INFORMS Journal on Computing
Vol. 23, No. 3, Summer 2011, pp. 470–482
ISSN 1091-9856 | EISSN 1526-5528 | 11 | 2303 | 0470

DOI 10.1287/ijoc.1100.0414

The Reliable Facility Location Problem: Formulations, Heuristics, and Approximation Algorithms

Zuo-Jun Max Shen
Department of Industrial Engineering and Operations Research, University of California, Berkeley, California 94720, shen@ieor.berkeley.edu

Roger Lezhou Zhan
Operations Research Group, CitationAir, Greenwich, Connecticut 06831, zhan@alumni.ufl.edu

Jiawei Zhang
Department of Information, Operations, and Management Sciences, Stern School of Business, New York University, New York 10012, jzhang@stern.nyu.edu

We study a reliable facility location problem wherein some facilities are subject to failure from time to time. If a facility fails, customers originally assigned to it have to be reassigned to other (operational) facilities. We formulate this problem as a two-stage stochastic program and then as a nonlinear integer program. Several heuristics that can produce near-optimal solutions are proposed for this NP-hard problem. For the special case where the probability that a facility fails is a constant (independent of the facility), we provide an approximation algorithm with a worst-case bound of 4. The effectiveness of our heuristics is tested by extensive computational studies, which also lead to some managerial insights.

Key words: reliable facility location problem; uncertainty; heuristics; approximation algorithm
History: Accepted by Karen Aardal, Area Editor for Design and Analysis of Algorithms; received July 2007; revised May 2009, December 2009; accepted July 2010. Published online in *Articles in Advance* December 2, 2010.

1. Introduction

Facility location models have been extensively studied in the literature. Different kinds of facilities have been modeled, such as routers or servers in a communication network, warehouses or distribution centers in a supply chain, and hospitals or airports in a public service system. Facility location models typically try to determine where to locate the facilities among a set of candidate sites and how to assign "customers" to the facilities, so that the total cost can be minimized or the total profit can be maximized (e.g., Daskin 1995, Mirchandani and Francis 1990, Shen 2005). Most models in the literature have treated facilities as if they would never fail; in other words, they are completely reliable. We will relax this assumption in this paper.

One of the most studied location models is the so-called uncapacitated facility location problem (UFLP). In the UFLP, we are given a set of demand points, a set of candidate sites, the cost of opening a facility at each location, and the cost of connecting each demand point to any facility. The objective is to open a set of facilities from the candidate sites and assign each demand point to an open facility so as to minimize the total facility opening and connection costs.

The UFLP and its generalizations are NP-hard; i.e., unless $P = NP$, they do not admit polynomial-time algorithms to find an optimal solution. There is a vast literature on these NP-hard facility location problems, and many solution approaches (integer programming, metaheuristics, approximation algorithms, etc.) have been developed in the last four decades. One common assumption in this literature is that the input parameters of the problems (costs, demands, facility capacities, etc.) are deterministic. However, such assumptions may not be valid in many realistic situations because many input parameters in the model are uncertain during the decision-making process.

The uncertainties can be generally classified into three categories: provider-side uncertainty, receiver-side uncertainty, and in-between uncertainty. The provider-side uncertainty may capture the randomness in facility capacity and the reliability of facilities, etc.; the receiver-side uncertainty can be the randomness in demands; and the in-between uncertainty may be represented by the random travel time, transportation cost, etc. Most stochastic facility location models focus on the receiver-side and in-between uncertainties (see Snyder 2006, Swamy and Shmoys 2006, and the references therein). The common feature of the receiver-side and in-between uncertainties is that the uncertainty does not change the topology of the provider–receiver network once the facilities have been built. However, this is not the case if the built facilities are subject to failure (provider-side uncertainty). If a facility fails, customers originally assigned to it have to be reassigned to other (operational) facilities, and thus the connection cost changes (usually increases).

We focus on the reliability issue of provider-side uncertainty in this paper. The uncertainty is modeled using two different approaches: (1) by a set of scenarios that specify which subset of the facilities will become nonoperational or (2) by an individual and independent failure probability inherent in each facility. Although each demand point needs to be served by one operational facility only, it should be assigned to a group of facilities that are ordered by levels: in the event of the lowest-level facility becoming nonoperational, the demand can then be served by the next level facility that is operational; and so on. If all operational facilities are too far away from a demand point, one may choose not to serve this demand point by paying a penalty cost. The objective is thus to minimize the facility opening cost plus the *expected* connection and penalty costs. This problem will be referred to as the uncapacitated reliable facility location problem (URFLP).

The URFLP is clearly NP-hard because it generalizes the UFLP. We propose several heuristics to solve the URFLP. We also provide a 4-approximation algorithm for a special case where the failure probability of a facility is independent of the facility. Designing approximation algorithms for the UFLP and its variations has recently received considerable attention from the research community. However, to the best of our knowledge, our paper presents the first approximation algorithm for stochastic facility location problems with provider-side uncertainty.

The rest of this paper is organized as follows. In §2, we review the related literature and provide some basic background for our model. In §3, we present two formulations for the model and discuss their properties. An approximation algorithm for a special case of the model is presented in §4. In §5, we discuss and evaluate different heuristics. In §6, we conclude the paper by suggesting several future research directions.

2. Literature Review

The importance of uncertainty in decision making has prompted a number of researchers to address stochastic facility location models (e.g., Owen and Daskin 1998, Snyder 2006, Swamy and Shmoys 2006). However, as we pointed out in the §1, a majority of the current literature mainly deals with the receiver-side and in-between uncertainties. This includes Daskin et al. (1997), França and Luna (1982), Santoso et al. (2005), and Zeng and Ward (2005), among others.

The reliability facility location model was first studied by Snyder and Daskin (2005), where the authors assume that some facilities are perfectly reliable while others are subject to failure with the *same* probability. They formulate their problem as a linear integer program and propose a Lagrangian relaxation solution method. No approximation algorithm has been proposed in Snyder and Daskin (2005). Another related model is proposed in Berman et al. (2007), which is based on the p-median problem rather than the UFLP. A recent paper by Cui et al. (2010) relaxes the uniform failure probability assumption in Snyder and Daskin (2005) and allows the failure probabilities to be facility-specific. The authors propose a compact linear mixed-integer program formulation and a continuum approximation formulation. The continuum approximation model predicts the total system cost without details about facility locations and customer assignments, and it provides a fast heuristic to find near-optimum solutions. Their computational results show that for large-scale problems, the continuum approximation method is very effective, and it avoids prohibitively long running times.

There is also a strand of literature addressing the fortification of reliability for existing facilities (e.g., Scaparra and Church 2005, 2008; Snyder et al. 2006). Their main focuses are to identify the *existing* critical facilities to protect under the events of disruption.

Our paper is also related to the literature on approximation algorithms for facility location problems (e.g., Shmoys et al. 1997, Guha et al. 2003, Jain et al. 2003, Mahdian et al. 2006, Byrka 2007, Swamy and Shmoys 2008). However, until very recently, these approximation algorithms mainly dealt with deterministic problems. Approximation algorithms for the UFLP with stochastic demand have recently been proposed; see the survey by Swamy and Shmoys (2006) and the references therein. Another related paper, Gabor and van Ommeren (2006), proposes an approximation algorithm for a facility location problem with stochastic demand and inventory. Our approximation algorithm makes use of the ideas from several papers, such as Shmoys et al. (1997), Guha et al. (2003), Jain et al. (2003), Mahdian et al. (2006), and Swamy and Shmoys (2008). In particular, our paper is closely related to Guha et al. (2003) and Swamy and Shmoys (2008), where approximation algorithms for the so-called fault-tolerant facility location problem (FTFLP) are presented. In FTFLP, every demand point must be served by several facilities where the number is specified, and a weighted linear combination is used to compute the connection costs. The FTFLP has been motivated by the reliability issue considered in this paper, but the failure probabilities of facilities are not explicitly modeled and no penalty cost is considered. In Swamy and Shmoys (2008), the authors show how FTFLP arises naturally in a setting with uniform failure probabilities if one insists on a certain "quality-of-service requirement" for each client. As for the worst-case bound of approximation algorithm, a 2.41-approximation algorithm is presented in Guha et al. (2003), and an improved approximation algorithm for (unweighted) FTFLP is shown in Swamy and Shmoys (2008).

In our model, the failure probabilities are facility-specific, which significantly complicates the problem when formulating it as a mathematical program. We propose two different models: a scenario-based stochastic programming (SP) model and a nonlinear integer programming (NIP) model. The scenario-based model is attractive because of its structural simplicity and its ability to model dependence among random parameters, but the model becomes computationally expensive as the number of scenarios increases. If the number of scenarios is too large, the nonlinear integer programming-based approach provides an alterative way to tackle the problem. We also consider the special case when the failure probabilities are not facility-specific. Notice that even under this assumption, our model still differs from that in Snyder and Daskin (2005) as we consider penalty cost. Furthermore, we also propose and analyze an approximation algorithm with constant worst-case bound guarantee for the model.

