Les Cahiers du GERAD ISSN: 0711–2440

Robust facility location under disruptions

C. Cheng, Y. Adulyasak,
L.-M. Rousseau

G–2018–91

October 2018

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

Citation suggérée: C. Cheng, Y. Adulyasak, L.-M. Rousseau (Octobre 2018). Robust facility location under disruptions, Rapport technique, Les Cahiers du GERAD G–2018–91, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2018-91) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Suggested citation: C. Cheng, Y. Adulyasak, L.-M. Rousseau (October 2018). Robust facility location under disruptions, Technical report, Les Cahiers du GERAD G–2018–91, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2018-91) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2018
– Bibliothèque et Archives Canada, 2018

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2018
– Library and Archives Canada, 2018

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Robust facility location under disruptions

Chun Cheng [a]

Yossiri Adulyasak [b]

Louis-Martin Rousseau [a]

[a] *CIRRELT & Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[b] *GERAD & Department of Logistics and Operations Management, HEC Montréal (Québec) Canada, H3T 2A7*

chun.cheng@polymtl.ca
yossiri.adulyasak@hec.ca
louis-martin.rousseau@cirrelt.net

October 2018
Les Cahiers du GERAD
G–2018–91

Copyright © 2018 GERAD, Cheng, Adulyasak, Rousseau

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract: Facility networks can be disrupted by, for example, power outages, poor weather conditions, or natural disasters, and the probabilities of these events may be difficult to estimate. This could lead to costly recourse decisions since the customers cannot be served by the planned facilities. In this paper, we study a fixed-charge location problem (FLP) that considers the risk of disruptions. We adopt a two-stage robust optimization method, 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 has been revealed. We implement a column-and-constraint generation (C&CG) algorithm to solve the robust models exactly. Instead of relying on dualization or reformulation techniques to deal with the subproblem, as is common in the literature, we use an exact enumeration method that allows us to take into account a discrete uncertainty set of facility failures. We also develop an approximation scheme for instances of a realistic size; it requires the adjustable decisions to be an affine function of the uncertain parameters. Numerical experiments show that the proposed C&CG algorithm outperforms existing methods for both the robust FLP and the robust p-median problem. We also introduce an enhancement to the formulation that allows the decision-maker to control the trade-off between the nominal cost increase and the robustness of the solution.

Keywords: Facility location, disruption risk, robust optimization, column-and-constraint generation, linear decision rule

---

Acknowledgments: The first author wishes to acknowledge the supporters of this study: the Institute for Data Valorization (IVADO) and the Interuniversity Research Center on Enterprise Networks, Logistics and Transportation (CIRRELT). We thank Calcul Quebéc for providing computing facilities for the experiments, and we are grateful to Erick Delage for valuable comments on the first draft of this paper.

1 Introduction

Decisions about facility locations are often strategic: the impact is long-lasting, and the decisions are difficult to reverse. During the lifetime of a facility the environment where it operates may experience dramatic changes. Events such as power outages, industrial accidents, or transportation infrastructure issues will disrupt the facility's operations. Natural disasters could leave the facility unable to function. The probability and impact of such events are often difficult to estimate because of a lack of high-quality historical data. Thus, it is important to consider uncertainties at the design phase to ensure that the facility location decisions are sufficiently robust to avoid high recourse costs at the operational stage.

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 (Snyder and Daskin 2005, Cui et al. 2010, Chen et al. 2011, Xie et al. 2015, Zhang et al. 2015). 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 (Gabrel et al. 2014). 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 (Jiang et al. 2012, Bertsimas et al. 2013) and the column-and-constraint generation (C&CG) algorithm (Zeng and Zhao 2013, An et al. 2014), have been developed to solve two-stage RO models exactly. Approximation schemes, such as affine decision rules (Ben-Tal et al. 2011, Simchi-Levi et al. 2016) and piecewise affine decision rules (Ardestani-Jaafari and Delage 2017), are also used.

In this paper, 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 the surviving facilities). The goal is to guarantee the system's performance under disruptive scenarios. The contributions of our work are fourfold:

(i) To the best of our knowledge, this paper is the first to apply the two-stage RO method to a fixed-charge location problem (FLP) under disruptions; the method can produce less conservative solutions with a higher reliability.

(ii) We develop a C&CG algorithm that solves the subproblem by exact enumeration. This approach allows us to take into account a discrete and combinatorial uncertainty set (i.e., the disrupted facilities). Numerical results show that the proposed approach outperforms existing C&CG algorithms and BD methods for both the robust UFLP/CFLP and the robust p-median problem (PMP). We also demonstrate that cuts generated from the second-worst scenarios can improve the convergence of the C&CG algorithm and reduce the computational time.

(iii) For realistically sized instances, we introduce an approximation scheme that restricts the adjustable decisions to be an affine function of the uncertain parameters. Numerical results show that this method can generate good solutions in a shorter time.

(iv) To reflect decision-makers' attitudes toward increased nominal cost when optimizing the worst-case performance, we impose an upper bound on the nominal cost. 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 when robustifying the system. Other managerial insights are also drawn from the numerical results.

The rest of this paper is organized as follows. Section 2 reviews previous work on facility location problems under uncertainty. Section 3 presents the deterministic and robust models for the FLP, and Section 4 describes the solution methods. Section 5 discusses the numerical results, and Section 6 provides concluding remarks.

2 Literature review

This section reviews papers on stochastic and robust facility location problems; for deterministic problems, see Daskin (2011). Table 1 presents a summary of the papers.

Most studies consider uncertain demand and transportation costs in the context of facility location. For early work, see the review by Snyder (2006). Baron et al. (2011) study a multi-period facility location problem with uncertain demand. They compare the solutions generated by two uncertainty sets: box and ellipsoid. Mišković et al. (2017) study a two-echelon capacitated facility location problem with uncertain transportation costs in both echelons. They use a budgeted uncertainty set and propose a memetic algorithm. Zetina et al. (2017) study a robust uncapacitated hub location problem, where interval uncertainty occurs in the demand, the transportation cost, and both simultaneously. They use a budgeted uncertainty set to control the level of conservatism and propose a branch-and-cut algorithm. Atamtürk and Zhang (2007) were the first to apply the two-stage RO approach to a location-transportation problem under demand uncertainty. They compare the solutions generated by the two-stage RO method with those provided by the stochastic programming model, and they find that the former approach offers a compromise between stochastic programming and a static RO approach. Ardestani-Jaafari and Delage (2017) study a multi-period robust location-transportation problem (MRLTP) with uncertain demand. They develop six approximation models using affine policies.

Facility and transportation network failures are also forms of supply-chain uncertainty. If the potential for such failures is ignored at the design stage, it could result in costly recourse decisions. Drezner (1987) was the first to consider disruptions in facility location models, introducing the unreliable 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. Snyder and Daskin (2005) introduce the reliable PMP and 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. Cui et al. (2010) 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 program (MIP) formulation and a continuous approximation model. The MIP is solved by Lagrangian relaxation. Chen et al. (2011) study a reliable inventory-location problem. They 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. Xie et al. (2015) present a reliable location-routing problem. Their problem setting is similar to that of Chen et al. (2011), except that they make routing decisions instead of forming inventory control policies. Shen et al. (2011) study a reliable UFLP. The problem is first formulated as a two-stage stochastic program and then as a nonlinear integer program. They propose a four-approximation algorithm for the case where the facilities have identical failure probabilities. Zhang et al. (2015) formulate the reliable location-routing problem as a scenario-based MIP and develop a simulated annealing heuristic.

