283.7	0.0	13.6	5184.9	2	347	37	2	5
		4	5106.7	5106.2	0.0	2578.8	4707.2	0	785	29	2	7	358.0	358.0	0.0	70.4	326.7	0	829	25	2	7	5464.7	5464.3	0.0	521.1	5121.0	0	1025	41	2	7
		5	6153.8	6153.8	0.0	11.4	6059.8	7	288	32	2	6	444.2	444.1	0.0	31.5	413.5	14	655	45	2	6	6598.5	6598.0	0.0	22.1	6455.3	11	753	30	2	6
	20	1	7652.9	7652.6	0.0	23.2	7470.0	0	246	39	4	11	547.9	547.9	0.0	22.4	506.9	0	567	30	4	11	8201.7	8201.7	0.0	22.5	8001.3	0	731	33	4	11
		2	8773.9	8773.1	0.0	349.4	8389.1	1	612	24	4	10	628.6	628.5	0.0	214.4	551.2	0	1162	22	4	10	9407.2	9406.3	0.0	154.1	8973.3	1	1163	35	4	10
		3	8113.2	8113.2	0.0	11.5	7990.0	0	478	42	4	10	583.9	583.8	0.0	38.9	562.8	1	1165	50	4	10	8698.5	8698.1	0.0	12.6	8604.2	0	283	48	4	10
		4	6177.6	6177.0	0.0	636.0	5864.3	1	939	38	3	9	430.6	430.5	0.0	470.3	381.8	0	1304	32	3	9	6608.2	6607.7	0.0	307.5	6346.0	0	849	41	3	9
		5	6909.9	6909.3	0.0	915.4	6589.7	14	878	34	3	9	485.5	485.4	0.0	257.5	464.4	7	1864	40	3	9	7403.0	7402.3	0.0	941.0	7054.8	6	675	30	3	9
	25	1	9807.5	9806.6	0.0	642.7	9631.9	0	762	31	3	12	703.8	703.8	0.0	791.1	660.2	2	1463	48	4	12	10512.2	10511.1	0.0	989.1	10277.1	1	1808	35	4	12
		2	7810.7	7809.9	0.0	11970.3	7505.2	10	1361	35	3	10	558.3	558.3	0.0	11311.1	468.1	15	2533	46	3	10	8372.8	8372.0	0.0	8181.0	8039.9	11	1950	48	3	10
		3	9228.6	9227.7	0.0	423.5	9084.6	1	784	28	4	10	658.2	658.2	0.0	459.7	574.6	3	1687	27	4	10	9888.3	9887.3	0.0	837.8	9731.1	3	1117	32	4	10
		4	9411.5	9410.5	0.0	2303.5	8975.4	2	780	45	3	13	674.2	674.2	0.0	1246.0	600.1	1	1034	29	4	13	10086.9	10085.9	0.0	899.9	9576.4	3	1418	56	4	13
		5	10383.3	10016.5	3.5	14400.0	9704.5	1	1291	47	4	12	734.5	719.2	2.1	14400.0	646.7	3	1604	36	3	11	11123.4	10673.9	4.0	14400.0	10366.7	0	1491	43	4	12
	30	1	9014.0	9013.1	0.0	4096.4	8618.5	0	1071	37	5	13	653.8	635.2	2.9	14400.0	576.1	0	3209	41	5	13	9669.6	9668.7	0.0	4740.8	9140.5	0	2514	41	5	13
		2	11779.5	11758.9	0.2	14400.5	11479.2	0	1299	44	4	14	856.1	845.6	1.2	14400.0	723.2	3	2242	24	5	14	12635.6	12554.2	0.6	14400.0	12216.2	1	2799	37	4	14
		3	11199.9	11114.0	0.8	14400.0	10742.4	1	631	42	5	14	804.5	804.4	0.0	8639.3	747.0	0	1638	43	5	14	12005.2	11882.8	1.0	14400.0	11375.2	0	1061	29	5	14
		4	11606.9	11584.3	0.2	14400.1	11258.2	0	1360	38	5	13	831.9	831.9	0.0	9215.6	740.0	0	1860	36	5	13	12444.3	12414.6	0.2	14400.0	11934.3	0	1680	37	5	13
		5	11249.0	11077.3	1.5	14400.0	10710.9	5	1415	35	5	13	799.6	782.0	2.2	14400.0	705.5	11	1884	34	5	13	12022.3	11798.3	1.9	14400.0	11406.5	5	2763	35	5	13
Set A_2	10	1	4475.2	4475.2	0.0	0.1	4464.8	1	82	6	2	5	307.2	307.2	0.0	0.2	306.8	1	80	7	2	5	4782.4	4782.4	0.0	0.1	4780.3	2	78	9	2	5
		2	5447.2	5447.2	0.0	0.4	5438.8	1	54	20	2	6	375.1	375.1	0.0	0.4	372.0	1	162	31	2	6	5822.3	5822.3	0.0	0.2	5810.7	2	96	18	2	6
		3	4946.7	4946.7	0.0	3.9	4582.0	1	208	23	2	5	318.7	318.7	0.0	2.1	290.8	1	151	23	2	5	5265.4	5265.4	0.0	2.9	4866.4	1	195	14	2	5
		4	4978.5	4978.5	0.0	2.4	4710.4	0	97	21	2	4	341.8	341.8	0.0	1.7	292.5	0	142	19	2	4	5321.6	5321.6	0.0	1.4	5137.7	0	206	24	2	4
