Submitted to *INFORMS Journal on Optimization*
manuscript (Please, provide the manuscript number!)

Robust Facility Location Under Disruptions

Chun Cheng
Institute of Supply Chain Analytics, Dongbei University of Finance and Economics, Dalian, China 116025, chun.cheng@polymtl.ca

Yossiri Adulyasak
HEC Montréal and GERAD, Montreal, Canada H3T 2A7, yossiri.adulyasak@hec.ca

Louis-Martin Rousseau
Polytechnique Montréal and CIRRELT, Montreal, Canada H3C 3A7, louis-martin.rousseau@cirrelt.net

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 customers cannot be served by the planned facilities. In this paper, we study a fixed-charge location problem (FLP) that considers disruption risks. 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 a linear programming-based enumeration method that allows us to take into account a discrete uncertainty set of facility failures. This also gives the flexibility to tackle cases when the dualization technique cannot be applied to the subproblem. We further develop an approximation scheme for instances of a realistic size. Numerical experiments show that the proposed C&CG algorithm outperforms existing methods for both the robust FLP and the robust p-median problem.

Key words: Facility location, disruption risk, robust optimization

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). 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, Billionnet et al. 2014, Bertsimas and Shtern 2018, Simchi-Levi et al. 2019b) and the column-and-constraint generation (C&CG) algorithm (Zeng and Zhao 2013, Ayoub and Poss 2016), 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. 2019a) and piecewise affine decision rules (Ardestani-Jaafari and Delage 2017), are also used.

Our contributions. 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 or leave them unmet). The goal is to guarantee the system's performance under disruptive scenarios. The contributions of our work are as follows: (1) We solve a fixed-charge location problem (FLP) under disruptions using the two-stage RO method together with a budgeted uncertainty set. We illustrate that for this class of problems, the adjustable robust models are necessary to produce less conservative solutions in comparison with the static RO method. (2) We compare the numerical efficiency of exhaustive scenario search to the usual mixed-integer linear programming (MILP) reformulation when solving the NP-hard adversarial problem that arises in a step of the C&CG algorithm. (3) We validate the numerical efficiency and quality of solutions obtained when employing affine decision rules on this class of problems. (4) To illustrate the use of a bi-objective approach to better trade off the reliability cost and the nominal cost, we impose an upper bound on the nominal cost when robustifying the system. The results demonstrate that the bound constraints can further reduce the conservativeness of the robust solutions and serve as a decision support tool indicating the trade-off between reliability and nominal cost.

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 adjustable robust models, 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 paper 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 conservativeness and propose a branch-and-cut algorithm. Atamtürk and Zhang (2007) are 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) is the first to consider disruptions in facility location models, introducing the unreliable p-median problem (PMP) and the (p, q)-center problem. In the former, each facility has a given probability of being inactive; in the latter, p facilities must be built to minimize the maximum cost when at most q facilities are disrupted. 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. Xie and Ouyang (2019) study a reliable service network design problem, where customers must pass certain network access points to reach facilities for services. They assume that each network access point is associated with a site-dependent failure probability and minimize the expected system cost.

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 distributionally RO (DRO) model for the UFLP with correlated disruptions. They assume that the exact probability distribution of disruption scenarios are unknown but lies in a distributional uncertainty set, such that the marginal disruption probability of a site is equal to a given value. The expected cost under the worst-case distribution is minimized. For more details 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 demand variation. Specifically, they introduce a parameter ϑ (not a random variable) to denote demand change and study its effect by setting it to a negative value, 0, and a positive value in numerical tests. Note that they assume the facility set J and the customer set I be the same, which makes it possible to incorporate ϑ into the model, because there should be a link between disruptions and

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
Xie and Ouyang (2019)	Service system design	Facility disruption	Two-stage		Min location cost and ETC	Lagrangian relaxation
An et al. (2014)	PMP	Facility disruption		Two-stage (Budgeted)	Min weighted NOC and WOC	C&CG
Lu et al. (2015)	UFLP	Facility disruption		Distributionally	Min expected cost	BD, search-and-cut
This paper	UFLP, CFLP	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.

demand variations. In their setting, since $J = I$, disruptions may occur at a customer site (which is also a facility site), resulting in 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 a linear programming (LP)-based enumeration method for the subproblem in order to evaluate the worst-case recourse scenario in the C&CG algorithm. This approach does not require to set big-M values, and it also provides information for other potential worst-case scenarios, which can be used to speed up the algorithm. Second, we introduce an approximation scheme based on the affine policy for large instances, and we provide conditions under which this scheme produces optimal solutions. We further introduce an enhancement to the robust formulations, which can effectively reduce the conservativeness of solutions. We emphasize that our modeling scheme is also able to incorporate demand variations for situations with $J = I$. Lu et al. (2015) consider the UFLP with correlated disruptions, which are characterized by a joint distribution. Therefore, the DRO instead of the two-stage RO framework is used. They first exploit the structural property of the DRO model (supermodularity) and then reformulate it as a stochastic program, where standard solution methods such as BD can be used. Their numerical tests focus on quantifying the benefits of considering disruptions that are correlated rather than independent.

3. Mathematical Models

This section introduces the notation and presents the deterministic and the adjustable robust models.

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

3.1. Deterministic UFLP and CFLP

The deterministic UFLP can be formulated as follows (Daskin 2011):

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

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

$$x_{ij} \leq y_j \qquad \forall i \in I, j \in J, \tag{1c}$$

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

$$x_{ij} \geq 0 \qquad \forall i \in I, j \in J. \tag{1e}$$

The objective (1a) minimizes the total cost, which includes the fixed facility location cost and the demand-weighted transportation cost. Constraints (1b) indicate that customer demand must be fully satisfied. Since it is a minimization problem, the equality relationship always holds at the optimum. Constraints (1c) impose that demand nodes can only be assigned to opened facilities. Constraints (1d)–(1e) impose the integrality and non-negativity constraints.

The formulation for the deterministic CFLP is as follows:

$$\min_{\mathbf{y},\mathbf{x}} \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij}, \tag{2a}$$

$$\text{s.t.} \quad \text{(1b)–(1e) and} \tag{2b}$$

$$\sum_{i \in I} h_i x_{ij} \leq C_j y_j \qquad \forall j \in J. \tag{2c}$$

The objective functions of the deterministic UFLP and CFLP are the same. Constraints (2c) ensure that once a facility is opened, its capacity is respected. They also ensure that customers are allocated to opened facilities, so constraints (1c) become redundant. However, we retain them because they can strengthen the linear programming relaxation (Daskin 2011).

3.2. Adjustable Robust UFLP and CFLP

In this section, we first introduce the uncertainty set and then present the two-stage adjustable robust counterpart (ARC) models for the UFLP and CFLP under disruptions.

Uncertainty set. We use a budgeted uncertainty set to characterize facilities' disruption risk (An et al. 2014):

$$\mathbb{Z}(k) = \left\{ \mathbf{z} \in \{0,1\}^{|J|} : \sum_{j \in J} z_j \leq k \right\}, \tag{3}$$

where random variable $z_j = 1$ if facility $j \in J$ is disrupted, and $z_j = 0$ otherwise. The budget constraint ensures that at most k (k is integer) facilities fail simultaneously in a disruptive scenario.

Before presenting the ARC models, we propose an extension to the deterministic models. Specifically, we introduce a variable u_i to the left-hand side of constraints (1b) for each $i \in I$, which produces

$$\sum_{j \in J} x_{ij} + u_i \geq 1 \qquad \forall i \in I.$$

Correspondingly, we modify the objective functions to

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

where each p_i can be explained as a marginal penalty for violating constraints (1b) or leaving demand unsatisfied. The introduction of $\mathbf{u}$ allows us to find a trade-off between reassigning demand or leaving it unmet when considering the adjustable robust formulations. Moreover, it helps to guarantee the feasibility of the second-stage problem under any first-stage decision and any uncertainty realization—a property termed as *relatively complete recourse*.

