Computers & Operations Research 40 (2013) 2128–2137

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Robust vertex *p*-center model for locating urgent relief distribution centers

Chung-Cheng Lu [a,c,*], Jiuh-Biing Sheu [b,1]

[a] Department of Transportation Technology and Management, National Chiao-Tung University, 1001, Ta-Hsueh Road, Hsinchu 300, Taiwan
[b] Department of Business Administration, National Taiwan University, 1, Section 4, Roosevelt Road, Taipei 106, Taiwan
[c] Institute of Information and Logistics Management, National Taipei University of Technology, 1, Section 3, Chung-Hsiao East Road, Taipei 106, Taiwan

ARTICLE INFO

Available online 27 February 2013

Keywords:
Emergency logistics
Facility location
Data uncertainty
Robust optimization

ABSTRACT

This work locates urgent relief distribution centers (URDCs) on a given set of candidate sites using a robust vertex *p*-center (RVPC) model. This model addresses uncertain travel times, represented using fixed intervals or ranges instead of probability distributions, between URDCs and affected areas. The objective of locating a predetermined number (p) of URDCs is to minimize worst-case deviation in maximum travel time from the optimal solution. To reduce the complexity of solving the RVPC problem, this work proposes a property that facilitates identification of the worst-case scenario for a given set of URDC locations. Since the problem is *NP*-hard, a heuristic framework is developed to efficiently obtain robust solutions. Then, a specific implementation of the framework, based on simulated annealing, is developed to conduct computational experiments. Experimental results show that the proposed heuristic is effective and efficient in obtaining robust solutions of interest. This work examines the impact of the degree of data uncertainty on the selected performance measures and the tradeoff between solution quality and robustness. Additionally, this work demonstrates the applicability of the proposed model to natural disasters based on a real-world instance. The result is compared with that obtained by a scenario-based, two-stage stochastic model. This work contributes significantly to the growing body of literature applying robust optimization approaches to emergency logistics.

1. Introduction

Emergency response to quick-onset disasters relies heavily on effective and efficient emergency logistics, which have drawn considerable attention in the last decade (e.g., [1–3]) due to the severe impacts of numerous natural disasters and the devastation and casualties they have caused [4]. To support emergency responses, such as evacuation of survivors, search and rescue of the injured, and distribution of medical and relief supplies, all components in an emergency logistics system must be designed and deployed optimally, along with mechanisms that trigger and coordinate activities in and among those components. Of vital importance to an emergency logistics system are urgent relief distribution centers (URDCs), because they serve as hubs with the aim of seamlessly integrating and coordinating inbound and outbound emergency logistics in response to relief demands from affected areas. These hubs also have an inventory management function (*i.e.*, risk pooling)—aggregating relief demands (or their forecasts) across several affected areas to reduce the adverse impact of relief demand variability and uncertainty on the system.

Recognizing that facility location is an essential design variable for URDCs, this work focuses on the URDC location problem. Specifically, this work aims at developing a robust URDC location model that explicitly accounts for uncertain travel times between URDCs and affected areas. Among various facility location models that have been presented in the literature (e.g., [5,6]), the *p*-center model, which aims to locate p facilities to minimize maximum distance (or travel time) between demand nodes and their closest facilities (e.g., [7]), is particularly suitable for emergency applications (e.g., [8–10]). The vertex *p*-center (VPC) model restricts the set of candidate sites to network nodes, while the absolute *p*-center model allows facilities to be anywhere along network arcs. In response to quick-onset disasters, government agencies typically designate existing public buildings (e.g., schools and stadiums) with little or no damage that can be promptly converted to shelters for survivors and/or warehouses for relief supplies as candidate sites instead of establishing new emergency facilities from scratch. Thus, this work considers the URDC location problem as a VPC problem (e.g., [11]).

A robust location model of URDCs must explicitly account for uncertain input data, such as travel times between URDCs and

* Corresponding author. Tel: +886 2 27712171x2306; fax: +886 2 87726946.
E-mail addresses: jasoncclu@gmail.com (C.-C. Lu), jbsheu@ntu.edu.tw (J.-B. Sheu).
[1] Tel: +886 2 33661069; fax: +886 2 23625379.

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.02.019

affected areas, mainly due to poor measurements based on limited information available during a disaster's aftermath or approximations in the modeling process choosing a distance norm. Two major categories of approaches have been adopted in the literature to deal with uncertain coefficients in facility location models [12], namely, stochastic programming (SP) and robust optimization (RO). The former has been used typically to deal with decision-making for facility locations in risk situations, in which the values of uncertain coefficients are governed by discrete or continuous probability distributions that are known to a decision-maker. The SP approach has been widely applied to emergency logistics for short-notice disasters (e.g., hurricanes, flooding, and wild fires) by assuming that possible impacts of these disasters can be estimated based on historical and meteorological data. A classical example of applying SP to disaster relief is the scenario-based, two-stage stochastic model proposed by Mete and Zabinsky [13], for medical supply location and distribution in disaster management. Other examples can be found, for instance, in [14–16]. A common goal of stochastic location models is to optimize the expected value of a given objective function.

On the other hand, the RO approach attempts to optimize the worst-case system performance in uncertain situations that lack any information about the probability distributions of uncertain coefficients; hence, the RO approach generally describes uncertain data using pre-specified intervals or ranges (e.g., [17–21]). Typical robustness measures include mini-max objective value and mini-max regret in an objective value. The RO approach may be more appropriate in response to quick-onset or no-notice disasters (e.g., earthquakes, tsunamis, and landslides); however, to the best of our knowledge, it has rarely been applied to this context. For quick-onset disasters, because of the difficulty in predicting disaster occurrence and impacts as well as a lack of historical data, probability distributions and scenario data are generally unavailable. For example, an extremely large earthquake, 9.1 on the Richter scale, which hit the northeastern coast of Japan on March 11, 2011, was never considered in that nation's preparedness planning for earthquakes, even though Japan is widely regarded as one of the most advanced countries in earthquake preparedness. Thus, in responding to such a disaster, decision-makers may prefer an alternative method for describing uncertain data (*i.e.*, using intervals to represent uncertain data). The selection of a solution technique (*i.e.*, SP or RO) depends mostly on data availability and the decision-maker's objective.

The p-center problems with interval-represented uncertain data tend to be very difficult because of the mini-max structure. Therefore, analytical results and exact algorithms for the p-center problems with interval data have only been attained in special cases, such as locating a single facility on general networks or multiple facilities on tree networks (e.g., [22,23]). To the best of our knowledge, only Averbakh and Berman [24] reported analytical results for an absolute weighted p-center problem with interval-represented node weights. No study has addressed absolute or vertex multi-center problems with interval-represented edge lengths.

This work develops a robust vertex p-center (RVPC) model for locating URDCs in an emergency logistics network. This model considers explicitly uncertain travel times between URDCs and affected areas. The objective of locating p URDCs is to minimize worst-case deviation in maximum travel time between URDCs and affected areas from the optimal solution. In this model, uncertain travel times are represented using prescribed, continuous intervals (or ranges), rather than probability distributions. This work also proposes a property that facilitates identification of the worst-case scenario for a given set of URDC locations, thereby reducing complexity of solving the problem. Since the problem is *NP*-hard [25], a local search-based algorithmic framework incorporating the property for identifying the worst-case scenarios is developed to find robust solutions within a reasonable amount of computational resources. Then, a specific framework implementation based on simulated annealing (SA) is developed to conduct numerical experiments, including a case study based on the Jiji Earthquake, which hit central Taiwan on September 21, 1999.

This study contributes significantly to literature by (i) modeling the URDC location problem as the vertex multi-center problem with interval-represented edge lengths on general networks; (ii) providing an effective and efficient algorithmic framework for solving these problems; and (iii) shedding light on the applicability and potential benefits of the proposed models to real-world instances.