		5	4845.3	4845.3	0.0	0.6	4654.7	1	67	28	2	5	326.5	326.5	0.0	0.4	313.4	1	136	28	2	5	5171.7	5171.7	0.0	0.4	5024.2	1	140	25	2	5
	15	1	6023.2	6023.2	0.0	2.1	5828.4	0	88	23	3	8	414.0	414.0	0.0	2.4	395.7	0	253	27	3	8	6437.2	6437.2	0.0	2.7	6327.2	0	253	48	3	8
		2	7368.8	7368.8	0.0	22.3	6910.5	1	287	31	2	7	499.9	499.9	0.0	8.0	457.3	2	302	26	2	8	7870.8	7870.5	0.0	21.4	7477.0	1	422	45	3	7
		3	6042.0	6042.0	0.0	1.3	5967.4	0	80	62	3	9	412.6	412.6	0.0	1.1	412.5	0	198	33	3	9	6454.6	6454.6	0.0	1.1	6363.8	0	266	44	3	9
		4	7111.2	7111.2	0.0	3.9	6863.5	0	32	25	2	7	479.6	479.6	0.0	4.4	450.6	0	311	37	2	7	7590.8	7590.8	0.0	7.3	7261.0	0	154	28	2	7
		5	8114.1	8113.6	0.0	14.7	7525.6	2	213	40	3	9	543.3	543.2	0.0	6.3	508.9	2	371	36	3	10	8673.0	8673.0	0.0	15.7	7996.7	2	585	25	3	9
	20	1	10078.9	10078.9	0.0	54.7	9573.2	1	224	30	4	12	703.5	703.5	0.0	8.2	684.0	0	296	68	4	13	10786.6	10785.5	0.0	31.7	10386.7	2	552	34	4	12
		2	8693.6	8693.6	0.0	104.0	8233.3	1	160	24	3	10	601.2	601.2	0.0	46.3	518.3	0	421	27	3	10	9294.8	9294.1	0.0	112.1	8771.2	4	514	23	3	10
		3	9334.0	9333.2	0.0	141.9	9018.6	7	362	33	3	10	644.5	644.5	0.0	113.0	619.6	5	1146	55	3	10	9978.5	9977.7	0.0	68.7	9633.2	5	559	34	3	10
		4	8858.5	8857.8	0.0	160.4	8500.1	1	471	36	4	10	610.9	610.9	0.0	15.8	588.1	0	436	37	4	11	9485.4	9484.7	0.0	120.5	9058.6	2	907	28	4	10
		5	7249.3	7249.3	0.0	26.0	7106.9	1	123	35	3	9	515.3	515.3	0.0	52.7	464.9	1	917	39	3	10	7764.8	7764.8	0.0	34.2	7560.0	0	325	44	3	9
	25	1	10219.2	10218.2	0.0	84.7	9801.4	0	257	31	4	12	717.8	717.8	0.0	59.0	628.8	5	907	39	4	12	10937.0	10937.0	0.0	49.0	10592.1	1	877	47	4	12
		2	11416.2	11415.1	0.0	4947.9	10767.5	4	461	40	4	13	785.8	785.8	0.0	375.0	697.5	5	734	35	4	13	12202.1	12200.9	0.0	4017.2	11504.2	3	819	46	4	13
		3	9677.8	9677.8	0.0	62.0	9364.0	2	544	35	4	12	675.2	675.2	0.0	26.2	643.4	2	781	50	3	12	10353.5	10352.9	0.0	81.2	10021.5	3	827	33	3	12
		4	11468.4	11467.2	0.0	766.6	10935.8	2	444	43	4	13	804.4	804.4	0.0	107.8	762.5	2	1020	39	4	14	12279.3	12278.1	0.0	810.7	11717.7	2	774	34	4	13
		5	10545.5	10544.7	0.0	54.4	10322.4	1	388	44	4	13	750.4	750.4	0.0	28.2	717.2	0	819	41	4	13	11296.8	11296.3	0.0	74.9	11007.6	1	863	36	4	13
	30	1	13944.2	13418.9	3.8	14400.0	12783.9	2	1237	36	5	16	971.6	971.5	0.0	5019.4	851.3	2	1746	33	5	17	14930.1	14535.8	2.6	14400.0	13634.9	3	1732	35	5	16
		2	11459.8	11458.7	0.0	1507.8	10988.3	1	920	44	4	13	812.6	812.5	0.0	1709.8	751.6	1	1714	40	3	13	12272.4	12271.2	0.0	3026.4	11717.8	0	1236	49	4	13
		3	11088.6	10992.3	0.9	14400.0	10368.7	5	1172	36	5	14	781.7	767.6	1.8	14400.0	661.8	6	2279	33	5	14	11871.0	11869.9	0.0	11014.3	11066.1	4	2012	39	5	14
		4	10717.3	10716.3	0.0	13811.0	10147.1	4	1364	30	4	12	747.8	741.8	0.8	14400.0	617.5	11	2168	29	4	12	11468.2	11467.1	0.0	6351.6	10632.9	2	1485	28	4	12
		5	11527.6	11526.7	0.0	127.8	11299.8	4	594	32	4	12	826.1	826.0	0.0	4316.1	777.8	0	1696	44	4	12	12356.8	12355.7	0.0	151.6	11933.6	2	1089	29	4	12