Most studies of the facility location problem with disruptions assume that the probability information is known perfectly a priori; only a few papers consider RO approaches. An et al. (2014) propose a two-stage RO scheme for the reliable PMP. They use a budgeted uncertainty set to characterize disruptions and develop C&CG algorithms. Lu et al. (2015) propose a distributional RO model for the UFLP with correlated disruptions. They assume that the distributions of the disruptions are unknown, and they minimize the expected cost under the worst-case distribution. For more details about the facility location problem under disruptions, see the review by Snyder et al. (2016).

Our work differs from the two related papers cited above as follows. An et al. (2014) study the reliable PMP and explore the modeling capability of two-stage RO by taking into account partial disruption and demand variation. They evaluate their C&CG algorithm by a comparison with the BD method. We study the reliable FLP and propose two solution methods. First, we use an exact enumeration method for the subproblem in order to evaluate the worst-case recourse scenario in the C&CG algorithm. This approach outperforms the basic C&CG implementation in An et al. (2014). 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. Lu et al. (2015) focus on the UFLP with correlated disruptions and use numerical tests to quantify the benefits of considering disruptions that are correlated rather than independent. They use distributional RO instead of the two-stage RO framework.

Table 1: Summary of the literature

Authors	Problem	Uncertainty type	SP	RO (Uncertainty set)	Objective function	Solution method
Baron et al. (2011)	Multi-period facility location	Demand		Static (Box/ellipsoidal)	Max worst-case profit	Mixed integer program
Mišković et al. (2017)	Two-echelon facility location	Transportation cost		Static (Budgeted)	Min WOC	Memetic algorithm
Zetina et al. (2017)	Hub location	Demand; transportation cost; and both		Static (Budgeted)	Min WOC	Branch-and-cut
Atamtürk and Zhang (2007)	Location-transportation	Demand		Two-stage (Budgeted)	Min WOC	Mathematical analysis
Ardestani-Jaafari and Delage (2017)	MRLTP	Demand		Multi-stage (Budgeted)	Max worst-case profit	Affine polices
Drezner (1987)	PMP, (p, q)-center	Facility disruption	Static		Min ETC	Heuristics
Snyder and Daskin (2005)	PMP, UFLP	Facility disruption	Static		Min weighted NOC and EFC	Lagrangian relaxation
Cui et al. (2010)	UFLP	Facility disruption	Static		Min location cost and ETC	Lagrangian relaxation, CA
Chen et al. (2011)	Inventory-location	Facility disruption	Static		Min location cost and EITC	Lagrangian relaxation
Xie et al. (2015)	Location-routing	Facility disruption	Static		Min location cost and ETC	Lagrangian relaxation, CG
Shen et al. (2011)	UFLP	Facility disruption	Two-stage		Min location cost and ETC	Three heuristics
Zhang et al. (2015)	Location-routing	Facility disruption	Two-stage		Min location cost and ETC	Simulated annealing heuristic
An et al. (2014)	PMP	Facility disruption		Two-stage (Budgeted)	Min weighted NOC and WOC	C&CG
Lu et al. (2015)	UFLP	Facility disruption		Distributional	Min expected cost	BD, search-and-cut
This paper	UFLP	Facility disruption		Two-stage (Budgeted)	Min WOC	C&CG, affine policy

SP: stochastic programming; **CA**: continuous approximation; **CG**: column generation; **WOC**: worst-case cost; **NOC**: nominal cost; **ETC**: expected transportation cost; **EFC**: expected failure cost; **EITC**: expected inventory and transportation costs.

3 Mathematical models

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

3.1 Notation

Let I and J be 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 unsatisfied demand at site $i \in I$.

3.2 Robust UFLP

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

$$\min_{\mathbf{y},\mathbf{x},\mathbf{u}} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} x_{ij} + \sum_{i\in I} p'_i h_i u_i, \tag{1}$$

$$s.t. \quad \sum_{j\in J} x_{ij} + u_i \geq 1 \qquad \forall i \in I, \tag{2}$$

$$x_{ij} \leq y_j \qquad \forall i \in I, \forall j \in J, \tag{3}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{4}$$

$$x_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J, \tag{5}$$

$$u_i \geq 0 \qquad \forall i \in I. \tag{6}$$

The objective (1) minimizes the total cost, which includes the fixed facility location cost, the demand-weighted transportation cost, and the penalty cost of unmet demand. Constraints (2) 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 4.2). Constraints (3) ensure that demand nodes are assigned to open facilities. Constraints (4)–(6) impose the integrality and non-negativity constraints. Note that, in contrast to the model presented in Daskin (2011), 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 (An et al. 2014):

$$\mathbb{Z}(k) = \{\mathbf{z} \in \{0,1\}^{|J|} : \sum_{j \in J} z_j \leq k\}, \tag{7}$$

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

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

$$\begin{aligned}
&\min_{\mathbf{y},\mathbf{x}(\cdot),\mathbf{u}(\cdot)} \sup_{\mathbf{z}\in\mathbb{Z}(k)} \sum_{j \in J} f_j y_j + \sum_{i \in I}\sum_{j \in J} h_i d_{ij} x_{ij}(\mathbf{z}) + \sum_{i \in I} p'_i h_i u_i(\mathbf{z}), && && (8)\\
&\quad s.t. \quad \sum_{j \in J} x_{ij}(\mathbf{z}) + u_i(\mathbf{z}) \geq 1 && \forall \mathbf{z} \in \mathbb{Z}(k), \forall i \in I, && (9)\\
&\qquad\qquad x_{ij}(\mathbf{z}) \leq y_j(1 - z_j) && \forall \mathbf{z} \in \mathbb{Z}(k), \forall i \in I, \forall j \in J, && (10)\\
&\qquad\qquad y_j \in \{0,1\} && \forall j \in J, && (11)\\
&\qquad\qquad x_{ij}(\mathbf{z}) \geq 0 && \forall \mathbf{z} \in \mathbb{Z}(k), \forall i \in I, \forall j \in J, && (12)\\
&\qquad\qquad u_i(\mathbf{z}) \geq 0 && \forall \mathbf{z} \in \mathbb{Z}(k), \forall i \in I. && (13)
\end{aligned}$$

The objective function (8) 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 (10) ensure that the demand nodes are assigned to open and surviving facilities in a disruptive scenario.

