AN INTEGER L-SHAPED ALGORITHM FOR THE INTEGRATED LOCATION AND NETWORK RESTORATION PROBLEM IN DISASTER RELIEF

Ece Sanci[a,b,*] (*Corresponding author)

E-mail address: es2138@bath.ac.uk

Mark S. Daskin[a]

E-mail address: msdaskin@umich.edu

[a] University of Michigan, Ann Arbor, MI 48109, USA

[b] University of Bath, Bath BA2 7AY, UK

Being prepared for potential disaster scenarios enables government agencies and humanitarian organizations to respond effectively once the disaster hits. In the literature, the two-stage stochastic programming models are commonly employed to develop preparedness plans before anticipated disasters. These models can be very difficult to solve as the complexity increases by several sources of uncertainty and interdependent decisions. In this study, we propose an integer L-shaped algorithm to solve the integrated location and network restoration model, which is a two-stage stochastic programming model determining the number and locations of the emergency response facilities and restoration resources under uncertainty. Our algorithm accommodates the second-stage binary decision variables which are required to indicate undamaged and restored roads of the network that can be used for relief distribution. Our computational results show that our algorithm outperforms CPLEX for the larger number of disaster scenarios as the solution time of our algorithm increases only linearly as the number of scenarios increases.

Keywords: Disaster preparedness, prepositioning, relief distribution, network restoration, integer L-shaped

1. INTRODUCTION

Every year, disasters threaten hundreds of thousands of lives and cause massive amounts of damage. The Indian Ocean Tsunami in 2004, Hurricane Katrina in 2005, the Haiti earthquake in 2010, the Japan earthquake and tsunami in 2011, and the Nepal earthquake in 2015 are only a few examples of the deadliest disasters in the last two decades. The high impact of these disasters has attracted the attention of many scholars in operations research. There has been a significant interest in using operations research in humanitarian applications to reduce loss of life and alleviate human suffering caused by disasters. Kara and Savaşer (2017) state that location problems constitute the majority of problems studied in the relief logistics literature published between 2007 and 2017. In particular, the warehouse location and inventory prepositioning problem has been extensively studied in recent years.

Prepositioning emergency relief items prior to an anticipated disaster is a preparedness strategy to effectively support the relief distribution capacity in the aftermath of the disaster. Studies on prepositioning in the literature mainly develop optimization models to determine the location and capacity of the emergency response facilities, which are warehouses to store the relief items before the disaster. After the disaster strikes, relief items kept in these facilities are distributed to the affected communities. It is usually very difficult to predict the location, time, and scale of disasters. Therefore, most of these studies consider potential disaster scenarios and the corresponding relief operations when locating the emergency response facilities (Grass and Fischer, 2016). The two-stage stochastic programming framework is the most common modeling approach since it allows the modeler to incorporate uncertainties following the disaster and time-dependent decisions.

Maintaining a holistic view is critical in disaster management (Çelik et al., 2014). However, the problem of locating emergency response facilities is approached by mainly focusing on the relief distribution in the post-disaster stage and ignoring network restoration efforts performed concurrently to reestablish connectivity with inaccessible demand regions. Kovacs and Moshtari (2019) draw attention to the need for integrated models considering the potential effect of network restoration decisions on the prepositioning plans and relief distribution decisions. There are several studies (Liberatore, Ortuno, Tirado, Vitoriano, and Scaparra, 2014; Çelik, Ergun, and Keskinocak, 2015; Ransikarbum and Mason, 2016) which consider the relief item distribution and network restoration problems jointly; however, these studies limit the problem environment to the post-disaster stage. A recent study by Aslan and Çelik (2019) proposes a prepositioning model that determines warehouse locations and inventory levels prior to a disaster in the presence of vulnerable roads. The model allows for the network restoration after the disaster to ensure the timely delivery of relief items. Nevertheless, this model fails to incorporate that restoration resources may be scarce, and they may not have access to some parts of the network due to limited connectivity.

Sanci and Daskin (2019) present a two-stage stochastic programming model which determines where to locate emergency response facilities and restoration equipment before an anticipated disaster in the face of uncertainty in demand, supply availability, and network availability. Decisions related to distributing relief items from the emergency response facilities to the disaster victims and decisions related to restoring damaged arcs of the underlying relief network are made jointly after the disaster as the uncertainties are resolved. To solve this integrated location and network restoration problem, Sanci and Daskin (2019) develop a sample average approximation (SAA) method with concentration sets motivated by Rosing and ReVelle (1997)'s Heuristic Concentration. In this approach, they construct concentration sets consisting of locations with high potential to be in the optimal solution, and they limit the first-stage solution space to concentration sets within the SAA scheme.

The computational results from Sanci and Daskin (2019) show that limiting the solution space in the first stage to concentration sets reduces the problem size without sacrificing the solution quality significantly; however, the quality of solutions cannot be verified unless the optimal solution is known. Moreover, this approach does not exploit the structure of the two-stage stochastic programming model which allows us to decompose the second-stage problem into subproblems for each disaster scenario once the first-stage decisions are fixed. A solution method

utilizing this property can effectively tackle the long run time and out-of-memory risks posed by the increasing number of scenarios required for the convergence of the SAA algorithm.

In this study, we develop an exact solution approach based on exploring the first-stage solution space on a branch-and-bound tree and decomposing the second-stage problem into subproblems for fixed first-stage solutions. Optimality cuts derived from these subproblems remove non-optimal solutions from further consideration. We use the integer optimality cuts, which are employed in the integer L-shaped method by Laporte and Louveaux (1993), since our problem includes binary decision variables in the second stage. Our approach mainly follows an improved version of the integer L-shaped method by Angulo, Ahmed, and Dey (2016) with two modifications to better exploit the structure of our integrated model. Furthermore, we start our algorithm by finding a feasible solution by locating emergency response facilities and restoration equipment in two steps. This feasible solution typically provides a tight upper bound for our branch-and-cut procedure. Our computational results show that the solution time of our algorithm increases only linearly as the number of scenarios increases.

The organization of the rest of the paper is as follows: in Section 2, we review the literature on prepositioning models using the stochastic programming framework and the literature on exact solution algorithms for two-stage stochastic programs with integer recourse. We present the two-step sequential approach and the integer L-shaped method with our modifications in Section 3 and Section 4, respectively. We discuss our computational results in Section 5. In Section 6, we conclude the paper along with our future research directions.

2. LITERATURE REVIEW

We review studies from two streams of literature related to our study. First, we review studies employing the stochastic programming framework for the prepositioning problem. Second, we review studies developing exact solution methods to solve two-stage stochastic programming models with integer decision variables in the second-stage (recourse) problem.

2.1. Stochastic Programming Models for the Prepositioning Problem

The seminal papers by Balcik and Beamon (2008), Rawls and Turnquist (2010), and Mete and Zabinsky (2010) propose stochastic programming models to determine the location of the emergency response facilities and their prepositioning levels in the pre-disaster stage while focusing on the relief distribution problem in the post-disaster stage. Balcik and Beamon (2008) use an objective function maximizing the expected covered demand, whereas Rawls and Turnquist (2010) and Mete and Zabinsky (2010) use objective functions minimizing the expected total cost including the expected unmet demand cost. These studies consider uncertainty in demand and uncertainty associated with the network condition through increased transportation time/cost and reduced transportation capacity. There are various prepositioning studies in the literature extending this line of work by incorporating other decisions in the pre- or post-disaster stages, objective functions, or sources of uncertainty (Salmeron and Apte, 2010; Duran, Gutierrez, and Keskinocak, 2011; Döyen, Aras, and Barbarosoğlu, 2012; Salman and Yücel, 2015; Ahmadi, Seifi, Tootooni, 2015). We refer the interested reader to Kara and Savaşer (2017) for a more detailed literature review on the prepositioning studies. In this section, we review several recent studies in this body of literature.

The damaged transportation network in the aftermath of a disaster can be a major obstacle to the effectiveness of relief distribution operations. Hong, Lejeune, and Noyan (2015) propose a two-stage chance-constrained stochastic programming model to ensure that a feasible flow of relief supplies exists on the post-disaster transportation network with high probability. Moreover, to improve the response time, they define regions and use chance constraints to guarantee that these regions can satisfy their own demand with high probability. Elci and Noyan (2018) also propose a chance-constrained model for the prepositioning problem to ensure responsive relief distribution. This model features a mean-risk objective that incorporates the conditional value-at-risk to reflect the decision-makers' risk preferences with respect to the total cost.

Several prepositioning studies address the relationship between relief distribution operations and the capacity of available vehicle fleets. Alem, Clark, and Moreno (2016) propose a two-stage stochastic programming model to determine the fleet sizing of multiple types of vehicles as well as the inventory levels of the prepositioned relief supplies on the relief network before the disaster. Paul and Zhang (2019) present a two-stage stochastic programming model for the hurricane preparedness problem to determine the transportation capacity in addition to the location of emergency response facilities and relief supply levels prior to the landfall. Hu, Han, Dong, and Meng (2019) propose a multi-stage stochastic programming model that decides on the facility locations at the first stage but updates the number and type of vehicles needed for the relief distribution at every stage of the disaster response. The authors take into consideration dynamic road capacity due to unpredictable secondary disasters and the fluctuation of vehicle rental costs. In a similar problem setting, Moreno, Alem, Ferreira, and Clark (2018) propose a two-stage stochastic programming model to determine the number of vehicles procured before and after the disaster occurs. The authors incorporate human suffering due to delayed access to relief supplies in the objective function via minimizing the deprivation cost (Holguín-Veras, Pérez, Jaller, Van Wassenhove, and Aros-Vera, 2013) in addition to minimizing the logistics cost.

The severity levels of casualties may deteriorate in the absence of relief supplies stemming from insufficient capacity or long transportation times, eventually leading to fatality. Paul and MacDonald (2016) incorporate uncertainty in the number of casualties of different severity levels and propose a model that minimizes the sum of facility and relief supply acquisition costs and the associated fatality cost given the expected fatalities. Paul and Wang (2019) propose two robust models for a similar problem using distribution-free uncertainty sets for stochastic parameters and using a relative regret limit.