Table D.6 Results for extended Solomon's instances of type 2

		R_e										E_e										$(R+E)_e$									
Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
25	c201	603.1	603.1	0.0	1.4	599.1	0	309	26	4	10	44.4	44.4	0.0	11.5	39.9	0	467	41	4	10	647.5	647.5	0.0	2.5	641.2	0	467	27	4	10
	c202	603.1	603.1	0.0	13.4	595.9	0	232	42	4	10	43.8	43.8	0.0	48.5	38.1	0	1242	47	4	10	646.9	646.9	0.0	19.7	634.9	0	865	44	4	10
	c203	603.1	603.1	0.0	56.4	593.9	0	430	48	4	10	43.5	43.5	0.0	86.5	37.7	0	1294	73	4	10	646.6	646.6	0.0	74.8	633.0	0	1470	45	4	10
	c204	603.1	603.1	0.0	48.2	592.1	0	451	30	4	10	43.4	43.4	0.0	240.8	37.4	0	1099	73	4	10	646.5	646.5	0.0	78.6	631.0	0	880	52	4	10
	c205	603.1	603.1	0.0	13.0	592.2	0	678	35	4	10	43.9	43.9	0.0	29.5	39.4	0	716	46	4	10	647.3	647.3	0.0	9.0	637.0	0	446	30	4	10
	c206	603.1	603.1	0.0	21.1	594.6	0	296	42	3	10	43.6	43.6	0.0	40.8	38.1	0	1383	32	4	10	646.7	646.7	0.0	21.7	634.5	0	504	35	4	10
	c207	603.1	603.1	0.0	32.0	594.1	0	593	38	4	10	43.3	43.3	0.0	55.3	37.5	0	1127	53	4	10	646.4	646.4	0.0	43.4	631.7	0	1091	41	4	10
	c208	603.1	603.1	0.0	23.2	593.5	0	523	37	4	10	43.4	43.4	0.0	34.7	37.8	0	1374	34	4	10	646.5	646.5	0.0	34.3	635.3	0	780	47	4	10
	r201	761.7	761.7	0.0	4.0	749.2	0	362	29	4	12	54.2	54.2	0.0	8.7	51.7	0	729	50	4	12	816.2	816.2	0.0	5.6	794.2	0	226	30	4	12
	r202	755.7	755.7	0.0	32.9	737.9	0	118	42	4	12	53.0	53.0	0.0	26.9	48.1	0	414	47	4	12	809.0	809.0	0.0	29.2	786.2	0	873	38	4	12
	r203	749.5	749.5	0.0	132.1	733.2	0	579	47	4	12	52.9	52.9	0.0	40.2	47.3	0	811	40	4	12	802.9	802.9	0.0	58.0	778.0	0	876	39	4	12
	r204	749.5	749.5	0.0	134.2	732.5	0	671	49	4	12	52.9	52.9	0.0	125.4	47.3	0	782	38	4	12	802.7	802.7	0.0	120.3	779.6	0	875	44	4	12
	r205	755.7	755.7	0.0	38.5	738.4	0	211	36	4	12	53.3	53.3	0.0	21.8	50.6	0	889	45	4	12	809.0	809.0	0.0	27.1	788.6	0	678	44	4	12
	r206	755.7	755.7	0.0	54.8	736.7	0	775	44	4	12	52.9	52.9	0.0	29.8	49.4	0	839	72	4	12	809.0	809.0	0.0	84.6	786.0	0	891	52	4	12
	r207	749.5	749.5	0.0	83.5	731.3	0	662	48	4	12	52.9	52.9	0.0	37.4	49.6	0	800	46	4	12	802.9	802.9	0.0	87.8	783.8	0	1298	56	4	12
	r208	749.5	749.5	0.0	75.7	731.0	0	358	42	4	12	52.9	52.9	0.0	47.9	49.4	0	689	37	4	12	802.7	802.7	0.0	106.7	780.7	0	1459	39	4	12
	r209	749.5	749.5	0.0	42.8	731.6	0	481	41	4	12	53.2	53.2	0.0	41.5	48.4	0	761	42	4	12	803.2	803.2	0.0	50.0	779.4	0	584	43	4	12
	r210	755.7	755.7	0.0	46.0	737.1	0	337	38	4	12	52.9	52.9	0.0	24.8	49.2	0	567	49	4	12	809.0	809.0	0.0	60.2	784.8	0	415	45	4	12
	r211	749.5	749.5	0.0	136.1	729.0	0	316	47	4	12	52.9	52.9	0.0	53.4	48.8	0	1154	59	4	12	802.7	802.7	0.0	102.5	777.8	0	874	37	4	12
	rc201	956.3	956.3	0.0	28.0	930.6	0	744	56	4	12	68.7	68.7	0.0	31.7	58.2	0	666	34	4	12	1025.4	1025.4	0.0	60.0	954.3	0	1074	30	4	12
	rc202	940.6	940.5	0.0	366.9	912.4	1	712	50	4	11	67.9	67.9	0.0	315.0	57.1	0	1517	51	4	12	1010.2	1010.2	0.0	125.1	969.3	1	1467	25	4	11
	rc203	934.4	934.4	0.0	15.7	931.5	0	266	26	4	11	67.4	67.4	0.0	29.9	60.3	0	780	40	4	12	1003.2	1003.1	0.0	56.8	974.0	1	697	27	4	11
	rc204	934.4	934.4	0.0	5.4	931.8	0	338	47	4	11	67.2	67.2	0.0	64.3	63.4	0	668	37	4	12	1003.2	1003.2	0.0	952.8	981.6	0	1144	59	4	11
	rc205	946.7	946.7	0.0	63.4	933.6	0	881	24	4	11	68.3	68.3	0.0	269.0	57.8	0	1446	49	4	12	1016.6	1016.6	0.0	100.6	993.3	1	363	45	4	11
	rc206	948.5	948.5	0.0	65.4	935.6	0	408	27	4	12	68.2	68.2	0.0	59.5	62.0	1	1708	53	4	12	1017.0	1016.9	0.0	755.0	977.8	0	1478	45	4	12
	rc207	940.7	940.6	0.0	1253.2	924.3	0	1053	36	4	12	67.6	67.6	0.0	58.3	59.6	1	816	41	4	12	1008.4	1008.3	0.0	7306.9	967.8	1	2029	29	4	12
	rc208	934.4	934.4	0.0	207.4	922.8	0	659	55	4	11	67.2	67.2	0.0	23.0	64.4	0	746	40	4	12	1002.9	1002.9	0.0	157.5	981.4	0	1101	33	4	11
40	c201	1032.8	1032.8	0.0	48.3	1017.2	0	1368	25	5	15	75.5	75.5	0.0	229.9	62.4	0	2236	29	5	15	1108.3	1108.3	0.0	43.8	1090.1	0	1181	43	5	15