The remainder of this paper is structured as follows. Section 2 describes the RVPC problem, the representation of data uncertainty, and the property of worst-case scenarios. Section 3 presents the generic heuristic framework and a specific implementation using SA. This is followed by the numerical experiments in Section 4. Section 5 provides a case study demonstrating the applicability of the proposed model to a real instance. Concluding remarks are given in Section 6.

2. Vertex *p*-center problem with data uncertainty

2.1. The deterministic problem

Consider a connected, undirected network $G(N, A)$, where N is the vertex set and A the arc (or edge) set. Let U be the set of candidate sites for URDC locations and V be the set of relief stations in affected areas; $U \cup V = N$, and $U \neq V$. Each possible pair of relief station $i \in V$ and URDC $j \in U$ is connected by an arc $(i, j) \in A$ that is associated with a positive (real or integer) number, t_{ij}, representing travel time between relief station i and URDC j. Each relief station is serviced only by a single URDC. For a given set of predetermined candidate sites, the VPC problem is to locate p ($p < |U|$) URDCs and assign relief stations to these centers, thereby minimizing maximum travel time between relief stations and URDCs. A mixed integer linear programming (MILP) formulation of the problem is as follows (e.g., [11])

$$\text{(VPC) Minimize } z \tag{1}$$

$$\text{Subject to } z \geq \sum_{j \in U} t_{ij} y_{ij}, \quad \forall i \in V \tag{2}$$

$$\sum_{j \in U} y_{ij} = 1, \quad \forall i \in V \tag{3}$$

$$y_{ij} - x_j \leq 0, \quad \forall i \in V, \quad j \in U \tag{4}$$

$$\sum_{j \in U} x_j = p \tag{5}$$

$$x_j \in \{0,1\}, \quad \forall j \in U \tag{6}$$

$$y_{ij} \in \{0,1\}, \quad \forall i \in V, \quad j \in \mathbf{U} \tag{7}$$

The decision variables are binary variables x_j, $\forall j \in U$ and y_{ij}, $\forall i \in V$, $j \in U$. $x_j = 1$ if candidate site j is selected; otherwise, $x_j = 0$. Additionally, $y_{ij} = 1$ if relief station i is serviced by URDC j; otherwise, $y_{ij} = 0$. The objective function (1) minimizes maximum travel time between each relief station and its closest URDC. Constraint (2) defines the lower bound of maximum travel time, which is being minimized. Constraint (3) requires that each relief station be assigned to exactly one URDC. Constraint (4) restricts relief station assignments only to open URDCs. Constraint (5) stipulates that p URDCs are to be located. Constraints (6) and (7) indicate that location and allocation decision variables are binary.

2.2. Representation of data uncertainty and the robust VPC problem

In the RVPC problem, both link travel times and nodal (relief) demands could be uncertain, but this current work only deals with uncertain link travel times. Uncertain travel times between relief stations and URDCs are described using intervals or ranges. Specifically, an interval $[tl_{ij}, tu_{ij}]$, $0 \le tl_{ij} \le tu_{ij}$, is used to capture the uncertainty of travel time between relief station i and URDC j. Let W be the Cartesian product of intervals $[tl_{ij}, tu_{ij}]$, $\forall i \in V, j \in U$. A scenario $w \in W$ is defined as a realization of travel times, $t_{ij}(w) \in [-tl_{ij}, tu_{ij}]$, $\forall i \in V$, and $j \in U$, where $t_{ij}(w)$ denotes the travel time between station i and URDC j in scenario w. Let $\boldsymbol{\tau} = (\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} = \{x_j, j \in U\}$ and $\boldsymbol{y} = \{y_{ij}, i \in V, j \in U\}$, be a feasible solution to the VPC problem (*i.e.*, satisfying Constraints (3)–(7)), and let Ω be the set of feasible solutions. In this study, $\boldsymbol{\tau} \in \Omega$ is called a URDC location plan. For a plan $\boldsymbol{\tau}$ and a relief station $i \in V$, this work defines the travel time between i and $\boldsymbol{\tau}$ in scenario w as

$$d(w,i,\tau) = \text{Min}_{x_j = 1, \forall j \in U}\{t_{ij}(w)\}. \tag{8}$$

This definition requires that each relief station be serviced by its closest URDC. For a plan $\boldsymbol{\tau}$ and scenario w, the maximum travel time between plan $\boldsymbol{\tau}$ and relief stations in scenario w, $z(w, \boldsymbol{\tau})$, is given as follows:

$$z(w, \boldsymbol{\tau}) = \text{Max}_{i \in V} d(w, i, \boldsymbol{\tau}). \tag{9}$$

For a given scenario w, the deterministic VPC problem, as presented in Section 2.1, can be written

$$\text{VPC}(w) = \text{Minimize}_{\boldsymbol{\tau} \in \Omega} z(w,\boldsymbol{\tau}) \tag{10}$$

This work defines the robust deviation of plan $\boldsymbol{\tau}$ in scenario w, $dev(w, \boldsymbol{\tau})$, as the difference between maximum travel time of plan $\boldsymbol{\tau}$ and that of the optimal plan, $\boldsymbol{\tau}^*(w)$, in scenario w.

$$dev(w,\boldsymbol{\tau}) = z(w,\boldsymbol{\tau}) - z(w,\boldsymbol{\tau}^*(w)). \tag{11}$$

This robust deviation represents the regret (or opportunity loss) of adopting plan $\boldsymbol{\tau}$ instead of the optimal plan $\boldsymbol{\tau}^*(w)$ in scenario w. Further, for a given plan $\boldsymbol{\tau} \in \Omega$, the maximum (or worst-case) deviation is called robustness cost, $rc(\boldsymbol{\tau})$, which can be obtained by solving the following sub-problem:

$$rc(\boldsymbol{\tau}) = \text{Max}_{w \in W} dev(w,\boldsymbol{\tau}). \tag{12}$$

The RVPC problem can be formally stated as

$$\text{(RVPC) Minimize}_{\boldsymbol{\tau} \in \Omega} rc(\boldsymbol{\tau}). \tag{13}$$

This is equivalent to finding $\boldsymbol{\tau}_{\text{robust}} = \text{argMin}_{\boldsymbol{\tau} \in \Omega}\, rc(\boldsymbol{\tau})$. That is, the RVPC problem is to find a robust solution $\boldsymbol{\tau}_{\text{robust}}$ that minimizes worst-case deviation in maximum travel time from the optimal solution.

2.3. A key property of the RVPC problem

One of the major difficulties in solving the RVPC problem, a min–max (or minimax) combinatorial optimization problem, is evaluating the robustness cost of a given plan $\boldsymbol{\tau}$, $rc(\boldsymbol{\tau})$. To determine robustness cost, one must identify the worst-case scenario for a given plan $\boldsymbol{\tau}$, which is particularly difficult when uncertain data are represented using (continuous) intervals, because interval representation implies an infinite number of possible scenarios. To facilitate the identification of the worst-case scenario for a given plan $\boldsymbol{\tau}$, this work presents a property that increases the tractability of the RVPC problem from a combinatorial perspective. Specifically, this property indicates that, when evaluating the robustness cost of a plan $\boldsymbol{\tau}$, even when uncertain travel times are specified as independent ranges, attention can be restricted to a finite set of discrete scenarios selected appropriately.

Theorem 1. Given a plan $\boldsymbol{\tau}$, the worst-case scenario, wc, which maximizes the robust deviation of $\boldsymbol{\tau}$, can be determined as follows: for each arc (i, j), $t_{ij}(wc) = tu_{ij}$ if $y_{ij} = 1$ and $x_j = 1$ (*i.e.*, station i is serviced by URDC j, which is denoted as $(i, j) \in \boldsymbol{\tau}$); otherwise, $t_{ij}(wc) = tl_{ij}$ (*i.e.*, $y_{ij} = 0$, or station i is not serviced by URDC j, which is denoted as $(i, j) \notin \boldsymbol{\tau}$).