3.3 Robust CFLP

The formulation for the deterministic CFLP is as follows:

$$\begin{aligned}
&\min_{\mathbf{y},\mathbf{x},\mathbf{u}} \sum_{j \in J} f_j y_j + \sum_{i \in I}\sum_{j \in J} h_i d_{ij} x_{ij} + \sum_{i \in I} p'_i h_i u_i, && && (14)\\
&s.t. \quad (2)\text{–}(6) \text{ and} && && (15)\\
&\qquad \sum_{i \in I} h_i x_{ij} \leq C_j y_j && \forall j \in J. && (16)
\end{aligned}$$

The objective functions of the deterministic UFLP and CFLP are the same. Constraints (16) ensure that once a facility is open, its capacity is respected. They also ensure that customers are allocated to open facilities, so constraints (3) become redundant. However, we retain them because they can strengthen the linear programming relaxation (Daskin 2011).

Similarly, we get the robust CFLP from (8)–(13) with the constraints

$$\sum_{i \in I} h_i x_{ij}(\mathbf{z}) \leq C_j y_j (1 - z_j) \qquad \forall \mathbf{z} \in \mathbb{Z}(k), \forall j \in J. \tag{17}$$

3.4 Properties of the robust formulations

Observation 1 *We can replace the uncertainty set* $\mathbb{Z}(k)$ *with its convex hull* $\mathbb{Z}'(k) = \{\mathbf{z} \in \mathbb{R}^{|J|} : 0 \leq \mathbf{z} \leq 1, \sum_{j \in J} z_j \leq k\}$*, because the worst-case scenario always occurs at the extreme points.*

This observation makes it possible to use the affine policy for the ARC models, because only when $z_j, \forall j \in J$ is relaxed to a continuous variable, we can apply duality theory to reformulate the models based on the affine policy.

Lemma 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 mitigation 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. Suppose the number of open facilities is m, i.e., $\sum_{j \in J} \hat{y}_j = m$. From Observation 1, the worst-case scenario has exactly k disrupted facilities. Figure 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 $B_1 = B_2$. ☐

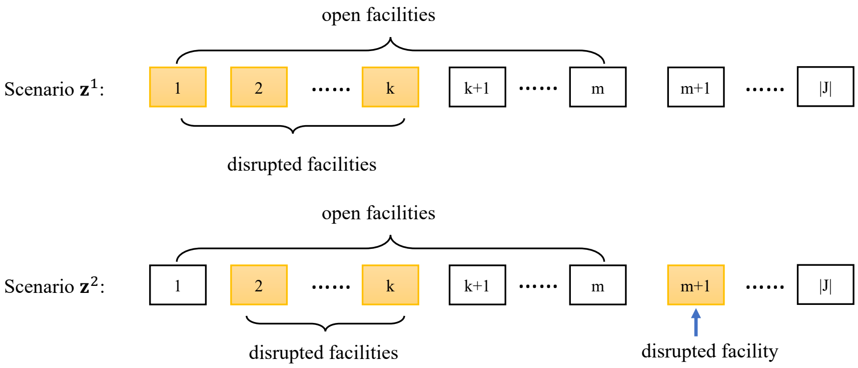


Figure 1: Illustration for the proof of Lemma 1

Lemma 2 *Given the facility location* $\hat{\mathbf{y}}$ $(\sum_{j \in J} \hat{y}_j = m)$ *and the uncertainty set* $\mathbb{Z}(k)$*, we have that if* $m > k$*, the worst-case disruptions occur at open facilities, i.e., those with* $\hat{y}_j = 1$*. If* $m \leq k$*, the worst-case disruptions occur at all the open facilities, and all the demand in the system will be left unsatisfied.*

Proof. This result follows from the proof of Lemma 1. ☐

Lemma 2 indicates that when $m > k$, we can enumerate all the potential worst-case scenarios by considering only the set of open 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.

4 Solution methods

In this section, we introduce an exact enumeration-based C&CG algorithm and an approximation scheme for the ARC models. For both models, we also implement the duality-based C&CG algorithm (An et al. 2014) and the BD method as benchmarks.

4.1 Enumeration-based 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 enumeration-based C&CG algorithm for the two robust models and then introduce an enhancement strategy to improve the computational performance.

4.1.1 Enumeration-based 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

$$\phi = \min_{\mathbf{y},s,\{\mathbf{x}^l\}_{l=1}^n,\{\mathbf{u}^l\}_{l=1}^n} s, \tag{18}$$

$$s.t. \quad s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} x_{ij}^l + \sum_{i\in I} p_i' h_i u_i^l \qquad \forall l \in \{1,...,n\}, \tag{19}$$

$$\sum_{j\in J} x_{ij}^l + u_i^l \geq 1 \qquad \forall l \in \{1,...,n\}, \forall i \in I, \tag{20}$$

$$x_{ij}^l \leq y_j(1-z_j^l) \qquad \forall l \in \{1,...,n\}, \forall i \in I, \forall j \in J, \tag{21}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{22}$$

$$x_{ij}^l \geq 0 \qquad \forall l \in \{1,...,n\}, \forall i \in I, \forall j \in J, \tag{23}$$

$$u_i^l \geq 0 \qquad \forall l \in \{1,...,n\}, \forall i \in I. \tag{24}$$

Subproblem. We use an exact enumeration method based on Lemma 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:

$$\psi = \min_{\mathbf{x},\mathbf{u}} \sum_{j\in J} f_j \hat{y}_j + \sum_{i\in I}\sum_{j\in \bar{J}} h_i d_{ij} x_{ij} + \sum_{i\in I} p_i' h_i u_i, \tag{25}$$

$$s.t. \quad \sum_{j\in \bar{J}} x_{ij} + u_i \geq 1 \qquad \forall i \in I, \tag{26}$$

$$x_{ij}, u_i \geq 0 \qquad \forall i \in I, \forall j \in \bar{J}. \tag{27}$$

To solve the MCFP, we use the Python NetworkX 2.0 package (NetworkX 2015), 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 open facilities are disrupted and all the customer demand is left unsatisfied.

4.1.2 Enumeration-based C&CG algorithm for robust CFLP

The **master problem** for the robust CFLP is defined by (18)–(24) and the constraints

$$\sum_{i\in I} h_i x_{ij}^l \leq C_j y_j (1-z_j^l) \qquad \forall l \in \{1,...,n\}, \forall j \in J. \tag{28}$$

Subproblem. 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; it is defined by (25)–(27) and the constraints

$$\sum_{i\in I} h_i x_{ij}^l \leq C_j \qquad \forall j \in \bar{J}. \tag{29}$$

If $\sum_{j\in J}\hat{y}_j \leq k$, in the worst-case scenario, all the open 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 Appendices A and B for the robust UFLP and CFLP respectively. The subproblems of both algorithms are obtained by applying duality theory. We note that our exact enumeration method has three advantages over solving the dualized subproblem. First, we do not need to set big-M values for the constraints, which helps to avoid the numerical issues that can arise with large parameter values. Second, our method is not sensitive to the type of the uncertain variable z_j, whereas the dualized method requires z_j to be binary, in order to linearize the nonlinear term in the objective function of the subproblem. If z_j is not binary, we have to apply the Karush–Kuhn–Tucker (KKT) conditions to derive the subproblem. Our preliminary tests in Section 5.2.2 show that the dualized model is easier to solve than the KKT model. Third, we have cost information for all the potential worst-case scenarios, not just the actual worst-case scenario.