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

$$\min_{\mathbf{y}} \ \sum_{j \in J} f_j y_j + \sup_{\mathbf{z} \in \mathbb{Z}(k)} \ g(\mathbf{y}, \mathbf{z}) \tag{4a}$$

$$\text{s.t.} \ \ y_j \in \{0,1\} \qquad \forall j \in J, \tag{4b}$$

where $g(\mathbf{y},\mathbf{z})$ is the optimal value of the second-stage problem, once location decisions $\mathbf{y}$ are made and facility statuses $\mathbf{z}$ are revealed, and is defined as

$$g(\mathbf{y},\mathbf{z}) = \min_{\mathbf{x},\mathbf{u}} \sum_{i\in I}\sum_{j\in J} h_i d_{ij} x_{ij} + \sum_{i\in I} p_i h_i u_i \tag{5a}$$

$$\text{s.t.} \quad \sum_{j\in J} x_{ij} + u_i \geq 1 \qquad \forall i \in I, \tag{5b}$$

$$x_{ij} \leq y_j(1 - z_j) \qquad \forall i \in I, j \in J, \tag{5c}$$

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

$$u_i \geq 0 \qquad \forall i \in I. \tag{5e}$$

The objective function (4a) minimizes the sum of the fixed location cost and the worst-case recourse cost. The second-stage model finds the least costly recourse decisions $(\mathbf{x},\mathbf{u})$ under a given location decision $\mathbf{y}$ and a realized scenario $\mathbf{z}$. Note that, besides $\mathbf{x}$, the auxiliary variables $\mathbf{u}$ are also adjustable over $\mathbf{z}$. Constraints (5c) ensure that demand nodes are assigned to opened and surviving facilities in a disruptive scenario.

Our ARC model can also incorporate disruption-caused demand deviations presented in An et al. (2014). Specifically, we first assume that $I = J$ and then change the objective function (5a) to

$$\min_{\mathbf{x},\mathbf{u}} \sum_{i\in I}\sum_{j\in J} (1 - \vartheta z_i) h_i d_{ij} x_{ij} + \sum_{i\in I} (1 - \vartheta z_i) p_i h_i u_i.$$

That being said, we will still focus on the models with objective (5a), due to the fact that ϑ is treated as a parameter instead of a random variable. Moreover, there are many applications, where candidate facility sites and customers are not exactly the same. For example, in global supply chains, upstream factories and downstream customers are normally located in different areas.

Adjustable robust CFLP. Similarly, we get the adjustable robust CFLP from (4a)–(4b), and (5a)–(5e) with the constraints

$$\sum_{i\in I} h_i x_{ij} \leq C_j y_j (1 - z_j) \qquad \forall j \in J. \tag{6}$$

In this robust CFLP, when facility $j \in J$ is disrupted, its capacity becomes 0 and its service capability is totally lost. We refer to this as *complete disruption*. Our framework is also able to incorporate *partial disruption*, where a damaged facility can still satisfy part of the demand (Peng et al. 2011, Liberatore et al. 2012). The reliable CFLP with partial disruption can be modeled with (4a)–(4b), (5a)–(5b), (5d)–(5e), and the constraints

$$\sum_{i \in I} h_i x_{ij} \leq C_j y_j (1 - \omega_j z_j) \qquad \forall j \in J,$$

where parameter ω_j $(0 < \omega_j \leq 1)$ is the proportion of lost capacity at location $j \in J$ when a disruption occurs. We give the model here to demonstrate the strong modeling capability of the two-stage RO approach, but in the following sections we focus on complete disruption, i.e., $\omega_j = 1, \forall j \in J$, to demonstrate the applicability of our approach when dealing with a discrete uncertainty set associated with this type of disruptions.

3.3. A Toy Example to Illustrate the Importance of the Adjustable Policy

We note that variables $\mathbf{x}$ and $\mathbf{u}$ must be adaptive to $\mathbf{z}$, otherwise the non-adaptive (or static) RO model always identifies solutions with $y_j = 0, \forall j \in J$ at optimality when $k \geq 1$. A logical explanation of this result is that when $k \geq 1$, the adversary can always choose an opened facility to disrupt, and thus rendering solutions with no opened facilities in the absence of recourse. A toy example (refer to Figure 1) is given in this section to demonstrate the importance of the adjustable policy.

We assume that there are three uncapacitated candidate facilities in the system, among which only facilities 1 and 2 are opened, i.e., $y_1 = 1, y_2 = 1, y_3 = 0$. There are also three customers. In the nominal disruption-free scenario, based on the distances between facilities and customers, customer 1 is served by facility 1, and customers 2 and 3 are served by facility 2. We further assume that $k = 1$, i.e., one facility is disrupted in the worst-case scenario.

In disruption scenarios, the flow variable $x_{ij}, i \in I, j \in J$ is a function of random variables $\mathbf{z}$. For simplicity, we assume that the function is linear (see Section 4.2 for more details) and formally

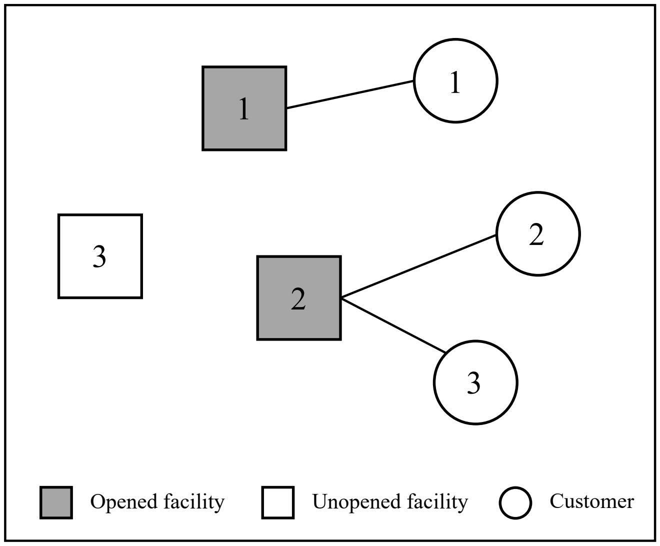


Figure 1 **Illustration for the importance of the adjustable policy**

define it as $x_{ij} = \sum_{j' \in J} W_{ij}^{j'} z_{j'} + w_{ij}$. According to constraints (5c) and (5d), the product flow x_{1j} at customer 1 can be expressed as

$$0 \leq W_{11}^1 z_1 + W_{11}^2 z_2 + W_{11}^3 z_3 + w_{11} \leq y_1(1 - z_1) \qquad \forall \mathbf{z} \in \mathbb{Z}(k), \tag{7a}$$

$$0 \leq W_{12}^1 z_1 + W_{12}^2 z_2 + W_{12}^3 z_3 + w_{12} \leq y_2(1 - z_2) \qquad \forall \mathbf{z} \in \mathbb{Z}(k), \tag{7b}$$

$$0 \leq W_{13}^1 z_1 + W_{13}^2 z_2 + W_{13}^3 z_3 + w_{13} \leq y_3(1 - z_3) \qquad \forall \mathbf{z} \in \mathbb{Z}(k). \tag{7c}$$

Then, under the static policy (i.e., $x_{1j} = w_{1j}, \forall j \in \{1,2,3\}$) and the given location decision, the flow at customer 1 can be rewritten as

$$0 \leq w_{11} \leq (1 - z_1) \qquad \forall \mathbf{z} \in \mathbb{Z}(k), \tag{8a}$$

$$0 \leq w_{12} \leq (1 - z_2) \qquad \forall \mathbf{z} \in \mathbb{Z}(k), \tag{8b}$$

$$0 \leq w_{13} \leq 0 \qquad \forall \mathbf{z} \in \mathbb{Z}(k), \tag{8c}$$

Since constraints (8a)–(8c) should hold for any realization in the uncertainty set, then under realizations $(z_1, z_2, z_3) = (1,0,0)$ and $(z_1, z_2, z_3) = (0,1,0)$, we will have

$$0 \leq w_{11} \leq 0, \qquad 0 \leq w_{12} \leq 1, \qquad 0 \leq w_{13} \leq 0,$$

$$0 \leq w_{11} \leq 1, \qquad 0 \leq w_{12} \leq 0, \qquad 0 \leq w_{13} \leq 0,$$

respectively, which indicate that $w_{11} = w_{12} = w_{13} = 0$ and $x_{11} = x_{12} = x_{13} = 0$. That is, the demand of customer 1 will be left unmet, leading to a penalty cost.