Proof. Let w be a generic scenario. One must prove that $dev(w, \boldsymbol{\tau}) \le dev(wc, \boldsymbol{\tau})$, $\forall w \in W$; that is, for plan $\boldsymbol{\tau}$, scenario wc is the worst-case scenario, which has the maximum deviation over all possible scenarios in W. According to the definition of robust deviation in Eqs. (9) and (11),

$$dev(w,\boldsymbol{\tau}) = z(w,\boldsymbol{\tau}) - z(w,\boldsymbol{\tau}^*(w)) = \text{Max}_{i \in V} d(w,i,\boldsymbol{\tau}) - \text{Max}_{i \in V} d(w,i,\boldsymbol{\tau}^*(w)),$$

and

$$dev(wc,\boldsymbol{\tau}) = z(wc,\boldsymbol{\tau}) - z(wc,\boldsymbol{\tau}^*(wc)) = \text{Max}_{i \in V} d(wc,i,t) - \text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}^*(wc)),$$

where $\boldsymbol{\tau}*(w)$ and $\boldsymbol{\tau}*(wc)$ are the optimal plan in scenario w and worst-case scenario wc, respectively. Firstly, by definition of the worst-case scenario wc given in Theorem 1

$$\tau_{ij}(wc) \ge \tau_{ij}(w), \forall (i,j) \in \boldsymbol{\tau} \backslash \boldsymbol{\tau}^*(w) \text{ and } \tau_{ij}(wc) \le \tau_{ij}(w), \forall (i,j) \in \boldsymbol{\tau}^*(w) \backslash \boldsymbol{\tau}.$$

Thus, the robust deviation of plan $\boldsymbol{\tau}$ in scenario w is smaller than or equal to the robust deviation of plan $\boldsymbol{\tau}$ in scenario wc. That is

$$dev(w,\boldsymbol{\tau}) = \text{Max}_{ii \in V} d(w,i,\boldsymbol{\tau}) - \text{Max}_{i \in V} d(w,i,\boldsymbol{\tau}^*(w)) \le \text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}) - \text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}^*(w)).$$

Furthermore, according to the definition in Eq. (8), the maximum travel time between plan $\boldsymbol{\tau}*(w)$ and relief stations in worst-case scenario wc is greater than or equal to the maximum travel time between the *optimal* plan $\boldsymbol{\tau}*(wc)$ in worst-case scenario wc, i.e.,

$$\text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}^*(w)) \ge \text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}^*(wc))$$

Thus,

$$\text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}) - \text{Max}_{ii \in V} d(wc,i,\boldsymbol{\tau}^*(w)) \le \text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}) - \text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}^*(wc)).$$

Finally, we can obtain that

$$dev(w,\boldsymbol{\tau}) \le \text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}) - \text{Max}_{i \in V} d(wc,i,\boldsymbol{\tau}^*(wc)) = z(wc,\boldsymbol{\tau}) - z(wc,\boldsymbol{\tau}^*(wc)) = dev(wc,\boldsymbol{\tau}).$$

This completes the proof. □

3. Solution algorithm

3.1. A local search-based algorithmic framework

Since the RVPC problem is *NP*-hard [25], to obtain robust solutions with a reasonable amount of computational resources for problem instances with practical sizes, heuristics or meta-heuristics are typically adopted. Based on the property presented in Theorem 1, the following local search-based algorithmic framework is proposed.

Step 1: Initialization

1.1 Generate randomly an initial solution $\boldsymbol{\tau}_0$; let $\boldsymbol{\tau} = \boldsymbol{\tau}_0$.
1.2 Evaluate the robustness cost of $\boldsymbol{\tau}$, $rc(\boldsymbol{\tau})$, based on Theorem 1.

Step 2: Local search

2.1 Generate a new solution, $\boldsymbol{\tau}_{\mathbf{new}}$, from the neighborhood of $\boldsymbol{\tau}$.
2.2 Evaluate the robustness cost of $\boldsymbol{\tau}_{\mathbf{new}}$, $rc(\boldsymbol{\tau}_{\mathbf{new}})$, based on Theorem 1.

Step 3: Solution acceptance or rejection

3.1 If the rules (or aspiration rules) of solution acceptance are adopted, then let $\tau = \tau_{\mathbf{new}}$.
3.2 Otherwise, decline the new solution, $\tau_{\mathbf{new}}$.

Step 4: Convergence check

4.1 If convergence criteria are satisfied, stop.
4.2 Otherwise, go to Step 2.

In this framework, each candidate solution has more than one neighbor solution, and the choice of which neighbor solution to move is determined using only the information about the neighborhood of the current solution (*i.e.*, local search). When no improvement mechanisms are designed for the neighborhood search, a local search may be stuck at local optima. This issue can be resolved by applying, for instance, restarts with different initial solutions or relatively more sophisticated schemes based on iterations (e.g., iterated greedy) or memory-less stochastic modifications (e.g., SA) in Step 3 of the proposed framework.

3.2. A specific implementation based on simulated annealing

This subsection presents a specific implementation of the proposed framework based on SA, which can escape from becoming trapped into a local optimum by accepting, with a small probability, worse solutions during iterations. Suman and Kumar [26] comprehensively reviewed SA-based optimization algorithms. The SA-based heuristic is presented as follows.

***Step* 0**. Input data and set parameters values p, T_0, T_F, $Ite_{\max}$, $Num_{\max}$, and β;

***Step* 1**. Initialization

1.1 Randomly generate the initial solution τ_0; $\tau = \tau_0$;
1.2 Initialize $Temp := T_0$, $Ite := 0$, $Num := 0$, $\tau_{\text{robust}} := \tau$, $rc(\tau_{\text{robust}}) := rc(\tau)$;

***Step* 2**. Generate a solution τ_{new}, and evaluate its robustness cost, $rc(\tau_{\text{new}})$; $Ite := Ite + 1$;

***Step* 3**. $\Delta E := rc(\tau_{\text{new}}) - rc(\tau)$; if $\Delta E \leq 0$, go to Step 3.1; otherwise, go to Step 3.2;

3.1 Let $\tau := \tau_{\text{new}}$;
3.2 Generate a random number $rand \sim U(0, 1)$;If $rand < (Temp/(Temp^2 + \Delta E^2))$, then $\tau = \tau_{\text{new}}$;

***Step* 4**. If $rc(\tau) < rc(\tau_{\text{robust}})$, then $\tau_{\text{robust}} := \tau$, $rc(\tau_{\text{robust}}) := rc(\tau)$, $Num := 0$;

***Step* 5**. If $Ite = Ite_{\max}$, then $Temp := Temp \times \beta$, $Ite := 0$, $Num := Num + 1$;

***Step* 6**. If $Temp < T_F$ or $Num := Num_{\max}$, then stop; otherwise, go to Step 2.

Note that the SA-based heuristic is provided only as a specific implementation for the framework, presented in Section 3.1. Any other local search-based heuristic can be adopted in the proposed framework, such as iterated greedy. Moreover, while SA is not a new approach, this work demonstrates a successful application of SA (or meta-heuristic) to solve robust combinatorial optimization problems, which is rarely seen in existing literature.