	c202	1031.8	1031.8	0.0	372.9	1013.2	0	1274	47	5	15	74.9	74.9	0.0	14073.8	61.8	0	2747	51	5	15	1106.8	1106.7	0.0	955.4	1076.8	0	1840	42	5	15
	c203	1031.1	1031.0	0.0	3881.6	1011.9	0	1968	60	5	15	74.7	74.0	1.0	43200.0	63.4	0	4554	55	5	15	1106.2	1106.1	0.0	10685.1	1072.4	0	3031	49	5	15
	c204	1031.1	1029.1	0.2	43200.0	1009.5	0	2076	55	5	15	74.5	73.3	1.5	43200.0	60.8	0	3838	61	5	15	1105.9	1105.8	0.0	16940.6	1075.0	0	2173	62	5	15
	c205	1031.1	1031.0	0.0	871.2	1011.7	0	1830	40	5	15	75.1	75.1	0.0	23630.3	60.9	0	2617	47	5	15	1106.4	1106.4	0.0	881.7	1073.7	0	1633	38	5	15
	c206	1031.1	1031.0	0.0	2011.9	1009.1	0	1726	33	5	15	74.8	73.4	1.8	43200.0	60.9	0	3321	49	5	15	1106.0	1105.9	0.0	8368.8	1074.1	0	2944	41	5	15
	c207	1031.1	1031.0	0.0	5757.3	1010.0	0	2628	51	5	15	74.4	74.4	0.0	5567.9	60.6	0	3439	52	5	15	1105.9	1105.8	0.0	8358.9	1074.9	0	2271	64	5	15
	c208	1031.1	1031.0	0.0	1936.7	1012.2	0	1990	49	5	15	74.5	74.0	0.6	43200.0	60.7	3	2665	53	5	15	1105.9	1105.8	0.0	3547.3	1071.5	0	1443	46	5	15
	r201	1204.8	1204.8	0.0	437.7	1160.3	1	1111	37	6	19	85.8	85.8	0.0	2730.5	74.4	0	3568	32	6	20	1291.6	1291.6	0.0	348.1	1233.7	0	1687	37	6	19
	r202	1169.9	1169.9	0.0	11666.6	1121.2	4	1566	53	6	19	83.2	82.2	1.2	43200.0	71.1	9	5734	40	6	19	1253.2	1253.2	0.0	8183.1	1188.0	0	2305	38	6	19
	r203	1155.4	1155.4	0.0	33302.4	1110.4	0	774	42	6	19	81.5	81.5	0.0	17013.0	71.9	0	1928	69	6	19	1237.0	1236.9	0.0	40920.9	1180.4	0	1563	53	6	19
	r204	1155.4	1148.7	0.6	43200.1	1108.9	1	291	45	6	19	81.1	81.1	0.0	37471.4	71.8	0	1914	57	6	19	1236.6	1221.5	1.2	43200.0	1177.2	0	1275	51	6	19
	r205	1163.5	1163.5	0.0	5921.7	1122.1	2	1024	45	6	19	82.4	82.3	0.0	7086.4	72.1	0	3252	44	6	19	1245.9	1245.9	0.0	8098.5	1196.3	0	3244	44	6	19
	r206	1161.0	1149.9	1.0	43200.0	1111.3	0	583	47	6	19	81.7	81.7	0.0	5167.2	74.9	0	2301	72	6	19	1242.8	1237.6	0.4	43200.0	1181.1	2	3483	55	6	19
	r207	1155.4	1141.9	1.2	43200.0	1107.0	0	797	50	6	19	81.5	81.5	0.0	22529.7	70.0	0	2488	42	6	19	1237.0	1236.9	0.0	39388.0	1178.9	0	771	54	6	19
	r208	1155.4	1144.3	1.0	43200.0	1103.9	0	376	41	6	19	81.1	80.9	0.4	43200.0	71.4	0	3191	50	6	19	1236.6	1228.4	0.7	43200.0	1182.9	0	1810	64	6	19
	r209	1155.4	1155.4	0.0	42044.0	1112.2	0	1497	37	6	19	82.1	77.3	5.9	43200.0	71.0	8	3092	46	6	19	1237.8	1220.5	1.4	43200.0	1183.4	2	1784	49	6	19
	r210	1161.0	1161.0	0.0	14821.3	1113.6	0	1302	48	6	19	81.7	81.7	0.0	2832.5	72.0	0	1623	58	6	19	1242.8	1234.4	0.7	43200.0	1179.7	0	2038	46	6	19
	r211	1155.4	1138.8	1.4	43200.0	1106.6	0	502	46	6	19	81.1	80.6	0.6	43200.0	71.4	0	3347	59	6	19	1236.6	1222.2	1.2	43200.0	1180.5	0	3028	54	6	19
	rc201	1582.9	1582.9	0.0	959.5	1529.2	0	1092	48	6	18	114.1	105.6	7.4	43200.0	94.2	2	1223	63	6	18	1697.1	1697.0	0.0	2540.3	1604.4	3	2397	42	6	18
	rc202	1533.9	1521.8	0.8	43200.0	1503.6	3	984	49	6	17	111.3	108.2	2.9	43200.0	93.6	1	3139	74	6	18	1647.2	1637.0	0.6	43200.0	1593.9	0	2844	45	6	17
	rc203	1527.7	1527.6	0.0	1542.6	1497.9	0	954	33	6	17	110.7	110.7	0.0	509.2	90.1	0	1170	42	6	18	1640.1	1640.0	0.0	5064.3	1595.5	2	2609	43	6	17
	rc204	1527.7	1527.7	0.0	110.3	1517.8	0	381	38	6	17	110.6	110.6	0.0	2663.9	96.9	0	2678	39	6	18	1640.1	1640.0	0.0	9900.8	1600.5	0	1752	39	6	17
	rc205	1543.5	1543.4	0.0	27642.5	1490.9	0	1592	38	6	18	111.9	96.7	13.6	43200.0	91.4	4	2668	40	6	18	1655.4	1629.2	1.6	43200.0	1600.7	7	2087	61	6	18
	rc206	1551.2	1551.1	0.0	4719.6	1527.2	1	2226	53	6	18	112.4	100.7	10.4	43200.0	93.8	0	2676	44	6	18	1663.8	1663.7	0.0	39338.2	1602.9	2	2373	41	6	18
	rc207	1533.6	1527.7	0.4	43200.0	1506.4	0	1472	41	6	17	110.8	109.6	1.1	43200.0	90.1	0	3426	48	6	18	1643.9	1628.9	0.9	43200.0	1598.9	0	3363	49	6	17
	rc208	1527.7	1527.7	0.0	1684.8	1498.3	0	803	46	6	17	110.6	110.6	0.0	237.6	97.0	0	1044	42	6	18	1639.8	1639.7	0.0	7961.0	1598.1	0	2476	40	6	17