3. Formulations

We first introduce some common notation that will be used throughout the paper. Let D denote the set of clients or demand points, and let F denote the set of facilities. $|F|$ is the number of the facilities. Let f_i be the facility cost to open facility i, let d_j be the demand of client j, and let c_{ij} be the service cost if j is serviced by facility i. The service costs, c_{ij}, are assumed to form a metric; i.e., they satisfy triangle inequalities. For each client $j \in D$ that is not served by any open and operational facility, a per-unit demand penalty cost of r_j is incurred.

3.1. Scenario-Based Model

We first discuss a scenario-based approach to model the URFLP. Given a finite set of scenarios, where each scenario specifies the set of operational facilities, we can formulate the URFLP as a two-stage stochastic program with recourse. The first-stage decisions consist of determining which facilities to open before knowing which facilities will be operational. When the uncertainty is resolved, each client (demand point) is either assigned to an operational facility or one incurs a penalty for the client; these constitute the second-stage decisions. In this model, we are not allowed to build new facilities in the second stage. In other words, no remedy can be made to the first-stage decision except optimally assigning the clients to the operational facilities. The objective is to minimize the total expected cost, which includes the first-stage cost and the expected second-stage cost. The expectation is taken over all scenarios according to a specific distribution.

Let $\mathcal{S}$ be the set of scenarios. For any $A \in \mathcal{S}$, let p_A be the probability that scenario A happens. Then the URFLP can be formulated as the following two-stage stochastic program:

(URFLP-2STG)

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in F} f_i y_i + \sum_{A\in\mathcal{S}} p_A g_A(y) \\ \text{subject to} \quad & y_i \in \{0,1\}, \end{aligned} \tag{1}$$

where

$$g_A(y) = \min \sum_{j\in D}\sum_{i\in F} d_j c_{ij} x_{ij}^A + \sum_{j\in D} d_j r_j z_j^A \tag{2}$$

$$\text{s.t.} \quad \sum_{i\in F} x_{ij}^A + z_j^A = 1 \quad \forall\, j \in D, \tag{3}$$

$$x_{ij}^A \le y_i \quad \forall\, i \in F,\ j \in D, \tag{4}$$

$$x_{ij}^A \le I_{A,i} \quad \forall\, i \in F,\ j \in D, \tag{5}$$

$$x_{ij}^A,\ z_j^A \in \{0,1\}. \tag{6}$$

In the above formulation, the binary variable y_i indicates whether facility i is opened in the first stage. Parameter $I_{A,i}$ indicates whether facility i is operational under scenario A, which is an input regardless of the value of y_i. Variable x_{ij}^A is the assignment variable that indicates whether client j is assigned to facility i in scenario A or not. Finally, variable z_j^A indicates whether client j receives service at all or is subject to penalty. The objective in the formulation (i.e., (1)) is to minimize the sum of the first-stage cost of opening facilities and the expected second-stage cost. The objective of the second stage (i.e., (2)) is to minimize the total service and penalty cost. Constraints (3) ensure that client j is either assigned to a facility or subject to a penalty in scenario A. Constraints (4) and (5) make sure that no client is assigned to an unopen facility or to a nonfunctional facility, respectively.

It is straightforward to see that the formulation (URFLP-2STG) is equivalent to the following mathematical program:

(URFLP-SP)

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in F} f_i y_i + \sum_{A\subseteq\mathcal{S}} p_A \left(\sum_{j\in D}\sum_{i\in F} d_j c_{ij} x_{ij}^A + \sum_{j\in D} d_j r_j z_j^A \right) \\ \text{s.t.} \quad & \sum_{i\in F} x_{ij}^A + z_j^A = 1 \quad \forall\, j\in D,\ A\subseteq\mathcal{S}, \\ & x_{ij}^A \le y_i I_{A,i} \quad \forall\, i\in F,\ j\in D,\ A\subseteq\mathcal{S}, \\ & y_i,\ x_{ij}^A,\ z_j^A \in \{0,1\}. \end{aligned}$$

One advantage of the scenario-based formulation is that it can easily capture the dependence of the failure probabilities of different facilities by properly defining the scenarios. If the number of scenarios is not too large, it is possible to solve the (URFLP-SP) using commercial software. However, when the failure probabilities are independent, the possible number of scenarios can be extremely large. Therefore, the number of variables and constraints in the (URFLP-SP) is exponentially large accordingly, which makes it extremely difficult to solve. Under this situation, we propose an alternative nonlinear integer programming formulation and an efficient solution algorithm. We discuss this alternative formulation in the next subsection.

3.2. Nonlinear Integer Programming Model

In this subsection, we assume that the failure probabilities of the facilities are independent. Let p_i denote the probability that facility i fails. Without loss of generality, we assume that $0 \le p_i < 1$. The major difficulty here is to compute the expected service cost for each client. To overcome this difficulty, we extend the formulation in Snyder and Daskin (2005) to a more general setting. Compared with Snyder and Daskin (2005), the URFLP can be interpreted slightly differently as follows. Given any solution to (URFLP-SP), one may assume that in each scenario A, each client j is assigned to the nearest operational facility i provided $c_{ij} \le r_j$; otherwise, one incurs the penalty for client j. Now, consider the collection of (operational) facilities to which a client is assigned in the different scenarios. Notice that this set completely determines the client–assignment decision and the corresponding assignment/penalty cost incurred in any scenario: we simply assign the client to the nearest operational facility from this set or incur the penalty if there is no such facility. The expected cost incurred for a client can also be easily calculated because the facilities fail independently with given probabilities.

This motivates the following alternative way of viewing URFLP. We now have a single-stage problem, where we need to determine which facilities to open and assign each client to a *set* of facilities. We view the facilities in a client's set as being classified into levels, with the interpretation that in a given scenario, a client is served by the lowest-level facility that is operational. (In the mapping described above, the level of a facility increases with its distance to the client, a property we will prove later.)

Mathematically, we define two types of new binary variables, x_{ij}^k, z_j^k, to capture different level of facilities for a client j. In particular, $x_{ij}^k = 1$ if facility i is the kth level backup facility of client j and $x_{ij}^k = 0$ otherwise; $z_j^k = 1$ if j has $(k-1)$th backup facility but has no kth backup facility, so that j incurs a penalty cost at level k.

Given variables x_{ij}^k and z_j^k, one can compute the expected total service cost. Consider a client j and its expected service cost at its level k facility. Client j is served by its level k facility only if all its lower-level assigned facilities become nonoperational. For any facility l, if it is on the lower level (i.e, less than k) for demand node j, then $\sum_{s=1}^{k-1} x_{lj}^s = 1$; otherwise, $\sum_{s=1}^{k-1} x_{lj}^s = 0$. It follows that for client j, the probability that all its lower-level facilities fail is $\prod_{l\in F} p_l^{\sum_{s=1}^{k-1} x_{lj}^s}$. If j is served by facility i, as j's level k backup facility, then facility i has to be operational, which occurs with probability $(1-p_i)$. Therefore, the expected service cost of client j at level k is

$$\sum_{i\in F} d_j c_{ij} x_{ij}^k (1-p_i) \prod_{l\in F} p_l^{\sum_{s=1}^{k-1} x_{lj}^s}.$$

Similarly, we can analyze the penalty cost of client j at level k, which is $\prod_{l\in F} p_l^{\sum_{s=1}^{k-1} x_{lj}^s} d_j r_j z_j^k$.

We use $|F|$ to denote the number of members in set F. To simplify the formulation, we generalize the notation of x_{ij}^k and z_j^k to $k = |F|+1$. Given $|F|$ facilities, it is impossible to have $(|F|+1)$th level backup facility of client j; therefore, $x_{ij}^{|F|+1}$ is always 0. On the other hand, $z_j^{|F|+1} = 1$, only if client j is served by all $|F|$ facilities.