3.3. Neighborhood description and evaluation of robustness cost

In Step 2 of the SA-based heuristic, a new solution τ_{new} is generated in each iteration from the neighborhood of the current solution τ. Whenever a new solution τ_{new} is generated, one must evaluate its robustness cost, $rc(\tau_{\text{new}})$. This work defines two neighborhood types. The first is called the *allocation* neighborhood, and involves only changes in allocation decision variables (*i.e.*, y_{ij}, $i \in V$, $j \in U$). Both location and allocation decision variables (*i.e.*, x_j, $j \in U$, and y_{ij}, $i \in V$, $j \in U$) change in the second type, called the *Location–Allocation* neighborhood. Specifically, the allocation neighborhood consists of feasible solutions obtained by randomly selecting two relief stations serviced by different URDCs in the current solution, and swapping their associated URDCs (Fig. 1(a)). Further, to accelerate the generation of a feasible solution from the allocation neighborhood, a dominance-checking rule is applied to eliminate impossible swaps. Consider a selected relief station i, which is to be reallocated from its currently associated URDC $j1$, to another URDC $j2$. This rule compares travel time intervals $[tl_{ij1}, tu_{ij1}]$ and $[tl_{ij2}, tu_{ij2}]$. If $tl_{ij2} > tu_{ij1}$, the latter interval is said to be dominated by the former interval and the intended reallocation is abandoned because relief stations are assumed to be serviced by their respective closest URDCs, such that i will never be connected to $j2$.

The new solutions in the Location–Allocation neighborhood are generated by randomly choosing two URDCs and swapping their associated groups of relief stations. In this neighborhood type, at least one chosen URDC must be an open facility. Moreover, if only one chosen URDC is open, this swap is equivalent to moving a group of relief stations from an open facility to another facility that was closed but is now open after the swap (Fig. 1(b)). In each iteration of the SA-based heuristic, a new solution τ_{new} is generated from either the allocation (with probability p_1) or Location–Allocation (with probability p_2) neighborhood of current solution τ. Selection between the two types of neighborhood is based on probabilities p_1 and p_2, and $p_1 + p_2 = 1$.

The robustness cost of a solution τ is the regret of using plan τ instead of the optimal solution in the worst-case scenario. Two steps exists for evaluating $rc(\tau)$: (i) identify the worst-case scenario of τ; and, (ii) find the optimal solution to this worst-case scenario. In the RVPC problem, although the interval representation of uncertain data implies an infinite number of possible

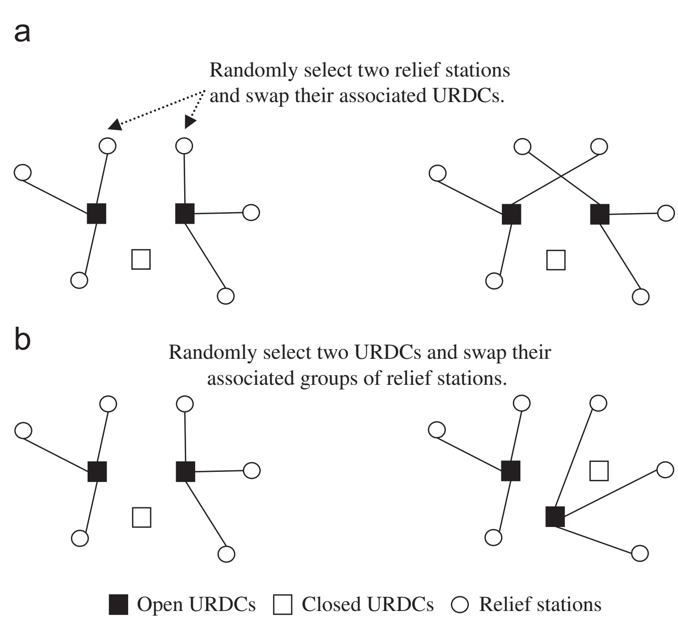

Fig. 1. Illustrations of different neighborhood types. (a) Illustration of type 1 neighborhood and (b) Illustration of type 2 neighborhood.

scenarios, the worst-case scenario can be identified easily using Theorem 1, which was presented in Section 2.3. That is, the worst-case scenario of (or induced by) a given plan $\boldsymbol{\tau}$ is obtained by setting travel times to the upper bounds for links connecting open URDCs and their associated relief stations, and to the lower bounds for the remaining links. The efficient algorithm proposed by Chen and Chen [27] can then be adopted to solve the deterministic VPC problem (*i.e.*, to obtain the optimal location plan URDCs) in the identified worst-case scenario.

4. Numerical experiments

4.1. Experimental design

A set of numerical experiments were conducted on a set of test instances, to examine the algorithmic performance of the SA-based heuristic, the tradeoff between robustness and optimality, and the impact of both number of facilities and data uncertainty on the solutions. The algorithm was coded in C++ computer language and tested on a personal computer with a Pentium Core 2 Duo 2.4 GHz CPU and 2 GB RAM.

To examine the performance of the SA-based heuristic, this work also implemented an enumeration approach and compared the effectiveness (solution quality) and efficiency (computational time) of the SA-based heuristic with those of the enumeration approach. This approach enumerates all possible plans and identifies the plan with minimal objective value. In spite of its computational inefficiency, this enumeration approach guarantees to find exact solutions that serve as the benchmark for evaluating the solution quality of the heuristic.

The test instances have different problem sizes, represented by the triplet $(|V|, |U|, p)$, where $|V|$, $|U|$, and p denote the numbers of relief stations, candidate URDC sites, and open URDC sites, respectively. Fifteen different sizes of instances were considered. For each problem size, 30 instances were generated, so there were 450 problem instances. For each instance, the two-dimensional coordinates of relief stations and candidate URDC sites were generated from the intervals (0, 100) and (40, 60), respectively. The nominal travel time t_{ij} between each pair of relief station i and candidate URDC site j is given as the Euclidean distance rounded to the nearest integer. Because of the likely degradation of road condition in the aftermath of disasters, the travel time interval $[tl_{ij}, tu_{ij}]$ is generated as $[t_{ij}, t_{ij}+\alpha t_{ij}]$, $\forall i, j$. The parameter α is used to control the degree of data uncertainty; the larger the value of α, the higher the degree of data uncertainty. Each problem instance was tested for five different uncertainty levels, α=0.5, 1.0, 1.5, 2.0, and 2.5, so there were 2250 (=450 × 5) tests.

In addition to the objective function values, defined in Eq. (12), the price of robustness $\eta(\boldsymbol{\tau}_{\text{robust}})$ and hedge value $H(\boldsymbol{\tau}_{\text{robust}})$ are the other two performance measures. $\eta(\boldsymbol{\tau}_{\text{robust}})$ is equal to the price that the decision-maker needs to pay for employing the robust plan $\boldsymbol{\tau}_{\text{robust}}$, instead of the optimal nominal plan $\boldsymbol{\tau}_{\text{nominal}}$, in the scenario of nominal travel times, w_{nominal}. Specifically

$$\eta(\boldsymbol{\tau}_{\text{robust}}) = z(w_{\text{nominal}}, \boldsymbol{\tau}_{\text{robust}}) - z(w_{\text{nominal}}, \boldsymbol{\tau}_{\text{nominal}}), \tag{14}$$

where $z(w_{\text{nominal}}, \boldsymbol{\tau}_{\text{nominal}})$ is the maximum travel time between the URDCs of the optimal nominal plan $\boldsymbol{\tau}_{\text{nominal}}$ and relief stations in the nominal scenario w_{nominal}. In the conducted numerical experiments, the nominal travel times were set as the mean or middle values of the generated travel time intervals.

$H(\boldsymbol{\tau}_{\text{robust}})$ is defined as the value gained from implementing the robust plan $\boldsymbol{\tau}_{\text{robust}}$, instead of the optimal nominal plan $\boldsymbol{\tau}_{\text{nominal}}$, in the worst-case scenarios. Specifically

$$H(\boldsymbol{\tau}_{\text{robust}}) = rc(\boldsymbol{\tau}_{\text{nominal}}) - rc(\boldsymbol{\tau}_{\text{robust}}). \tag{15}$$

In the definitions, $\eta(\boldsymbol{\tau}_{\text{robust}})$ represents the tradeoff between robustness and optimality, while $H(\boldsymbol{\tau}_{\text{robust}})$ can be viewed as the regret of employing the plan $\boldsymbol{\tau}_{\text{nominal}}$ in the worst-case scenario.