4.1.3 Algorithm enhancement

Our exact enumeration method provides information about all the potential worst-case scenarios, and this section introduces an enhancement that makes use of this information 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 5.2.1, we test four ways of adding the scenarios. An et al. (2014) 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 An et al. (2014) applies only to the case where $I = J$. However, our method can be used in any situation. In addition, since the exact 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 enumeration-based C&CG.

Algorithm 1: Enumeration-based C&CG algorithm

```
Step 1: Initialization
    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 $\frac{UB-LB}{UB} \leq \epsilon$ :
            an $\epsilon$–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.
```

4.2 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 (Ben-Tal et al. 2004, Gorissen et al. 2015). This restriction often leads to tractable robust models for realistically sized problems. Our ARC models have a fixed recourse, all the observations are linear functions of $\mathbf{z}$, and the uncertainty set $\mathbb{Z}'(k)$ is nonempty and bounded, so we can use the LDR to approximate them.

4.2.1 Affine policy for robust UFLP

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

$$\min_{\mathbf{y},\mathbf{W},\mathbf{w},\mathbf{A},\mathbf{a}} \sup_{\mathbf{z}\in\mathbb{Z}'(k)} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij}(\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) + \sum_{i\in I} p'_i h_i(\mathbf{A}_i^T\mathbf{z} + a_i), \tag{30}$$

$$\begin{aligned}
s.t.\quad & \sum_{j\in J}(\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) + (\mathbf{A}_i^T\mathbf{z} + a_i) \geq 1 && \forall \mathbf{z} \in \mathbb{Z}'(k), \forall i \in I, && (31)\\
& \mathbf{W}_{ij}^T\mathbf{z} + w_{ij} \leq y_j(1 - z_j) && \forall \mathbf{z} \in \mathbb{Z}'(k), \forall i \in I, \forall j \in J, && (32)\\
& y_j \in \{0,1\} && \forall j \in J, && (33)\\
& \mathbf{W}_{ij}^T\mathbf{z} + w_{ij} \geq 0 && \forall \mathbf{z} \in \mathbb{Z}'(k), \forall i \in I, \forall j \in J, && (34)\\
& \mathbf{A}_i^T\mathbf{z} + a_i \geq 0 && \forall \mathbf{z} \in \mathbb{Z}'(k), \forall i \in I. && (35)
\end{aligned}$$

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 (Gorissen et al. 2015), which produces a mixed-integer linear programming (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 Ardestani-Jaafari and Delage (2017). We have implemented both methods, and our tests show that for our problem it is more efficient to solve the MILP directly. This reformulation is as follows:

$$\min_{\substack{\mathbf{y},\mathbf{W},\mathbf{w},\mathbf{A},\mathbf{a},s,\delta,\\ \boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\theta},\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\varsigma},\boldsymbol{\pi},\boldsymbol{\nu}}} s, \tag{36}$$

$$\begin{aligned}
s.t.\quad & s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} w_{ij} + \sum_{i\in I} p'_i h_i a_i + k\delta + \sum_{e\in J}\alpha_e, && && (37)\\
& \delta + \alpha_e \geq \sum_{i\in I}\sum_{j\in J} h_i d_{ij} W_{ije} + \sum_{i\in I} p_i h_i A_{ie} && \forall e \in J, && (38)\\
& \sum_{j\in J} w_{ij} + a_i - k\xi_i - \sum_{e\in J}\eta_{ie} \geq 1 && \forall i \in I, && (39)\\
& \xi_i + \eta_{ie} \geq -\sum_{j\in J} W_{ije} - A_{ie} && \forall i \in I, \forall e \in J, && (40)\\
& -k\theta_{ij} - \sum_{e\in J}\mu_{ije} \geq -y_j + w_{ij} && \forall i \in I, \forall j \in J, && (41)\\
& \theta_{ij} + \mu_{ije} \geq W_{ije} + y_j && \forall i \in I, \forall j \in J, \forall e \in J, e = j, && (42)\\
& \theta_{ij} + \mu_{ije} \geq W_{ije} && \forall i \in I, \forall j \in J, \forall e \in J, e \neq j, && (43)\\
& w_{ij} - k\sigma_{ij} - \sum_{e\in J}\varsigma_{ije} \geq 0 && \forall i \in I, \forall j \in J, && (44)\\
& \sigma_{ij} + \varsigma_{ije} \geq -W_{ije} && \forall i \in I, \forall j \in J, \forall e \in J, && (45)\\
& -k\pi_i - \sum_{e\in J}\nu_{ie} + a_i \geq 0 && \forall i \in I, && (46)\\
& \pi_i + \nu_{ie} \geq -A_{ie} && \forall i \in I, \forall e \in J, && (47)\\
& y_j \in \{0,1\} && \forall j \in J, && (48)\\
& \delta, \alpha_e, \xi_i, \eta_{ie}, \theta_{ij}, \mu_{ije}, \sigma_{ij}, \varsigma_{ije}, \pi_i, \nu_{ie} \geq 0 && \forall i \in I, \forall j \in J, \forall e \in J. && (49)
\end{aligned}$$

4.2.2 Affine policy for robust CFLP

The AARC model for the robust CFLP is defined by (30)–(35) and the constraints

$$\sum_{i\in I} h_i(\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) \leq C_j y_j(1 - z_j) \qquad \forall \mathbf{z} \in \mathbb{Z}'(k), \forall j \in J. \tag{50}$$

After applying duality to each robust constraint, we get the MILP reformulation, which consists of (36)–(49) with the constraints

$$k\rho_j + \sum_{e\in J}\Gamma_{ej} + \sum_{i\in I} h_i w_{ij} \leq C_i y_j \qquad \forall j \in J, \tag{51}$$

$$\rho_j + \Gamma_{ej} \geq \sum_{i\in I} h_i W_{ije} + C_j y_j \qquad \forall e \in J, \forall j \in J, e = j \tag{52}$$

$$\rho_j + \Gamma_{ej} \geq \sum_{i\in I} h_i W_{ije} \qquad \forall e \in J, \forall j \in J, e \neq j \tag{53}$$

$$\rho_j, \Gamma_{ej} \geq 0 \qquad \forall e \in J, \forall j \in J. \tag{54}$$

According to Bertsimas and Goyal (2012), 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 $k = |J|$, both the exact algorithm and the approximation scheme identify solutions with no open facilities.

5 Numerical results

In this section, we present the instances and explore:

- The efficiency of the multiple-scenario technique and the performance of the enumeration-based 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 5.2).
- The impact of the LDR on the computational complexity and solution quality (Section 5.3).
- 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 5.4).

5.1 Instances

We consider a 49-site data set (Daskin 2011), 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 $d_{ij}, \forall i \in I, \forall j \in J$ 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 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. The problem is strategic in nature, and we can afford to let the algorithm run for sufficient time to ensure a high-quality solution. This time limit is in line with other papers considering facility location under uncertainty (Ardestani-Jaafari and Delage 2017, Zetina et al. 2017). The optimality tolerance ϵ is set to 0.01% for all the exact algorithms unless otherwise specified.

5.2 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.

5.2.1 Performance of multiple-scenario technique

Each time, after solving the enumeration-based 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 2.

Table 2: 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 2 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.