The above discussion and notation lead to a nonlinear integer programming formulation for the URFLP as follows:

(URFLP-NIP)

$$\text{minimize} \quad \sum_{i\in F} f_i y_i + \sum_{j\in D} \sum_{k=1}^{|F|} \sum_{i\in F} d_j c_{ij} x_{ij}^k (1-p_i) \prod_{l\in F} p_l^{\sum_{s=1}^{k-1} x_{lj}^s} + \sum_{j\in D} \sum_{k=1}^{|F|+1} \prod_{l\in F} p_l^{\sum_{s=1}^{k-1} x_{lj}^s} d_j r_j z_j^k \tag{7}$$

$$\text{subject to} \quad \sum_{i\in F} x_{ij}^k + \sum_{t=1}^{k} z_j^t = 1 \quad \forall\, j \in D,\ k = 1, \dots, |F|+1, \tag{8}$$

$$x_{ij}^k \le y_i \quad \forall\, i \in F,\ j \in D,\ k = 1, \dots, |F|, \tag{9}$$

$$\sum_{k=1}^{|F|} x_{ij}^k \le 1 \quad \forall\, i \in F,\ j \in D, \tag{10}$$

$$x_{ij}^k,\ z_j^k,\ y_i \in \{0, 1\}. \tag{11}$$

The decision variables x_{ij}^k, z_j^k are defined earlier. The indicator variable $y_i = 1$ if facility i is open in the first stage; otherwise, $y_i = 0$. The objective function (7) is the summation of the facility cost, the expected service cost, and the expected penalty cost. Constraints (8) ensure that client j is either assigned to a facility or subject to a penalty at each level k. Constraints (9) make sure that no client is assigned to an unopen facility. Constraints (10) prohibit a client from being assigned to a specific facility at more than one level. Note that Constraints (9) and (10) can be tightened as

$$\sum_{k=1}^{|F|} x_{ij}^k \le y_i \quad \forall\, i \in F,\ j \in D. \tag{12}$$

In the Online Supplement to this paper (available at http://joc.pubs.informs.org/ecompanion.html), we show that the following statement is true.

Theorem 1. *Formulations* (*URFLP-NIP*) *and* (*URFLP-SP*) *are equivalent.*

3.3. Model Properties

In formulation (URFLP-NIP), we do not explicitly require that a closer open facility be assigned as a lower-level facility to a particular demand point. However, according to the following proposition, it is true that the level assignments among the open facilities are based on the relative distances between the demand point and the facilities regardless of the failure probabilities.

Proposition 1. *In any optimal solution to the* (*URFLP-NIP*), *for any client* j, *if* $x_{uj}^k = x_{vj}^{k+1} = 1$, *then* $c_{uj} \le c_{vj}$.

Proof. We prove the proposition by contradiction. Supposing that $c_{uj} > c_{vj}$, we will show that by "swapping" the assignment of u and v, the objective function will strictly decrease.

In particular, if we set $x_{uj}^{k+1} = 1$ and $x_{vj}^k = 1$ with the values of other variables unchanged, we can literally compute the difference between the new objective value and the original one; that is,

$$\begin{aligned} &d_j\big(c_{vj}(1-p_v)\bar{p}_k + c_{uj}(1-p_u)\bar{p}_k p_v\big)\\ &\quad - d_j\big(c_{uj}(1-p_u)\bar{p}_k + c_{vj}(1-p_v)\bar{p}_k p_u\big)\\ &\quad = d_j(c_{vj}-c_{uj})(1-p_v)(1-p_u)\bar{p}_k < 0, \end{aligned}$$

where $\bar{p}_k = \prod_{l\in F} p_l^{\sum_{s=1}^{k-1} x_{lj}^s}$. In other words, the two incur different costs only in the event where all the facilities l such that $\sum_{s=1}^{k-1} x_{lj}^s$ fail and facilities u, v do not fail, in which case the difference is $c_{vj} - c_{uj}$. Thus the difference in the expected cost is precisely $d_j(c_{vj}-c_{uj})(1-p_v)(1-p_u)\bar{p}_k$.

The negativity of the difference holds because $\bar{p}_k > 0$, and it is assumed that $p_u < 1$, $p_v < 1$, and $c_{vj} < c_{uj}$. This is clearly a contradiction to the optimality of the original solution. Therefore, $c_{uj} \le c_{vj}$. □

An implication of Proposition 1 is that if the set of open facilities is determined, then it is trivial to solve the level assignment problem for each client: assigning levels according to the relative distances of different facilities to the client. If at some level the distance is beyond the penalty cost, then no facility will be assigned at this level (and higher ones), and the demand node simply takes the (cheaper) penalty.

We would like to point out Theorem 1 implies that formulation (URFLP-SP) and formulation (URFLP-NIP) are just two ways of modeling the same problem. They should have the same minimum cost as long as the inputs to the two models are consistent. In formulation (URFLP-NIP), each facility i has independent failure probability p_i. This implies that there are $2^{|F|}$ scenarios, and the probability that each of the scenarios will occur can be calculated. The corresponding values serve as inputs to formulation (URFLP-SP). We notice that, in formulation (URFLP-SP), it is straightforward to obtain an optimal second-stage solution for a given first-stage solution. In particular, at the second stage, every client will be assigned to and be served by an open and operational facility that is closest to the client at the lowest possible level; if the service cost is higher than the penalty cost, then it takes the penalty.

Although the two integer programs (URFLP-SP) and (URFLP-NIP) are equivalent, it is not clear to us if this still holds if we relax the binary constraints. It would be interesting to know whether this is the case, or whether the relaxation of one integer program is stronger than the other. It is also interesting to know whether Proposition 1 can be extended to the case where the binary constraints are relaxed. That is, if $x_{uj}^k > 0$ and $x_{vj}^{k+1} > 0$, then $c_{uj} \le c_{vj}$. Answers to these questions may improve our understanding of these formulations and their relaxations, and could be useful in a computational study.

3.4. Uniform Failure Probabilities

Now we consider a special case of the URFLP where all facilities have the same failure probability; i.e., $p_i = p$, $\forall\, i \in F$. This assumption simplifies formulation (URFLP-NIP) considerably based on the following observation. Because $p_i = p$, $\forall\, i \in F$, it is straightforward that $\prod_{l\in F} p_l^{\sum_{s=1}^{k-1} x_{lj}^s} = p^{k-1}$, which is independent of the values of x_{lj}^s. This property is implicitly used in a multiobjective formulation proposed in Snyder and Daskin (2005).

Based on the above observation, we are able to reduce formulation (URFLP-NIP) to a linear integer program as follows:

(URFLP-IP-I)

$$\text{minimize} \quad \sum_{i\in F} f_i y_i + \sum_{j\in D}\sum_{k=1}^{|F|}\sum_{i\in F} d_j c_{ij} x_{ij}^k (1-p)p^{k-1} + \sum_{j\in D}\sum_{k=1}^{|F|+1} p^{k-1} d_j r_j z_j^k$$

$$\text{subject to} \quad \sum_{i\in F} x_{ij}^k + \sum_{t=1}^{k} z_j^t = 1 \quad \forall j\in D, k=1,\ldots,|F|+1,$$
$$\sum_{k=1}^{|F|} x_{ij}^k \le y_i \quad \forall i\in F,\ j\in D,$$
$$x_{ij}^k, z_j^k, y_i \in \{0,1\}.$$

In the next section, we shall present an approximation algorithm for (URFLP-IP-I). We find that it is more convenient to deal with a slightly different formulation of (URFLP-IP-I). In the new formulation, we introduce a new set of decision variables θ_j^k to replace z_j^k. Define $\theta_j^k = 1$ if j is not assigned to any facility as its kth backup facility. That is, $\theta_j^k = \sum_{t=1}^k z_j^t$. Also, for simplicity, we let $\theta_j^k = 1$ for all $k \ge |F|+1$. In other words, θ_j^k are not decision variables when $k > |F|$.

(URFLP-IP-II)

$$\text{minimize} \quad \sum_{i\in F} f_i y_i + \sum_{j\in D}\sum_{k=1}^{|F|}\sum_{i\in F} d_j c_{ij} x_{ij}^k p^{k-1}(1-p) + \sum_{j\in D}\sum_{k=1}^{\infty} d_j r_j \theta_j^k p^{k-1}(1-p)$$
$$\text{subject to} \quad \sum_{i\in F} x_{ij}^k + \theta_j^k = 1 \quad \forall j\in F,\ k=1,\ldots,|F|,$$
$$\sum_{k=1}^{|F|} x_{ij}^k \le y_i \quad \forall i\in F,\ j\in D,$$
$$x_{ij}^k, \theta_j^k, y_i \in \{0,1\}.$$

We prove that the above integer program is equivalent to the formulation (URFLP-IP-I) as stated in Theorem 2. We refer the reader to the Online Supplement to this paper for a proof.

Theorem 2. *Formulations* (*URFLP-IP-I*) *and* (*URFLP-IP-II*) *are equivalent. Furthermore, the LP relaxations of these two formulations are also equivalent.*

4. Approximation Algorithms: Uniform Probabilities

In this section, we aim to propose a 4-approximation algorithm for the special case where the failure probabilities are uniform. We call it an R-approximation algorithm if the algorithm produces a solution with cost no more than R times the optimal total cost.