However, under the adjustable policy, we will have the following results based on the aforementioned uncertainty realizations

$$0 \leq W^1_{11} + w_{11} \leq 0, \qquad 0 \leq W^1_{12} + w_{12} \leq 1, \qquad 0 \leq W^1_{13} + w_{13} \leq 0,$$
$$0 \leq W^2_{11} + w_{11} \leq 1, \qquad 0 \leq W^2_{12} + w_{12} \leq 0, \qquad 0 \leq W^2_{13} + w_{13} \leq 0.$$

Thus, under the uncertainty realization $(z_1, z_2, z_3) = (1, 0, 0)$, the flow variable $x_{12} = W^1_{12} + w_{12}$ can be 0 or 1, depending on the transportation cost d_{12} and the penalty cost p_1. Under the uncertainty realization $(z_1, z_2, z_3) = (0, 1, 0)$, the flow variable $x_{11} = W^2_{11} + w_{11}$ can take a value of 1 as in the nominal scenario. Therefore, we can conclude that under the adjustable policy, the system has more flexibility in satisfying customers' demand and controlling costs, instead of simply penalizing the demand as does the static policy.

In addition, one may argue that we can use other models to produce here-and-now decisions and then reoptimize the allocation decisions after a disruption, besides adopting the adjustable robust models. However, in our case, the static robust models will generate first-stage solutions with no opened facilities as illustrated in the toy example, which leave no space for reoptimization after a disruption, i.e., all the demand will be penalized in a disruption scenario. We may also consider taking the deterministic models' solutions as the here-and-now decision. Nevertheless, our experiments in Section 5.4.1 show that the deterministic solutions lead to significantly higher post-disruption worst-case costs in comparison with the adjustable robust solutions.

3.4. Properties of the Adjustable Robust Formulations

In this section, we present two properties of the adjustable robust formulations, based on which we develop the solution methods in next section.

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

Proof. See Appendix A.1.

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

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

Lemma 2 indicates that when $m > k$, we can enumerate all the potential worst-case scenarios by considering only the set of opened facilities instead of set J. This helps to reduce the number of minimum cost flow problems to be solved in the subproblem of the C&CG framework.

4. Solution Methods

In this section, we introduce a new C&CG algorithm and an approximation scheme for the ARC models. For both models, we also implement the duality-based C&CG algorithm (An et al. 2014) and the BD method as benchmarks. We close this section by discussing potential extensions of our modeling and solution schemes to other applications.

4.1. Column-and-Constraint-Generation Algorithm

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

C&CG algorithm for adjustable 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 adjustable robust UFLP is

$$\phi = \min_{\mathbf{y}, s, \{\mathbf{x}^l\}_{l=1}^n, \{\mathbf{u}^l\}_{l=1}^n} s, \tag{9a}$$

$$\text{s.t.} \quad s \geq \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij}^l + \sum_{i \in I} p_i h_i u_i^l \qquad \forall l \in \{1, \dots, n\}, \tag{9b}$$

$$\sum_{j \in J} x_{ij}^l + u_i^l \geq 1 \qquad \forall l \in \{1, \dots, n\}, i \in I, \tag{9c}$$

$$x_{ij}^l \leq y_j (1 - z_j^l) \qquad \forall l \in \{1, \dots, n\}, i \in I, j \in J, \tag{9d}$$

$$y_j \in \{0, 1\} \qquad \forall j \in J, \tag{9e}$$

$$x_{ij}^l \geq 0 \qquad \forall l \in \{1, \dots, n\}, i \in I, j \in J, \tag{9f}$$

$$u_i^l \geq 0 \qquad \forall l \in \{1, \dots, n\}, i \in I. \tag{9g}$$

We use a LP-based enumeration method derived from 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:

$$\psi = \min_{\mathbf{x}, \mathbf{u}} \sum_{j \in J} f_j \hat{y}_j + \sum_{i \in I} \sum_{j \in \bar{J}} h_i d_{ij} x_{ij} + \sum_{i \in I} p_i h_i u_i, \tag{10a}$$

$$\text{s.t.} \quad \sum_{j \in \bar{J}} x_{ij} + u_i \geq 1 \qquad \forall i \in I, \tag{10b}$$

$$x_{ij}, u_i \geq 0 \qquad \forall i \in I, j \in \bar{J}. \tag{10c}$$

To solve the MCFP, we can use three methods: (i) Directly solving model (10) using an off-the-shelf solver; (ii) Using the Python NetworkX 2.0 package (NetworkX 2015), which applies a primal network simplex algorithm; (iii) Solving a set of separated models of model (10) by customer index i. Specifically, for a customer $i \in I$, if its distance to the nearest surviving facility is equal to or smaller than p_i (i.e., $\min_{j\in\bar{J}} d_{ij} \le p_i$), then it will be severed by that facility; Otherwise, a penalty cost is incurred. We have tested these three methods on some randomly selected instances, and results show that the average computing time of the second method is shorter than that of the first one, but very close to that of the third method. Since method (iii) based on the separated models can only be applied to the case when facilities are uncapacitated, we choose to use the NetworkX package for solving the MCFPs.

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

C&CG algorithm for adjustable robust CFLP. The master problem for the adjustable robust CFLP is defined by (9a)–(9g) and the constraints

$$\sum_{i\in I} h_i x_{ij}^l \le C_j y_j (1 - z_j^l) \qquad \forall l \in \{1,\dots,n\}, j \in J. \tag{11}$$

Being similar 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 (10a)–(10c) and the constraints

$$\sum_{i\in I} h_i x_{ij} \le C_j \qquad \forall j \in \bar{J}. \tag{12}$$

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

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

Multiple scenario generation. At each iteration we add multiple scenarios instead of just one. Any replicated scenarios are eliminated before we solve the master problem. In Section 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. Moreover, since the LP-based enumeration method evaluates all the potential scenarios, no extra computational effort is needed to produce and evaluate an alternative scenario in this framework. In addition, when solving the subproblem of the duality-based C&CG via an optimization solver, we can also extract scenarios associated with suboptimal solutions, and then add them to the master problem. However, our preliminary tests show that it is more computationally efficient to use the LP-based C&CG algorithm with the multiple-scenario generation technique for our problem.

Algorithm 1 describes the proposed C&CG algorithm.

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. The LDR is commonly used as a heuristic method and to provide computational insights for exact algorithms. Before introducing the LDR for our ARC models, we first present Lemma 3, which is a sufficient condition for the adoption of LDR.

Lemma 3. *In the ARC models, the uncertainty set $\mathbb{Z}(k)$ has an integrality property when k is integer, that is, the optimal value of the ARC models will not be affected if we replace $\mathbb{Z}(k)$ with its convex hull*

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

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

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

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

Step 3: Iterate until the algorithm terminates:

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

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

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

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

Proof. See Appendix A.2.

The proof process of Lemma 3 also indicates that when $\sum_{j\in J}\hat{y}_j \ge k$, the worst-case scenario always occurs at the extreme points. And if $\sum_{j\in J}\hat{y}_j < k$, all the opened facilities would be disrupted as indicated in Lemma 2. Note that our proposed C&CG algorithm does not take advantage from the integrality property here even though it does exist for both the UFLP and the CFLP. Thus, the proposed C&CG framework can generally be applied to the problems where such property does not hold (e.g., see Section 4.3).