Although the prepositioning literature recognizes the adverse effect of the damaged transportation network on the promptness of deliveries to the disaster victims, it neglects the network restoration efforts supporting relief distribution activities. Aslan and Çelik (2019) made the first attempt to incorporate network restoration decisions in the prepositioning problem. They propose a two-stage stochastic programming model to design a three-echelon network for the prepositioning problem by considering the damage in the road network and the interdependence of road repair and relief distribution activities in the aftermath of the disaster. The authors show that ignoring network restoration efforts would affect the timeliness of relief distribution dramatically. Although this model is a closer representation of the post-disaster problem where road repair and relief distribution take place concurrently, it has

two strong assumptions in terms of the restoration requirements. It assumes that there is a sufficient number of restoration resources, and these resources have no limitation in accessing any part of the network. In addition to these restrictive assumptions, Aslan and Çelik (2019) also point out the need to improve the solution time of the resulting mixed-integer programs for larger scenario sets within the SAA scheme.

Sanci and Daskin (2019) avoid the assumptions made in Aslan and Çelik (2019) by a two-stage stochastic programming model which determines the location and number of pieces of restoration equipment along with the location and size of emergency response facilities in the first stage. The second-stage problem allows for the flow of restoration equipment and the flow of relief items only on the operational (either undamaged or restored) roads of the underlying network. Therefore, this integrated model helps decision-makers to plan for the restoration requirements before the disaster while ensuring that the restoration resources repair damaged roads after the disaster only if they can reach these roads from their initial position. Sanci and Daskin (2019) propose a heuristic solution approach coupled with the SAA algorithm to solve this problem. In this paper, we develop an exact solution approach based on the integer L-shaped algorithm to solve the same problem. Note that our algorithm can be used to solve the resulting problems generated within the SAA scheme, or the deterministic equivalent MIPs if there is a limited set of discrete disaster scenarios.

2.2. Exact Solution Methods for the Two-Stage Stochastic Programs with Integer Recourse

Grass and Fischer (2016) claim that a high-quality solution should be obtained from the stochastic programming models within a reasonable time to support the decision-making process in disaster management. They state that even the decisions made in the pre-disaster stage have become time-sensitive with the need for agile humanitarian supply chains which can quickly respond to short-term forecasts. Their literature survey shows that most of the studies in this field use a general-purpose solver like CPLEX which cannot take advantage of the special structure of the stochastic programs and run into long computational times and memory issues. They also observe that for large-scale instances, these studies often propose a heuristic algorithm that does not provide any guarantee on the solution quality. The high level of uncertainty in data with the questionable solution quality can lead to fatally wrong decisions. In this paper, we develop an exact algorithm to solve the integrated location and network restoration model introduced by Sanci and Daskin (2019), which has binary decision variables in the second-stage (recourse) problem. Therefore, we review the exact algorithms for solving two-stage stochastic programs with an integer recourse in this section.

First, we briefly discuss the two-stage stochastic programming framework to lay the foundation. A standard formulation of the two-stage stochastic program can be presented as follows:

$$(1) \qquad \min f^T z + \mathbb{E}[Q(z, \boldsymbol{\xi})]$$
$$\text{s.t.} \qquad z \in \mathcal{Z}$$

where $\boldsymbol{\xi}$ denotes the uncertain data with a known probability distribution P and $\mathbb{E}[.]$ is the expectation operator taken with respect to $\boldsymbol{\xi}$. Then, for a particular realization ξ of $\boldsymbol{\xi}$, $Q(z, \xi)$ is defined as

(2) $$Q(z,\xi) = \min q(\xi)^T y$$
$$\text{s.t.} \quad W(\xi)y + T(\xi)z \geq h(\xi)$$
$$y \in \mathcal{Y}$$

Problem (1) is the first-stage problem where the first-stage decision vector z is determined before the realization of the uncertain data so that the sum of the first-stage cost and expected second-stage cost is minimized. Problem (2) constitutes the second-stage problem with the second-stage decision vector y. $\mathcal{Z}$ and $\mathcal{Y}$ are generally described by linear constraints. In some cases, integrality restrictions are also included in these sets. The second-stage matrix $W(\xi)$ is called the recourse matrix. The impact of the first-stage decisions on the second-stage problem is reflected through the technology matrix $T(\xi)$.

When $\boldsymbol{\xi}$ follows a discrete distribution with a finite support, it is possible to calculate the expected second-stage cost as a simple weighted sum. Moreover, Schultz (1995) shows that the continuous distribution of $\boldsymbol{\xi}$ can be approximated by a discrete distribution under mild conditions even if the stochastic program has an integer recourse. Therefore, the uncertain data can be represented by a finite number of scenarios each corresponding to a realization ξ. Off-the-shelf solvers typically run into memory problems as the set of scenarios gets large. The L-shaped method (Van Slyke and Wets, 1969) is a common solution approach to overcome this difficulty.

For given first-stage decisions and the realization of the uncertain parameters, the second-stage problem decomposes into independent subproblems corresponding to each scenario. The L-shaped method exploits this structure by approximating the convex function of the expected second-stage cost using the duals obtained from the subproblems when the second-stage problem has only continuous decisions.

Let us reformulate Problem (1) by introducing a new decision variable θ, which underestimates $\mathbb{E}[Q(z,\boldsymbol{\xi})]$:

(3) $$\min f^T z + \theta \qquad (3.1)$$
$$\text{s.t.} \quad z \in \mathcal{Z} \qquad (3.2)$$
$$\theta \geq o_j - O_j z \quad j = 1,2,\dots,J \qquad (3.3)$$

Problem (3) is the so-called master problem and the constraint set (3.3) is the set of optimality cuts. The L-shaped method starts with an empty set of optimality cuts. At every iteration, a new z is obtained by solving the master problem and one optimality cut is generated using the dual variables obtained by solving the subproblems for given z. The L-shaped method terminates when it finds an optimal first-stage decision vector z^*; i.e. θ is equal to $\mathbb{E}[Q(z^*,\boldsymbol{\xi})]$. Note that if every feasible z results in a feasible second-stage problem, the problem has a (relatively) complete recourse. For problems which do not have this property, feasibility cuts should be also added to the master problem in every iteration the first-stage decision vector creates an infeasible second-stage problem. Note that we do not consider the feasibility cuts in our discussion since our problem has this property.

Now, let us assume that the second-stage decision variables are continuous and let us rewrite Problem (2) as follows:

(4) $$Q(z,\xi) = \min q(\xi)^T y \qquad (4.1)$$

$$\text{s.t.} \quad W(\xi)y \geq h(\xi) - T(\xi)z \qquad (4.2)$$

$$y \geq 0 \qquad (4.3)$$

Let $\pi(\xi)$ be the optimal dual variables associated with the constraint set (4.2). Then, the optimal objective function value for the subproblem corresponding to realization ξ is $[h(\xi) - T(\xi)z]'\pi(\xi)$. Assuming that the set of scenarios is denoted by Ω and the probability of scenario ξ_ω is pr^ω, the expected second-stage cost for z is $\sum_{\omega\in\Omega} pr^\omega [h(\xi_\omega) - T(\xi_\omega)z]'\pi(\xi_\omega)$. Then, the matrix O_j and the vector o_j in j^{th} optimality cut are $\sum_{\omega\in\Omega} pr^\omega \pi(\xi_\omega)'T(\xi_\omega)$ and $\sum_{\omega\in\Omega} pr^\omega \pi(\xi_\omega)'h(\xi_\omega)$, respectively.

Note that the L-shaped algorithm is also applicable for the two-stage stochastic programming models with integer first-stage variables and continuous second stage variables. The expected second stage value function is still convex for these models. On the other hand, the existence of integer decisions in the second stage problem leads to an objective function which is non-convex and discontinuous in general (Ahmed, 2010).

Caroe and Tind (1998) generalize the L-shaped method to accommodate integer variables in the second stage. Drawing upon IP duality, their method generates nonlinear feasibility and optimality cuts resulting in a non-convex master problem. Caroe and Tind (1998) show that it is conceptually possible to use the L-shaped method framework for stochastic integer programs; however, their approach does not address the complexity arising from non-convexity.

Caroe and Schultz (1999) propose a Lagrangian relaxation based approach which deploys a scenario-wise decomposition instead of a stage-wise decomposition as in the L-shaped method. This is accomplished by introducing copies of first-stage variables for each scenario and relaxing the non-anticipativity constraint which restricts them to be identical across all scenarios. Then, the Lagrangian dual problem becomes separable by scenarios for a given set of Lagrange multipliers. Note that the subproblems of the Lagrangian dual include decision variables and constraints from both stages. Therefore, it is harder to solve a subproblem in this approach compared to a subproblem in the L-shaped method which only corresponds to the second-stage problem for a given first-stage decision and a particular scenario. Another limitation is that a non-smooth optimization technique such as the subgradient method is required to solve the Lagrangian dual problem.

It is worth mentioning that Caroe and Tind (1998) and Caroe and Schultz (1999) do not impose any restriction on whether the integer variables are in the first stage and/or second stage. They also assume a general structure for the recourse and technology matrices. More efficient solution approaches can be developed for stochastic integer programs with certain specific structures. Sen (2005) and Kucukyavuz and Sen (2017) review a collection of algorithms developed to solve stochastic integer programming problems, which mainly differ based on (i) the integrality restrictions in the first stage variables and/or recourse variables, and (ii) the structure of the recourse and technology matrices.

Laporte and Louveaux (1993) develop the integer L-shaped algorithm, which is a general branch-and-cut procedure, for two-stage stochastic programs with binary first-stage variables and mixed-integer second stage variables. The integer L-shaped method uses a new optimality cut which removes a non-optimal solution from further

consideration. In this study, the classical L-shaped method is modified in a way that the 'continuous L-shaped optimality cuts' generated from the linear relaxation of the second stage problems are still valid. These valid cuts are used to improve lower bounds in the branch-and-cut procedure.