5.2.2 Exact algorithms for robust UFLP

In the following sections, *C&CG-E* indicates the enumeration-based C&CG algorithm and *C&CG-D* indicates the C&CG algorithm with the dualized subproblem. Table 3 presents the average results, and the detailed results are given in Tables 10–12 in Appendix D. In the last six columns of Table 3, we give the preliminary results generated by the C&CG algorithm and the BD method with the KKT subproblem (C&CG-KKT and BD-KKT, respectively).

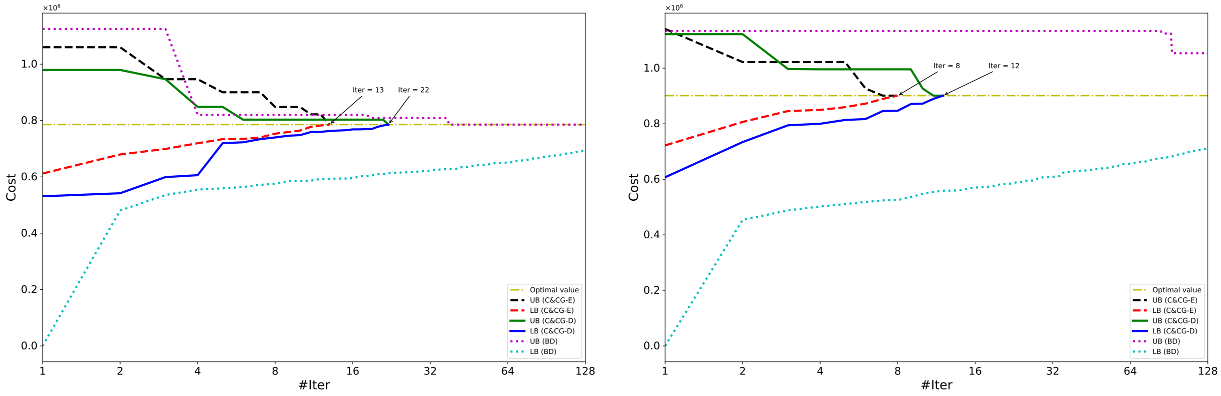
Duality-based C&CG and BD versus C&CG-KKT and BD-KKT. Table 3 shows that the duality-based algorithms perform better than the algorithms with the KKT subproblem. Specifically, C&CG-D consumes less time than C&CG-KKT on average, and the duality-based BD method can solve six more instances to optimality than BD-KKT. Therefore, in the following sections, we use the duality-based algorithms as benchmarks for C&CG-E, and no further experiments are conducted on the KKT algorithms.

Table 3: Average results for the robust UFLP

		C&CG-E				C&CG-D				BD (dualized subproblem)				C&CG-KKT			BD-KKT		
k	p'	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU	Gap	#Opt	#Iter	CPU	Gap	#Opt	CPU	Gap	#Opt	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	**0.0**	35/35	11186.3	17.6	12/35	63990.0
	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	N/A			N/A		
3	$p^{0.8}$	**3.4**	19/35	94.6	43930.6	4.0	19/35	121.3	44616.9	N/A				N/A			N/A		
	p^{max}	**3.3**	**18/35**	87.3	46206.2	3.9	17/35	120.6	46726.7	N/A				N/A			N/A		
4	$p^{0.8}$	**9.8**	**14/35**	153.2	57359.4	10.9	13/35	165.0	58191.8	N/A				N/A			N/A		
	p^{max}	11.0	17/35	122.9	52245.8	**10.4**	17/35	150.0	53551.9	N/A				N/A			N/A		

N/A: No further experiments are performed.

C&CG-E versus C&CG-D and BD. 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 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 2(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.



(a) Robust UFLP

(b) Robust CFLP

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

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 optimaility gap while the CPU time and the number of iterations are greater.

Table 3 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 10, 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.

5.2.3 Exact algorithms for robust CFLP

We present the summarized results in Table 4 and the detailed results in Tables 13–15 in Appendix D. Table 4 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 2(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.

Table 4: 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			

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.

5.2.4 Enumeration-based C&CG algorithm for other variants of the facility location problem under disruptions

In this section, we apply C&CG-E to the uncapacitated/capacitated p-median problem (UPMP/CPMP). For both models, the objective function optimizes the weighted sum of the nominal cost and the worst-case cost. The detailed models (An et al. 2014) are given in Appendix C. Since our preliminary experiments as well as those of An et al. (2014) 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 take the same values as in An et al. (2014), and the optimality tolerance ϵ is set to 0.1%. The results for the two models are given in Tables 5 and 6, where $J = I$ and ϱ is the weight of the worst-case cost.

Table 5: Results for uncapacitated p-median problem under disruptions

				$p' = 15$						$p' = p^{max}$					
				C&CG-E			An et al. (2014)			C&CG-E			An et al. (2014)		
$\lvert J\rvert$	ϱ	p	k	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU
25	0.2	8	1	0.0	1	**0.9**	0.0	4	1.8	0.0	1	**0.3**	0.0	4	1.9
			2	0.0	5	36.3	0.0	5	**2.8**	0.0	5	12.2	0.0	5	**3.2**
			3	0.0	21	**216.7**	0.0	58	324.8	0.0	35	**353.4**	0.0	70	1082.8
		10	1	0.0	1	1.0	0.0	1	**0.5**	0.0	1	1.9	0.0	1	**0.5**
			2	0.1	3	2.9	0.0	5	**2.5**	0.1	3	8.1	0.0	5	**3.2**
			3	0.0	5	7.3	0.0	9	**6.4**	0.0	7	31.9	0.0	11	**11.2**
	0.4	8	1	0.0	3	10.6	0.0	4	**1.8**	0.0	3	2.7	0.0	4	**2.2**
			2	0.0	8	63.1	0.0	13	**12.1**	0.0	8	16.8	0.0	13	**15.3**
			3	0.0	46	**660.8**	0.0	96	1332.1	0.0	64	**3535.5**	0.1	109	9942.8
		10	1	0.0	1	1.5	0.0	1	**0.5**	0.0	1	0.9	0.0	1	**0.6**
			2	0.0	4	26.6	0.0	6	**3.3**	0.0	4	17.0	0.0	6	**5.2**
			3	0.0	13	195.9	0.0	22	**35.0**	0.0	14	69.5	0.0	25	**56.7**
49	0.2	8	1	0.0	1	5.7	0.0	3	**5.3**	0.0	1	**1.8**	0.0	3	6.4
			2	0.0	6	67.5	0.0	9	**30.2**	0.0	27	**3797.9**	0.0	75	27180.3
			3	0.0	48	**12651.5**	0.0	85	31218.0	**5.5**	55	88248.9	7.6	94	88555.6
		10	1	0.0	1	5.7	0.0	1	**1.6**	0.0	1	3.9	0.0	1	**1.7**
			2	0.0	6	107.4	0.0	11	**56.8**	0.0	6	**68.9**	0.0	11	78.8
			3	0.0	27	**4081.5**	0.0	40	4821.1	**0.0**	40	**29917.8**	5.5	122	89319.2
	0.4	8	1	0.0	2	9.5	0.0	3	**5.4**	0.0	2	9.1	0.0	3	**8.0**
			2	0.0	17	**627.3**	0.0	29	1114.7	**5.7**	42	91967.6	9.4	99	88954.9
			3	**7.9**	63	92404.0	11.5	95	89076.8	**16.2**	45	88745.0	19.3	69	88781.1
		10	1	0.0	2	13.1	0.0	3	**4.8**	0.0	2	5.0	0.0	3	**4.8**
			2	0.0	13	**718.2**	0.0	30	1205.8	0.0	13	**586.2**	0.0	30	2224.0
			3	7.1	59	88010.8	**5.3**	87	89078.4	**12.6**	41	91181.5	14.5	85	90145.5
Average				**0.6**$^{(2)}$	14.8	8330.2	0.7$^{(2)}$	25.8	9097.6	**1.7**$^{(4)}$	17.5	16607.6	2.3$^{(5)}$	35.4	20266.1