Affine policy for adjustable robust UFLP. The integrality property of the uncertainty set makes it possible to reformulate the ARC models based on the LDR. We set $x_{ij} = \mathbf{W}_{ij}^T\mathbf{z} + w_{ij}$ and $u_i = \mathbf{A}_i^T\mathbf{z} + a_i$, 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 adjustable robust UFLP is

$$\min_{\mathbf{y},\mathbf{W},\mathbf{w},\mathbf{A},\mathbf{a},s} s \tag{13a}$$

$$\text{s.t.} \quad s \ge \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij}(\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) + \sum_{i\in I} p_i h_i(\mathbf{A}_i^T\mathbf{z} + a_i) \quad \forall \mathbf{z} \in \mathbb{Z}'(k), \tag{13b}$$

$$\sum_{j\in J}(\mathbf{W}_{ij}^T\mathbf{z}+w_{ij})+(\mathbf{A}_i^T\mathbf{z}+a_i)\geq 1 \qquad \forall \mathbf{z}\in\mathbb{Z}'(k), i\in I, \tag{13c}$$

$$\mathbf{W}_{ij}^T\mathbf{z}+w_{ij}\leq y_j(1-z_j) \qquad \forall \mathbf{z}\in\mathbb{Z}'(k), i\in I, j\in J, \tag{13d}$$

$$y_j\in\{0,1\} \qquad \forall j\in J, \tag{13e}$$

$$\mathbf{W}_{ij}^T\mathbf{z}+w_{ij}\geq 0 \qquad \forall \mathbf{z}\in\mathbb{Z}'(k), i\in I, j\in J, \tag{13f}$$

$$\mathbf{A}_i^T\mathbf{z}+a_i\geq 0 \qquad \forall \mathbf{z}\in\mathbb{Z}'(k), i\in I. \tag{13g}$$

To solve the AARC model, we can make use of two methods: (1) We apply duality to each robust constraint (Gorissen et al. 2015), which produces an MILP model. We then feed the resulting MILP model directly to an optimization solver. (2) We develop a BD method for an equivalent reformulation of model (13) following the approach of Ardestani-Jaafari and Delage (2017). We have implemented both methods, and our preliminary tests show that for our problem, the first method is more efficient. The corresponding MILP reformulation with respect to this method is provided in Appendix D.

Affine policy for adjustable robust CFLP. The AARC model for the robust CFLP is defined by (13) and the constraints

$$\sum_{i\in I}h_i(\mathbf{W}_{ij}^T\mathbf{z}+w_{ij})\leq C_j y_j(1-z_j) \qquad \forall \mathbf{z}\in\mathbb{Z}'(k), j\in J. \tag{14}$$

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

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 $k=|J|$, the affine policy with $\mathbf{W}_{ij}=0,\ w_{ij}=0,\ \mathbf{A}_i=0,\ a_i=1,\ \forall i\in I, j\in J$ achieves the same worst-case cost as the optimal worst-case cost. In this situation, both the exact algorithm and the approximation scheme identify the solutions with no opened facilities.

4.3. Extension of Our Modeling and Solution Schemes for Other Applications

Multiple uncertainty sets. An and Zeng (2014) and Cheng et al. (2018) suggest that multiple uncertainty sets can be used to further reduce the conservativeness of robust solutions. To be specific, each uncertainty set with a different budget is assigned a weight to characterize decision makers' risk preference, and the overall cost of facility location and the weighted sum of the worst-case cost is minimized. We note that our C&CG algorithm and the LDR can still be used in this context. In the former method, one subproblem is solved using the LP-based enumeration method for each uncertainty set. All the identified worst-case scenarios, recourse variables, and corresponding constraints are added to the master problem in each iteration. In the latter method, the constraints containing variables $\mathbf{z}$ in model (13) will be required to hold for all the uncertainty sets. Duality theory can still be used to derive the reformulation.

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

$$
\begin{aligned}
\min_{\mathbf{y},\mathbf{x}(\cdot),\mathbf{u}(\cdot),\mathbf{v}(\cdot)} \quad & \sup_{\mathbf{z}\in\mathbb{Z}(k)} \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} d_{ij} v_{ij}(\mathbf{z}) + \sum_{i\in I} p_i h_i u_i(\mathbf{z}), \\
\text{s.t.} \quad & h_i x_{ij}(\mathbf{z}) \le Q v_{ij}(\mathbf{z}) && \forall \mathbf{z}\in\mathbb{Z}(k), i\in I, j\in J, \\
& v_{ij}(\mathbf{z}) \ge 0, \text{ integer} && \forall \mathbf{z}\in\mathbb{Z}(k), i\in I, j\in J, \\
& \text{and } \; (5b)\text{–}(5e).
\end{aligned}
$$

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

5. Numerical Results

In this section, we present the instances and explore: (i) The efficiency of the multiple-scenario technique and the performance of the proposed C&CG algorithm, compared to existing exact algorithms. We also compare the C&CG algorithm with other variants of facility location problems under disruptions (Section 5.2). (ii) The impact of the LDR on the computational complexity and solution quality (Section 5.3). (iii) The trade-off between the nominal cost and worst-case performance. We enhance our robust formulations with an additional set of constraints to evaluate this trade-off (Section 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., $p = p_i, i \in I$. To represent systems with different penalty costs, we set two values for p. For each instance, we first calculate the transportation costs $d_{ij}, i \in I, j \in J$ and then rank them in nondecreasing order. The two values for p are the maximal value and the $(\lceil 0.8 \times |I| \times |J| \rceil)th$ value in the order. For convenience, we denote these values p^{max} and $p^{0.8}$. 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 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

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.

RAM under Linux 6.3. Each experiment is conducted on a four-core processor of one node. The computational time limit is set to 24 hours, which is in line with other papers considering facility location under uncertainty (48 hours in Ardestani-Jaafari and Delage (2017) and 96 hours in 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 is 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 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 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.

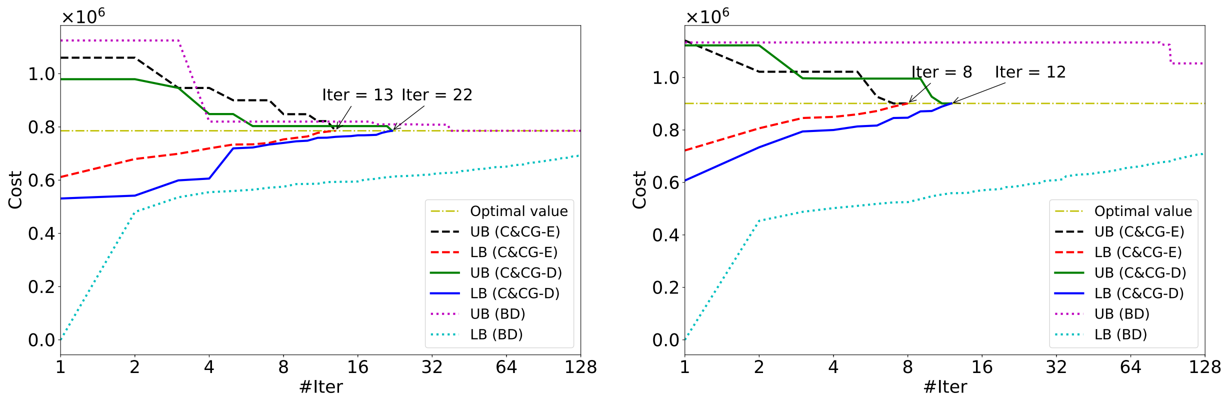
5.2.2. Exact Algorithms for Adjustable Robust UFLP. In the following sections, C&CG-E indicates the proposed C&CG algorithm (where the LP-based enumeration method is applied for the subproblem) and C&CG-D indicates the C&CG algorithm with the dualized subproblem. Table 3 presents the average results.

Table 3 Average results for the adjustable robust UFLP

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

N/A: No further experiments are performed.

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. When $k = 3$, one more instance can be optimally solved by C&CG-E. Moreover, the average gap is smaller, there are fewer iterations, and the CPU time is shorter. For $k = 4$ and $p = p^{max}$, C&CG-D provides a smaller optimality gap while the CPU time and the number of iterations are greater. 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.



(a) Robust UFLP (b) Robust CFLP

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

5.2.3. Exact Algorithms for Adjustable Robust CFLP. We present the summarized results in Table 4. It 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 adjustable 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.