4.2. Numerical results of solving the RVPC problem

4.2.1. Effectiveness and efficiency of the SA-based heuristic

The computational results of solving the RVPC problem instances with the number of relief stations $|V|$=10, 15, and 20 are displayed in Tables 1, 2, and 3, respectively. In these tables, Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$ denotes the average robustness cost, over 30 instances, for each problem size and uncertainty level. As shown in the third and fifth columns of these tables, for most problem sizes and uncertainty levels (α), the Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$ obtained using the SA-based heuristic is the same as that obtained using the enumeration approach. For each problem size and each uncertainty level, the number of instances (out of 30 instances) where the SA-based heuristic fails to obtain optimal solution is reported in the parenthesis of the third column in the tables. In all, but 19, of the 500 instances (excluding the instances of problem sizes (20, 4, 3) and (20, 5, 3)), the objective value obtained using the SA-based heuristic is identical to that obtained using the enumeration approach, indicating that the proposed heuristic is able to obtain optimal, or near-optimal, solutions. In the 19 instances, where the heuristic failed to obtain optimal solution, the maximum gap between heuristic solutions and optimal solutions (obtained by the enumeration) is about 10% of the corresponding optimal solution. The average gap of the 19 instances is about 6%. These small deviations further highlight the effectiveness of the proposed heuristic.

Note that, since the estimated computational time for the enumeration approach to optimally solve one instance with size

Table 1
Comparison of algorithmic performance for the RVPC instances with $|V|$=10.

Problem size	α	SA-based heuristic		Enumeration	
		Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$	CPU time (s)	Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$	CPU time (s)
(10, 4, 2)	0.5	21.02 (0)	1.35	21.02	3.89
	1.0	46.83 (0)		46.83	
	1.5	72.65 (0)		72.65	
	2.0	98.47 (1)		98.23	
	2.5	124.28 (0)		124.28	
(10, 4, 3)	0.5	20.90 (0)	1.46	20.90	13.20
	1.0	46.07 (0)		46.07	
	1.5	71.52 (1)		71.43	
	2.0	97.30 (1)		97.20	
	2.5	123.47 (0)		123.47	
(10, 5, 2)	0.5	21.13 (0)	1.38	21.13	4.37
	1.0	46.60 (0)		46.60	
	1.5	72.07 (0)		72.07	
	2.0	97.53 (0)		97.53	
	2.5	123.00 (0)		123.00	
(10, 5, 3)	0.5	21.03 (0)	1.72	21.03	59.48
	1.0	46.57 (2)		46.43	
	1.5	70.92 (2)		70.85	
	2.0	97.23 (1)		97.17	
	2.5	121.63 (1)		121.53	
(10, 5, 4)	0.5	19.20 (1)	2.39	19.15	407.79
	1.0	46.10 (0)		46.10	
	1.5	70.40 (1)		70.15	
	2.0	96.20 (0)		96.20	
	2.5	120.60 (1)		120.25	

equal to (20, 4, 3) or (20, 5, 3) is more than one week, it is very difficult to use the enumeration approach to obtain exact solutions for all the larger instances. Alternatively, a maximum computation time (5 days) was set to allow the enumeration approach to solve one instance with size equal to (20, 4, 3) or (20, 5, 3) and report the incumbent solution (i.e., an upper bound) obtained within that maximum computation time. As shown in Table 3, for the tests on these larger instances, the solution obtained using the SA-based heuristic is significantly better than that obtained using the enumeration approach, with the improvement in objective value ranging from 15 to 50%.

Table 2
Comparison of algorithmic performance for the RVPC instances with $|V|=15$.

Problem size	α	SA-based heuristic		Enumeration	
		Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$	CPU time (s)	Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$	CPU time (s)
(15, 4, 2)	0.5	21.67 (0)	2.51	21.67	22.072
	1.0	48.33 (0)		48.33	
	1.5	75.03 (0)		75.03	
	2.0	101.73 (0)		101.73	
	2.5	128.43 (0)		128.43	
(15, 4, 3)	0.5	20.95 (1)	3.56	20.85	6161.899
	1.0	46.90 (0)		46.90	
	1.5	74.50 (1)		74.00	
	2.0	100.10 (0)		100.10	
	2.5	125.90 (1)		125.20	
(15, 5, 2)	0.5	22.00 (0)	2.71	22.00	49.554
	1.0	48.50 (0)		48.50	
	1.5	74.98 (0)		74.98	
	2.0	101.50 (0)		101.50	
	2.5	127.32 (0)		127.32	
(15, 5, 3)	0.5	21.70 (1)	3.77	21.60	25649.025
	1.0	47.10 (1)		46.90	
	1.5	72.40 (0)		72.40	
	2.0	98.20 (1)		97.90	
	2.5	123.75 (1)		123.40	

Table 3
Comparison of algorithmic performance for the RVPC instances with $|V|=20$.

Problem size	α	SA-based heuristic		Enumeration	
		Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$	CPU time (s)	Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$	CPU time (s)
(20, 4, 2)	0.5	23.65 (0)	3.68	23.65	779.588
	1.0	52.50 (0)		52.50	
	1.5	81.35 (0)		81.35	
	2.0	110.20 (0)		110.20	
	2.5	139.05 (0)		139.05	
(20, 4, 3)	0.5	21.50 (N/A)	4.74	28.50[a]	5 days
	1.0	49.00 (N/A)		61.00[a]	
	1.5	76.50 (N/A)		91.50[a]	
	2.0	104.00 (N/A)		122.00[a]	
	2.5	131.50 (N/A)		152.50[a]	
(20, 5, 2)	0.5	23.75 (0)	3.93	23.75	1875.083
	1.0	52.30 (0)		52.30	
	1.5	80.85 (0)		80.85	
	2.0	109.40 (0)		109.40	
	2.5	137.95 (0)		137.95	
(20, 5, 3)	0.5	18.50 (N/A)	5.57	26.50[a]	5 days
	1.0	43.00 (N/A)		67.00[a]	
	1.5	69.00 (N/A)		97.50[a]	
	2.0	95.00 (N/A)		128.00[a]	
	2.5	121.00 (N/A)		158.50[a]	

[a] Incumbent solution obtained using the enumeration approach in 5 days

Regarding the computational efficiency of the proposed heuristic, as shown in the fourth and sixth columns of these tables, the SA-based heuristic requires less computational time than the enumeration approach in all the conducted tests. In particular, although the computational time of the enumeration approach increases dramatically as the problem size gets larger, the increase in computational time of the SA-based heuristic is not significant. For instance, for the test with the same numbers of $|V|=15$ and $|U|=5$, when p increases from two to three, the computational time of the enumeration approach increases, on average, by more than 500 times, but the SA-based heuristic requires an average of one more second. In summary, the SA-based heuristic obtains optimal or near-optimal solutions using much less computational time than the enumeration approach.

4.2.2. *Impacts of data uncertainty*

The tests conducted on the problem instances of larger size, i.e., (30, 5, 3), (40, 8, 4), and (50, 10, 5), aim to examine the impact of data uncertainty on the selected performance measures. The average optimal objective value (Ave. $rc(\boldsymbol{\tau}_{\text{robust}})$), robustness price (Ave. $\eta(\boldsymbol{\tau}_{\text{robust}})$), and hedge value (Ave. $H(\boldsymbol{\tau}_{\text{robust}})$), for different uncertainty levels, are reported in Table 4. The nominal values used in evaluating the robustness price and hedge value are the middle (or mean) values of the generated travel time intervals. As observed in the table, both the average robustness cost and hedge value increase as the uncertainty level becomes higher. Because of the larger travel time intervals (due to a larger α), the worst-case scenario deviates further from the nominal scenario, resulting in larger robust deviations and the increase in hedge value, which highlights the advantage of implementing robust solutions in the presence of data uncertainty. It is also shown in the table that, although the average robustness price, $\eta(\boldsymbol{\tau}_{\text{robust}})$, increases slightly with problem size and uncertainty level, the value is small (about 10% of $z(w_{\text{nominal}}, \boldsymbol{\tau}_{\text{nominal}})$) for all problem sizes and uncertainty levels, indicating that the robust plans determined by the proposed method do not sacrifice much solution quality (or optimality) for robustness.