We first solve a linear programming relaxation of formulation (URFLP-IP-II). Note that $\theta_j^k = 1$ for all $k \ge |F|+1$, and $\sum_{k=|F|+1}^{\infty} p^{k-1}(1-p) = p^{|F|}$. Therefore, this linear programming can be expressed as the following:

(URFLP-IP-II-P)

$$\text{minimize} \quad \sum_{i\in F} f_i y_i + \sum_{j\in D}\sum_{k=1}^{|F|}\sum_{i\in F} d_j c_{ij} x_{ij}^k p^{k-1}(1-p) + \sum_{j\in D}\sum_{k=1}^{|F|} d_j r_j \theta_j^k p^{k-1}(1-p) + \sum_{j\in D} d_j r_j p^{|F|}$$
$$\text{subject to} \quad \sum_{i\in F} x_{ij}^k + \theta_j^k = 1 \quad \forall j\in D,\ k=1,\ldots,|F|, \tag{13}$$
$$\sum_{k=1}^{|F|} x_{ij}^k \le y_i \quad \forall i\in F,\ j\in D, \tag{14}$$
$$y_i \le 1 \quad \forall i\in F, \tag{15}$$
$$x_{ij}^k \ge 0, \quad \theta_j^k \ge 0. \tag{16}$$

The dual of this linear program is

(URFLP-IP-II-D)

$$\text{maximize} \quad \sum_{j\in D}\sum_{k=1}^{|F|} \alpha_j^k - \sum_{i\in F}\gamma_i + \sum_{j\in D} d_j r_j p^{|F|} \tag{17}$$
$$\text{subject to} \quad \alpha_j^k \le d_j r_j p^{k-1}(1-p) \quad \forall j\in D,\ k=1,\ldots,|F|, \tag{18}$$
$$\alpha_j^k - \beta_{ij} \le d_j c_{ij} p^{k-1}(1-p) \tag{19}$$
$$\forall i\in F,\ j\in D,\ k=1,\ldots,|F|, \tag{20}$$
$$\sum_{j\in D}\beta_{ij} - \gamma_i \le f_i \quad \forall i\in F, \tag{21}$$
$$\beta_{ij}\ge 0, \quad \gamma_i \ge 0 \quad \forall i\in F,\ j\in D. \tag{22}$$

Assume that (x, y, θ) is an optimal solution to (URFLP-IP-II-P) and (α, β, γ) is an optimal solution to the dual (URFLP-IP-II-D). Our algorithm rounds the fractional solution (x, y, θ) to an integer solution $(\bar{x}, \bar{y}, \bar{\theta})$ that is feasible to formulation (URFLP-aIP-II).

Our algorithm is based on several properties of the optimal solutions (x, y, θ) and (α, β, γ), which are formalized in the following lemmas.

Lemma 1. *There exists an optimal solution of* (*URFLP-IP-II-P*) *such that, for each* $j \in D$, *the following two statements are true.*

(i) $\theta_j^k \le \theta_j^{k'}$ *for any* $1 \le k \le k'$.

(ii) *If there exists* k *such that* $0 < \theta_j^k < 1$, *then* $\theta_j^{k'} = 0$ *and* $\theta_j^{k''} = 1$ *for* $k' < k$ *and* $k'' > k$.

Proof. The proof is intuitive and similar to the proof of Theorem 2. We just sketch the key idea here.

Assume there exists j and k such that $0 < \theta_j^k \le 1$, and $\theta_j^{k'} < 1$ for some $k' > k$. Then there must exist an i_0 such that $x_{i_0 j}^{k'} > 0$. We change the solution as follows for some $\varrho > 0$ (the other variables are the same as before):

$$\theta_j^{k'} \leftarrow \theta_j^{k'} + \varrho, \quad \theta_j^k \leftarrow \theta_j^k - \varrho$$
$$x_{i_0 j}^{k'} \leftarrow x_{i_0 j}^{k'} - \varrho, \quad x_{i_0 j}^{k} \leftarrow x_{i_0 j}^{k} + \varrho.$$

It is clear that we can choose $\varrho > 0$ so that the new solution is still feasible (all the constraints of the linear program are satisfied). We can compare the cost corresponding to the new solution and the optimal cost. The difference should be greater than zero; that is,

$$\varrho p^{k-1}(1-p) d_j c_{i_0 j} - \varrho p^{k'-1}(1-p) d_j c_{i_0 j} - d_j r_j \varrho p^{k-1}(1-p) + d_j r_j \varrho p^{k'-1}(1-p) \ge 0.$$

This implies that $r_j \le c_{i_0 j}$. Thus $r_j \le c_{ij}$ for any $i \in F$ such that $x_{ij}^{k'} > 0$. Then we change the original optimal solution as follows (the other variables are the same as before):

$$\theta_j^{k'} \leftarrow 1, \quad x_{i_0 j}^{k'} \leftarrow 0.$$

This is clearly a feasible solution, and the cost will not increase; i.e., it is an optimal solution. This proves both (i) and (ii). □

Lemma 2. *There exists an optimal solution of* (*URFLP-IP-II-P*) *such that, for any $i \in F$ and $j \in D$, there exist at most two values of k with $x_{ij}^k > 0$. If two such values exist, they must be consecutive in k.*

Proof. Consider any $j \in D$. Assume that there exist $i \in F$ and k_1, k_2 with $k_2 \ge k_1 + 2$ such that $x_{ij}^{k_1} > 0$ and $x_{ij}^{k_2} > 0$. It follows from (13) that $\theta_j^{k_1} < 1$ and $\theta_j^{k_2} < 1$. Now consider any k with $k_1 < k < k_2$. By Lemma 1(i), we have $\theta_j^k \le \theta_j^{k_2}$. If $\theta_j^k > 0$, then $0 < \theta_j^k \le \theta_j^{k_2} < 1$, which cannot hold in view of Lemma 1(ii). Therefore, we must have $\theta_j^k = 0$ or $\sum_{i' \in F} x_{i'j}^k = 1$.

However, $x_{ij}^k \le 1 - x_{ij}^{k_1} - x_{ij}^{k_2} < 1$. Thus, there must exist $l \in F$, $l \ne i$ such that $x_{lj}^k > 0$. Let L be the set of such ls. It is evident that $c_{lj} = c_{ij}$ for any $l \in L$. For each $l \in L \cup \{i\}$ (it is not possible that $i \in L$), let $\chi_l = \sum_{k=k_1}^{k_2} x_{lj}^k \in (0, y_l]$ and $\phi^k = \sum_{l \in L \cup \{i\}} x_{lj}^k$ for $k_1 \le k \le k_2$. Notice that $\phi^k = 1$ for $k_1 < k < k_2$. Then we do the following.

Step 1. Relabel the facilities in $L \cup \{i\}$ as $1, 2, \ldots, |L \cup \{i\}|$.

Step 2. Set $k = k_1$, $l = 1$.

Step 3. While $k < k_2$ or $l < |L \cup \{i\}|$, do the following

Let $x_{lj}^k \leftarrow \min\{\chi_l, \phi^k\}$, $\chi_l \leftarrow \chi_l - x_{lj}^k$, $\phi^k \leftarrow \phi^k - x_{lj}^k$.

If $\phi^k = 0$, let $k \leftarrow k + 1$.

If $\chi_l = 0$, let $l \leftarrow l + 1$.

After this procedure, if there exist $i \in F$ and k_1', k_2' with $k_2' \ge k_1' + 2$ such that $x_{ij}^{k_1'} > 0$ and $x_{ij}^{k_2'} > 0$, then $k_1' \ge k_2$ or $k_2' \le k_1$. By repeating the above procedure at most $|F|$ times, the lemma holds for $j \in D$. Notice this procedure will not affect the solution of other clients. This completes the proof. □

Finally, by linear programming complementarity conditions, we have the following.

Lemma 3. *The following statements hold.*

- *If $x_{ij}^k > 0$, then $\alpha_j^k = \beta_{ij} + d_j c_{ij} p^{k-1}(1-p) \ge d_j c_{ij} p^{k-1}(1-p)$.*
- *If $\theta_j^k > 0$, then $\alpha_j^k = d_j r_j p^{k-1}(1-p)$.*
- *If $\gamma_i > 0$, then $y_i = 1$ and thus $\sum_{j \in D} \beta_{ij} - \gamma_i = f_i$.*
- *If $\beta_{ij} > 0$, then $y_i = \sum_{k=1}^{|F|} x_{ij}^k$.*

With the help of these lemmas, we take advantage of results for the fault-tolerant version of the UFLP, where every demand point j must be served by k_j distinct facilities, a concept close to our level assignment. Guha et al. (2003) and Swamy and Shmoys (2008) present several approximation algorithms for the fault-tolerant facility location problem.

The basic idea is to first determine how many open facilities a client should be assigned to. This is done based on the values of the primal variable θ_j^k. For each $j \in D$, assume k_j is the smallest integer such that $\theta_j^{k_j} > 0$. Once this is determined, we will solve an appropriately defined fault-tolerant facility location problem.

Phase 1

- For every $j \in D$ and $k \ge k_j$, let $\bar{\theta}_j^k = 1$.
- In the following phases, each client j shall be assigned to $k_j - 1$ open facilities.