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

Table 6: Results for capacitated p-median problem under disruptions

				$p' = 15$						$p' = p^{max}$					
				C&CG-E			An et al. (2014)			C&CG-E			(An et al. 2014)		
$\|J\|$	ϱ	p	k	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU
25	0.2	8	1	0.0	2	**2.0**	0.0	3	5.3	0.0	2	**2.3**	0.0	3	**2.3**
			2	0.0	5	12.6	0.0	7	**11.2**	0.0	10	**82.5**	0.0	20	112.4
			3	0.0	25	**328.0**	0.0	41	418.1	0.0	62	**19529.7**	0.0	120	40543.4
		10	1	0.0	1	**0.4**	0.0	1	1.0	0.0	1	**0.4**	0.0	1	0.5
			2	0.0	2	**2.2**	0.0	3	4.3	0.0	3	4.6	0.0	4	**4.3**
			3	0.0	7	**19.9**	0.0	12	26.4	0.0	13	159.6	0.0	19	**98.1**
	0.4	8	1	0.0	3	6.5	0.0	4	**4.6**	0.0	3	12.0	0.0	4	**4.4**
			2	0.0	8	59.3	0.0	15	**50.5**	0.0	12	**138.2**	0.0	23	190.4
			3	0.0	37	**1002.3**	0.0	74	3349.7	0.0	72	**30259.1**	0.0	130	71842.8
		10	1	0.0	1	**0.4**	0.0	1	0.7	0.0	1	**0.5**	0.0	1	0.6
			2	0.0	4	**6.3**	0.0	10	17.0	0.0	5	**8.8**	0.0	10	18.4
			3	0.0	12	158.8	0.0	21	**94.4**	0.0	21	**401.7**	0.0	35	507.7
49	0.2	8	1	0.0	1	**3.3**	0.0	3	7.0	0.0	2	13.1	0.0	3	**7.9**
			2	0.0	10	**355.4**	0.0	15	460.6	**0.0**	36	75769.5	3.3	56	89353.4
			3	**4.9**	56	89826.1	5.7	87	88889.0	22.3	32	90474.1	**21.7**	50	88703.5
		10	1	0.0	1	**1.8**	0.0	2	4.4	0.0	1	**3.7**	0.0	2	5.0
			2	0.0	8	**247.4**	0.0	21	1005.9	0.0	11	**880.5**	0.0	21	1791.4
			3	0.0	31	**17207.5**	0.1	43	26352.0	**11.1**	29	94667.5	11.3	50	91420.6
	0.4	8	1	0.0	2	10.7	0.0	3	**7.2**	0.0	2	13.9	0.0	3	**10.0**
			2	0.0	27	**6816.4**	0.1	59	50219.2	**13.3**	32	93588.1	17.0	60	95364.5
			3	14.5	40	93032.9	**14.2**	56	90059.0	37.8	29	94312.5	**36.8**	43	91086.0
		10	1	0.0	3	31.0	0.0	4	**20.6**	0.0	3	32.8	0.0	4	**15.6**
			2	0.0	27	**24130.0**	0.0	46	29568.7	0.0	26	25861.3	0.1	39	**16645.4**
			3	13.2	38	90096.3	**9.7**	63	87510.1	**23.4**	31	96423.2	26.9	46	92089.4
Average				$1.4^{(3)}$	14.6	13473.2	$\mathbf{1.2}^{(3)}$	24.8	15753.6	$\mathbf{4.5}^{(\mathbf{5})}$	18.3	25943.3	$4.9^{(6)}$	31.1	28325.7

Table 5 shows that for the UPMP, C&CG-E has better performance, in terms of average gap, number of iterations, and CPU time. In particular, when $p' = p^{max}$, one more instance can be solved to optimality by C&CG-E. Table 6 shows that for the CPMP, when $p' = 15$, C&CG-D provides a slightly better optimality gap while C&CG-E consumes less CPU time and requires much lower number of iterations on average. When $p' = p^{max}$, C&CG-E is more efficient: one more instance can be solved, the optimality gap is smaller, there are fewer iterations, and the average CPU time is lower. From both tables, we can see that C&CG-E generally works better for instances with a large budget. For a small budget, C&CG-E requires extra computational time to probe all the potential worst-case scenarios and to solve a larger master problem with more scenarios.

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

5.3 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 $k = 1, 2, 3, 4$ 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 7, and the detailed results are given in Tables 16–17 in Appendix D.

Table 7: Average results of the linear decision rule 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 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 10–17, 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 $p' = p^{0.8}$.

5.4 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.

5.4.1 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 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 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.

Table 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

5.4.2 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 robustfying the system. To reflect this, we introduce another group of constraints:

$$
\begin{aligned}
& f_j y_j + \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij}^0 + \sum_{i \in I} p_i h_i u_i^0 \leq (1+q) c_0^*, && && (55)\\
& \sum_{j \in J} x_{ij}^0 + u_i^0 \geq 1 && \forall i \in I, && (56)\\
& x_{ij}^0 \leq y_j && \forall i \in I, \forall j \in J, && (57)\\
& \sum_{i \in I} h_i x_{ij}^0 \leq C_j y_j && \forall j \in J, && (58)\\
& x_{ij}^0 \geq 0 && \forall i \in I, \forall j \in J, && (59)\\
& u_i^0 \geq 0 && \forall i \in I, && (60)
\end{aligned}
$$

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 constant $q \geq 0$ indicates the decision-makers' tolerance of increased nominal cost when robustifying the system.

We study the impact of constraints (55)–(60) by varying the value of q. The experiments are conducted on two randomly selected instances, and the results are presented in Table 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 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 open. 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.

Table 9: Impact of imposing an upper bound on the nominal cost ($k = 2, p' = p^{0.8}$)

			ARC model with constraints (55)–(60)			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

6 Conclusions

We have solved a reliable fixed-charge location problem, where each facility is exposed to the risk of disruptions. We used a budgeted uncertainty set to characterize the risk and applied a two-stage RO method. To solve the ARC models exactly, we developed a C&CG algorithm where an exact enumeration method is used for the subproblem. We also used the LDR to approximate the ARC models, in order to solve large instances in a reasonable time. Our numerical experiments show that the enumeration-based 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.