5. Numerical example based on a real case

This numerical example demonstrates the application of the proposed RVPC model to locate URDCs in a relief distribution

Table 4
Computational results of the SA-based heuristic on solving larger RVPC instances.

Problem size	α	$rc(\boldsymbol{\tau}_{\text{robust}})$	$\eta(\boldsymbol{\tau}_{\text{robust}})$	$H(\boldsymbol{\tau}_{\text{robust}})$	CPU time
(30, 5, 3)	0.5	24.5	6.3	44.7	6.48
	1.0	53.6	6.3	75.3	
	1.5	82.5	6.5	108.5	
	2.0	111.5	6.7	147.1	
	2.5	140.9	6.9	175.2	
(40, 8, 4)	0.5	27.9	6.5	42.3	16.68
	1.0	56.7	6.6	74.9	
	1.5	85.5	6.7	107.1	
	2.0	115.8	6.9	139.8	
	2.5	144.8	7.6	174.5	
(50, 10, 5)	0.5	27.9	6.9	43.8	59.36
	1.0	57.9	7.3	77.3	
	1.5	87.3	7.5	110.8	
	2.0	118.7	7.9	145.3	
	2.5	148.6	8.4	178.4	

system responding to the massive earthquake that hit central Taiwan on September 21, 1999—the 921 Jiji Earthquake. This earthquake, which measured 7.3 on the Richter scale, mostly affected Taichung and Nantou counties, causing more than 2500 deaths and 8000 injuries and destroying (completely or partially) 39,000 buildings. The focus of this example is on the three-tier relief distribution system established in Nantou County immediately after this earthquake. Specifically, relief supplies were transported from six unaffected counties (Taipei, Taoyuan, Hsinchu, Changhua, Tainan, and Kaohsiung counties) and sent to the 51 relief stations in the 11 townships in Nantou county. To centralize relief distribution to relief stations in the relief distribution system, the rescue and relief agency selected Nantou Stadium and Jiji Town Hall as URDCs. In this case study, these two buildings were also considered candidate sites for URDCs. Five other candidate sites were selected based on an earthquake preparedness report by Taiwan's Ministry of the Interior [28]. The triplet $(|V|, |U|, p)=(51, 7, 2)$ denotes the problem size in this numerical example, where $|V|$, $|U|$ and p denote the number of relief stations, candidate URDC sites, and open URDC sites, respectively.

Fig. 2 shows the locations of the seven candidate sites and 51 relief stations. Table 5 lists the average travel times between the seven candidate sites and 51 relief stations. These travel time data were collected by Sheu [2,3] to evaluate an emergency logistics distribution approach. In the earthquake's aftermath, the maximum travel time for sending relief supplies from the two URDCs (*i.e.*, Nantou Stadium and Jiji Town Hall) to the 51 relief stations was 75 min. Without considering travel time uncertainty, this

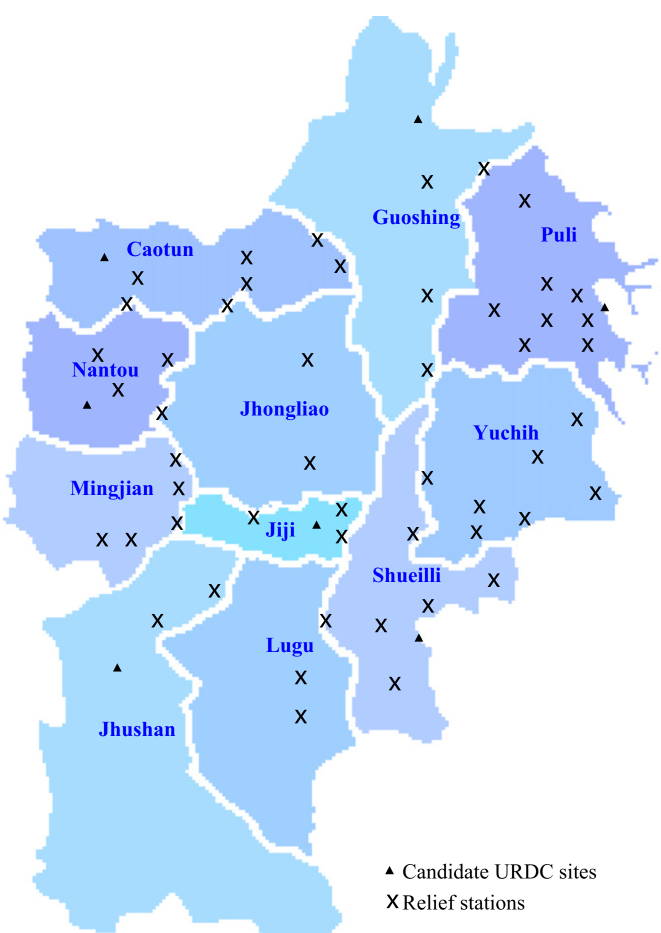

Fig. 2. Candidate URDC sites and relief stations.

work solved a deterministic VPC problem based on the travel times listed in Table 5. In the solution, Puli High School and Jiji Town Hall are selected as the URDCs, with a maximum travel time of 54 min. These two URDCs significantly reduce (75−54=21 min) the maximum travel time, compared to that of the two URDCs set up in the earthquake's aftermath. This indicates a potential improvement in relief distribution system efficiency by locating URDCs at the sites suggested by the solution of the deterministic VPC problem.

Because earthquakes are quick-onset disasters, predictions are generally very limited. Additionally, historical data may be unavailable when designing travel time scenarios. Thus, the proposed RVPC model represented uncertain travel times using intervals. Let t_{ij} be the average travel time (see Table 5) between URDC candidate site j and relief station i, $\forall i, j$. The travel time interval was generated as $[t_{ij}-\alpha t_{ij},\ t_{ij}+\alpha t_{ij}]$, $\forall i, j$. Three distinct values of α (*i.e.*, 0.25, 0.5, and 0.75) were considered to reflect different degrees of travel time uncertainty, such that three instances of the RVPC problem existed. Note that the travel times $t_{ij}, \forall i, j$ collected by Sheu [2] were the estimated mean travel times in the aftermath of the JiJi earthquake on September 21, 1999, rather than the typical travel times before the earthquake. That is, the impact of the earthquake on the road network had been taken into account in these travel times. It is possible that some actual travel times were less than the estimated mean travel times. Thus, we used these travel times as the mean travel times of the travel time intervals.

Table 6 lists the numerical results of solving these three RVPC problem instances. The average CPU time for the implemented SA-based heuristic to solve one instance is 73 s. The URDCs selected in the RVPC model differ from those in the deterministic VPC problem. Particularly, Shueili Middle School was selected as a URDC in all three test instances of the RVPC problem but not in the VPC problem. This difference highlights the importance of considering data uncertainty in choosing URDC sites. All the three performance measures ($rc(\tau_{\text{robust}})$, $\eta(\tau_{\text{robust}})$, and $H(\tau_{\text{robust}})$) increase as the uncertainty level, α, increases. For the different uncertainty levels, $\eta(\tau_{\text{robust}})$ is about 20–25% of the optimal objective value (54 min) in the scenario of average travel times. This finding indicates that seeking solution robustness would trade off a significant amount of solution quality. The reason is likely due to the spatial distribution of candidate sites. In this case study, because candidate sites are scattered throughout Nantou county, the objective value changes dramatically when a different set of candidate sites is selected to locate URDCs.