Lemma 4. *If each client will be assigned to $k_j - 1$ open facilities, then the total penalty cost is $\sum_{j \in D} \sum_{k=k_j}^{|F|} \alpha_j^k + \sum_{j \in D} d_j r_j p^{|F|}$.*

Proof. Notice that for $k \ge k_j$, $\theta_j^k > 0$, and thus $\alpha_j^k = d_j r_j p^{k-1}(1-p)$. Therefore, the total penalty cost is

$$\begin{aligned}\sum_{j \in D} d_j r_j p^{k_j - 1} &= \sum_{j \in D} \sum_{k=k_j}^{|F|} d_j r_j p^{k-1}(1-p) + \sum_{j \in D} d_j r_j p^{|F|} \\ &= \sum_{j \in D} \sum_{k=k_j}^{|F|} \alpha_j^k + \sum_{j \in D} d_j r_j p^{|F|}.\end{aligned}$$

This completes the proof. □

The next phase is based on an idea proposed by Swamy and Shmoys (2008) for the fault-tolerant facility location problem.

Phase 2

- Denote $F_1 = \{i \in F\colon y_i = 1\}$. Open all facilities in F_1; i.e., $\bar{y}_i = 1$ for $i \in F_1$.
- For each $j \in D$, repeat the following for each $i \in F_1$. Let k^{ji} be the smallest k such that $x_{ij}^k > 0$ if such k exists; otherwise, let $k^{ji} = 0$. In view of Lemma 2, we also know that $x_{ij}^k = 0$ for $k > k^{ji}+1$ when $k^{ji} \ne 0$ (it is possible that $x_{ij}^{k^{ji}+1} > 0$). Then we let $\bar{x}_{ij}^{k^{ji}} = 1$ and $\bar{\theta}_j^{k^{ji}} = 0$; i.e., facility i is the k^{ji}th level assignment of client j. In the original solution, the contribution of facility i to the $(k^{ji}+1)$th level of client j is $x_{ij}^{k^{ji}+1} \le 1 - x_{ij}^{k^{ji}} = \sum_{i' \ne i} x_{i'j}^{k^{ji}}$ (in fact, given that $i \in F_1$ and thus $y_i = 1$, there must exist an optimal solution such that the inequality holds as equality). Therefore, to compensate for the loss of this contribution, we do the following: if there exists $i' \ne i$ such that $x_{i'j}^{k^{ji}} > 0$ and $k^{ji} + 1 \le k_j$, then let $x_{i'j}^{k^{ji}+1} = x_{i'j}^{k^{ji}}$. This procedure will maintain the property that in the fractional solution, j is assigned to at least one facility for the $(k^{ji}+1)$th level.
- For each $j \in D$, denote

$$R_j = \{k\colon 1 \le k \le |F|, \exists i \in F_1, \text{ s.t. } k = k^{ji}\}.$$

Let $\hat{x}_{ij}^k = x_{ij}^k$ for $i \notin F_1$, $j \in D$, $1 \le k \le k_j - 1$ and $k \notin R_j$.

Remark. Notice that if $k^{ji}=0$ for all $i\in F_1$, then $R_j=\varnothing$. Therefore, Phase 2 of the algorithm only addresses clients j with $R_j\neq\varnothing$.

Lemma 5. *The cost of Phase* 2 *is*

$$\sum_{i\in F_1} f_i+\sum_{j\in D}\sum_{k\in R_j} d_j c_{ij} p^{k-1}(1-p)\bar{x}_{ij}^k=\sum_{j\in D}\sum_{k\in R_j}\alpha_j^k-\sum_{i\in F}\gamma_i\geq 0.$$

Proof. For each $i\in F_1$, $y_i=1$ and thus $f_i=\sum_{j\in D}\beta_{ij}-\gamma_i$. Moreover, for each $i\in F$, $z_i>0$ only if $i\in F_1$. The total facility cost is

$$\sum_{i\in F_1} f_i=\sum_{i\in F_1}\sum_{j\in D}\beta_{ij}-\sum_{i\in F_1}\gamma_i=\sum_{j\in D}\sum_{i\in F_1}\beta_{ij}-\sum_{i\in F}\gamma_i.$$

The total assignment cost is

$$\sum_{j\in D}\sum_{i\in F_1} d_j c_{ij} p^{k^{ji}-1}(1-p)=\sum_{j\in D}\sum_{i\in F_1}(\alpha_j^{k^{ji}}-\beta_{ij}).$$

However, notice that for each $j\in D$, $\sum_{i\in F_1}\alpha_j^{k^{ji}}=\sum_{k\in R_j}\alpha_j^k$. The lemma follows. □

Lemma 6. *After Phase* 2, *it holds that for* $k\notin R_j$ *and* $k\leq k_j-1$,

- $\sum_{i\in F\setminus F_1}\hat{x}_{ij}^k\geq 1$,
- $\sum_{k\notin R_j,\,k\leq k_j-1}\hat{x}_{ij}^k\leq y_i\leq 1$, *for every* $i\in F\setminus F_1$, *and*
- *if* $\hat{x}_{ij}^k>0$ *then* $c_{ij}\leq\alpha_j^k/(d_j p^{k-1}(1-p))$.

Proof. The first item follows from the description and discussion of the algorithm. The second item follows because

$$\sum_{k\notin R_j,\,k\leq k_j-1}\hat{x}_{ij}^k\leq\sum_{k\notin R_j,\,k\leq k_j-1}x_{ij}^k\leq y_i\leq 1.$$

For the last item, we notice that by Lemma 3, before Phase 2, if $x_{ij}^k>0$, then $c_{ij}\leq\alpha_j^k/(d_j p^{k-1}(1-p))$. If $\hat{x}_{ij}^k>0$ and $\hat{x}_{ij}^k\neq x_{ij}^k$, then $\hat{x}_{ij}^k=x_{ij}^{k-1}>0$, and there exists i' such that $x_{i'j}^{k-1}>0$ and $x_{i'j}^k>0$. It is clear that $c_{ij}\leq c_{i'j}$. However, since $x_{i'j}^k>0$, we must have $c_{i'j}\leq\alpha_j^k/(d_j p^{k-1}(1-p))$. This completes the proof. □

In Phase 3, we shall use one of the key results from Guha et al. (2003) and the algorithm in §2 of Swamy and Shmoys (2008) for the fault-tolerant facility location problem. The result can be stated as follows.

Lemma 7. *Consider any vector* (x,y) *satisfying the following inequalities* (*the dimension of* (x,y) *should be clear from the inequalities*):

$$\sum_{i\in F}x_{ij}^k\geq 1\quad\forall j\in D,\ k\leq k_j,$$

$$\sum_{k=1}^{k_j}x_{ij}^k\leq y_i\quad\forall i\in F,\ j\in D,$$

$$x_{ij}^k\geq 0,$$

$$0\leq y_i\leq 1.$$

(*These inequalities define the feasible set of the LP relaxation of the fault-tolerant facility location problem.*) *Also, if there exists a vector* $\tilde{C}$ *such that* $x_{ij}^k>0$ *implies* $c_{ij}\leq\tilde{C}_j^k$, *then one can find, for any* $t_j\leq k_j$ *and* w_j^k, *an integer solution* $(\tilde{x},\tilde{y})$ *satisfying the above inequalities so that*

$$\sum_{i\in F}f_i\tilde{y}_i+\sum_{j\in D}\sum_{k=1}^{t_j}\sum_{i\in F}w_j^k d_j c_{ij}\tilde{x}_{ij}^k\leq\sum_{i\in F}f_i y_i+3\sum_{j\in D}\sum_{k=1}^{t_j}w_j^k d_j\tilde{C}_j^k.$$

Phase 3

Using algorithms from Guha et al. (2003) or Swamy and Shmoys (2008) to solve a fault-tolerant facility location problem, we open a set of facilities in $F\setminus F_1$ so that for each $j\in D$, it is assigned to $k_j-1-|R_j|$ open facilities in $F\setminus F_1$. After this phase, let $\bar{y}_i=1$ if facility i is open, and let $\bar{x}_{ij}^k=1$ if client j is assigned to facility i for its kth level with $k\leq k_j-1$ and $k\notin R_j$.

Lemma 8. *The cost of Phase* 3 *is at most* $\sum_{i\in F\setminus F_1}f_i y_i+3\sum_{j\in D}\sum_{1\leq k\leq k_j-1,\,k\notin R_j}\alpha_j^k$.

Proof. It follows directly from Lemmas 6 and 7. □

Phases 1, 2, and 3 output a feasible integer solution $(\bar{x},\bar{y},\bar{\theta})$ to the URFLP. The performance guarantee of this algorithm is given by the next theorem.