Appendix A Duality-based C&CG algorithm and Benders decomposition method for robust UFLP

A.1 Duality-based C&CG algorithm

Master Problem. The master problem is defined by Equations (18)–(24).

Subproblem. Let λ and β be the dual variables of constraints (9) and (10), respectively. The dual problem of the inner minimization problem can be written as

$$\psi = \max_{\mathbf{x},\mathbf{u},\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} \sum_{j\in J} f_j \hat{y}_j + \sum_{i\in I} \lambda_i - \sum_{i\in I}\sum_{j\in J} \hat{y}_j(1-z_j)\beta_{ij}, \tag{61}$$

$$\begin{aligned}
s.t.\quad & \lambda_i - \beta_{ij} \le h_i d_{ij} && \forall i\in I, \forall j \in J, && (62)\\
& \lambda_i \le p_i' h_i && \forall i \in I, && (63)\\
& \sum_{j\in J} z_j \le k, && && (64)\\
& z_j \in \{0,1\} && \forall i\in I, && (65)\\
& \lambda_i \ge 0 && \forall i \in I, && (66)\\
& \beta_{ij} \ge 0 && \forall i\in I, j\in J. && (67)
\end{aligned}$$

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:

$$\begin{aligned}
& \pi_{ij} \ge 0 && \forall i\in I, j\in J, && (68)\\
& \pi_{ij} \le \beta_{ij} && \forall i\in I, j\in J, && (69)\\
& \pi_{ij} \le \mathbb{M}_{ij} z_j && \forall i\in I, j\in J, && (70)\\
& \pi_{ij} \ge \beta_{ij} + \mathbb{M}_{ij}(z_j - 1) && \forall i\in I, j\in J, && (71)
\end{aligned}$$

where $\mathbb{M}_{ij} = \max\{h_i(p_i' - d_{ij}), 0\}$.

Therefore, the full subproblem is

$$\psi = \max_{x,u,z,\lambda,\beta} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(\beta_{ij} - \pi_{ij}), \tag{72}$$

subject to constraints (62)–(71).

A.2 Benders decomposition method

Master problem. The master problem of the Benders decomposition method is

$$\phi = \min_{\mathbf{y},s} s, \tag{73}$$

$$\begin{aligned}
s.t.\quad & s \ge \sum_{j\in J} f_j y_j + \sum_{i\in I}\hat{\lambda}_i^l - \sum_{i\in I}\sum_{j\in J}\hat{\beta}_{ij}^l y_j (1-\hat{z}_j^l) && \forall l \in \{1,...,n\}, && (74)\\
& y \in \{0,1\}, && && (75)
\end{aligned}$$

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 (62)–(72).

Appendix B Duality-based C&CG algorithm and Benders decomposition method for robust CFLP

B.1 Duality-based C&CG algorithm

Master problem. The master problem is defined by Equations (18)–(24) and (28).

Subproblem. Let γ be the dual variable of constraints (17). The resulting dual problem can be written as follows:

$$\psi = \max_{\mathbf{x},\mathbf{u},\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(1-z_j)\beta_{ij} - \sum_{j\in J} C_j\hat{y}_j\gamma_j(1-z_j), \tag{76}$$

$$s.t.\ \ \lambda_i - \beta_{ij} - h_i\gamma_j \le h_i d_{ij} \qquad \forall i\in I, \forall j\in J, \tag{77}$$

$$\gamma_j \ge 0 \qquad \forall j\in J, \tag{78}$$

$$\text{and } (63)\text{–}(67). \tag{79}$$

There are two nonlinear terms in the objective function (76), i.e., $z_j\beta_{ij}$ and $\gamma_j z_j$. We can use the technique of Appendix A.1 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:

$$\zeta_j \ge 0 \qquad \forall j\in J, \tag{80}$$

$$\zeta_j \le \gamma_j \qquad \forall j\in J, \tag{81}$$

$$\zeta_j \le \mathbb{M}_j z_j \qquad \forall j\in J, \tag{82}$$

$$\zeta_j \ge \gamma_j + \mathbb{M}_j(z_j - 1) \qquad \forall j\in J, \tag{83}$$

where $\mathbb{M}_j = \max\{\max_i(p_i - d_{ij}), 0\}$.

Therefore, the full subproblem is

$$\psi = \max_{x,u,z,\lambda,\beta} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(\beta_{ij}-\pi_{ij}) - \sum_{i\in J} C_j\hat{y}_j(\gamma_j-\zeta_j), \tag{84}$$

subject to constraints (77)–(83).

B.2 Benders decomposition method

Master problem. The master problem of the Benders decomposition method is

$$\phi = \min_{\mathbf{y},s} s, \tag{85}$$

$$s.t.\ \ s \ge \sum_{j\in J} f_j y_j + \sum_{i\in I}\hat{\lambda}_i^l - \sum_{i\in I}\sum_{j\in J} y_j(1-\hat{z}_j^l)\hat{\beta}_{ij}^l - \sum_{j\in J} C_j y_j\hat{\gamma}_j^l(1-\hat{z}_j^l) \qquad \forall l\in\{1,...,n\}, \tag{86}$$

$$y \in \{0,1\}, \tag{87}$$

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 (77)–(84).

Appendix C Robust uncapacitated/capacitated p-median problem

Parameters:

ϱ = weight of the worst-case cost.

b_{ij} = fraction of demand from customer $i \in I$ that is satisfied by facility $j \in J$ in a disruptive scenario.

g_i = 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 3.1.

The uncapacitated p-median problem under disruptions can be formulated as

$$\min_{\mathbf{x},\mathbf{y}}(1-\varrho)\sum_{i\in I}\sum_{j\in J}d_{ij}h_i x_{ij} + \varrho \max_{\mathbf{z}\in\mathbb{Z}(k)} \min_{(\mathbf{b},\mathbf{g})\in S(\mathbf{y},\mathbf{z})}(\sum_{i\in I}\sum_{j\in J}d_{ij}h_i b_{ij} + \sum_{i\in I}p'_i h_i g_i), \tag{88}$$

$$\begin{aligned}
s.t.\quad & x_{ij} \le y_j && \forall i\in I, j\in J, && (89)\\
& \sum_{j\in J} x_{ij} = 1 && \forall i \in I, && (90)\\
& \sum_{j\in J} y_j = p, && && (91)\\
& x_{ij} \ge 0, && \forall i\in I, j\in J, && (92)\\
& y_j \in \{0,1\} && \forall j\in J, && (93)\\
\text{where}\quad & S(\mathbf{y},\mathbf{z}) = \{b_{ij} \le 1 - z_j && \forall i\in I, \forall j\in J, && (94)\\
& b_{ij} \le y_j && \forall i\in I, \forall j\in J, && (95)\\
& \sum_{j\in J} b_{ij} + g_i = 1 && \forall i\in I, && (96)\\
& b_{ij} \ge 0, && \forall i\in I, \forall j\in J, && (97)\\
& g_i \ge 0, && \forall i\in I\}. && (98)
\end{aligned}$$

The capacitated p-median problem under disruptions is

$$\min_{\mathbf{x},\mathbf{y}}(1-\varrho)\sum_{i\in I}\sum_{j\in J}d_{ij}h_i x_{ij} + \varrho \max_{\mathbf{z}\in\mathbb{Z}(k)} \min_{(\mathbf{b},\mathbf{g})\in S^C(\mathbf{y},\mathbf{z})}(\sum_{i\in I}\sum_{j\in J}d_{ij}h_i b_{ij} + \sum_{i\in I}p'_i h_i g_i) \tag{99}$$

$$\begin{aligned}
s.t.\quad & (89)\text{–}(93) && && (100)\\
& \sum_{i\in I} h_i x_{ij} \le C_j y_j && \forall j\in J, && (101)\\
\text{where}\quad & S^C(\mathbf{y},\mathbf{z}) = \{(94)\text{–}(98), && && (102)\\
& \sum_{i\in I} h_i b_{ij} \le C_j y_j && \forall j\in J\}. && (103)
\end{aligned}$$

Appendix D Detailed numerical results

The detailed results of the numerical experiments are presented in Tables 10–17.