The algorithm proposed by Laporte and Louveaux (1993) requires the exact computation of the expected second stage value function every time a new first-stage solution is obtained in the branch-and-bound tree. Even though the second-stage problem decomposes to smaller subproblems for each scenario for fixed first-stage decisions, this still means that a MIP should be solved for each scenario in the presence of mixed-integer second-stage variables. Angulo et al. (2016) improve the integer L-shaped algorithm by using the LP relaxation of subproblems whenever the continuous L-shaped optimality cut is enough to cut off a solution which is not optimal.

The main idea behind the integer L-shaped method is to approximate the expected second-stage value function sequentially by generating linear cuts. In the same spirit, the methods proposed by Sherali and Fraticelli (2002) and Sen and Higle (2005) sequentially construct the partial representation of the convex hull, which heavily relies on the theory of disjunctive programming. Sherali and Fraticelli (2002) develop a solution method for two-stage stochastic programs with binary first-stage decisions and 0/1 mixed-integer subproblems. The authors modify the classical L-shaped method by using a cutting plane scheme to describe the convex hull of the subproblems before generating the continuous L-shaped optimality cut. Note that Sherali and Fraticelli (2002) do not assume fixed recourse or technology matrices, as is usually assumed in the literature. Sen and Higle (2005) also use a cutting plane method for the convexification of the second-stage problem which takes advantage of the fixed recourse property to eliminate the need to store cuts separately for each scenario. Sen and Sherali (2006) extend this line of research by coupling the use of disjunctive cuts with a branch-and-cut procedure. Gade, Kucukyavuz, and Sen (2014) and Zhang and Kucukyavuz (2014) use Gomory cuts instead of disjunctive cuts within a decomposition algorithm to solve two-stage stochastic programs with pure integer recourse.

The studies presented so far assume that the first-stage decision variables are integer variables. This assumption is mainly important to guarantee that the proposed algorithms converge to the optimal solution in a finite number of steps. Ahmed, Tawarmalani, and Sahinidis (2004) propose a branch-and-bound algorithm for stochastic integer programs with mixed-integer first-stage variables and pure integer second-stage variables. When there exist continuous variables in the first stage, infinitely many branching operations may be needed before the lower and upper bounds converge; therefore, finiteness of the algorithm becomes a major concern. The algorithm proposed by Ahmed et al. (2004) overcomes this difficulty by using a variable transformation which is applicable when the technology matrix is deterministic. Sherali and Zhu (2006) also propose a branch-and-bound strategy to solve stochastic programs with integer recourse when continuous first-stage variables are present. In the proposed algorithm, the lower bounds are computed by deploying a similar decomposition approach as in Sherali and Fraticelli (2002).

Although the interest in developing algorithms to solve the two-stage stochastic programs with integer recourse is increasing, the number of application papers is still very limited. The integer L-shaped method by Laporte and Louveaux (1993) is applied to solve a stochastic location problem (Laporte, Louveaux, and van Hamme, 1994),

a stochastic traveling salesperson problem (Laporte, Louveaux, and Mercure, 1994), and stochastic vehicle routing problems (Gendreau, Laporte, and Seguin, 1995; Laporte, Louveaux, and van Hamme, 2002). More recent studies Miller-Hooks, Zhang, and Faturechi (2012), Noyan, Balcik, and Atakan (2016), and Noyan and Kahvecioglu (2018) employ the integer L-shaped method to solve problems from disaster management. The two-stage stochastic programming model proposed by Miller-Hooks, Zhang, and Faturechi (2012) determines the optimal set of pre-disaster preparedness activities on the transportation network to maximize the expected post-disaster flow between origin and destination pairs. Miller-Hooks et al. (2012) solve this model using a dynamic list of optimality cuts within the integer L-shaped method. Noyan, Balcik, and Atakan (2016) consider the problem of determining the locations/capacities of the distribution points in the relief networks at the post-disaster stage to achieve high levels of accessibility and equity in the aftermath of a disaster. They employ the lazy constraints when implementing the multi-cut version of the integer L-shaped algorithm. Noyan and Kahvecioglu (2018) extend this work by considering a three-echelon relief network design problem. They follow the solution procedure in Noyan et al. (2016) with the improvements proposed by Angulo et al. (2016).

3. TWO-STEP SEQUENTIAL APPROACH

Sanci and Daskin (2019) present a two-stage stochastic programming model for the integrated location and network restoration problem to locate emergency response facilities and restoration equipment prior to a disaster. The model determines the number and locations of the facilities and restoration resources in the first stage in the face of demand uncertainty, damage uncertainty, and repair time uncertainty. The decisions related to relief item distribution and network restoration are made jointly in the second stage for the corresponding first-stage decisions and realizations of uncertain parameters. The model assumes that the relief items prepositioned in the emergency response facilities before the disaster are transported to disaster victims at demand nodes using the operational arcs of the network after the disaster. Meanwhile, restoration resources repair damaged arcs so that demand nodes may be accessed subsequently if there does not exist any operational path from an open facility right after the disaster.

To solve this integrated model, Sanci and Daskin (2019) propose a heuristic procedure to obtain near-optimal solutions in a reasonable time by reducing the first-stage solution space using concentration sets comprised of promising nodes. This idea is inspired by Heuristic Concentration (Rosing and ReVelle, 1997) where concentration sets are formed utilizing information from multiple runs of an exchange heuristic. Sanci and Daskin (2019) construct the concentration sets by solving the integrated model for small subsets of scenarios multiple times. In this study, we use another approach to reduce the first-stage solution space to handle the problem in two steps where decisions for emergency response facilities and restoration equipment are made in the first step and second step, respectively. In the first step, a two-stage stochastic programming model is solved to determine the location and size of the emergency response facilities. The network restoration problem is disregarded completely in this step, so a demand node which cannot be reached by any of the open facilities immediately after the disaster will remain inaccessible. Therefore, there is no need to consider multiple time periods in the second stage unlike the original integrated model where people in need may be reached in the future time periods following the disaster as the damaged roads are restored. Once the location/size of the facilities are determined in the first step, another two-stage stochastic programming

model is solved to locate restoration equipment in the second step. This second model is almost identical to the original model; however, the decisions regarding the location and size of the emergency response facilities are taken as an input this time.

Let us introduce the two-stage stochastic programming model solved in the first step of this sequential approach. Before we present the formulation, we go over the notation introduced in Sanci and Daskin (2019). Note that we provide the complete list of notations and the model formulation in Appendix A. We consider a network (N, A) where N is the set of nodes and A is the set of arcs. $N(i)$ denotes the set of neighbor nodes of node i, which includes node $j \in N$ if $a = (i, j)$ or $\bar{a} = (j, i)$ is in set A. Moreover, N^F and L^F are the set of candidate nodes and set of levels for emergency response facilities, respectively. The operating cost of an emergency response facility of level $l \in L^F$ during the preparedness phase is f_l which enables the facility to have q_l units of capacity during this phase. Moreover, c_a is the travel cost per unit emergency relief item flowing on arc $a \in A$, and b is the penalty cost per unit unmet demand. Note that the parameters presented so far are deterministic. As before, $\boldsymbol{\xi}$ denotes the uncertain data and ξ denotes a particular realization of the uncertain data. The uncertain parameters are demand d_i^{ξ} at node $i \in N$, damage ratio ρ_i^{ξ} at node $i \in N$, and repair time p_a^{ξ} of arc $a \in A$ in realization ξ. Note that the damage ratio refers to the ratio of the lost capacity after the disaster to the initial capacity of an open facility. Also, note that although it is necessary to incorporate the repair time of arcs to model the uncertainty in the network condition for the original model, we use repair time information here just to check whether an arc is operational or not in the aftermath of the disaster. That is, if $p_a^{\xi} > 0$, arc a is not operational; if $p_a^{\xi} = 0$, arc a is operational in realization ξ.

Let us also go over the decision variables in this model. Z_{il} is the binary decision variable which is 1 if an emergency response facility of level l is open at node i, and 0 otherwise. Z_{il} is the only first-stage decision variable in this model, and z stands for the decision vector for facility locations. Y_{ak}^{ξ} and $Y_{\bar{a}k}^{\xi}$ denote the fraction of demand at node k flowing on arc a in realization ξ. Note that $\bar{a}$ denotes the reverse direction of arc $a \in A$. W_{ik}^{ξ} is the fraction of demand at node k served by the facility at node i, and U_k^{ξ} is the unmet demand at node k in realization ξ. Finally, X_a^{ξ} is the binary decision variable denoting whether arc a is operational or not in realization ξ.