Theorem 3. *The above algorithm is a* 4*-approximation algorithm for the URFLP with uniform failure probabilities.*

Proof. Combining Lemmas 4, 5, and 8, the total cost of Phases 1, 2, and 3 is

$$\begin{aligned}&\sum_{j\in D}\sum_{k=k_j}^{|F|}\alpha_j^k+\sum_{j\in D}d_j r_j p^{|F|}+\left(\sum_{j\in D}\sum_{k\in R_j}\alpha_j^k-\sum_{i\in F}\gamma_i\right)\\&\quad+\sum_{i\in F\setminus F_1}f_i y_i+3\sum_{j\in D}\sum_{1\leq k\leq k_j-1,\,k\notin R_j}\alpha_j^k\\&\leq 3\sum_{j\in D}\sum_{k=k_j}^{|F|}\alpha_j^k+3\sum_{j\in D}d_j r_j p^{|F|}+3\left(\sum_{j\in D}\sum_{k\in R_j}\alpha_j^k-\sum_{i\in F}\gamma_i\right)\\&\quad+\sum_{i\in F\setminus F_1}f_i y_i+3\sum_{j\in D}\sum_{1\leq k\leq k_j-1,\,k\notin R_j}\alpha_j^k\\&=3\sum_{j\in D}\sum_{k=1}^{|F|}\alpha_j^k-3\sum_{i\in F}\gamma_i+3\sum_{j\in D}d_j r_j p^{|F|}+\sum_{i\in F\setminus F_1}f_i y_i.\end{aligned}$$

Then the theorem follows by noticing that $\sum_{i\in F\setminus F_1}f_i y_i$ is bounded above by the optimal total cost. □

5. Heuristics and Computational Tests

In this section, we propose several heuristics to solve the general URFLP: sample average approximation heuristic (SAA-H), greedy adding heuristic (GAD-H), and greedy adding and substitution heuristic (GADS-H). To evaluate the performance of these three heuristics, we apply them to the special case of URFLP with uniform failure probabilities. The reason is that the special case admits an integer programming formulation that can be solved to optimality by using commercial solvers such as CPLEX, so that we can compare the heuristic results with the exact solutions.

The test data set is generated as follows. The coordinates of the sites are drawn from $U[0,1] \times U[0,1]$, demand of each site is drawn from $U[0,1{,}000]$ and rounded to the nearest integer, fixed facility costs are drawn from $U[500,1{,}500]$ and rounded to the nearest integer, and penalty costs are drawn from $U[0,15]$. Furthermore, the transportation cost c_{ij} is set to be the Euclidean distance between points i and j. The number of sites are varied from 10 to 100.

All the algorithms were coded in C++ and tested on a Dell Optiplex GX620 computer running the Windows XP operating system with a Pentium IV 3.6 GHz processor and 1.0 GB RAM.

5.1. Sample Average Approximation Heuristic

The sample average approximation method is widely used for solving complicated stochastic discrete optimization problems (e.g., Kleywegt et al. 2001, Santoso et al. 2005, Verweij et al. 2003) and is also in the design of approximation algorithms for such problems (Swamy and Shmoys 2006). The basic idea of this method is to use a sample average function to estimate the expected value function. We apply the following procedures to solve the (URFLP-SP).

The SAA Heuristic

Step 1. Randomly generate a sample of N scenarios $\{A_1, \ldots, A_N\}$ and solve the following SAA problem:

$$\begin{aligned}
\text{minimize}\quad & \sum_{i\in F} f_i y_i + \sum_{s=1}^{N} \frac{1}{N}\left(\sum_{j\in D}\sum_{i\in F} d_j c_{ij} x_{ij}^{A_s} + \sum_{j\in D} d_j r_j z_j^{A_s}\right)\\
\text{s.t.}\quad & \sum_{i\in F} x_{ij}^{A_s} + z_j^{A_s} = 1 \quad \forall\, j\in D,\ s=1,\ldots,N,\\
& x_{ij}^{A_s} \le y_i I_{A_s,i} \quad \forall\, i\in F,\ j\in D,\ s=1,\ldots,N,\\
& y_i, x_{ij}^{A_s}, z_j^{A_s} \in [0,1].
\end{aligned}$$

Repeat this step M times. For each $m = 1, 2, \ldots, M$, let (x^m, y^m, z^m) and v^m be the corresponding optimal solution and its optimal objective value to the above SAA formulation, respectively. Note that formulations (URFLP-SP) and (URFLP-NIP) are equivalent. Instead of using v^m to estimate the expected objective value denoted by $\hat{v}^m$ of (URFLP-SP), one can compute such expected value directly under the objective function of (URFLP-NIP), i.e., (7) using the value of (x^m, y^m, z^m).

Step 2. Among the M solutions obtained in the first step, output the one with the minimum objective value; i.e., $\hat{v}^{\min} = \min_{m=1,\ldots,M} \hat{v}^m$.

Two remarks are in order. First, in a standard SAA approach (e.g., Santoso et al. 2005, Verweij et al. 2003), one more independent sample is needed to estimate the true expected value $\hat{v}^m$ of solution y^m. But in our case, the form of the underlying distribution allows us to use an analytical formula to compute the true expected value. Second, in contrast to the results in Santoso et al. (2005), the average of the v^m values, i.e., $\bar{v}^M = (\sum_{m=1}^{M} v^m)/\mathrm{M}$, does not provide a lower statistical bound for the optimal value. The reason is that different samples in our case may lead to different solution spaces of the problem, which is due to the provider-side uncertainty. Nonetheless, empirically, $\bar{v}^M$ may still serve as a good indication of the quality of the solution from the SAA approach, as we will illustrate in the following computational tests.

We first test how the sample size (N) affect (1) the quality of the solution and (2) the efficiency of the program for a 50-node data set with the uniform failure probabilities varying from 0 to 1. Table 1 lists the objective values obtained from SAA-H when $M = 1$ and the sample size varies from 10 to 50.

It is clear from Table 1 that the solution quality can be improved by increasing the sample size. The ratios of the objective value obtained from the SAA-H to the optimal value are plotted in Figure 1, and the computation times are plotted in Figure 2.

Figure 1 shows that the SAA-H with $M = 1$ obtains fairly good solution when the failure probability is small ($p \le 0.3$) but not very good solution when the failure probability is big ($p > 0.3$). The following could be a possible explanation. When the failure probability is small, the majority of the facilities are candidate sites for opening in each sample, so the sets of candidate facilities are similar in different samples. Thus, any individual sample can capture the characteristics of the system pretty well, and the corresponding solution obtained from SAA-H is close to optimal. For the extreme case where $p = 0$, all facilities are available to open in each sample, so the sets of available facilities are the same in each sample, and the SAA-H can produce the exact solution in this case.

Table 1 Objective Values from SAA-H for 50-Node Data Set

Failure prob.	N10	N20	N30	N40	N50	Exact
0	7,197.27	7,197.27	7,197.27	7,197.27	7,197.27	7,197.27
0.1	7,956.03	7,956.03	7,763.80	7,763.80	7,763.80	7,763.80
0.2	9,429.49	8,770.31	8,425.99	8,425.99	8,425.99	8,425.99
0.3	9,908.52	10,059.90	9,669.49	10,201.50	10,095.20	9,275.99
0.4	11,546.90	11,753.50	11,984	13,887.60	11,937.60	10,253.90
0.5	18,727.50	15,128.10	13,120.60	15,052.80	13,546.4	11,603
0.6	27,429	18,161.70	17,946.40	17,946.40	17,946.40	13,416.80
0.7	32,965.30	33,139.20	27,374.80	23,659.30	23,690.40	16,157.20
0.8	62,876.90	46,387.50	35,743.60	35,743.60	35,743.60	21,500.70
0.9	84,406.10	69,255.40	54,722.30	54,728.10	55,647.40	35,987.70
1.0	128,009	128,009	128,009	128,009	128,009	128,009

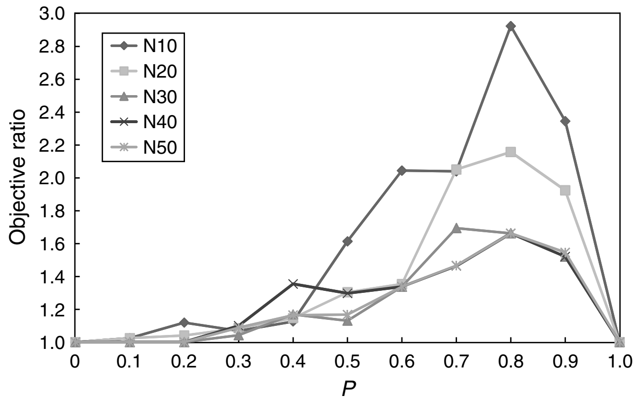

Figure 1 **Objective Ratio Across Different Sample Sizes for 50-Node Data Set**

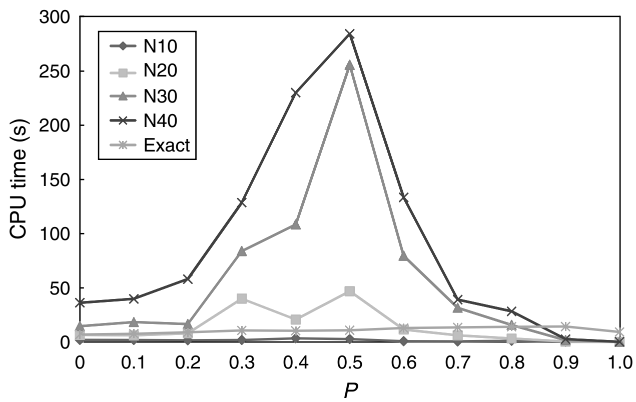

Figure 2 **CPU Time Across Different Sample Sizes for 50-Node Data Set**

Figure 2 shows that the case with $N = 10$ is the only one that requires slightly less time than the exact algorithm, whereas others require more time. Another interesting pattern in Figure 2 is the tail effect of the CPU time in terms of the failure probability. SAA-H spends more time to obtain a solution when the failure probability is approximately 0.5. One possible way to explain this phenomenon is the following: when the failure probability is approximately 0.5, the constraints $x_{ij}^s \le y_i I_{A_s,i}$ among different samples are quite different. As a result, the problem size increases, and so does the computation time.