We present the statistical results of the CPU time spent on the master and sub problems in Appendix E. It shows that the main time reduction of the C&CG-E comes from the computing time for the master problem, which indicates that the cuts generated from the second-worst scenarios can help to improve the convergence of the C&CG algorithm. We also apply the C&CG-E algorithm to the adjustable robust uncapacitated/capacitated PMP (UPMP/CPMP) and the detailed results are presented in Appendix F. The results demonstrate that for both the reliable UPMP and CPMP, the C&CG-E shows better performances in terms of the average optimality gap and the number of iterations. On the other hand, from Tables 3 and 4, we observe that the CPU times of both exact algorithms increase quickly with the uncertainty budget. When $k=4$, both algorithms consume around 15 hours on average; therefore, approximation algorithms are expected to be used for cases with a large value of k.

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

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 $k=1,\ldots,4$ and $p=p^{0.8}$, p^{max}, which are solved to optimality by C&CG-E. There are 208 instances for the UFLP and 231 instances for the CFLP. The average results are reported in Table 5.

Table 5 Average results of the linear decision rule for the instances solved to optimality by C&CG-E

Model	p	k	#Opt	CPU time C&CG-E	CPU time LDR	Opt. gap	Model	p	k	#Opt	CPU time C&CG-E	CPU time 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 5 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 the detailed solutions, we find 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 solve the subproblem of the C&CG algorithm to identify the worst-case scenario and its corresponding 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

Table 6 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

system's nominal cost. Table 6 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 6, 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.

5.4.2. An Enhancement for Trade-Off between Reliability and Nominal Cost. Sometimes, decision-makers may want to both reduce the recourse cost under disruption scenarios and control the increase in the nominal cost when robustifying the system. To reflect this, we introduce another group of constraints:

$$
\begin{aligned}
& f_j y_j + \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij}^0 + \sum_{i \in I} p_i h_i u_i^0 \le (1+q) c_0^*, && \\
& \sum_{j \in J} x_{ij}^0 + u_i^0 \ge 1 && \forall i \in I, \\
& x_{ij}^0 \le y_j && \forall i \in I, j \in J, \\
& \sum_{i \in I} h_i x_{ij}^0 \le C_j y_j && \forall j \in J, \\
& x_{ij}^0 \ge 0 && \forall i \in I, j \in J, \\
& u_i^0 \ge 0 && \forall i \in I,
\end{aligned} \tag{15}
$$

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

We study the impact of constraints (15) by varying the value of q. The experiments are conducted on two randomly selected instances, and the results are presented in Table 7, 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 7 Impact of imposing an upper bound on the nominal cost ($k=2, p=p^{0.8}$)

			ARC model with constraints (15)			Deterministic model	Cost gap (%)	
Model	Instance	q	Location decision	Worst-case cost	Nominal cost	Nominal cost	Nominal	Worst-case
UFLP	F10-C30	0.00	[1, 5, 6]	1208972	435528	435528	0.00	39.99
		0.06	[1, 3, 5, 6]	892768	459163		5.15	18.74
		0.08	[1, 5, 6, 8]	759502	466159		6.57	4.48
		0.26	[1, 5, 6, 7]	743641	475435		8.39	2.44
		0.30	[3, 5, 6, 8]	725463	555996		21.67	0.00
	F10-C49	0.00	[1, 5, 6]	1358803	469866	469866	0.00	42.19
		0.06	[1, 3, 5, 6]	957321	491532		4.41	17.94
		0.08	[1, 5, 6, 8]	828318	500497		6.12	5.16
		0.10	[1, 5, 6, 7]	821814	508753		7.64	4.41
		0.12	[1, 3, 5, 6, 8]	816383	522163		10.02	3.77
		0.28	[1, 3, 5, 6, 7]	811487	530419		11.42	3.19
		0.30	[3, 5, 6, 8]	785576	602896		22.07	0.00
CFLP	F10-C30	0.00	[1, 3, 6, 7, 9]	1043678	548704	548704	0.00	19.97
		0.02	[1, 3, 5, 6, 8, 9]	925575	555901		1.29	9.76
		0.04	[1, 3, 5, 7, 8, 9]	898898	569228		3.61	7.09
		0.06	[1, 3, 6, 7, 8, 9]	869826	580916		5.55	3.98
		0.30	[1, 5, 6, 7, 8, 9]	862084	588988		6.84	3.12
		0.32	[3, 4, 5, 6, 7, 8, 9]	835208	718517		23.63	0.00
	F10-C49	0.00	[1, 3, 4, 8, 9]	1141237	571443	571443	0.00	21.03
		0.02	[1, 4, 5, 8, 9]	1052860	581790		1.78	14.40
		0.04	[1, 3, 4, 5, 8, 9]	969084	590197		3.18	7.00
		0.06	[1, 4, 5, 6, 8, 9]	932736	594405		3.86	3.38
		0.30	[1, 3, 4, 5, 6, 8, 9]	901229	612904		6.76	0.00

Table 7 shows that imposing an upper bound on the nominal cost impacts the location decision of the ARC models, i.e., different facilities are chosen or different numbers of sites are opened. We also observe that the bound constraints can help the decision-makers to further control the conservativeness of the robust solutions. For the given instances, sometimes the nominal cost can be significantly decreased with a slight increase in the worst-case cost. For example, for the UFLP with F10-C49, when q changes from 0.30 to 0.08, the increase in the nominal cost drops from 22.07% to 6.12%; however, the worst-case cost increases by only 5.16%. Similarly, for the CFLP with F10-C30, when q changes from 0.32 to 0.30, the increase in the nominal cost drops from 23.63% to 6.84%, and the worst-case cost increases by only 3.12%. Managers can also use the bound constraints as a decision support tool to see the trade-off between reliability and nominal cost, and to decide the extent to which the nominal cost can be controlled when robustifying the system.

6. Conclusions

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

One potential future research direction is to apply the technique in Iancu and Trichakis (2014) to determine Pareto robustly optimal solutions for the two-stage FLP. This technique can be applied to the AARC formulations in case of partial disruptions (i.e., random parameters can be fractional), which can potentially improve the quality of the (heuristic) solution determined by the AARC models.

Acknowledgements

The authors gratefully acknowledge the support of the Institute for Data Valorization (IVADO), the Interuniversity Research Center on Enterprise Networks, Logistics and Transportation (CIRRELT), and the Natural Sciences and Engineering Research Council of Canada (Grant 2016-05822). We thank Calcul Québec for providing computing facilities for the experiments, and we are grateful to Erick Delage for valuable comments on this paper.

References

An Y, Zeng B (2014) Exploring the modeling capacity of two-stage robust optimization: Variants of robust unit commitment model. *IEEE transactions on Power Systems* 30(1):109–122.

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.

Ayoub J, Poss M (2016) Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science* 13(2):219–239.

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.

Bertsimas D, Shtern S (2018) A scalable algorithm for two-stage adaptive linear optimization. *Working paper,* `https://arxiv.org/abs/1807.02812`.

Billionnet A, Costa MC, Poirion PL (2014) 2-Stage Robust MILP with continuous recourse variables. *Discrete Applied Mathematics* 170:21–32.

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.

Cheng C, Qi M, Zhang Y, Rousseau LM (2018) A two-stage robust approach for the reliable logistics network design problem. *Transportation Research Part B: Methodological* 111:185–202.

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.

Iancu DA, Trichakis N (2014) Pareto efficiency in robust optimization. *Management Science* 60(1):130–147.

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)).

Király Z, Kovács P (2012) Efficient implementations of minimum-cost flow algorithms. *arXiv preprint arXiv:1207.6381* .

Liberatore F, Scaparra MP, Daskin MS (2012) Hedging against disruptions with ripple effects in location analysis. *Omega* 40(1):21–30.

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`.

Peng P, Snyder LV, Lim A, Liu Z (2011) Reliable logistics networks design with facility disruptions. *Transportation Research Part B: Methodological* 45(8):1190–1211.

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 (2019a) Designing response supply chain against bioattacks. *Operations Research* 67(5):1246–1268.

Simchi-Levi D, Wang H, Wei Y (2019b) Constraint generation for two-stage robust network flow problems. *INFORMS Journal on Optimization* 1(1):49–70.

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 S, Ouyang Y (2019) Reliable service systems design under the risk of network access failures. *Transportation Research Part E: Logistics and Transportation Review* 122:1–13.

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.