With this notation, the two-stage stochastic programming model in the first step is as follows:

(5)
$$min \quad \sum_{i \in N^F} \sum_{l \in L^F} f_l Z_{il} + \mathbb{E}[Q_1(z, \boldsymbol{\xi})] \tag{5.1}$$

s.t.

$$\sum_{l \in L^F} Z_{il} \leq 1 \qquad \forall i \in N^F \tag{5.2}$$

$$Z_{il} \in \{0,1\} \qquad \forall i \in N^F, l \in L^F \tag{5.3}$$

where

(6)
$$Q_1(z, \xi) = min \sum_{a \in A} \sum_{k \in N} c_a d_k^{\xi} (Y_{ak}^{\xi} + Y_{\bar{a}k}^{\xi}) + b \sum_{k \in N} d_k^{\xi} U_k^{\xi} \tag{6.1}$$

s.t.
$$\sum_{j\in N(i)} Y^{\xi}_{\bar{a}i} + W^{\xi}_{ii} + U^{\xi}_{i} = \sum_{j\in N(i)} Y^{\xi}_{ai} + 1, \quad \forall i \in N \tag{6.2}$$
$$\sum_{j\in N(i)} Y^{\xi}_{\bar{a}k} + W^{\xi}_{ik} = \sum_{j\in N(i)} Y^{\xi}_{ak}, \quad \forall i \in N, \forall k \in N\backslash\{i\} \tag{6.3}$$
$$\sum_{k\in N} d^{\xi}_{k} W^{\xi}_{ik} \leq (1-\rho^{\xi}_{i}) \sum_{l\in L^F} q_l Z_{il}, \quad \forall i \in N^F \tag{6.4}$$
$$W^{\xi}_{ik} \leq \sum_{l\in L^F} Z_{il}, \quad \forall i \in N^F, \forall k \in N \tag{6.5}$$
$$W^{\xi}_{ik} = 0, \quad \forall i \in N\backslash N^F, \forall k \in N \tag{6.6}$$
$$Y^{\xi}_{ak} \leq X^{\xi}_{a}, \quad \forall a \in A, \forall k \in N \tag{6.7}$$
$$Y^{\xi}_{\bar{a}k} \leq X^{\xi}_{a}, \quad \forall a \in A, \forall k \in N \tag{6.8}$$
$$p^{\xi}_{a} X^{\xi}_{a} = 0, \quad \forall a \in A \tag{6.9}$$
$$0 \leq Y^{\xi}_{ak}, Y^{\xi}_{\bar{a}k} \leq 1, \quad \forall a \in A, \forall k \in N \tag{6.10}$$
$$0 \leq W^{\xi}_{ik} \leq 1, \quad \forall i,k \in N \tag{6.11}$$
$$0 \leq U^{\xi}_{k} \leq 1, \quad \forall k \in N \tag{6.12}$$
$$X^{\xi}_{a} \in \{0,1\}, \quad \forall a \in A \tag{6.13}$$

If the two-stage stochastic programming model solved in this first step is compared with the original model, it is easy to recognize that the model presented above is a reduced version in which the decision variables and constraints related to network restoration are removed. In this step, emergency response facilities are located based on the network availability right after the disaster; therefore, the multi-period second stage is also reduced to a single period.

The model decides on the location and size of emergency response facilities to minimize the objective function (5.1) which is comprised of the facility costs and expected second-stage cost. Constraints (5.2) ensure that at most one facility of any available size can be established at candidate nodes. Constraints (5.3) defines the first-stage decision variables as binary decisions.

The second-stage cost function (6.1) consists of the relief distribution cost and unmet demand cost. Constraints (6.2) and (6.3) are the relief item flow balance constraints when $i = k$ and $i \neq k$, respectively. Two sets of flow balance constraints are needed as constraints (6.2) control the flow balance of relief items at node $i \in N$ when the flow is designated for this node. On the other hand, constraints (6.3) ensure that the flow balance is preserved at node $i \in N$ for the relief item flow designated for another node $k \in N\backslash\{i\}$ which is transshipped through node i. Constraints (6.4) guarantee that the capacity limitation is not violated. Constraints (6.5) and (6.6) together make sure that only open facilities are used to serve the demand for relief items. Constraints (6.7) and (6.8) forbid using the damaged arcs for relief transportation. Constraints (6.9) define the operational arcs and damaged arcs. Finally, constraints (6.10) - (6.13) define the domain of the second-stage decision variables.

Let us now present the additional notation used in the two-stage stochastic programming model solved in the second step of the sequential approach. N^E and L^E are the set of candidate nodes and set of levels for restoration equipment, respectively. T denotes the set of time periods representing multiple time periods in the second stage problem. In addition to the deterministic parameters presented above, e_l is the cost of acquiring restoration equipment

at level $l \in L^E$ with w_l pieces of restoration equipment at this level. c'_a is the travel cost per piece of restoration equipment flowing on arc $a \in A$.

In this step, information on the location/size of facilities becomes a parameter of the model. Let us define $\bar{z}_{il}$ to represent whether there is an open facility at node $i \in N^F$ of level $l \in L^F$. This time V_{il} is the only first-stage decision variable, which is 1 if node i is chosen to locate level l of restoration resources, and 0 otherwise. Additionally, v denotes the decision vector for the restoration equipment. In this model, we keep the second-stage decision variables defined above with a slight modification to reflect the time perspective in the response phase. Therefore, we add time index t to Y^{ξ}_{akt}, $Y^{\xi}_{\bar{a}kt}$, W^{ξ}_{ikt}, and U^{ξ}_{kt}. Moreover, the decision variables YR^{ξ}_{at} and $YR^{\xi}_{\bar{a}t}$ represent the restoration equipment flow on arc a at time t, WR^{ξ}_{it} represents the number of pieces of restoration equipment available at node i at time t, and finally H^{ξ}_{at} and $H^{\xi}_{\bar{a}t}$ represent the number of pieces of restoration equipment repairing arc a at time t in realization ξ.

The two-stage stochastic programming model in the second step is as follows:

(7) $$min \quad \sum_{i\in N^F}\sum_{l\in L^F} f_l \bar{z}_{il} + \sum_{i\in N^E}\sum_{l\in L^E} e_l V_{il} + \mathbb{E}[Q_2(\bar{z}, v, \boldsymbol{\xi})] \tag{7.1}$$

s.t.

$$\sum_{l\in L^E} V_{il} \le 1 \qquad \forall i \in N^E \tag{7.2}$$

$$V_{il} \in \{0,1\} \qquad \forall i \in N^E, l \in L^E \tag{7.3}$$

where

(8) $$Q_2(\bar{z}, v, \xi) = min \sum_{t\in T}\Big(\sum_{a\in A}\sum_{k\in N} c_a d^{\xi}_k (Y^{\xi}_{akt} + Y^{\xi}_{\bar{a}kt}) + \sum_{a\in A} c'_a\,(YR^{\xi}_{at} + YR^{\xi}_{\bar{a}t}) + b\sum_{k\in N} d^{\xi}_k U^{\xi}_{kt}\Big) \tag{8.1}$$

s.t.

$$\sum_{j\in N(i)} Y^{\xi}_{\bar{a}it} + W^{\xi}_{iit} + U^{\xi}_{it} = \sum_{j\in N(i)} Y^{\xi}_{ait} + 1, \qquad \forall i \in N, t = 1 \tag{8.2}$$

$$\sum_{j\in N(i)} Y^{\xi}_{\bar{a}it} + U^{\xi}_{it} = \sum_{j\in N(i)} Y^{\xi}_{ait} + U^{\xi}_{i,t-1}, \qquad \forall i \in N, \forall\, t \in T\backslash\{1\} \tag{8.3}$$

$$\sum_{j\in N(i)} Y^{\xi}_{\bar{a}kt} + W^{\xi}_{ikt} = \sum_{j\in N(i)} Y^{\xi}_{akt}, \qquad \forall i \in N, \forall k \in N\backslash\{i\}, \forall\, t \in T \tag{8.4}$$

$$\sum_{k\in N}\sum_{t\in T} d^{\xi}_k W^{\xi}_{ikt} \le (1 - \rho^{\xi}_i)\sum_{l\in L^F} q_l \bar{z}_{il}, \qquad \forall i \in N^F \tag{8.5}$$

$$W^{\xi}_{ikt} \le \sum_{l\in L^F} \bar{z}_{il}, \qquad \forall i \in N^F, \forall k \in N, \forall\, t \in T \tag{8.6}$$

$$W^{\xi}_{ikt} = 0, \qquad \forall i \in N\backslash N^F, \forall k \in N, \forall\, t \in T \tag{8.7}$$

$$Y^{\xi}_{akt} \le X^{\xi}_{at}, \qquad \forall a \in A, \forall k \in N, \forall\, t \in T \tag{8.8}$$

$$Y^{\xi}_{\bar{a}kt} \le X^{\xi}_{at}, \qquad \forall a \in A, \forall k \in N, \forall\, t \in T \tag{8.9}$$

$$\sum_{j\in N(i)} YR^{\xi}_{\bar{a}t} + \sum_{l\in L^E} w_l V_{il} = \sum_{j\in N(i)} YR^{\xi}_{at} + WR^{\xi}_{it}, \qquad \forall i \in N^E, \forall\, t \in T \tag{8.10}$$

$$\sum_{j\in N(i)} YR^{\xi}_{\bar{a}t} = \sum_{j\in N(i)} YR^{\xi}_{at} + WR^{\xi}_{it}, \qquad \forall i \in N\backslash N^E, \forall\, t \in T \tag{8.11}$$

$$YR_{at}^{\xi} \leq MX_{at}^{\xi}, \qquad \forall a \in A, \forall t \in T \qquad (8.12)$$

$$YR_{\bar{a}t}^{\xi} \leq MX_{at}^{\xi}, \qquad \forall a \in A, \forall t \in T \qquad (8.13)$$

$$\sum_{j \in N(i)} H_{at}^{\xi} \leq WR_{it}^{\xi}, \qquad \forall i \in N, \forall\, t \in T \qquad (8.14)$$

$$p_a^{\xi} X_{at}^{\xi} = 0, \qquad \forall a \in A, t = 1 \qquad (8.15)$$

$$p_a^{\xi} X_{at}^{\xi} \leq \sum_{t'=1}^{t-1} (H_{at'}^{\xi} + H_{\bar{a}t'}^{\xi}), \qquad \forall a \in A, \forall\, t \in T \backslash \{1\} \qquad (8.16)$$

$$\sum_{t'=1}^{t} (H_{at'}^{\xi} + H_{\bar{a}t'}^{\xi}) \leq p_a^{\xi}, \qquad \forall a \in A, \forall\, t \in T \qquad (8.17)$$

$$H_{at}^{\xi} + H_{\bar{a}t}^{\xi} \leq p_a^{\xi} \sum_{l \in N} U_{lt}^{\xi}, \qquad \forall a \in A, \forall\, t \in T \qquad (8.18)$$

$$0 \leq Y_{akt}^{\xi}, Y_{\bar{a}kt}^{\xi} \leq 1, \qquad \forall a \in A, \forall k \in N, \forall\, t \in T \qquad (8.19)$$

$$0 \leq W_{ikt}^{\xi} \leq 1, \qquad \forall i, k \in N, \forall\, t \in T \qquad (8.20)$$

$$0 \leq U_{kt}^{\xi} \leq 1, \qquad \forall k \in N, \forall t \in T \qquad (8.21)$$

$$YR_{at}^{\xi}, YR_{\bar{a}t}^{\xi} \in \mathbb{Z}^{+}, \qquad \forall a \in A, \forall\, t \in T \qquad (8.22)$$

$$WR_{it}^{\xi} \in \mathbb{Z}^{+}, \qquad \forall i \in N, \forall\, t \in T \qquad (8.23)$$

$$X_{at}^{\xi} \in \{0,1\}, \qquad \forall a \in A, \forall\, t \in T \qquad (8.24)$$

$$H_{at}^{\xi}, H_{\bar{a}t}^{\xi} \in \mathbb{Z}^{+}, \qquad \forall a \in A, \forall\, t \in T \qquad (8.25)$$

In this model, the objective function (7.1) is to minimize the sum of the facility and restoration equipment costs and expected second-stage cost. Constraints (7.2) guarantee that at most one level is chosen from the set of levels for restoration equipment at any candidate node and constraints (7.3) define the first-stage decision variables as binary decisions.