Next, we examine the effect of replication number (M) on the solution quality by fixing $N = 30$. Table 2 provides the objective values obtained when $M = 5, 10, 15$, and 20. From the objective values obtained in different replication numbers, we can see that the increase of the replication number has not affected the solution quality too much. The gap in this table is defined as $((\hat{v}^{\min} - \bar{v}^M)/\bar{v}^M) \times 100\%$. The negative numbers in the "Gap (%)" column show that $\bar{v}^M$ is not always a lower bound for $\hat{v}^{\min}$. However, it is a good indication for the quality of the solution from SAA-H. In this particular case, if the gap is within $\pm 10\%$, the obtained objective value is close to the optimal value.

Overall, SAA-H spends a lot of time to obtain some poor-quality solutions when the failure probability is fairly large. We defer presenting the computational results for the general case to §5.3, where a comparison is done between SAA-H and the greedy methods.

5.2. Greedy Methods: GAD-H and GADS-H

One of the most commonly used heuristics in optimization is local search and iterative improvement algorithms. In formulation (URFLP-NIP), Proposition 1 ensures that the level assignments can be easily derived for a given set of open facilities. Therefore, one can concentrate on selecting a set of open facilities without worrying too much on the level assignment decisions. Let $v(T)$ denote the objective function value given by the set of open facilities, T. Let T^t be the set of open facilities at Step t, and let Φ be the empty set.

Greedy Adding Heuristic

Step 1. Initially the set of open facilities is empty. Set $t = 0$ and $T_{\text{open}}^t = \Phi$.

Step 2. Choose a facility from the remaining candidates to open such that it can reduce the total cost the most. Add this facility to the facility set. That is,

$t = t + 1$,

$j_t = \arg_{j \in F \setminus T^{t-1}} \min(v(T^{t-1} \cup \{j\}))$,

$T^t = T^{t-1} \cup \{j_t\}$.

Table 2 **Runs from SAA-H for 50-Node Data Set**

	$M = 5$			$M = 10$			$M = 15$			$M = 20$			
Failure prob.	$\hat{v}^{\min}$	$\bar{v}^5$	Gap (%)	$\hat{v}^{\min}$	$\bar{v}^{10}$	Gap (%)	$\hat{v}^{\min}$	$\bar{v}^{15}$	Gap (%)	$\hat{v}^{\min}$	$\bar{v}^{20}$	Gap (%)	Exact
0	7,197.27	7,197.27	0.00	7,197.27	7,197.27	0.00	7,197.27	7,197.27	0.00	7,197.27	7,197.27	0.00	7,197.27
0.1	7,763.80	7,687.03	1.00	7,763.80	7,760.14	0.05	7,763.80	7,784.17	−0.26	7,763.80	7,768.50	−0.06	7,763.80
0.2	8,425.99	8,315.65	1.33	8,425.99	8,436.50	−0.12	8,425.99	8,484.24	−0.69	8,425.99	8,453.66	−0.33	8,425.99
0.3	9,414.40	9,054.28	6.79	9,378.06	9,112.31	3.32	9,378.06	9,138.80	2.62	9,275.99	9,152.36	2.47	9,275.99
0.4	10,872.60	9,740.45	23.03	10,479.80	9,814.09	10.79	10,479.80	9,826.75	6.65	10,259.90	9,842.41	6.48	10,253.90
0.5	11,932	10,457	25.47	11,932	10,497.60	13.66	11,932	10,507.40	13.56	11,932	10,530.80	13.31	11,603
0.6	17,825.90	11,377.60	57.73	17,335.50	11,475.90	55.33	17,335.50	11,523.50	50.44	17,291.50	11,563	49.92	13,416.80
0.7	27,157.40	12,758.80	114.56	23,227.40	12,816.20	111.90	22,894.20	12,766.10	81.95	22,894.20	12,747.70	79.59	16,157.20
0.8	34,912.50	14,758.30	142.19	31,284.20	14,761.90	136.50	31,284.20	14,736.80	112.29	31,284.20	14,746	112.15	21,500.70
0.9	54,722.30	19,703.50	177.73	54,628.80	19,428.60	181.66	54,628.80	19,752	176.57	53,343.90	19,811.80	175.74	35,987.70
1	128,009	128,009	0.00	128,009	128,009	0.00	128,009	128,009	0.00	128,009	128,009	0.00	128,009

Table 3 50-Node Uniform Case: Greedy Adding and Exact Method

	Greedy adding heuristic		Exact algorithm		
P	Objective	Time (s)	Objective	Time (s)	Gap (%)
0	7,551.02	0.00	7,197.27	6.94	4.92
0.1	8,053.11	0.00	7,763.80	7.61	3.73
0.2	8,637.46	0.00	8,425.99	8.94	2.51
0.3	9,309.50	0.00	9,275.99	10.62	0.36
0.4	10,253.90	0.02	10,253.90	10.38	0.00
0.5	11,622.80	0.02	11,603.00	10.86	0.17
0.6	13,416.80	0.02	13,416.80	12.84	0.00
0.7	16,157.20	0.02	16,157.20	13.47	0.00
0.8	21,500.70	0.02	21,500.70	14.08	0.00
0.9	35,987.70	0.05	35,987.70	14.27	0.00
1	128,009.00	0.00	128,009.00	9.27	0.00

Step 3. Repeat Step 2 until the current solution cannot be improved further.

In general, as we can see from the computational tests, the greedy adding heuristic is able to find a high-quality solution very efficiently. Table 3 lists the computational results of 50-node data set when the failure probability varies from 0 to 1. The first column, P, is the failure probability at each facility. The "Gap (%)" column is defined as the percentage difference between the cost of the solution obtained by GAD-H and the optimal cost.

As we can see from Table 3, GAD-H finds optimal solutions in most cases in less than 0.05 seconds. Compared with to the exact method using CPLEX, it spends much less time. In the case where an obtained solution is not optimal, it is close to the optimal solution. The greedy adding algorithm seems to perform better when the facility failure probability is high. It actually finds optimal solutions when the failure probability exceeds 0.5. This is in contrast to the performance of the SAA-H, which works better when the failure probability is low.

The greedy adding heuristic can be further improved by propositioning a greedy substitution heuristic: after the greedy adding procedure, an additional process is performed that substitutes the open facility. At each iteration, a substitution facility is chosen to replace the existing open one if it reduces the total cost the most. We repeat this procedure until no cost improved substitution facility exists. The substitution includes to simply close the existing open facility. That is, a null facility is the best substitution for the open facility. After the substitution process, another greedy adding procedure is performed to further improve the solution. The whole process is called the greedy adding and substitution heuristic (GADS-H).

Our computational experiments show that GADS-H is very effective. It actually finds all the optimal solutions for the instances in Table 3 without much extra computational effort. The results are summarized in Table 4.

It is interesting to compare the sets of open facilities in Table 4. One might conclude that more facilities should be open as the facilities get more vulnerable, that is, when the failure probability increases. Although this claim is usually valid, it is not always true. An extreme case is when failure probability is 1 so that no facility should open. One can also consider the following counterexample where there is only one single facility to open. If $f_1 + d_{11} r_1 p_1 < d_{11} r_1$, then this facility should be open. If p_1 increases to p_1' such that $f_1 + d_{11} r_1 p_1' > d_{11} r_1$, then this facility should not be open. In other words, fewer facilities are open, when the failure probability increases in this counterexample.