Table D.7 Results using multicore processors for Set A instances with 35–50 customers

			R_e										E_e										$(R+E)_e$									
	Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	35	1	11719.2	11237.2	4.1	43200.0	10962.9	12	5471	54	5	13	846.2	807.7	4.6	43200.0	691.9	4	2734	39	5	13	12510.2	12104.5	3.2	43200.0	11693.5	0	3485	47	5	13
		2	12145.5	11927.6	1.8	43200.0	11495.5	9	3031	54	5	15	880.0	848.5	3.6	43200.0	735.6	4	3748	38	5	15	13025.5	12663.3	2.8	43200.0	12218.1	5	1695	50	5	15
		3	12346.4	12345.2	0.0	20642.0	11955.2	11	2866	37	5	15	889.5	889.4	0.0	21866.9	762.9	19	3273	37	5	15	13236.1	13106.4	1.0	43200.0	12706.7	10	2637	44	5	15
		4	12925.3	12924.1	0.0	30214.8	12528.6	5	3409	37	5	15	941.7	941.6	0.0	36895.7	808.6	20	4282	30	5	15	13868.2	13838.5	0.2	43200.0	13305.6	10	4865	40	5	15
		5	12384.3	12383.3	0.0	29126.0	11900.5	0	1144	48	6	18	894.5	894.4	0.0	15121.7	752.6	1	2687	36	6	18	13286.7	13285.7	0.0	20446.6	12709.6	1	1495	43	6	18
	40	1	14546.6	14055.3	3.4	43200.0	13792.5	4	1916	58	6	18	1050.1	1019.4	2.9	43200.0	899.9	6	5416	40	6	18	15546.0	14931.9	4.0	43200.0	14608.2	12	5103	53	6	18
		2	15741.8	15740.9	0.0	13947.5	15181.1	1	1370	46	7	20	1144.1	1133.8	0.9	43200.0	1036.8	3	3824	42	7	20	16887.0	16787.9	0.6	43200.1	16156.1	0	3204	55	7	20
		3	13230.2	12735.2	3.7	43200.1	12415.0	7	3905	52	6	17	960.9	915.4	4.7	43200.0	759.8	6	4500	63	6	18	14183.7	13648.1	3.8	43200.0	13238.9	3	4583	59	6	17
		4	15189.5	15122.6	0.4	43200.0	14560.9	16	3179	47	7	20	1099.0	1098.9	0.0	23208.1	984.3	14	5721	43	7	20	16292.1	16229.3	0.4	43200.0	15563.0	2	4024	64	7	20
		5	14554.9	14448.5	0.7	43200.0	14075.2	0	2293	46	6	19	1056.1	1042.4	1.3	43200.0	996.1	3	4015	59	6	19	15625.8	15264.1	2.3	43200.0	14965.9	3	4185	59	6	19
	45	1	13832.5	13245.7	4.2	43200.1	12999.7	3	3065	64	6	18	991.7	955.6	3.6	43200.0	840.5	14	8605	37	6	19	14760.2	14175.2	4.0	43200.0	13817.3	4	5690	54	6	18
		2	18423.5	18026.8	2.2	43200.0	17692.1	4	1753	67	7	22	1343.0	1307.2	2.7	43200.0	1203.8	8	7610	36	7	22	19747.2	19340.7	2.1	43200.0	18942.6	3	7054	67	7	22
		3	17548.7	17283.6	1.5	43200.0	16809.4	1	1768	50	7	21	1284.5	1236.2	3.8	43200.0	1053.9	1	8810	50	7	22	18899.1	18434.2	2.5	43200.0	17974.2	1	4837	54	7	22
		4	15231.5	14767.5	3.0	43200.0	14429.8	10	3801	64	7	21	1104.8	1055.9	4.4	43200.0	928.9	29	8422	52	7	21	16341.1	15826.7	3.1	43200.0	15398.8	8	5617	49	7	21
		5	17467.0	17126.7	1.9	43200.0	16781.1	12	2274	65	8	23	1254.4	1232.0	1.8	43200.0	1116.2	31	6849	42	8	23	18736.6	18307.2	2.3	43200.0	17850.3	18	4315	56	8	23
Set A_2	35	1	13338.1	13000.9	2.5	43200.0	12340.2	11	1607	42	5	14	941.9	941.8	0.0	31873.4	804.1	29	4738	30	5	15	14287.9	13909.9	2.6	43200.0	13108.8	18	2912	38	5	14
		2	16285.8	16285.7	0.0	1755.4	15114.2	5	759	42	6	20	1158.5	1158.4	0.0	18306.0	986.1	5	2471	47	6	20	17450.5	17449.3	0.0	3397.5	16138.0	15	3099	32	6	20
		3	13709.3	13707.9	0.0	8732.7	13091.3	4	1814	45	5	14	982.4	954.6	2.8	43200.0	822.4	6	2328	33	5	14	14696.5	14695.1	0.0	11645.0	13828.6	9	3204	36	5	14
		4	16511.4	16510.1	0.0	9765.1	15590.8	9	1233	45	6	17	1171.3	1171.3	0.0	38648.4	997.3	13	2774	47	6	18	17694.9	17693.2	0.0	25076.3	16599.0	6	1778	37	6	17
		5	15706.0	15704.8	0.0	9491.5	14796.8	8	1676	48	5	16	1108.8	1108.7	0.0	42292.8	946.2	6	2742	34	5	16	16817.4	16815.8	0.0	18041.8	15740.4	9	2594	43	5	16
	40	1	15866.1	15865.0	0.0	32162.3	15050.8	2	813	47	6	19	1131.8	1131.7	0.0	6219.5	982.1	7	1930	35	6	20	17008.8	17007.3	0.0	21628.1	16050.1	10	3197	49	6	19
		2	16751.9	16563.7	1.1	43200.0	15858.9	4	1167	36	6	18	1189.6	1172.6	1.4	43200.0	1010.3	3	3376	37	6	18	17954.7	17952.9	0.0	41897.4	16900.3	12	2304	50	6	18
		3	16867.9	16866.9	0.0	3298.6	16156.5	9	1854	50	7	20	1202.4	1202.3	0.0	30554.3	1023.7	2	1955	47	7	21	18085.0	18083.4	0.0	5495.7	17039.1	6	3058	50	7	20
		4	17327.7	16959.0	2.1	43200.0	16506.2	14	1771	52	6	19	1233.5	1233.4	0.0	7308.4	1111.1	10	3392	25	6	20	18565.5	18164.3	2.2	43200.0	17526.7	6	5158	45	6	20
		5	12989.9	12333.8	5.1	43200.0	11903.8	19	3308	41	5	15	920.0	861.2	6.4	43200.0	763.2	17	3425	42	5	17	13886.7	13218.2	4.8	43200.0	12651.4	24	4235	39	5	16
	45	1	17395.5	17393.9	0.0	6142.2	16659.3	9	2285	50	8	23	1242.9	1242.9	0.0	1802.5	1076.3	1	2044	50	8	24	18662.0	18660.4	0.0	8093.6	17684.0	6	2824	41	8	23
		2	18299.8	18298.0	0.0	41018.0	17418.5	17	1752	66	7	23	1288.6	1271.8	1.3	43200.0	1085.5	14	4237	36	7	23	19596.7	19337.3	1.3	43200.0	18204.9	14	2576	45	7	22
		3	18854.9	18627.9	1.2	43200.0	18056.7	8	1878	44	8	23	1355.5	1326.3	2.2	43200.0	1169.6	5	3513	38	8	23	20214.0	20212.5	0.0	7452.0	19205.3	2	2111	38	8	23
		4	16156.7	15830.2	2.0	43200.0	15187.7	7	3575	38	6	19	1142.6	1121.5	1.8	43200.0	979.6	6	4761	31	6	19	17322.9	16859.6	2.7	43200.0	16154.7	7	2342	38	6	19
		5	22695.8	22693.7	0.0	37956.5	21613.5	5	1321	57	8	26	1617.5	1583.3	2.1	43200.0	1403.6	3	4321	50	8	26	24318.9	24075.2	1.0	43200.0	22998.6	3	3652	46	8	26
	50	1	23045.5	22630.7	1.8	43200.0	22117.4	17	3932	43	8	26	1649.2	1637.1	0.7	43200.0	1513.3	6	5118	43	8	26	24723.8	24141.0	2.4	43200.0	23646.8	5	3907	51	8	26
		2	20734.0	19945.0	3.8	43200.0	19248.1	16	3064	39	8	24	1461.9	1429.1	2.2	43200.0	1271.5	26	5673	30	8	25	21947.0	21112.5	3.8	43200.0	20476.4	14	4361	43	8	24
		3	18482.6	17942.6	2.9	43200.0	17595.5	3	3278	47	7	20	1304.5	1249.7	4.2	43200.0	1149.9	6	4313	36	7	21	19707.0	19170.5	2.7	43200.0	18708.9	3	4128	40	7	21
		4	18537.6	18293.1	1.3	43200.0	17778.5	5	1269	44	7	23	1309.1	1266.1	3.3	43200.1	1128.8	5	2372	52	7	24	19849.8	19707.0	0.7	43200.0	19078.5	2	4748	46	7	23
		5	22160.8	21761.2	1.8	43200.0	20922.1	3	1607	43	8	25	1578.2	1553.6	1.6	43200.0	1454.6	8	3312	35	8	26	23729.1	23252.5	2.0	43200.0	22417.4	0	1800	46	8	25

APPENDIX E SUPPLEMENT TO CHAPTER 7

We provide the proofs of Theorem 7.4.1 and Theorem 7.4.2 here.