Appendix A Proofs of Lemmas

A.1 Proof of Lemma 1

Suppose the number of opened facilities is m, i.e., $\sum_{j\in J}\hat{y}_j = m$. Since the worst-case scenario always occurs at the extreme points, there will be exactly k disrupted facilities in a worst-case scenario. Figure A1 gives an example where $m > k$. We index the facilities so that the first m are opened 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 opened 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 $B_1 = B_2$.

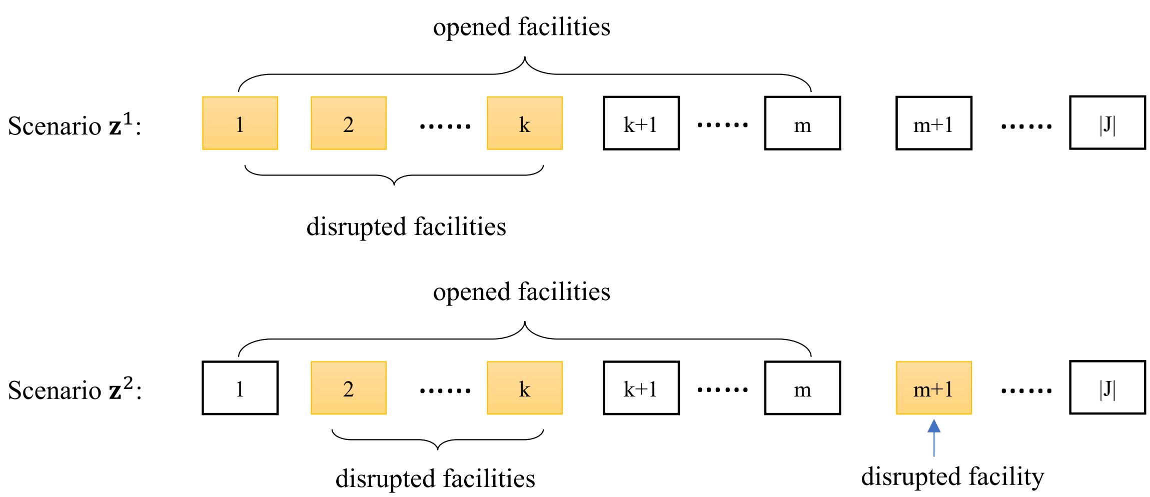


Figure A1 Illustration for the proof of Lemma 1

A.2 Proof of Lemma 3

To prove $\mathbb{Z}'(k)$ is equivalent to $\mathbb{Z}(k)$, we need to show that even though we relax the variables $\mathbf{z}$ to be continuous, the optimal values of $\mathbf{z}$ in a worst-case scenario still take integer values (0 or 1). This can be proved through the subproblems of the duality-based C&CG algorithms (refer to B.1 and C.1). Specifically, for the UFLP, we can rewrite the objective function (B1) as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{j\in J}\left(\sum_{i\in I}\hat{y}_j\beta_{ij}(1-z_j)\right). \tag{A1}$$

Constraints (B2)–(B7) indicate that variables $\mathbf{z}$ are not linked to other variables (i.e., $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$). Since the subproblem is a maximization problem, the third term subtracted from Equation (A1) is expected to be as small as possible. To achieve this, we can rank $\sum_{i\in I}\hat{y}_j\beta_{ij}(\geq 0)$ for each j, and for the first k largest values, z_j would take the value of 1 (even though z_j is relaxed to be continuous) at optimum. For other values, z_j would be 0. Similarly, for the CFLP, we can rewrite the objective function (C1) as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{j\in J}\left(\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j(1-z_j)\right). \tag{A2}$$

If $\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j \leq 0$, z_j would be 0 at optimum. For those with $\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j > 0$, the same method describe above for the UFLP applies.

Appendix B Duality-Based C&CG Algorithm and Benders Decomposition Method for Adjustable Robust UFLP

B.1 Duality-Based C&CG Algorithm

Master Problem. The master problem is defined by Equations (9a)–(9g).

Subproblem. Let λ and β be the dual variables of constraints (5b) and (5c), respectively. The dual problem of the inner minimization problem can be written as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(1-z_j)\beta_{ij}, \tag{B1}$$

$$\text{s.t.}\quad \lambda_i - \beta_{ij} \leq h_i d_{ij} \qquad \forall i\in I, j\in J, \tag{B2}$$

$$\lambda_i \leq p_i h_i \qquad \forall i\in I, \tag{B3}$$

$$\sum_{j\in J} z_j \leq k, \tag{B4}$$

$$z_j \in \{0,1\} \qquad \forall i\in I, \tag{B5}$$

$$\lambda_i \geq 0 \qquad \forall i\in I, \tag{B6}$$

$$\beta_{ij} \geq 0 \qquad \forall i\in I, j\in J. \tag{B7}$$

Since the nonlinear term $z_j\beta_{ij}$ is the product of a binary variable z_j and a continuous variable β_{ij}, we can linearize it by introducing a new variable $\pi_{ij} = z_j\beta_{ij}$ and adding the following constraints:

$$\begin{aligned} &\pi_{ij} \geq 0 && \forall i \in I, j \in J, \\ &\pi_{ij} \leq \beta_{ij} && \forall i \in I, j \in J, \\ &\pi_{ij} \leq \mathbb{M}_{ij} z_j && \forall i \in I, j \in J, \\ &\pi_{ij} \geq \beta_{ij} + \mathbb{M}_{ij}(z_j - 1) && \forall i \in I, j \in J, \end{aligned} \tag{B8}$$

where $\mathbb{M}_{ij} = \max\{h_i(p'_i - d_{ij}), 0\}$. Therefore, the full subproblem is

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\pi}} \sum_{j\in J} f_j \hat{y}_j + \sum_{i\in I} \lambda_i - \sum_{i\in I}\sum_{j\in J} \hat{y}_j(\beta_{ij} - \pi_{ij}), \tag{B9}$$

subject to constraints (B2)–(B8).

B.2 Benders Decomposition Method

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

$$\begin{aligned} \phi = &\min_{\mathbf{y},s} s, \\ \text{s.t.}\quad & s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I} \hat{\lambda}_i^l - \sum_{i\in I}\sum_{j\in J} \hat{\beta}_{ij}^l y_j (1 - \hat{z}_j^l) \qquad \forall l \in \{1, ..., n\}, \\ & y_j \in \{0,1\} \qquad \forall j \in J, \end{aligned} \tag{B10}$$

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 (B2)–(B9).

Appendix C Duality-Based C&CG Algorithm and Benders Decomposition Method for Adjustable Robust CFLP

C.1 Duality-Based C&CG Algorithm

Master problem. The master problem is defined by Equations (9a)–(9g) and (11).

Subproblem. Let γ be the dual variable of constraints (6). The resulting dual problem can be written as follows:

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}} \sum_{j\in J} f_j \hat{y}_j + \sum_{i\in I} \lambda_i - \sum_{i\in I}\sum_{j\in J} \hat{y}_j (1-z_j)\beta_{ij} - \sum_{j\in J} C_j \hat{y}_j \gamma_j (1-z_j), \tag{C1}$$

$$\begin{aligned} \text{s.t.}\quad & \lambda_i - \beta_{ij} - h_i\gamma_j \le h_i d_{ij} && \forall i\in I, j\in J, \\ & \gamma_j \ge 0 && \forall j \in J, \\ & \text{and (B3)–(B7).} \end{aligned} \tag{C2}$$

There are two nonlinear terms in the objective function (C1), i.e., $z_j\beta_{ij}$ and $\gamma_j z_j$. We can use the technique in B.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:

$$\begin{aligned} & \zeta_j \ge 0 && \forall j\in J, \\ & \zeta_j \le \gamma_j && \forall j\in J, \\ & \zeta_j \le \mathbb{M}_j z_j && \forall j\in J, \\ & \zeta_j \ge \gamma_j + \mathbb{M}_j (z_j - 1) && \forall j\in J, \end{aligned} \tag{C3}$$

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

Therefore, the full subproblem is

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\zeta}} \sum_{j\in J} f_j \hat{y}_j + \sum_{i\in I} \lambda_i - \sum_{i\in I}\sum_{j\in J} \hat{y}_j (\beta_{ij} - \pi_{ij}) - \sum_{i\in J} C_j \hat{y}_j (\gamma_j - \zeta_j), \tag{C4}$$

subject to constraints (B8) and (C2)–(C3).