5.3. General Case: SAA-H, GAD-H, and GADS-H

Dropping the uniformity assumption on the facility failure probabilities introduces more challenges to solve the URFLP exactly, because of the nonlinearity in formulation (URFLP-NIP). In this section, we apply SAA-H, GAD-H, and GADS-H to the general problem and compare their performance. Several data sets

Table 4 50-Node Uniform Case: Greedy Add-Drop and Exact Solution

	GADS-H		Exact algorithm			
P	Objective	Time (s)	Objective	Open facilities	Time (s)	Gap (%)
0	7,197.27	0.02	7,197.27	15 31 40 41 48	6.94	0.00
0.1	7,763.80	0.02	7,763.80	15 22 31 40 41 48	7.61	0.00
0.2	8,425.99	0.03	8,425.99	15 22 31 40 41 48	8.94	0.00
0.3	9,275.99	0.02	9,275.99	15 22 31 40 41 48	10.62	0.00
0.4	10,253.90	0.02	10,253.90	15 22 31 40 41 42 48	10.38	0.00
0.5	11,603.00	0.03	11,603.00	15 22 31 35 40 41 42 48	10.86	0.00
0.6	13,416.80	0.06	13,416.80	2 12 15 22 31 35 40 41 43 48	12.84	0.00
0.7	16,157.20	0.09	16,157.20	2 12 14 15 19 22 31 35 40 41 43 45 48	13.47	0.00
0.8	21,500.70	0.20	21,500.70	2 12 14 15 19 20 21 22 26 31 35 36 40 41 43 45 48	14.08	0.00
0.9	35,987.70	0.48	35,987.70	1 2 10 11 12 14 15 17 19 20 21 22 23 24 26 27 31 35 36 40 41 43 45 48 49	14.27	0.00
1	128,009.00	0.00	128,009.00	No open facility	9.27	0.00

Table 5 General Case: SAA-H vs. Greedy Methods

	GAD-H		GADS-H		SAA-H ($N = 100$, $M = 1$)	
Date set no.	Objective	Time (s)	Objective	Time (s)	Objective	Time (s)
Node-20	5,761.79	0.00	5,761.79	0.00	5,761.79	3.09
Node-30	7,420.86	0.00	7,382.04	0.00	7,622.22	50.67
Node-40	7,474.92	0.00	7,474.92	0.02	7,474.92	54.20
Node-50	8,763.75	0.00	8,641.28	0.00	8,641.28	185.56
Node-60	9,357.37	0.00	9,357.37	0.02	9,394.87	159.48
Node-70	10,337.60	0.00	10,337.60	0.03	10,391.80	250.85
Node-80	11,054.30	0.00	11,054.30	0.05	11,054.30	320.82
Node-90	13,030.90	0.00	12,405.50	0.06	12,405.50	659.57
Node-100	14,463.40	0.02	13,820.87	0.09	14,028.10	2,164.03

are derived from the 100-node data set. Table 5 lists the objective values of solutions obtained by these three heuristics. The table contains only results from a one-run SAA-H with a sample size of $N = 100$. Multiple runs of SAA-H shows little improvement on the solution.

We can see that SAA-H spends much more time to find a few better solutions than GAD-H. GADS-H finds the best solutions among all three heuristics using far less time than SAA-H. We conclude that GADS-H is the most suitable heuristic for the URFLP according to our computational studies.

6. Conclusions

In this paper, we propose and analyze two formulations for the reliable facility location problem, where every open facility can fail with a certain probability. If a facility fails, customers originally assigned to it have to be reassigned to other facilities that are operational. This problem is NP-hard, and we propose a 4-approximation algorithm for the uniform case where all facilities have the same failing probability. We also propose several heuristics that can produce near-optimal solutions for the general problem.

It would be interesting to know whether there is a constant factor approximation algorithm for the URFLP with nonuniform failure probabilities. One possible approach is to solve the LP relaxation of (URFLP-2STG) and then round its optimal solution to integers. However, it is not clear if the LP relaxation itself can be solved to optimality in polynomial-time. Nonetheless, it might be possible to solve the LP relaxation approximately, possibly obtaining a polynomial time approximation scheme, following the approach proposed by Swamy and Shmoys (2006). This approach depends on the ability to compute an approximate subgradient of the objective function of (URFLP-2STG). This approach deserves further research.

There are several other interesting future research directions. We note that the main limitation of the current models is the assumption that the facilities are uncapacitated. Although the assumption itself is very common in the facility location models, it may be unrealistic in practice. In the capacitated case, customers of failed facilities can be assigned to the next-level backup facilities only if they have sufficient capacity to satisfy the additional demand. This makes the capacitated model very complex. We plan to investigate this direction in the near future. In addition, some new measurements of the reliability concept, especially in the facility location problem settings, are worth pursuing. Finally, we think it is important to design efficient and powerful metaheuristics tailored for the complex reliable facility location problems.

Acknowledgments

The authors thank the two anonymous reviewers and editor Karen Aardal for their helpful comments and suggestions, which have significantly improved the presentation of this paper. This work was supported in part by National Science Foundation CAREER Awards CMMI-0621433 and CMMI-0727640. This support is gratefully acknowledged.

References

Berman, O., D. Krass, M. B. C. Menezes. 2007. Facility reliability issues in network p-median problems: Strategic centralization and co-location effects. *Oper. Res.* **55**(2) 332–350.

Byrka, J. 2007. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. M. Charikar, K. Jansen, O. Reingold, J. D. P. Rolim, eds. *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*. Lecture Notes in Computer Science, Vol. 4627. Springer, Berlin, 29–43.

Cui, T., Y. Ouyang, Z. J. Shen. 2010. Reliable facility location design under the risk of disruptions. *Oper. Res.* **58**(4, Part 1) 998–1011.

Daskin, M. S. 1995. *Network and Discrete Location: Models, Algorithms, and Applications*. John Wiley & Sons, New York.

Daskin, M. S., K. Hogfan, C. ReVelle. 1997. α-Reliable p-minimax regres: A new model for strategic facility locaton modeling. *Location Sci.* **5**(4) 227–246.

França, P. M., H. P. L. Luna. 1982. Solving stochastic transportation-location problems by generalized benders decomposition. *Transportation Sci.* **16**(2) 113–126.

Gabor, A. F., J. C. W. van Ommeren. 2006. An approximation algorithm for a facility location problem with stochastic demands and inventory. *Oper. Res. Lett.* **34**(3) 257–263.

Guha, S., A. Meyerson, K. Munagala. 2003. A constant factor approximation algorithm for the fault-tolerant facility location problem. *J. Algorithms* **48**(2) 429–440.

Jain, K., M. Mahdian, E. Markakis, A. Saberi, V. V. Vazirani. 2003. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM* **50**(6) 795–824.

Kleywegt, A. J., A. Shapiro, T. Homem-de-Mello. 2001. The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.* **12**(2) 479–502.

Mahdian, M., Y. Ye, J. Zhang. 2006. Approximation algorithms for metric facility location problems. *SIAM J. Comput.* **36**(2) 411–432.

Mirchandani, R., R. Francis, eds. 1990. *Discrete Location Theory*. John Wiley & Sons, New York.

Owen, S. H., M. S. Daskin. 1998. Strategic facility location: A review. *Eur. J. Oper. Res.* **111**(3) 423–447.

Santoso, T., S. Ahmed, M. Goetshalckx, A. Shapiro. 2005. A stochastic programming approach for supply chain network design under uncertainty. *Eur. J. Oper. Res.* **167**(1) 96–115.

Scaparra, P. M., R. L. Church. 2005. An optimal approach for the interdiction median problem with fortification. Working paper, Kent Business School, University of Kent, Kent, UK.

Scaparra, M. P., R. L. Church. 2008. A bilevel mixed-integer program for critical infrastructure protection planning. *Comput. Oper. Res.* **35**(6) 1905–1923.

Shen, Z.-J. M. 2005. A profit-maximizing supply chain network design model with demand choice flexibility. *Oper. Res. Lett.* **34**(6) 673–682.

Shmoys, D. B., É. Tardos, K. Aardal. 1997. Approximation algorithms for facility location problems. *Proc. 29th ACM STOC, El Paso, TX*, ACM, New York, 265–274.

Snyder, L. V. 2006. Facility location under uncertainty: A review. *IIE Trans.* **38**(7) 537–554.

Snyder, L. V., M. S. Daskin. 2005. Reliability models for facility location: The expected failure cost case. *Transportation Sci.* **39**(3) 400–416.

Snyder, L. V., P. M. Scaparra, M. S. Daskin, R. L. Church. 2006. Planning for disruptions in supply chain networks. M. P. Johnson, B. Norman, N. Secomandi, eds. *TutORials in Operations Research: Models, Methods and Applications for Innovative Decision Making*. INFORMS, Hanover, MD, 234–257.

Swamy, C., D. B. Shmoys. 2006. Approximation algorithms for 2-stage stochastic optimization problems. *ACM SIGACT News* **37**(1) 33–46.

Swamy, C., D. B. Shmoys. 2008. Fault-tolerant facility location. *ACM Trans. Algorithms* **4**(4) Article 51.

Verweij, B., S. Ahmed, A. J. Kleywegt, G. Nemhauser, A. Shapiro. 2003. The sample average approximation method applied to stochastic routing problems: A computational study. *Comput. Optim. Appl.* **24**(2–3) 289–333.

Zeng, T., J. E. Ward. 2005. The stochastic location-assignment problem on a tree. *Ann. Oper. Res.* **136**(1) 81–97.