E.1 Proof of Theorem 7.4.1

Because of the feasibility and the linearity of the ambiguity set (see, for instance, [116, 159,160]), strong duality holds, thus we can reformulate Problem (7.17) as the following minimization problem, with $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\epsilon}, \eta$ as the dual variables associated with the expectations and probability in $\mathcal{G}_k^1$:

$$
\begin{array}{ll}
\min & (\boldsymbol{\mu}_k^1)'\boldsymbol{\alpha} + (\boldsymbol{\nu}_k^1)'\boldsymbol{\beta} + (\boldsymbol{\sigma}_k^1)'\boldsymbol{\delta} + (\boldsymbol{\varsigma}_k^1)'\boldsymbol{\epsilon} + \eta \\
\text{s.t.} & (\boldsymbol{u}^1)'\boldsymbol{\alpha} + (\boldsymbol{v}^1)'\boldsymbol{\beta} + (\boldsymbol{a}^1)'\boldsymbol{\delta} + (\boldsymbol{b}^1)'\boldsymbol{\epsilon} + \eta \\
& \quad \geq \sum\limits_{i \in \bar{\mathcal{C}}^1} (x_{id}^1(u_i^1 + v_i^1) + y_{id}^1 u_i^1) - \bar{\tau}^1 \qquad \forall (\boldsymbol{u}^1, \boldsymbol{a}^1, \boldsymbol{v}^1, \boldsymbol{b}^1) \in \Xi_k^1, \\
& (\boldsymbol{u}^1)'\boldsymbol{\alpha} + (\boldsymbol{v}^1)'\boldsymbol{\beta} + (\boldsymbol{a}^1)'\boldsymbol{\delta} + (\boldsymbol{b}^1)'\boldsymbol{\epsilon} + \eta \geq -\gamma_d^1 \quad \forall (\boldsymbol{u}^1, \boldsymbol{a}^1, \boldsymbol{v}^1, \boldsymbol{b}^1) \in \Xi_k^1, \\
& \boldsymbol{\delta}, \boldsymbol{\epsilon} \geq \mathbf{0}, \\
& \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\epsilon}, \in \mathbb{R}^N, \eta \in \mathbb{R}.
\end{array}
\tag{E.1}
$$

Using the duality result of RO, i.e., the dual of the robust counterpart ('primal worst') is equal to the optimistic counterpart of the dual problem ('dual best') [161], Problem (E.1) is equivalent to the following maximization problem:

$$
\begin{array}{lll}
\max\limits_{(\boldsymbol{u}_j^1, \boldsymbol{a}_j^1, \boldsymbol{v}_j^1, \boldsymbol{b}_j^1) \in \Xi_k^1 \forall j \in [2]} & \min & (\boldsymbol{\mu}_k^1)'\boldsymbol{\alpha} + (\boldsymbol{\nu}_k^1)'\boldsymbol{\beta} + (\boldsymbol{\sigma}_k^1)'\boldsymbol{\delta} + (\boldsymbol{\varsigma}_k^1)'\boldsymbol{\epsilon} + \eta \\
& \text{s.t.} & (\boldsymbol{u}_1^1)'\boldsymbol{\alpha} + (\boldsymbol{v}_1^1)'\boldsymbol{\beta} + (\boldsymbol{a}_1^1)'\boldsymbol{\delta} + (\boldsymbol{b}_1^1)'\boldsymbol{\epsilon} + \eta \\
& & \quad \geq \left(\sum\limits_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)\boldsymbol{e}_i \right)' \boldsymbol{u}_1^1 + \left(\sum\limits_{i \in \bar{\mathcal{C}}^1} x_{id}^1 \boldsymbol{e}_i \right)' \boldsymbol{v}_1^1 - \bar{\tau}^1 \\
& & (\boldsymbol{u}_2^1)'\boldsymbol{\alpha} + (\boldsymbol{v}_2^1)'\boldsymbol{\beta} + (\boldsymbol{a}_2^1)'\boldsymbol{\delta} + (\boldsymbol{b}_2^1)'\boldsymbol{\epsilon} + \eta \geq -\gamma_d^1 \\
& & \boldsymbol{\delta}, \boldsymbol{\epsilon} \geq \mathbf{0}, \\
& & \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\epsilon}, \in \mathbb{R}^N, \eta \in \mathbb{R}.
\end{array}
$$

By duality of the inner linear optimization problem we have, equivalently,

$$
\max \left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)\boldsymbol{e}_i \right)' \boldsymbol{u}_1^1 p_1 + \left(\sum_{i \in \bar{\mathcal{C}}^1} x_{id}^1 \boldsymbol{e}_i \right)' \boldsymbol{v}_1^1 p_1 - \bar{\tau}^1 p_1 - \gamma_d^1 p_2
$$

$$
\begin{aligned}
\text{s.t. } & \boldsymbol{u}_1^1 p_1 + \boldsymbol{u}_2^1 p_2 = \boldsymbol{\mu}_k^1 \\
& \boldsymbol{v}_1^1 p_1 + \boldsymbol{v}_2^1 p_2 = \boldsymbol{\nu}_k^1 \\
& \boldsymbol{a}_1^1 p_1 + \boldsymbol{a}_2^1 p_2 \leq \boldsymbol{\sigma}_k^1 \\
& \boldsymbol{b}_1^1 p_1 + \boldsymbol{b}_2^1 p_2 \leq \boldsymbol{\varsigma}_k^1 \\
& p_1 + p_2 = 1 \\
& \underline{\boldsymbol{u}}_k^1 \leq \boldsymbol{u}_j^1 \leq \bar{\boldsymbol{u}}_k^1 && \forall j \in [2] \\
& \boldsymbol{a}_j^1 \geq |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1| && \forall j \in [2] \\
& \underline{\boldsymbol{v}}_k^1 \leq \boldsymbol{v}_j^1 \leq \bar{\boldsymbol{v}}_k^1 && \forall j \in [2] \\
& \boldsymbol{b}_j^1 \geq |\boldsymbol{v}_j^1 - \boldsymbol{\mu}_k^1| && \forall j \in [2] \\
& p_1, p_2 \geq 0 \\
& \boldsymbol{u}_j^1, \boldsymbol{a}_j^1, \boldsymbol{v}_j^1, \boldsymbol{v}_j^1 \in \mathbb{R}^N && \forall j \in [2].
\end{aligned}
$$

By perspective transformation, i.e., $\boldsymbol{u}_1^1 p_1 \to \boldsymbol{u}_1^1$, $\boldsymbol{v}_1^1 p_1 \to \boldsymbol{v}_1^1$, we obtain the linear optimization model presented in Equation (7.18), noting that the equivalence holds because the feasiblity $\mathcal{Y}_{kd}^1$ requires $\boldsymbol{u}_j^1 = \boldsymbol{a}_j^1 = \boldsymbol{v}_j^1 = \boldsymbol{b}_j^1 = \mathbf{0}$ whenever $p_j = 0$.