As mentioned above, the model in the first step is a reduced version of the original model. Moreover, the model in the second step is just a small modification of the original one which takes the first-stage facility location decisions as an input from the first step. These two models show similarity to a great extent for the relief item distribution constraints. Constraints (6.2) – (6.8) in the first model correspond to constraints (8.2) – (8.9) in the second model. The main difference is that these common constraints are stated for each time period in the second model. In addition, the second model involves network restoration decisions and constraints. Constraints (8.10) and (8.11) are the flow balance constraints for the restoration equipment. These constraints help to identify the number of pieces of restoration equipment available at every node. Constraints (8.12) and (8.13) ensure that only operational arcs are used to carry the equipment flow. Here, M denotes an adequately large number. Constraints (8.14) allow only available pieces of restoration equipment to be allocated to repair damaged arcs. Constraints (8.15) and (8.16) define the operational arcs by ensuring that an arc is not operational at a certain time period if it has not received the required amount of repair time by that time period. Constraints (8.17) state that no additional repair work is necessary after the number of pieces of restoration equipment allocated to an arc equals the repair time of that arc. Constraints (8.18) ensure that the efforts for network restoration stop as soon as the system-wide demand is satisfied. Note that the

optimal objective function value would not change if constraints (8.17) and (8.18) are removed; however, they help to reduce the number of alternative optimal solutions and consequently the computational efforts.

The two-step sequential approach locates emergency response facilities with a focus on satisfying more demand in the first time period. However, we can fail to find solutions satisfying more demand in the long run if we restrict our attention only to the first time period. We can illustrate how the integrated model outperforms the two-step approach on a small example.

Let us consider the network given in Figure 1. The fixed cost of opening a facility is \$100,000 at any node and the cost of acquiring a unit of restoration equipment is \$20,000. Note that there is only one size of facility and one level of restoration equipment in this example. The travel cost per unit flow (both relief item and restoration equipment) is \$1 for each arc. The penalty cost per unit unmet demand per time period is given as \$100. Let us assume we have a deterministic problem and demand at each node is given in Figure 1. The remaining capacity of any candidate facility is enough to meet all demand. All of the arcs in the network are damaged and the repair time of each arc is one time period.

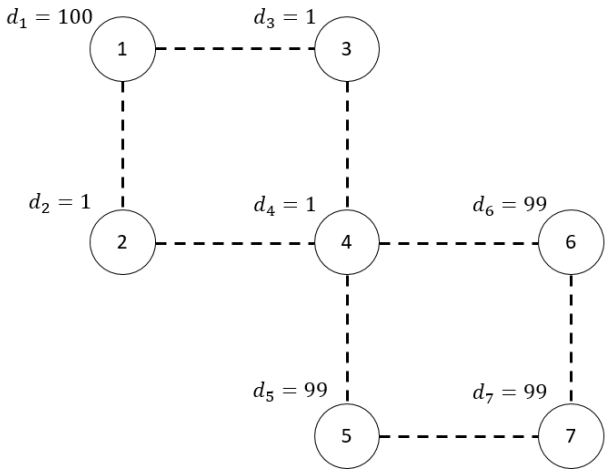

Figure 1 Network of the example problem

The optimal decision obtained from the integrated model in the first stage is to locate one emergency response facility at node 5, one piece of restoration equipment at node 3 and one piece of restoration equipment at node 7. Note that all of the nodes are disconnected from each other right after the disaster. Therefore, only demand at node 5 can be met in the first time period. The repaired arcs are shown in Figure 2 for the first four time periods. Figure 2 shows that node 7 can be reached from node 5 in the second time period since arc (5,7) becomes operational in this time period. Similarly, node 1, node 3, and node 4 are accessible from node 5 in the third time period. Finally, in the fourth time period, node 2 and node 6 can be accessed from node 5. As soon as a node can be reached from node 5 where the emergency response facility is located, emergency relief items are transported to meet the demand fully.

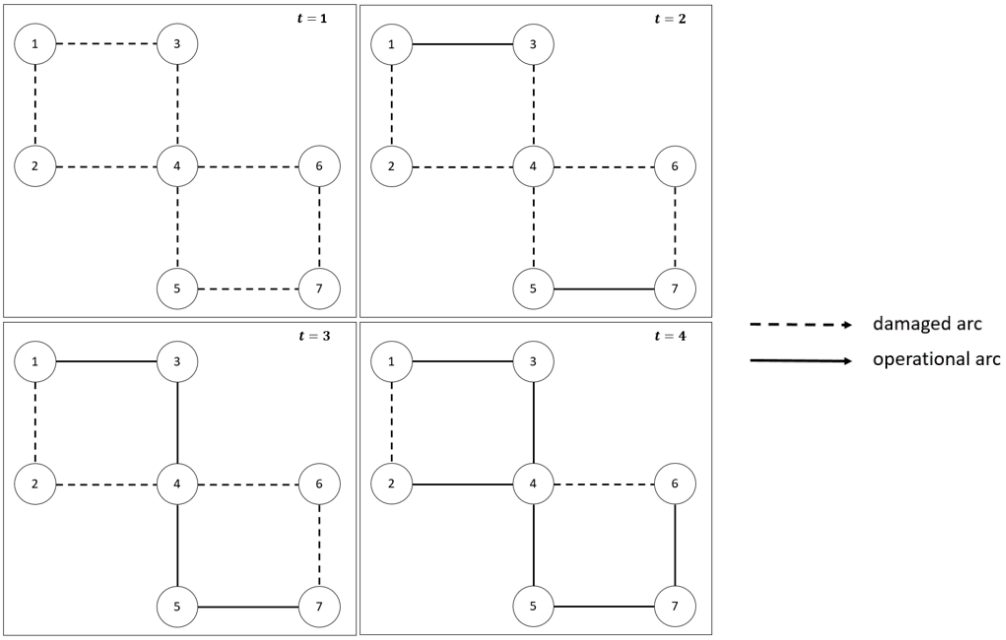

Figure 2 Condition of the network throughout time for the integrated model

Now, let us discuss what would happen differently if we use the two-step approach instead of the integrated model. The two-step approach chooses node 1 to locate an emergency response facility in the first step since it covers the maximum demand that can be satisfied in the first time period. For the given location of the emergency response facility, one piece of restoration equipment is located at node 1 and another piece of restoration equipment is located at node 4 in the second step. As a result, demand at node 1 is met in the first time period, demands at node 3 and node 4 are met in the second time period, demands at node 5 and node 6 are met in the third time period, and demands at node 2 and node 7 are met in the fourth time period. Similar to the integrated model, all demand is satisfied in four time periods. However, by the end of the second time period, the optimal solution obtained by the integrated model satisfies almost double the demand compared to the solution from the two-step approach. To put it another way, the solution from the integrated model satisfies 24% of the total demand one time period earlier. Figure 3 shows the percentage of satisfied demand throughout the time periods for both solutions.

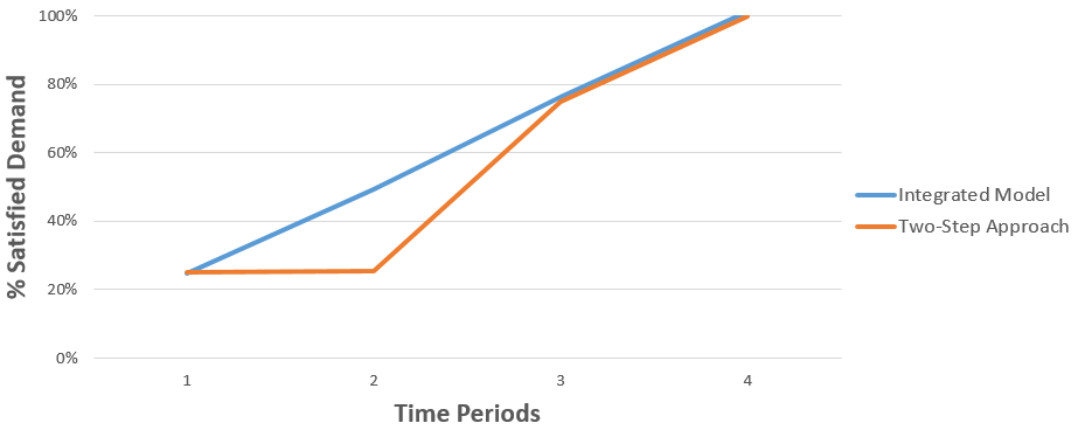

Figure 3 Percentage of satisfied demand for both solutions

Although the two-step approach may result in a suboptimal solution, it can still provide a good upper bound in a reasonable time. Note that any feasible solution for emergency response facilities obtained in the first step is also feasible for the integrated model. Moreover, the model solved in the second step is the same as the original model except that facility decisions are fixed. Therefore, the objective function of the model solved in the second step is a valid upper bound. Solving these two models in consecutive steps is expected to be much faster than solving the original model since the first model considers only a single time period in the second-stage problem by ignoring the network restoration activities. Also, the second model takes the facility locations as an input which reduces the computational burden to a great extent. In the next section, we propose an integer L-shaped algorithm which is an exact solution approach to solve the original model. We start our algorithm with the upper bound found through the two-step sequential procedure.