C.2 Benders Decomposition Method

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

$$\begin{aligned} \phi = & \min_{\mathbf{y},s} s, \\ \text{s.t.}\quad & s \ge \sum_{j\in J} f_j y_j + \sum_{i\in I} \hat{\lambda}_i^l - \sum_{i\in I}\sum_{j\in J} y_j (1-\hat{z}_j^l)\hat{\beta}_{ij}^l - \sum_{j\in J} C_j y_j \hat{\gamma}_j^l (1-\hat{z}_j^l) && \forall l \in \{1,...,n\}, \\ & y_j \in \{0,1\} && \forall j\in J, \end{aligned} \tag{C5}$$

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 (B8) and (C2)–(C4).

Appendix D Reformulation of the AARC Models

The AARC model for the robust UFLP can be reformulated as

$$
\begin{aligned}
\min_{\substack{\mathbf{y},\mathbf{W},\mathbf{w},\mathbf{A},\mathbf{a},s,\delta,\\ \boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\theta},\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\varsigma},\boldsymbol{\pi},\boldsymbol{\nu}}} \quad & s, \\
\text{s.t.} \quad & s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} w_{ij} + \sum_{i\in I} p_i h_i a_i + k\delta + \sum_{e\in J}\alpha_e, \\
& \delta + \alpha_e \geq \sum_{i\in I}\sum_{j\in J} h_i d_{ij} W_{ije} + \sum_{i\in I} p_i h_i A_{ie} && \forall e\in J, \\
& \sum_{j\in J} w_{ij} + a_i - k\xi_i - \sum_{e\in J}\eta_{ie} \geq 1 && \forall i\in I, \\
& \xi_i + \eta_{ie} \geq -\sum_{j\in J} W_{ije} - A_{ie} && \forall i\in I, e\in J, \\
& -k\theta_{ij} - \sum_{e\in J}\mu_{ije} \geq -y_j + w_{ij} && \forall i\in I, j\in J, \\
& \theta_{ij} + \mu_{ije} \geq W_{ije} + y_j && \forall i\in I, j\in J, e\in J, e=j, \\
& \theta_{ij} + \mu_{ije} \geq W_{ije} && \forall i\in I, j\in J, e\in J, e\neq j, \\
& w_{ij} - k\sigma_{ij} - \sum_{e\in J}\varsigma_{ije} \geq 0 && \forall i\in I, j\in J, \\
& \sigma_{ij} + \varsigma_{ije} \geq -W_{ije} && \forall i\in I, j\in J, e\in J, \\
& -k\pi_i - \sum_{e\in J}\nu_{ie} + a_i \geq 0 && \forall i\in I, \\
& \pi_i + \nu_{ie} \geq -A_{ie} && \forall i\in I, e\in J, \\
& y_j \in \{0,1\} && \forall j\in J, \\
& \delta, \alpha_e, \xi_i, \eta_{ie}, \theta_{ij}, \mu_{ije}, \sigma_{ij}, \varsigma_{ije}, \pi_i, \nu_{ie} \geq 0 && \forall i\in I, j\in J, e\in J.
\end{aligned}
\tag{D1}
$$

The AARC model for the robust CFLP can be reformulated with (D1) and the constraints

$$
\begin{aligned}
& k\rho_j + \sum_{e\in J}\Gamma_{ej} + \sum_{i\in I} h_i w_{ij} \leq C_i y_j && \forall j\in J, \\
& \rho_j + \Gamma_{ej} \geq \sum_{i\in I} h_i W_{ije} + C_j y_j && \forall e\in J, j\in J, e=j \\
& \rho_j + \Gamma_{ej} \geq \sum_{i\in I} h_i W_{ije} && \forall e\in J, j\in J, e\neq j \\
& \rho_j, \Gamma_{ej} \geq 0 && \forall e\in J, j\in J.
\end{aligned}
\tag{D2}
$$

Appendix E Statistics of CPU Time for Master and Sub Problems

Table E1 presents the statistical results of the CPU time spent on the master and sub problems respectively. The *time reduction factor* is calculated as the time spent by the C&CG-D algorithm divided by the time spent by the C&CG-E algorithm. A time factor equals to 2 indicates that the problem spends only half the time using the C&CG-E compared to using the C&CG-D. Table E1 shows that the main time reduction comes from the computing time for the master problem, which indicates that the cuts generated from the second-worst scenarios can improve the convergence of the C&CG algorithm. When $k=3$ and $k=4$, the time factor of the master problem is only a bit larger than 1, this is because some instances are not solved to optimality and the time limit is reached. We also notice that the solution process of the subproblem of the C&CG-E consumes little time although we need to solve a large number of MCFPs. This process is efficient because we use the network simplex algorithm (via the NetworkX package) which works well for the MCFP model (Király and Kovács 2012).

Table E1 Statistics of average CPU time for master and sub problems

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

Appendix F Results of Adjustable Robust UPMP and CPMP

In this section, we apply the C&CG-E to the adjustable robust uncapacitated/capacitated PMP (UPMP/CPMP). We first introduce the models and then report the numerical results.

F.1 The Adjustable Robust UPMP and CPMP Models

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 as follows.

Notation. The parameter ϱ is the weight of the worst-case cost. The variable b_{ij} is the fraction of demand from customer $i \in I$ that is satisfied by facility $j \in J$ in a disruptive scenario. The variable g_i is the unsatisfied portion of demand at customer $i \in I$ in a disruptive scenario. The definitions of the other parameters and variables are the same as in Section 3.

The UPMP under disruptions can be formulated as

$$\min_{\mathbf{x},\mathbf{y}}(1-\varrho)\sum_{i\in I}\sum_{j\in J} d_{ij}h_i x_{ij} + \varrho \max_{\mathbf{z}\in\mathbb{Z}(k)} \min_{(\mathbf{b},\mathbf{g})\in S(\mathbf{y},\mathbf{z})} (\sum_{i\in I}\sum_{j\in J} d_{ij}h_i b_{ij} + \sum_{i\in I} p_i h_i g_i), \quad \text{(F1)}$$

$$\begin{aligned} \text{s.t.}\quad & x_{ij} \le y_j && \forall i \in I, j \in J, \\ & \sum_{j\in J} x_{ij} = 1 && \forall i \in I, \\ & \sum_{j\in J} y_j = p, && && \text{(F2)} \\ & x_{ij} \ge 0, && \forall i \in I, j \in J, \\ & y_j \in \{0,1\} && \forall j \in J, \end{aligned}$$

$$\begin{aligned} \text{where } S(\mathbf{y},\mathbf{z}) = \{ & b_{ij} \le 1 - z_j && \forall i \in I, j \in J, \\ & b_{ij} \le y_j && \forall i \in I, j \in J, \\ & \sum_{j\in J} b_{ij} + g_i = 1 && \forall i \in I, && \text{(F3)} \\ & b_{ij} \ge 0, && \forall i \in I, j \in J, \\ & g_i \ge 0, && \forall i \in I \} \end{aligned}$$

The CPMP under disruptions is

$$\min_{\mathbf{x},\mathbf{y}}(1-\varrho)\sum_{i\in I}\sum_{j\in J} d_{ij}h_i x_{ij} + \varrho \max_{\mathbf{z}\in\mathbb{Z}(k)} \min_{(\mathbf{b},\mathbf{g})\in S^C(\mathbf{y},\mathbf{z})} (\sum_{i\in I}\sum_{j\in J} d_{ij}h_i b_{ij} + \sum_{i\in I} p_i h_i g_i) \quad \text{(F4)}$$

$$\text{s.t.}\quad \text{(F2) and } \sum_{i\in I} h_i x_{ij} \le C_j y_j, \quad \forall j \in J, \quad \text{(F5)}$$

$$\text{where}\quad S^C(\mathbf{y},\mathbf{z}) = \{\text{(F3) and } \sum_{i\in I} h_i b_{ij} \le C_j y_j, \quad \forall j \in J\}. \quad \text{(F6)}$$

F.2 Numerical Results of Adjustable Robust UPMP and CPMP

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, the optimality tolerance ($\epsilon = 0.1\%$), and the time limit (2 hours, and we allow the last iteration before reaching the time limit to terminate) take the same values as in An et al. (2014). The results for the two models are given in Tables F2 and F3, where $J = I$ and ϱ is the weight of the worst-case cost.