Table 10: **Performance of exact algorithms for the reliable UFLP ($k = 2$)**

		$p' = p^{0.8}$										$p' = p^{max}$									
		Enumeration-based C&CG				Duality-based C&CG			Benders decomposition			Enumeration-based C&C				Duality-based C&CG			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 11: Performance of C&CG algorithms for the reliable UFLP ($k = 3$)

		$p' = p^{0.8}$										$p' = p^{max}$									
		Enumeration-based C&CG					Duality-based C&CG					Enumeration-based C&CG					Duality-based C&CG				
$\|J\|$	$\|I\|$	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 12: Performance of C&CG algorithms for the reliable UFLP ($k = 4$)

		$p' = p^{0.8}$										$p' = p^{max}$									
		Enumeration-based C&CG					Duality-based C&CG					Enumeration-based C&CG					Duality-based C&CG				
$\|J\|$	$\|I\|$	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**$^{(\mathbf{21})}$	**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**$^{(\mathbf{18})}$	150.0	53551.9

Table 13: Performance of exact algorithms for the reliable CFLP ($k = 2$)

		$p' = p^{0.8}$											$p' = p^{max}$										
		Enumeration-based C&CG				Duality-based C&CG			Benders decomposition			Enumeration-based C&CG				Duality-based C&CG			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 14: Performance of C&CG algorithms for the reliable CFLP ($k = 3$)

		$p' = p^{0.8}$										$p' = p^{max}$									
		Enumeration-based C&CG					Duality-based C&CG					Enumeration-based C&CG					Duality-based C&CG				
$\|J\|$	$\|I\|$	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 15: Performance of C&CG algorithms for the reliable CFLP ($k = 4$)

		$p' = p^{0.8}$										$p' = p^{max}$									
		Enumeration-based C&CG					Duality-based C&CG					Enumeration-based C&CG					Duality-based C&CG				
$\|J\|$	$\|I\|$	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 16: 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 17: 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

References

An Y, Zeng B, Zhang Y, Zhao L (2014) Reliable p-median facility location problem: Two-stage robust models and algorithms. Transportation Research Part B: Methodological 64:54–72.

Ardestani-Jaafari A, Delage E (2017) The value of flexibility in robust location–transportation problems. Transportation Science 52(1):189–209.

Atamtürk A, Zhang M (2007) Two-stage robust network flow and design under demand uncertainty. Operations Research 55(4):662–673.

Baron O, Milner J, Naseraldin H (2011) Facility location: A robust optimization approach. Production and Operations Management 20(5):772–785.

Ben-Tal A, Do Chung B, Mandala SR, Yao T (2011) Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. Transportation Research Part B: Methodological 45(8):1177–1189.

Ben-Tal A, Goryashko A, Guslitzer E, Nemirovski A (2004) Adjustable robust solutions of uncertain linear programs. Mathematical Programming 99(2):351–376.

Bertsimas D, Goyal V (2012) On the power and limitations of affine policies in two-stage adaptive optimization. Mathematical Programming 134(2):491–531.

Bertsimas D, Litvinov E, Sun XA, Zhao J, Zheng T (2013) Adaptive robust optimization for the security constrained unit commitment problem. IEEE Transactions on Power Systems 28(1):52–63.

Chen Q, Li X, Ouyang Y (2011) Joint inventory-location problem under the risk of probabilistic facility disruptions. Transportation Research Part B: Methodological 45(7):991–1003.

Cui T, Ouyang Y, Shen ZJM (2010) Reliable facility location design under the risk of disruptions. Operations Research 58(4-part-1):998–1011.

Daskin MS (2011) Network and Discrete Location: Models, Algorithms, and Applications (John Wiley & Sons).

Drezner Z (1987) Heuristic solution methods for two location problems with unreliable facilities. Journal of the Operational Research Society 38(6):509–514.

Gabrel V, Murat C, Thiele A (2014) Recent advances in robust optimization: An overview. European Journal of Operational Research 235(3):471–483.

Gorissen BL, Yanıkoğlu İ, den Hertog D (2015) A practical guide to robust optimization. Omega 53:124–137.

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

Lu M, Ran L, Shen ZJM (2015) Reliable facility location design under uncertain correlated disruptions. Manufacturing & Service Operations Management 17(4):445–455.

Mišković S, Stanimirović Z, Grujičić I (2017) Solving the robust two-stage capacitated facility location problem with uncertain transportation costs. Optimization Letters 11(6):1169–1184.

NetworkX (2015) NetworkX documentation. Accessed September 17, 2018, https://networkx.github.io/documentation/networkx-1.10/index.html.

Shen ZJM, Zhan RL, Zhang J (2011) The reliable facility location problem: Formulations, heuristics, and approximation algorithms. INFORMS Journal on Computing 23(3):470–482.

Simchi-Levi D, Trichakis N, Zhang PY (2016) Designing response supply chain against bioattacks. Working paper, https://pdfs.semanticscholar.org/ceb6/be21bbcbbda3ef985690bd629bbd8b69c7cc.pdf.

Snyder LV (2006) Facility location under uncertainty: A review. IIE Transactions 38(7):547–564.

Snyder LV, Atan Z, Peng P, Rong Y, Schmitt AJ, Sinsoysal B (2016) OR/MS models for supply chain disruptions: A review. IIE Transactions 48(2):89–109.

Snyder LV, Daskin MS (2005) Reliability models for facility location: The expected failure cost case. Transportation Science 39(3):400–416.

Xie W, Ouyang Y, Wong SC (2015) Reliable location-routing design under probabilistic facility disruptions. Transportation Science 50(3):1128–1138.

Zeng B, Zhao L (2013) Solving two-stage robust optimization problems using a column-and-constraint generation method. Operations Research Letters 41(5):457–461.

Zetina CA, Contreras I, Cordeau JF, Nikbakhsh E (2017) Robust uncapacitated hub location. Transportation Research Part B: Methodological 106:393–410.

Zhang Y, Qi M, Lin WH, Miao L (2015) A metaheuristic approach to the reliable location routing problem under disruptions. Transportation Research Part E: Logistics and Transportation Review 83:90–110.