E.2 Proof of Theorem 7.4.2

The proof is similar to the proof of Theorem 7.4.1. Problem (7.20) is equivalent to the following minimization problem:

$$
\begin{aligned}
\min\ & (\boldsymbol{\mu}_k^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{\nu}_k^1)'\boldsymbol{\beta}^1 + (\boldsymbol{\sigma}_k^1)'\boldsymbol{\delta}^1 + (\boldsymbol{\varsigma}_k^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{\mu}_{sg}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{\nu}_{sg}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{\sigma}_{sg}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{\varsigma}_{sg}^2)'\boldsymbol{\epsilon}^2 + \eta \\
\text{s.t. } & (\boldsymbol{u}^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2)'\boldsymbol{\epsilon}^2 + \eta \\
& \geq \sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)(u_i^1 + v_i^1) + \sum_{i \in \bar{\mathcal{C}}^2} \Big((z_{id}^2(s) + x_{id}^2(s))(u_i^2 + v_i^2) + y_{id}^2(s) u_i^2 \Big) - \bar{\tau}^2 \\
& \qquad\qquad \forall (\boldsymbol{u}^1, \boldsymbol{v}^1, \boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{u}^2, \boldsymbol{v}^2, \boldsymbol{a}^2, \boldsymbol{b}^2) \in \Xi_{kgs}^2 \\
& (\boldsymbol{u}^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2)'\boldsymbol{\epsilon}^2 + \eta \\
& \geq \bar{\tau}^1 + \sum_{i \in \bar{\mathcal{C}}^2} \Big(z_{id}^2(s) v_i^2 + x_{id}^2(s)(u_i^2 + v_i^2) + y_{id}^2(s) u_i^2 \Big) - \bar{\tau}^2 \quad \forall (\boldsymbol{u}^1, \boldsymbol{v}^1, \boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{u}^2, \boldsymbol{v}^2, \boldsymbol{a}^2, \boldsymbol{b}^2) \in \Xi_{kgs}^2 \\
& (\boldsymbol{u}^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2)'\boldsymbol{\epsilon}^2 + \eta \geq -\gamma_d^2 \\
& \qquad\qquad \forall (\boldsymbol{u}^1, \boldsymbol{v}^1, \boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{u}^2, \boldsymbol{v}^2, \boldsymbol{a}^2, \boldsymbol{b}^2) \in \Xi_{kgs}^2 \\
& \boldsymbol{\delta}^n, \boldsymbol{\epsilon}^n \geq 0 \qquad\qquad \forall n \in [2] \\
& \boldsymbol{\alpha}^n, \boldsymbol{\beta}^n, \boldsymbol{\delta}^n, \boldsymbol{\epsilon}^n \in \mathbb{R}^N, \eta \in \mathbb{R} \qquad \forall n \in [2].
\end{aligned}
$$

Using the duality result of robust optimization we have, equivalently

$$\max_{(\boldsymbol{u}_j^1,\boldsymbol{v}_j^1,\boldsymbol{a}_j^1,\boldsymbol{b}_j^1,\boldsymbol{u}_j^2,\boldsymbol{v}_j^2,\boldsymbol{a}_j^2,\boldsymbol{b}_j^2)\in\Xi_{kgs}^2\,\forall j\in[3]} \min \Big((\boldsymbol{\mu}_k^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{\nu}_k^1)'\boldsymbol{\beta}^1 + (\boldsymbol{\sigma}_k^1)'\boldsymbol{\delta}^1 + (\boldsymbol{\varsigma}_k^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{\mu}_{sg}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{\nu}_{sg}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{\sigma}_{sg}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{\varsigma}_{sg}^2)'\boldsymbol{\epsilon}^2 + \eta\Big)$$

$$\text{s.t. } (\boldsymbol{u}_1^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}_1^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}_1^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}_1^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}_1^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}_1^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}_1^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}_1^2)'\boldsymbol{\epsilon}^2 + \eta$$
$$\geq \left(\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1 + y_{id}^1)\boldsymbol{e}_i\right)'(\boldsymbol{u}_1^1 + \boldsymbol{v}_1^1) + \left(\sum_{i\in\bar{\mathcal{C}}^2}(z_{id}^2(s) + x_{id}^2(s))\boldsymbol{e}_i\right)'(\boldsymbol{u}_1^2 + \boldsymbol{v}_1^2) + \left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{u}_1^2 - \bar{\tau}^2$$
$$(\boldsymbol{u}_2^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}_2^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}_2^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}_2^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}_2^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}_2^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}_2^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}_2^2)'\boldsymbol{\epsilon}^2 + \eta$$
$$\geq \bar{\tau}^1 + \left(\sum_{i\in\bar{\mathcal{C}}^2} z_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{v}_2^2 + \left(\sum_{i\in\bar{\mathcal{C}}^2} x_{id}^2(s)\boldsymbol{e}_i\right)'(\boldsymbol{u}_2^2 + \boldsymbol{v}_2^2) + \left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{u}_2^2 - \bar{\tau}^2$$
$$(\boldsymbol{u}_3^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}_3^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}_3^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}_3^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}_3^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}_3^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}_3^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}_3^2)'\boldsymbol{\epsilon}^2 + \eta \geq -\gamma_d^2$$
$$\boldsymbol{\delta}^n, \boldsymbol{\epsilon}^n \geq 0 \qquad \forall n \in [2]$$
$$\boldsymbol{\alpha}^n, \boldsymbol{\beta}^n, \boldsymbol{\delta}^n, \boldsymbol{\epsilon}^n \in \mathbb{R}^N, \eta \in \mathbb{R} \qquad \forall n \in [2].$$