Table F2 Results for uncapacitated p-median problem under disruptions

				Unit penalty cost = 15						Unit penalty cost = p^{max}					
				C&CG-E			C&CG-D in An et al. (2014)			C&CG-E			C&CG-D in 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	1	**0.3**	0.0	4	2.6	0.0	1	**0.3**	0.0	4	2.0
			2	0.0	5	3.7	0.0	5	**2.8**	0.0	5	7.6	0.0	5	**3.2**
			3	0.0	21	**58.8**	0.0	58	578.2	0.0	35	**322.5**	0.0	70	1364.8
		10	1	0.0	1	**0.2**	0.0	1	0.6	0.0	1	**0.4**	0.0	1	0.6
			2	0.1	3	**2.4**	0.0	5	3.8	0.1	3	**2.9**	0.0	5	3.3
			3	0.0	5	**6.4**	0.0	9	10.4	0.0	7	15.0	0.0	11	**12.1**
	0.4	8	1	0.0	3	**1.0**	0.0	4	2.0	0.0	3	3.7	0.0	4	**2.2**
			2	0.0	8	**7.3**	0.0	13	12.7	0.0	8	**9.7**	0.0	13	15.7
			3	0.0	46	**492.5**	0.0	96	1718.8	**0.0**	64	**4159.7**	6.9	92	7359.1
		10	1	0.0	1	**0.3**	0.0	1	0.5	0.0	1	1.0	0.0	1	**0.6**
			2	0.0	4	**3.3**	0.0	6	3.5	0.0	4	16.4	0.0	6	**4.4**
			3	0.0	13	**27.2**	0.0	22	36.3	0.0	14	**55.6**	0.0	25	59.5
49	0.2	8	1	0.0	1	**1.2**	0.0	3	4.9	0.0	1	**3.8**	0.0	3	9.1
			2	0.0	6	**22.4**	0.0	9	29.0	**0.0**	27	**4604.8**	5.0	44	7429.3
			3	**1.6**	38	7618.4	3.6	61	7426.4	**9.0**	33	7544.5	10.2	43	7661.3
		10	1	0.0	1	**0.9**	0.0	1	1.5	0.0	1	**1.4**	0.0	1	2.2
			2	0.0	6	**36.3**	0.0	11	71.9	0.0	6	**58.9**	0.0	11	110.0
			3	0.0	27	**4280.2**	0.0	40	7241.1	**7.1**	28	7223.8	12.1	52	7272.8
	0.4	8	1	0.0	2	**4.1**	0.0	3	4.8	0.0	2	**5.5**	0.0	3	11.2
			2	0.0	17	**747.0**	0.0	29	1334.7	**8.5**	24	7661.8	17.5	46	7458.9
			3	**11.0**	32	7227.2	15.6	44	7821.3	**21.5**	27	7507.7	23.1	33	7834.5
		10	1	0.0	2	**3.8**	0.0	3	5.0	0.0	2	**4.8**	0.0	3	5.1
			2	0.0	13	**683.0**	0.0	30	1468.7	0.0	13	**788.0**	0.0	30	3069.0
			3	11.7	24	7355.0	**10.4**	39	7512.4	**15.9**	22	8057.3	18.4	44	7753.2
Average				**1.0**$^{(3)}$	11.7	1190.9	1.2$^{(3)}$	20.7	1470.6	**2.6**$^{(5)}$	13.8	2002.4	3.9$^{(7)}$	22.9	2393.5

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

Tables F2 and F3 show that for both the UPMP and CPMP, C&CG-E has better performance, in terms of the average gap and the number of iterations. In particular, for the UPMP, when the unit penalty cost is p^{max}, two more instances can be solved to optimality by C&CG-E with less than 4700 seconds; however, C&CG-D produces solutions with optimality gaps over 5.0% when reaching the time limit. For the UPMP, the average CPU time of C&CG-E is also lower. For the CPMP, the average CPU time of the two algorithms is quite close, while C&CG-E provides better optimality gaps. From both tables, we observe that C&CG-E generally works better for instances with a large budget. Take $|J| = 25, \varrho = 0.4, p = 8, k = 3$ and unit penalty cost = 15 for example, for

Table F3 Results for capacitated p-median problem under disruptions

				Unit penalty cost = 15						Unit penalty cost = p^{max}					
				C&CG-E			C&CG-D in An et al. (2014)			C&CG-E			C&CG-D in 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	**1.5**	0.0	3	3.4	0.0	2	3.5	0.0	3	**3.0**
			2	0.0	5	**7.2**	0.0	7	9.4	0.0	10	**51.5**	0.0	20	104.1
			3	0.0	25	**236.5**	0.0	41	354.4	**15.4**	43	7521.6	19.8	72	7296.4
		10	1	0.0	1	**0.4**	0.0	1	0.5	0.0	1	**0.3**	0.0	1	0.5
			2	0.0	2	**2.2**	0.0	3	4.0	0.0	3	5.4	0.0	4	**4.1**
			3	0.0	7	**17.9**	0.0	12	23.2	0.0	13	111.4	0.0	19	**82.9**
	0.4	8	1	0.0	3	**3.2**	0.0	4	4.3	0.0	3	**3.4**	0.0	4	3.9
			2	0.0	8	**19.2**	0.0	15	39.8	0.0	12	**99.0**	0.0	23	168.0
			3	0.0	37	**1051.7**	0.0	74	2451.2	**28.4**	43	7352.8	36.8	63	7272.3
		10	1	0.0	1	0.8	0.0	1	**0.6**	0.0	1	**0.4**	0.0	1	0.6
			2	0.0	4	**5.5**	0.0	10	13.8	0.0	5	**9.8**	0.0	10	17.7
			3	0.0	12	**51.9**	0.0	21	63.7	0.0	21	**398.2**	0.0	35	460.6
49	0.2	8	1	0.0	1	**1.5**	0.0	3	7.5	0.0	2	**6.2**	0.0	3	9.3
			2	0.0	10	**310.5**	0.0	15	414.4	**5.8**	22	7819.2	12.4	37	7370.9
			3	**9.0**	25	7884.2	9.8	41	7676.8	24.4	16	8390.7	**24.3**	25	7396.5
		10	1	0.0	1	**1.9**	0.0	2	5.1	0.0	1	**2.1**	0.0	2	4.9
			2	0.0	8	**213.0**	0.0	21	840.2	0.0	11	**1121.3**	0.0	21	1307.6
			3	**2.2**	23	7258.5	3.0	35	8083.0	14.6	17	7292.3	**14.0**	27	9613.7
	0.4	8	1	0.0	2	8.0	0.0	3	**6.2**	0.0	2	**7.0**	0.0	3	9.6
			2	**0.7**	26	7815.3	9.5	35	7858.8	**18.3**	18	8751.8	27.0	29	7295.0
			3	**20.5**	18	8668.8	26.3	30	7347.6	**44.1**	15	7852.8	46.4	23	7379.5
		10	1	0.0	3	17.2	0.0	4	**11.5**	0.0	3	16.5	0.0	4	**14.1**
			2	**4.6**	21	8289.5	4.7	34	7257.8	10.9	19	7993.0	**6.8**	36	8071.1
			3	16.2	22	8742.2	**14.6**	31	7343.0	**28.7**	17	8477.9	29.9	27	7410.8
Average				**2.2**$^{(6)}$	11.1	2108.7	2.8$^{(6)}$	18.6	2075.8	**7.9**$^{(9)}$	12.5	3053.7	9.1$^{(9)}$	20.5	2970.7

the UPMP, C&CG-E consumes 492.5 seconds while C&CG-D takes 1718.8 seconds. Similarly, for the CPMP, the computing time is 1051.7 seconds versus 2451.2 seconds.