For comparison purposes, the conventional two-stage SP technique was applied to develop a stochastic VPC (SVPC) model for determining URDC locations in the case study. For the SVPC model, travel time uncertainty was represented using a finite set of discrete scenarios, S, where $t_{ij}(s)$ denotes travel time between station i and URDC j in scenario $s \in S$. Moreover, each scenario s is associated with a probability of occurrence, P_s ($\Sigma_{s\in S} P_s=1$). First-stage decisions involve locations of URDCs (*i.e.*, binary variables x_j, $\forall j \in U$), whereas assignment decisions are made during the second stage, in which binary variables $y_{ij}(s)=1$ when relief station i is assigned to URDC j in scenario s; otherwise, $y_{ij}(s)=0$. The SVPC model, which aims to minimize maximum expected travel time between URDCs and relief stations over all scenarios, can be formulated by adapting stochastic location models from the literature (e.g., [29,30])

$$\text{(SVPC) Minimize } z \tag{16}$$

$$\text{Subject to } z \geq \sum_{s\in S} P_s\Big[\sum_{j\in U} t_{ij}(s)y_{ij}(s)\Big], \quad \forall i \in V \tag{17}$$

Table 5
Travel times (min) between URDC candidate sites and relief stations.

Township	Relief stations	URDC candidate sites						
		Nantou Stadium	Puli High School	Caotun Middle School	Jhushan Elementary School	Jiji Town Hall	Guoshing Town Hall	Shueili Middle School
Nantou	NT-A	4	68	22	33	33	57	43
	NT-B	3	65	19	35	35	56	45
	NT-C	6	67	20	35	37	56	47
	NT-D	2	66	20	32	33	56	43
Puli	PL-A	65	7	54	80	77	38	60
	PL-B	56	4	54	80	78	40	62
	PL-C	69	7	57	84	80	42	64
	PL-D	65	9	53	80	76	38	60
	PL-E	71	20	59	85	91	40	75
	PL-F	61	9	49	76	72	34	56
	PL-G	65	7	53	79	76	38	60
	PL-H	67	6	55	82	78	40	62
Caotun	CT-A	18	59	6	37	43	42	53
	CT-B	32	42	13	51	57	25	67
	CT-C	36	50	16	55	56	33	71
	CT-D	27	53	8	46	52	35	62
	CT-E	39	54	20	58	60	37	75
	CT-F	48	41	28	67	72	24	85
	CT-G	39	41	20	58	64	25	74
Jhushan	JS-A	23	71	33	12	22	61	34
	JS-B	27	74	36	6	27	64	40
Jiji	JJ-A	36	79	50	36	4	78	17
	JJ-B	36	77	50	35	5	75	15
	JJ-C	27	79	41	30	19	67	29
Mingjia	MJ-A	15	67	29	21	19	57	29
	MJ-B	13	64	32	34	32	64	42
	MJ-C	17	73	35	31	28	63	39
	MJ-D	18	75	37	34	32	65	42
	MJ-E	23	75	37	31	29	62	39
Lugu	LG-A	41	88	50	19	37	78	44
	LG-B	50	97	59	28	46	87	54
	LG-C	58	103	73	42	34	101	42
Jhongliao	JL-A	28	65	35	54	38	47	53
	JL-B	21	72	35	39	18	62	33
Yuchih	YC-A	76	29	65	84	57	49	41
	YC-B	61	47	75	62	35	68	19
	YC-C	91	39	79	98	72	64	55
	YC-D	84	27	72	98	73	56	56
	YC-E	68	40	76	69	43	61	26
	YC-F	92	51	86	93	66	71	50
Guoshing	GS-A	57	41	38	71	77	1	72
	GS-B	71	37	52	85	60	26	46
	GS-C	75	43	56	89	95	18	90
	GS-D	62	28	42	76	70	17	56
Shueilli	SL-A	49	61	63	50	24	67	10
	SL-B	71	58	85	72	45	64	31
	SL-C	57	58	71	57	31	64	17
	SL-D	72	86	86	73	46	102	30
	SL-E	65	85	79	66	39	96	24
	SL-F	58	82	72	57	33	91	21
	SL-G	51	65	65	52	25	81	9

$$\sum_{j \in U} y_{ij}(s) = 1, \quad \forall i \in V, \quad s \in S \tag{18}$$

$$y_{ij}(s) - x_j \le 0, \quad \forall i \in V, \quad j \in U, \quad s \in S \tag{19}$$

$$\sum_{j \in U} x_j = p \tag{20}$$

$$x_j \in \{0,1\}, \quad y_{ij}(s) \in \{0,1\}, \quad \forall i \in V, \quad j \in U, \quad s \in S \tag{21}$$

This model can be solved by a commercial solver, such as CPLEX or DECIS in GAMS, or solution techniques in the literature (e.g., [30]).

In this case study, travel times in each scenario were generated based on the average travel times (t_{ij}, $\forall i$, j) given in Table 1. Specifically, in a scenario s, $t_{ij}(s)$ is generated from a uniform distribution, $U[t_{ij}-\alpha \ t_{ij},\ t_{ij}+\alpha t_{ij}]$, $\forall i$, j. Three SVPC problem instances were tested with $\alpha=0.25$, 0.50, and 0.75, and 20 scenarios were generated in each instance.

Table 7 lists the numerical results of solving the three SVPC problem instances. The URDCs selected in the SVPC model differ from those in the VPC model. In comparison with RVPC model results, the same set of URDCs was selected by the two models when $\alpha=0.25$, but different sets of URDCs were selected by the two models when $\alpha=0.50$ and 0.75. This observation may indicate that the difference between the results of the two models increases when the degree of data uncertainty increases. For other performance measures, both the objective value, $z(\boldsymbol{\tau}_{\text{robust}})$, and robustness price, $\eta(\boldsymbol{\tau}_{\text{robust}})$, increase as parameter α increases. The values of $\eta(\boldsymbol{\tau}_{\text{robust}})$ in the SVPC model (*i.e.*, 2, 5, and 11) are markedly smaller than those in the RVPC model (*i.e.*, 15, 20, and 16) for different degrees of travel time uncertainty, indicating that the SVPC model gives up much less solution quality to achieve solution robustness than the RVPC model.

Notably, comparison results for this case study should not be used to claim superiority of the RVPC model over the SVPC model or vice versa, because the two models differ fundamentally. Specifically, the SVPC model typically assumes availability of the probability distribution of uncertain travel times, whereas the RVPC model considers information about the probability distribution is unavailable and, hence, represents uncertain travel times using intervals or ranges. Moreover, the RVPC model minimizes the deviation from the optimal solution in the worst-case scenario, while the SVPC model optimizes the expected system performance. Thus, the RVPC model generally produces more conservative solutions than those acquired by the SVPC model. Accordingly, the SVPC model may be more appropriate for preparation for short-notice disasters, while the RVPC model is well-suited in response to no-notice or quick-onset disasters.

6. Concluding remarks

With particular emphasis on addressing data uncertainty when locating URDCs in response to quick-onset natural disasters, this work developed the RVPC model and its solution algorithm. The model aims to minimize the worst-case deviation in maximum travel time between URDCs and relief stations from the optimal solution. Rather than using probability distributions to describe uncertain travel times, the model represents uncertain travel times using prescribed fixed intervals. The useful property in Theorem 1 was proposed to facilitate the determination of the worst-case scenario among an infinite number of possible scenarios, due to the (continuous) interval representation of uncertain data. Because of the complexity of solving the models, the SA-based heuristic, which incorporates the properties described in Theorem 1 to evaluate robustness costs efficiently, was developed to obtain robust solutions.

A large number of problem instances, with various problem sizes and different degrees of data uncertainty, were generated and solved using the SA-based heuristic. The numerical results show that the proposed heuristic is able to efficiently obtain optimal, or near-optimal, solutions. It was also found that the robust solutions determined by the proposed method do not trade off much quality (or optimality) for robustness.