By duality of the inner optimization problem we have, equivalently,

$$\max \left(\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1 + y_{id}^1)\boldsymbol{e}_i\right)'(\boldsymbol{u}_1^1 + \boldsymbol{v}_1^1)p_1 + \left(\sum_{i\in\bar{\mathcal{C}}^2}(z_{id}^2(s) + x_{id}^2(s))\boldsymbol{e}_i\right)'(\boldsymbol{u}_1^2 + \boldsymbol{v}_1^2)p_1$$
$$+ \left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{u}_1^2 p_1 - \bar{\tau}^2 p_1 + \bar{\tau}^1 p_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2} z_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{v}_2^2 p_2$$
$$+ \left(\sum_{i\in\bar{\mathcal{C}}^2} x_{id}^2(s)\boldsymbol{e}_i\right)'(\boldsymbol{u}_2^2 + \boldsymbol{v}_2^2)p_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{u}_2^2 p_2 - \bar{\tau}^2 p_2 - \gamma_d^2 p_3$$
$$\text{s.t. } \boldsymbol{u}_1^1 p_1 + \boldsymbol{u}_2^1 p_2 + \boldsymbol{u}_3^1 p_3 = \boldsymbol{\mu}_k^1$$
$$\boldsymbol{v}_1^1 p_1 + \boldsymbol{v}_2^1 p_2 + \boldsymbol{v}_3^1 p_3 = \boldsymbol{\nu}_k^1$$
$$\boldsymbol{a}_1^1 p_1 + \boldsymbol{a}_2^1 p_2 + \boldsymbol{a}_3^1 p_3 \leq \boldsymbol{\sigma}_k^1$$
$$\boldsymbol{b}_1^1 p_1 + \boldsymbol{b}_2^1 p_2 + \boldsymbol{b}_3^1 p_3 \leq \boldsymbol{\varsigma}_k^1$$
$$\boldsymbol{u}_1^2 p_1 + \boldsymbol{u}_2^2 p_2 + \boldsymbol{u}_3^2 p_3 = \boldsymbol{\mu}_{sg}^2$$
$$\boldsymbol{v}_1^2 p_1 + \boldsymbol{v}_2^2 p_2 + \boldsymbol{v}_3^2 p_3 = \boldsymbol{\nu}_{sg}^2$$
$$\boldsymbol{a}_1^2 p_1 + \boldsymbol{a}_2^2 p_2 + \boldsymbol{a}_3^2 p_3 \leq \boldsymbol{\sigma}_{sg}^2$$
$$\boldsymbol{b}_1^2 p_1 + \boldsymbol{b}_2^2 p_2 + \boldsymbol{b}_3^2 p_3 \leq \boldsymbol{\varsigma}_{sg}^2$$
$$p_1 + p_2 + p_3 = 1$$
$$\boldsymbol{u}_k^1 \leq \boldsymbol{u}_j^1 \leq \bar{\boldsymbol{u}}_k^1 \qquad \forall j \in [3]$$

$$
\begin{aligned}
&\underline{\boldsymbol{v}}_k^1 \le \boldsymbol{v}_j^1 \le \bar{\boldsymbol{v}}_k^1 && \forall j \in [3]\\
&\boldsymbol{a}_j^1 \ge |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1| && \forall j \in [3]\\
&\boldsymbol{b}_j^1 \ge |\boldsymbol{v}_j^1 - \boldsymbol{\nu}_k^1| && \forall j \in [3]\\
&\underline{\boldsymbol{u}}_{sg}^2 \le \boldsymbol{u}_j^2 \le \bar{\boldsymbol{u}}_{sg}^2 && \forall j \in [3]\\
&\underline{\boldsymbol{v}}_{sg}^2 \le \boldsymbol{v}_j^2 \le \bar{\boldsymbol{v}}_{sg}^2 && \forall j \in [3]\\
&\boldsymbol{a}_j^2 \ge |\boldsymbol{u}_j^2 - \boldsymbol{\mu}_{sg}^2| && \forall j \in [3]\\
&\boldsymbol{b}_j^2 \ge |\boldsymbol{v}_j^2 - \boldsymbol{\nu}_{sg}^2| && \forall j \in [3]\\
&p_1, p_2, p_3 \ge 0
\end{aligned}
$$

By perspective transformation, it is equivalent to the following linear optimization problem,

$$
\begin{aligned}
\max\ & \left(\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1 + y_{id}^1)\boldsymbol{e}_i\right)'(\boldsymbol{u}_1^1 + \boldsymbol{v}_1^1) + \left(\sum_{i\in\bar{\mathcal{C}}^2}(z_{id}^2(s) + x_{id}^2(s))\boldsymbol{e}_i\right)'(\boldsymbol{u}_1^2 + \boldsymbol{v}_1^2)\\
&+ \left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{u}_1^2 - \bar{\tau}^2 p_1 + \bar{\tau}^1 p_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2} z_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{v}_2^2\\
&+ \left(\sum_{i\in\bar{\mathcal{C}}^2} x_{id}^2(s)\boldsymbol{e}_i\right)'(\boldsymbol{u}_2^2 + \boldsymbol{v}_2^2) + \left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{u}_2^2 - \bar{\tau}^2 p_2 - \gamma_d^2 p_3\\
\text{s.t. }& \boldsymbol{u}_1^1 + \boldsymbol{u}_2^1 + \boldsymbol{u}_3^1 = \boldsymbol{\mu}_k^1\\
&\boldsymbol{v}_1^1 + \boldsymbol{v}_2^1 + \boldsymbol{v}_3^1 = \boldsymbol{\nu}_k^1\\
&\boldsymbol{a}_1^1 + \boldsymbol{a}_2^1 + \boldsymbol{a}_3^1 \le \boldsymbol{\sigma}_k^1\\
&\boldsymbol{b}_1^1 + \boldsymbol{b}_2^1 + \boldsymbol{b}_3^1 \le \boldsymbol{\varsigma}_k^1\\
&\boldsymbol{u}_1^2 + \boldsymbol{u}_2^2 + \boldsymbol{u}_3^2 = \boldsymbol{\mu}_{sg}^2\\
&\boldsymbol{v}_1^2 + \boldsymbol{v}_2^2 + \boldsymbol{v}_3^2 = \boldsymbol{\nu}_{sg}^2\\
&\boldsymbol{a}_1^2 + \boldsymbol{a}_2^2 + \boldsymbol{a}_3^2 \le \boldsymbol{\sigma}_{sg}^2\\
&\boldsymbol{b}_1^2 + \boldsymbol{b}_2^2 + \boldsymbol{b}_3^2 \le \boldsymbol{\varsigma}_{sg}^2\\
&p_1 + p_2 + p_3 = 1\\
&\underline{\boldsymbol{u}}_k^1 p_j \le \boldsymbol{u}_j^1 \le \bar{\boldsymbol{u}}_k^1 p_j && \forall j \in [3]\\
&\underline{\boldsymbol{v}}_k^1 p_j \le \boldsymbol{v}_j^1 \le \bar{\boldsymbol{v}}_k^1 p_j && \forall j \in [3]\\
&\boldsymbol{a}_j^1 \ge |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1 p_j| && \forall j \in [3]\\
&\boldsymbol{b}_j^1 \ge |\boldsymbol{v}_j^1 - \boldsymbol{\nu}_k^1 p_j| && \forall j \in [3]\\
&\underline{\boldsymbol{u}}_{sg}^2 p_j \le \boldsymbol{u}_j^2 \le \bar{\boldsymbol{u}}_{sg}^2 p_j && \forall j \in [3]\\
&\underline{\boldsymbol{v}}_{sg}^2 p_j \le \boldsymbol{v}_j^2 \le \bar{\boldsymbol{v}}_{sg}^2 p_j && \forall j \in [3]
\end{aligned}
$$

$$\boldsymbol{a}_j^2 \geq |\boldsymbol{u}_j^2 - \boldsymbol{\mu}_{sg}^2 p_j| \qquad \forall j \in [3]$$

$$\boldsymbol{b}_j^2 \geq |\boldsymbol{v}_j^2 - \boldsymbol{\nu}_{sg}^2 p_j| \qquad \forall j \in [3]$$

$$p_1, p_2, p_3 \geq 0.$$