4. INTEGER L-SHAPED METHOD

As discussed in Section 2, the L-shaped method is an efficient approach to solve two-stage stochastic programs if the second-stage problem is a linear program. However, the integrated model has integer second-stage decision variables which makes the classical L-shaped method not applicable. Sanci and Daskin (2019) show that most of the integrality restrictions in the second-stage problem can be relaxed except for the binary decision variables which determine whether an arc is operational at a certain time period or not. This binary decision variable is necessary to ensure that relief items and restoration equipment can only flow on operational arcs.

In this section, we present a decomposition-based branch-and-cut algorithm to solve the integrated model. This algorithm is adapted from the integer L-shaped method which is developed by Laporte and Louveaux (1993) and later improved by Angulo et al. (2016). To describe our solution approach, let us represent the integrated model by a general formulation as follows:

$$(9) \quad \min f^T z + e^T v + E[Q(z, v, \boldsymbol{\xi})] \tag{9.1}$$

$$\text{s.t.} \quad z \in \mathcal{Z} \cap \{0,1\}^{n_z} \tag{9.2}$$

$$v \in \mathcal{V} \cap \{0,1\}^{n_v} \tag{9.3}$$

$$(10) \quad Q(z, v, \xi) = \min q(\xi)^T y \tag{10.1}$$

$$\text{s.t.} \quad W_y(\xi)y + W_x(\xi)x + T_z(\xi)z + T_v(\xi)v \geq h(\xi) \tag{10.2}$$

$$y \in \mathbb{R}_+^{n_y} \tag{10.3}$$

$$x \in \{0,1\}^{n_x} \tag{10.4}$$

In this model z and v vectors represent the capacity/location decisions for emergency response facilities and restoration equipment, respectively. These decisions belong to the sets $\mathcal{Z}$ and $\mathcal{V}$, which assure that at most one capacity level is chosen at any candidate node. The second-stage decisions are represented by x and y vectors corresponding to binary and continuous variables, respectively.

Now, let us reformulate the first-stage problem as follows:

$$(11) \quad \min f^T z + e^T v + \theta \tag{11.1}$$

$$\text{s.t.} \quad z \in \mathcal{Z} \cap \{0,1\}^{n_z} \qquad (11.2)$$

$$v \in \mathcal{V} \cap \{0,1\}^{n_v} \qquad (11.3)$$

$$\mathcal{Q}(z,v) \leq \theta \qquad (11.4)$$

where $E[Q(z,v,\boldsymbol{\xi})]$ is shown by $\mathcal{Q}(z,v)$ for simplicity.

In the L-shaped method, the idea is to relax $\mathcal{Q}(z,v) \leq \theta$, successively add optimality cuts to better approximate $\mathcal{Q}(z,v)$ until an optimal solution satisfying $\theta^* = \mathcal{Q}(z^*,v^*)$ is found. When the second-stage problem is a linear program, the optimality cuts are simply generated using optimal dual solutions as discussed in Section 2. However, when the second-stage problem is an integer program, the optimality cuts cannot be obtained using LP duality anymore. Laporte and Louveaux (1993) define a new integer optimality cut for stochastic integer programs with binary first-stage decision variables as follows:

$$\theta \geq (\mathcal{Q}(s) - L)\left(\sum_{i \in \mathrm{K}(s)} s_i - \sum_{i \notin \mathrm{K}(s)} s_i - |\mathrm{K}(s)|\right) + \mathcal{Q}(s) \qquad (12)$$

where s is a binary first-stage solution, set $\mathrm{K}(s)$ is defined as $\mathrm{K}(s) \coloneqq \{i\colon\ s_i = 1\}$ and L is a lower bound on $\mathcal{Q}(s)$. Note that we can assume L is zero for our problem since there is no second-stage cost if there is a facility with an adequate size at every node. This optimality cut ensures $\theta \geq \mathcal{Q}(s)$ for solution s since $\sum_{i \in \mathrm{K}(s)} s_i - \sum_{i \notin \mathrm{K}(s)} s_i - |\mathrm{K}(s)|$ is equal to zero at that point. Note that this integer optimality cut is not tight for any other solution.

The solution algorithm by Laporte and Louveaux (1993) utilizes the integer optimality cuts as well as a branch-and-cut scheme in the master problem. That is, the algorithm proceeds by fixing the value of first-stage decision variables on a branch-and-bound tree and the master problem is solved by considering these fixed values in the corresponding node of the tree. In this procedure, when a solution (z,v,θ) is found satisfying (11.2) and (11.3), the algorithm checks whether (11.4) is also satisfied; i.e. $\mathcal{Q}(z,v) \leq \theta$. If this condition is not satisfied, then the optimality cut of the form (12) is added to the master problem of all pending nodes of the tree. Their approach requires the exact evaluation of $\mathcal{Q}(z,v)$ every time a feasible first-stage solution (z,v) is obtained; however, $\mathcal{Q}(z,v)$ may not be easy to compute since it requires solving a MIP for each realization of uncertain parameters. Angulo et al. (2016) modify the procedure so that the same solution can be cut off by checking whether θ is less than $\mathcal{Q}_{LP}(z,v)$, i.e. the expected value of the LP relaxation of the second-stage problem for the solution. If this condition holds, this solution is removed by adding 'the continuous L-shaped optimality cuts' obtained from optimal dual solutions without exactly evaluating $\mathcal{Q}(z,v)$. In case θ is greater than or equal to $\mathcal{Q}_{LP}(z,v)$, the algorithm continues comparing θ with $\mathcal{Q}(z,v)$ as originally proposed by Laporte and Louveaux (1993).

The key point in Angulo at al. (2016) is to use $\theta < \mathcal{Q}_{LP}(z,v)$ as a sufficient condition to reject solution (z,v,θ) since $\mathcal{Q}_{LP}(z,v) \leq \mathcal{Q}(z,v)$. Then, this solution can be simply removed by the continuous L-shaped optimality cuts. Note that these cuts alone are not sufficient for the convergence of the algorithm; however, unlike the integer optimality cuts which are only non-trivial at a specific solution, they provide valid lower bounds on $\mathcal{Q}(z,v)$ for several values of (z,v).

We employ this improved version of the integer L-shaped method to solve our integrated two-stage stochastic programming model. We also made several alterations to the implementation of the method to attain improved solution times. We discuss these changes after we formally describe the algorithm as suggested by Angulo et al. (2016).

The master problem solved at the current node of the branch-and-bound tree takes the form as follows:

$$\text{(13)} \quad \min f^T z + e^T v + \theta \tag{13.1}$$

$$\text{s.t.} \quad z \in \mathcal{Z} \tag{13.2}$$

$$v \in \mathcal{V} \tag{13.3}$$

$$\theta \geq o_j - O_j^F z - O_j^E v \qquad j = 1, 2, \dots, J \tag{13.4}$$

$$\theta \geq Q(z^k, v^k)\left(\sum_{i \in \mathrm{K}(z^k)} z_i - \sum_{i \notin \mathrm{K}(z^k)} z_i + \sum_{i \in \mathrm{K}(v^k)} v_i - \sum_{i \notin \mathrm{K}(v^k)} v_i - \left|\mathrm{K}(z^k)\right| - \left|\mathrm{K}(v^k)\right| + 1 \right)$$

$$k = 1, 2, \dots, K \tag{13.5}$$

In this formulation, J and K denote the number of continuous L-shaped optimality cuts and integer optimality cuts generated until the current node, respectively. Note that we discuss the derivation of the continuous optimality cuts in Section 2. Also, note that we do not include feasibility cuts here since the problem has the relative complete recourse property.

Step 0: Define UB as the upper bound on the optimal objective function value and set UB to the objective function value of the solution found by the two-step sequential approach. Initialize the branch-and-bound tree by defining the only pendant node as the root node (The master problem at the root node is Problem (13) without constraints (13.4) and (13.5)).

Step 1: Select a pendant node. If none exists, stop.

Step 2: Solve the master problem corresponding to the current node. If the master problem is infeasible, fathom this node and go to Step 1. Else, let (z, v, θ) denote the optimal solution of the current master problem and go to Step 3.

Step 3: If $f^T z + e^T v + \theta > UB$, fathom this node and go to Step 1. Else, check whether $z \in \{0,1\}^{n_z}$ and $v \in \{0,1\}^{n_v}$. If all decision variables are binary, go to Step 4. Otherwise, choose a decision variable violating the binary restriction, create two new branches and append the new nodes to the list of pendant nodes, go to Step 1.

Step 4: Compute $Q_{LP}(z, v)$. If $Q_{LP}(z, v) \leq \theta$, go to Step 5. Else, add the corresponding continuous L-shaped optimality cut (13.4) to the master problem and go to Step 2.

Step 5: Compute $Q(z, v)$. If $f^T z + e^T v + Q(z, v) < UB$, update $UB = f^T z + e^T v + Q(z, v)$. If $Q(z, v) \leq \theta$, fathom this node and go to Step 1. Else, generate the corresponding integer optimality cut (13.5) and go to Step 2.

We change two aspects of the algorithm given above.

Modification 1: Instead of branching on z and v, we consider only z for branching. That is, the branch-and-bound tree nodes only impose restrictions on z. Whenever the first-stage variables for the emergency response facilities are all fixed, the model in the second step of the two-step sequential approach is solved to determine the first-stage variables for the restoration equipment. This solution provides the best upper bound that could be obtained by branching on v. Therefore, we do not need to continue branching from this point on. This allows us to reduce the number of branch-and-bound tree nodes; however, this modification requires us to solve the second-step model every time z is fixed. Nevertheless, this model is much easier to solve compared to the integrated model as discussed in Section 3.