To demonstrate the applicability of the proposed RVPC model to real-world instances, a case study of the Jiji Earthquake, which hit central Taiwan on September 21, 1999, was conducted. In this case study, the URDCs selected in the RVPC model differ from those in the deterministic VPC problem, highlighting the importance of considering data uncertainty when choosing URDC sites. This work also compared the RVPC model with the two-stage SVPC model. Comparison results reveal that the two models select different sets of URDCs, especially when the degree of data uncertainty is high. Further, the SVPC model sacrifices much less solution quality to achieve solution robustness than the RVPC model. However, comparison results obtained in this case study cannot be used to conclude that the RVPC model is superior to the SVPC model or vice versa, as the two models differ fundamentally and are best suited to distinct disaster types.

This work contributes significantly to the growing body of literature developing robust optimization approaches to emergency logistics. Particularly, this research is the first in the literature to address the RVPC problem in general networks with uncertain link travel times represented by intervals. Theorem 1, which facilitates evaluating robustness costs, could serve as a main building block for developing solution algorithms (e.g., heuristics or approximation algorithms) to the RVPC problem. As described in Section 3, this research developed the SA-based heuristic which utilizes this theorem. While the SA-based heuristic has satisfactory solution quality and computational efficiency, it would be desired to look for other more effective and efficient solution algorithms to the RVPC problem. Our future works will involve the development of other meta-heuristics,

Table 6
Numerical results of the RVPC problem in the case study.

Size	α	$rc(\boldsymbol{\tau}_{\text{robust}})$	$\eta(\boldsymbol{\tau}_{\text{robust}})$	$H(\boldsymbol{\tau}_{\text{robust}})$	Max. avg. travel time	Selected URDCs
(51, 7, 2)	0.25	57	15	27	74	Guoshing Town Hall, Shueili Middle School
	0.50	92	20	38	79	Puli High School, Shueili Middle School
	0.75	133	16	43	75	Caotun Middle School, Shueili Middle School

Table 7
Numerical results of the SVPC problem in the case study.

Size	α	$z(\boldsymbol{\tau}_{\text{robust}})$	$\eta(\boldsymbol{\tau}_{\text{robust}})$	Max. avg. travel time	Selected URDCs
(51, 7, 2)	0.25	71.69	2	56	Guoshing Town Hall, Shueili Middle School
	0.50	90.54	5	59	Caotun Middle School, Shueili Middle School
	0.75	105.95	11	66	Jiji Town Hall, Guoshing Town Hall

such as Genetic Algorithm and Tabu Search, to the RVPC problem, and compare their performance with that of the SA-based heuristic.

Some other interesting topics that may be addressed based on the proposed robust emergency facility location models and algorithms are briefly outlined as follows. For instance, practical constraints (such as facility capacity and budget constraints) can be included when determining emergency facility locations. The model can also be extended by a weighted vertex p-center model that considers not only uncertain travel times but also relief demands represented as nodal weights. We also hope this work can stimulate some other more advanced methods to the RVPC problem and emergency logistics.

Acknowledgments

This paper is partially based on a project (NSC 99-2410-H-027-007-MY3) sponsored by the National Science Council, Taiwan. The authors are grateful to three anonymous reviewers for many helpful and constructive comments and suggestions to improve the quality of this paper. The authors are solely responsible for the content of this paper.

References

[1] Altay N, Green WG. OR/MS research in disaster operations management. European Journal of Operational Research 2006;175:475–93.
[2] Sheu J-B. An emergency logistics distribution approach for quick response to urgent relief demand in disasters. Transportation Research Part E 2007;43:687–709.
[3] Sheu J-B. Dynamic relief-demand management for emergency logistics operations under large-scale disasters. Transportation Research Part E 2010;46:1–17.
[4] IFRC (International Federation of Red Cross and Red Crescent Societies) World disaster report: focus on information in disasters, 〈http://www.ifrc.org/publicat/wdr2005/〉; 2005.
[5] Mirchandani PB, Francis RL. Discrete location theory. New York: Wiley; 1990.
[6] Daskin MS. Network and discrete location: models, algorithms and applications. New York: John Wiley and Sons, Inc; 1995.
[7] Kariv O, Hakimi SL. An algorithmic approach to network location problems I: the p-centers. SIAM Journal on Applied Mathematics 1979;37:513–38.
[8] Jia H, Ordonez F, Dessouky MM. A modeling framework for facility location of medical services for large-scale emergencies. IIE Transactions 2007;39:41–55.
[9] Jia H, Ordonez F, Dessouky MM. Solution approaches for facility location of medical supplies for large-scale emergencies. Computers & Industrial Engineering 2007;52:257–76.
[10] Huang R, Kim S, Menezes MBC. Facility location for large-scale emergencies. Annals of Operations Research 2010;181:271–86.
[11] Current J, Daskin MS, Schilling D. Discrete network location models. In: Drezner Z, Hamacher HW, editors. Facility Location: Applications and Theory. New York: Springer-Verlag; 2002. p. 81–118.
[12] Snyder LV. Facility location under uncertainty: a review. IIE Transactions 2006;38:547–64.
[13] Mete HO, Zabinsky ZB. Stochastic optimization of medical supply location and distribution in disaster management. International Journal of Production Economics 2010;126:76–84.
[14] Chang M-S, Tseng Y-L, Chen J-W. A scenario planning approach for the flood emergency logistics preparation problem under uncertainty. Transportation Research Part E 2007;43:737–54.
[15] Balcik B, Beamon BM. Facility location in humanitarian relief. International Journal of Logistics: Research and Applications 2008;11:101–21.
[16] Noyan N. Risk-averse two-stage stochastic programming with an application to disaster management. Computers & Operations Research 2012;39:541–59.
[17] Kouvelis P, Yu G. Robust discrete optimization and its applications. Boston: Kluwer Academic Publishers; 1997.
[18] Montemanni R, Gambardella LM. An exact algorithm for the robust shortest path problem with interval data. Computers & Operations Research 2004;31:1667–80.
[19] Bertsimas D, Pachamanova D. Robust multiperiod portfolio management in the presence of transaction costs. Computers & Operations Research 2008;35:3–17.
[20] Alem DJ, Morabito R. Production planning in furniture settings via robust optimization. Computers & Operations Research 2012;39:139–50.
[21] Lu C-C, Lin S-W, Ying K-C. Robust scheduling on a single machine to minimize total flow time. Computers & Operations Research 2012;39:1682–91.
[22] Averbakh I, Berman O. Algorithms for the robust 1-center problem on a tree. European Journal of Operational Research 2000;123:292–302.
[23] Burkard RE, Dollani H. A note on the robust 1-center problem on trees. Annals of Operations Research 2002;110:69–82.
[24] Averbakh I, Berman O. Minimax regret p-center location on a network with demand uncertainty. Location Science 1997;5:247–54.
[25] Averbakh I. Complexity of robust single facility location problems on networks with uncertain edge lengths. Discrete Applied Mathematics 2003;127:505–22.
[26] Suman B, Kumar P. A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of the Operational Research Society 2006;57:1143–60.
[27] Chen D, Chen R. New relaxation-based algorithms for the optimal solution of the continuous and discrete p-center problems. Computers & Operations Research 2009;36:1646–55.
[28] Ministry of the Interior, Taiwan. National earthquake preparedness report; 2000.
[29] Daskin MS. Location, dispatching and routing models for emergency services with stochastic travel times. In: Ghosh A, Rushton G, editors. Spatial Analysis and Location–Allocation Models. New York: Van Nostrand Reinhold; 1987. p. 224–65.
[30] Hochbaum DS, Pathria A. Locating centers in a dynamically changing network and related problems. Location Science 1998;6:243–56.