Modification 2: We keep the binary restrictions on z and v in the master problem. In the original algorithm, if the first-stage decision variables obtained from the master problem are binary, i.e. $z \in \{0,1\}^{n_z}$ and $v \in \{0,1\}^{n_v}$, the next step is to compute $\mathcal{Q}_{LP}(z,v)$ and check whether $\mathcal{Q}_{LP}(z,v) \leq \theta$. In our modified version of the algorithm, the master problem never returns a fractional first-stage decision variable. Therefore, we check whether $\mathcal{Q}_{LP}(z,v) \leq \theta$ for any solution (z,v,θ) obtained by solving the master problem. Until this condition is satisfied, we add the continuous L-shaped optimality cuts to the master problem. We only start branching when $\mathcal{Q}_{LP}(z,v) = \theta$ at the root node. This means that before we start our branch-and-cut procedure, we obtain the optimal solution for the stochastic program with the LP relaxation of the second-stage problem. This modification results in a 0/1 mixed-integer master problem which is more challenging to solve, especially as the size of the problem gets larger with the added optimality cuts. On the other hand, the lower bound obtained from the master problem is improved which helps to remove more nodes of the branch-and-bound tree from our consideration. In this modified version of the algorithm, we also use another branching scheme instead of branching on a fractional decision variable. We first detect the candidate nodes in the network which are chosen to locate an emergency response facility in the solution obtained from the master problem. This facility can be of any size. Without loss of generality, let us assume there are two levels available for the facilities: small and large facilities. Then, we append three nodes to the list of pendant branch-and-bound nodes: one node for imposing there is a small facility, one node for imposing there is a large facility, and one node for imposing there is no facility. If we cannot detect any candidate node in the network with an open facility which is not fixed in the branch-and-bound tree before, we move on to the candidate nodes without any open facilities. Then, we apply the same branching rule.

We illustrate the difference in the branch-and-bound trees generated without and with these two modifications on a small example. Let us consider a network with m candidate nodes to locate emergency response facilities and m candidate nodes to locate restoration equipment. There are two levels available for both emergency response facilities and restoration equipment. In the original branch-and-cut scheme as applied by Laporte and Louveaux (1993) and Angulo et al. (2016), the master problem solved at the root node may result in a non-integer solution. Let us assume that Z_{11} (indicating whether node 1 is selected to locate a facility of level 1) has a fractional value in the solution obtained from this master problem. Then, two branches are created from the root node. One branch enforces Z_{11} to be 0 and the other branch enforces Z_{11} to be 1. Let us continue with the branch-and-bound

node fixing the value of Z_{11} to 0 in the master problem. This time Z_{12} has a fractional value between 0 and 1. We continue branching in this manner any time the master problem returns a first-stage decision variable with a fractional value. Figure 4 depicts a part of this branch-and-bound tree.

Now, let us examine the branch-and-bound tree generated after our two modifications. Let us assume that the solution found at the root node satisfying $Q_{LP}(z, v) = \theta$ locates one small facility at node 1 and one large facility at node 3. There are no facilities at the other $m - 2$ nodes of the network. In the first level of our branch-and-bound tree, we create three branches: one branch enforcing there is a small facility at node 1 ($Z_{11} = 1$), one branch enforcing there is a large facility at node 1($Z_{12} = 1$), and one branch enforcing there are no facilities at node 1 ($Z_{11} = 0$ and $Z_{12} = 0$). We continue with the branch-and-bound node generated by the first branch. In the second level, we again create three branches: one branch enforcing there is a large facility at node 3 ($Z_{32} = 1$), one branch enforcing there is a small facility at node 3 ($Z_{31} = 1$), and one branch enforcing there are no facilities at node 3 ($Z_{31} = 0$ and $Z_{32} = 0$). As observed from these two levels, we start branching on the nodes with facilities in the solution obtained at the root node. Then, we continue branching on the other nodes of the network with no facilities as partially illustrated in Figure 4.

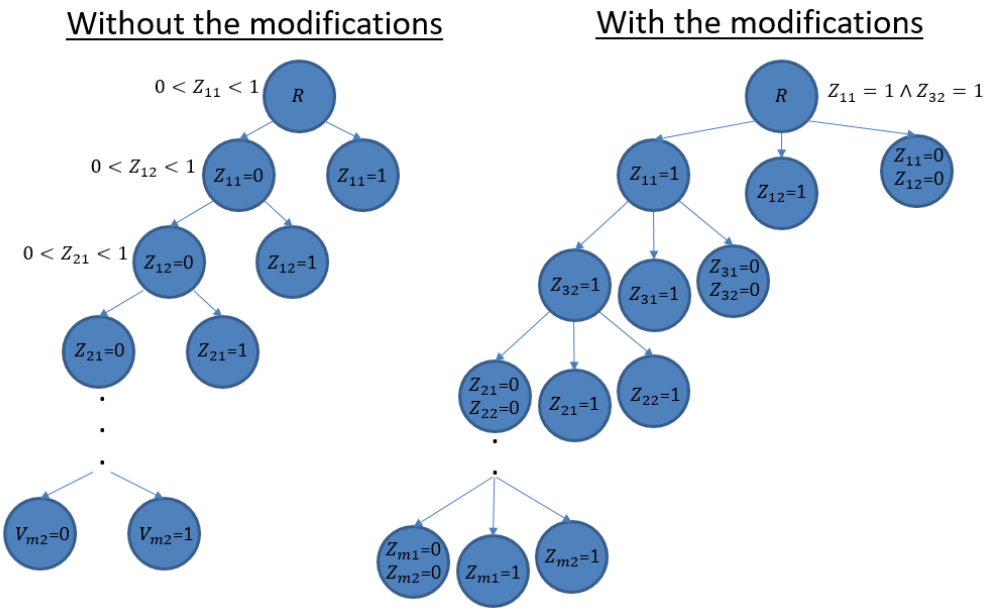

Figure 4 Comparison between the branch-and-bound trees without/with modifications

5. COMPUTATIONAL RESULTS

Sanci and Daskin (2019) apply the integrated model to generate a preparedness strategy for an anticipated earthquake in Istanbul. This case study shows that the model locates facilities with larger capacity at the centers of intense demand regions, and facilities with smaller capacity at the remote demand nodes with intense damage. Moreover, the restoration resources are either located in close proximity to the emergency response facilities or in remote regions. Sanci and Daskin (2019) also study the value of integrating the network restoration problem with the prepositioning problem. The results show that the increase in the satisfied demand will be very limited even if the

pre-disaster investment for emergency response facilities is raised immensely. On the other hand, the total demand can be satisfied within a short time by investing in purchasing restoration equipment, which is typically less costly than establishing facilities.

In this study, our computational results focuses on the solution time. We compare the performance of the CPLEX solver and the integer L-shaped algorithm with and without our modifications. We code the algorithms in C ++ using Microsoft Visual Studio 2015, and solve the mathematical models by IBM ILOG Cplex 12.7.1 on Intel Xeon E3-1505M v5 2.80 GHz, 32 GB RAM using Windows 10.

Before we present our results, we restate the sets and parameters in the case study introduced in Sanci and Daskin (2019) with a few changes. Istanbul has 39 districts with borders defined by the Grand National Assembly of Turkey (See Appendix B for the Istanbul map). We assume that each of these districts is a demand node in the network. We also assume that there is an arc connecting two demand nodes if the corresponding districts share a border and the distance between the district centers is less than or equal to 100 km. We define two demand nodes connected by an arc as neighbor nodes. We assume that the decision makers have identified 10 nodes out of 39 nodes for the sets of candidate facility nodes and candidate restoration equipment nodes separately. As mentioned in Sanci and Daskin (2019), the solution time of the CPLEX solver is extremely long even for a small number of scenarios when the candidate node sets include all 39 nodes. CPLEX also runs into memory problems as the number of scenarios increases when all these nodes are candidate nodes to locate emergency response facilities and restoration equipment. In this paper, we limit the candidate node sets to contain 10 nodes to be able to obtain a solution from CPLEX for scenario sets with up to 50 scenarios and compare its performance with our algorithm. Furthermore, in some cases, the decision makers can indeed pinpoint a subset of nodes which are more suitable to locate emergency response facilities and restoration equipment. In this section, we choose these nodes by generating 1,000 scenarios, solving the deterministic equivalent MIP for each scenario independently, and identifying 10 nodes with the highest number of times selected in these 1,000 solutions.

The sets of levels for facilities and restoration equipment both include two levels. This means that there are two capacity levels available for an open facility. Similarly, if a node is chosen for restoration resources, either one piece or two pieces of restoration equipment can be located at this node. Finally, the number of time periods is limited to three days, since it is widely accepted that the immediate response activities mainly takes place in the first three days after the disaster. We summarize the deterministic parameters in Table 1.

Table 1 Deterministic parameters and their values in the Istanbul case

Parameter Name	Parameter Description	Value
f_1	fixed cost of opening a small facility	\$7.5 million
f_2	fixed cost of opening a large facility	\$10 million
q_1	capacity of a small facility	500,000 units of items
q_2	capacity of a large facility	1,500,000 units of items

e_1	cost of acquiring one piece of restoration equipment	\$0.50 million
e_2	cost of acquiring two pieces of restoration equipment	\$1 million
w_1	number of pieces of restoration equipment for level one	1
w_2	number of pieces of restoration equipment for level two	2
c_a	travel cost per unit relief item flowing on arc a	\$0.50 × length of arc a
c'_a	travel cost per unit equipment flowing on arc a	\$0.50 × length of arc a
b	cost per unit unmet demand per time period	\$170

Sanci and Daskin (2019) also present a scenario generation algorithm to generate a joint realization of the uncertain parameters in every iteration of the algorithm. This algorithm is called n times within the SAA method to generate n disaster scenarios. The proposed algorithm considers the spatial correlation between neighbor nodes through the logistic normal distribution. The main input that the scenario generation algorithm utilizes is the number of damaged buildings for the most probable scenario and the worst-case scenario for the anticipated Istanbul earthquake. These two scenarios were developed as a part of the research project of the Istanbul Metropolitan Municipality with the Japan International Cooperation Agency (JICA, 2002).

In every iteration of the algorithm, the first step is to generate damage ratios of the demand nodes using the data in JICA (2002). Then, demand for relief items are generated by taking into consideration the damage ratios and population of the districts. Finally, the repair time of each arc is generated using the damage ratios of the two corresponding nodes. Sanci and Daskin (2019) assume that the probability that arc (i, j) is damaged is equal to the maximum of the damage ratios; i.e. $max\{\rho_i, \rho_j\}$. Furthermore, they assume that the repair time of a damaged arc has a discrete uniform distribution between one day and five days; however, we take the repair time of a damaged arc as one day for simplicity in this paper. Note that the repair time is zero if an arc is not damaged.

Before we start our analysis, we emphasize that the problem instances solved in this paper and Sanci and Daskin (2019) are not the same due to the differences in the set of candidate nodes and the probability distribution of the repair time. Therefore, the results in these two papers are not comparable.

Using the scenario generation algorithm, we generate scenario samples with varying sizes. Note that Sanci and Daskin (2019) show that the SAA algorithm converges for a sample size of 50 scenarios for a larger instance of the same problem. Therefore, we do not present our SAA analysis in this paper. Instead, we report our results for sample size of $|\Omega| = 10$, 20, 30, 40 and 50. Also, note that we use 30 replications for each sample size. We solve the corresponding problems by the CPLEX solver, the integer L-shaped algorithm and the integer L-shaped algorithm with our modifications. Figure 5 shows the box plots of the CPU times to find a solution whose objective value is within 1% of the optimal objective value for each sample size. This figure illustrates the dramatic increase in the CPU time of CPLEX as the number of scenarios increases. Moreover, the variance of the CPU time is also high when CPLEX is used. The integer L-shaped algorithm is able to reduce the CPU time to an extent; however, the variance is still high for larger scenario sets. On the other hand, the CPU time increases in a linear fashion as the number of

scenarios increases after our modifications in the integer L-shaped algorithm. Next, we analyze the performance of each solution approach in more detail.

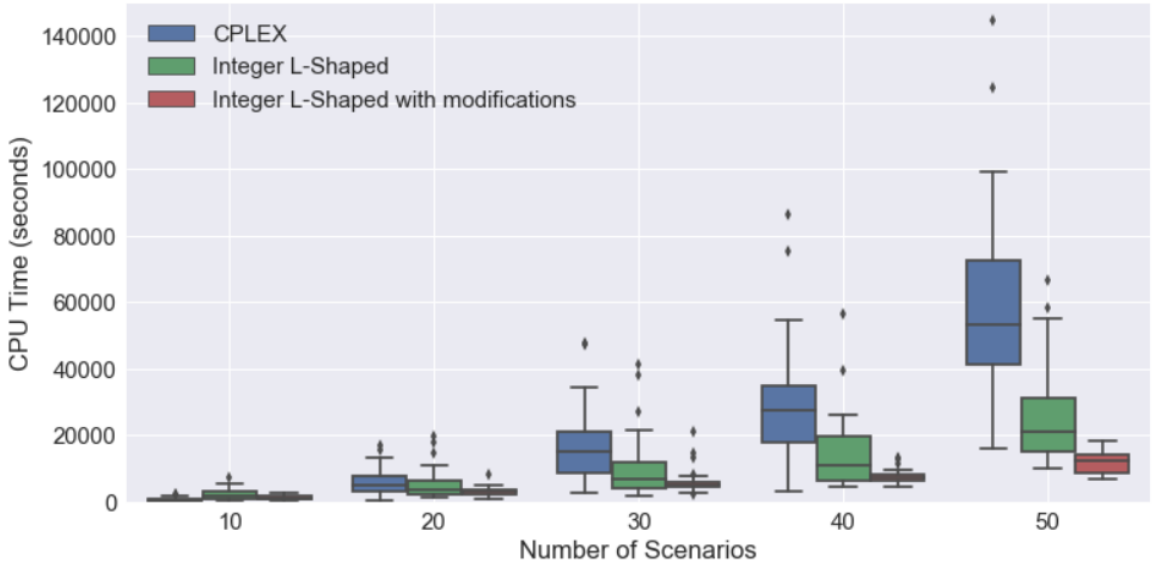

Figure 5 Box plots of the CPU times for different sample sizes

5.1. Results for the CPLEX Solver

We start our analysis with solving the deterministic equivalent MIP of our integrated model by CPLEX. Table 2 gives the average/standard deviation/minimum/maximum CPU time (in seconds) to find a solution whose objective value is within 1% of the optimal objective value (Table C.1 in Appendix C presents the CPU times for all replications). As we can observe from this table, the solution times increase significantly as we increase the number of scenarios in the second-stage problem. To illustrate, the average solution time when $|\Omega| = 50$ is approximately 91 times of the average solution time when $|\Omega| = 10$.

Table 2 CPU time (seconds) to solve the integrated model by CPLEX

	Average CPU Time	**St. Dev CPU Time**	**Minimum CPU Time**	**Maximum CPU Time**
$\|\Omega\|$=10	653	616	59	2399
$\|\Omega\|$=20	6030	4083	473	16862
$\|\Omega\|$=30	16440	11177	2896	47767
$\|\Omega\|$=40	29522	18322	3297	86582
$\|\Omega\|$=50	59692	30265	16209	144962

We perform regression analysis to find the relationship between the CPU time (y_{CPLEX}) and the number of scenarios using the data provided in Table C.1. Our analysis shows that Box-Cox transformation with $\lambda = 1/3$ fits the following regression equation with $R^2 = 0.96$:

$$y_{CPLEX}^{1/3} = 0.77 \times number\ of\ scenarios \text{ (equivalently } y_{CPLEX} = 0.46 \times number\ of\ scenarios\ ^3)$$

The fitted regression equation with the corresponding data points are shown in Figure 6.

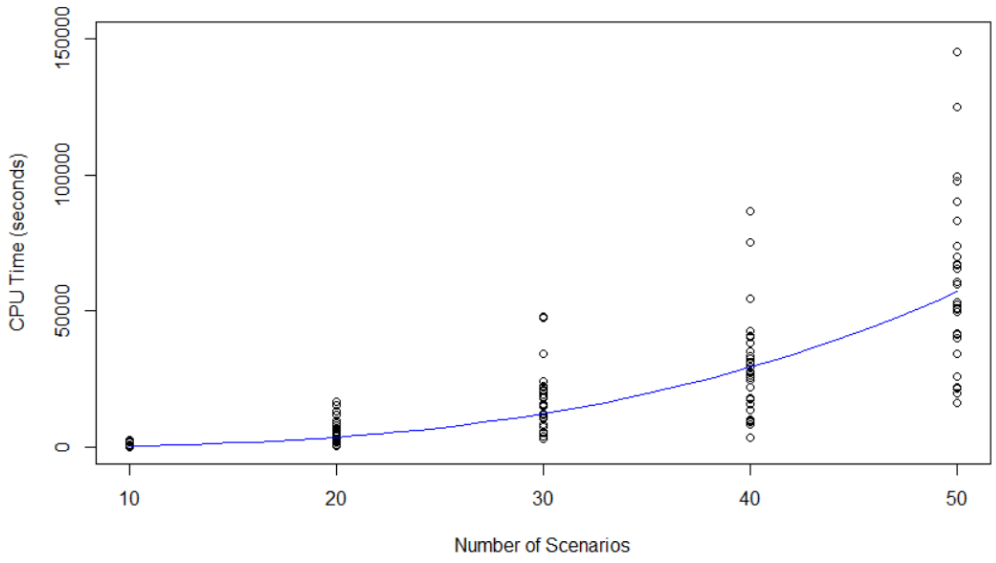

Figure 6 Relationship between the CPU time of CPLEX and the number of scenarios

5.2. Results for the Integer L-shaped Algorithm

The growth in the problem size results in a dramatic increase in the solution time in MIPs. However, the L-shaped method decomposes the second-stage problem into subproblems for each scenario. Therefore, the effect of having larger scenario sets on the solution time is expected to be less significant. In Table 3 we report the average/standard deviation/minimum/maximum CPU time (in seconds) to find a solution with again 1% optimality gap using the integer L-shaped method without the first and second modifications proposed in Section 4 (Table C.2 in Appendix C presents the CPU times for all replications). Therefore, both facility location and restoration equipment location decisions are considered for branching, and the master problem does not impose binary restrictions on these decisions. Let us name this algorithm as ILS I. One can observe that the solution time of CPLEX is shorter than the solution time of ILS I for smaller scenario sets; however, as we increase the number of scenarios, ILS I becomes more advantageous in terms of the solution time.

Table 3 CPU time (seconds) to solve the integrated model by ILS I

	Average CPU Time	**St. Dev CPU Time**	**Minimum CPU Time**	**Maximum CPU Time**
$\|\Omega\|$=10	2232	1660	392	7073
$\|\Omega\|$=20	5449	4754	1108	19824
$\|\Omega\|$=30	10524	9936	1882	41192
$\|\Omega\|$=40	14822	11521	4642	56635
$\|\Omega\|$=50	25657	14410	10053	66442

We report the average number of branch-and-bound nodes generated (average #nodes), the average number of nodes fathomed (average #fathomed), and the average number of nodes at the final level of the branch-and-bound tree (average #final) in Table 4. Note that the final level corresponds to the case in which all first-stage decision variables are fixed and the expected second-stage cost can be computed by solving the subproblems for the given first-stage decisions. Therefore, #final denotes how many times this computation is performed.

Table 4 Branch-and-bound metrics for ILS I

	Average #nodes	**Average #fathomed**	**Average #final**
$\|\Omega\|$=10	6292	4195	20
$\|\Omega\|$=20	11365	7577	26
$\|\Omega\|$=30	13509	9006	42
$\|\Omega\|$=40	14130	9420	25
$\|\Omega\|$=50	20143	13429	29

We again perform regression analysis to find the relationship between the CPU time (y_{ILS1}) and the number of scenarios using the data provided in Table C.2. This time we use $\lambda = 1/2.5 = 0.4$ for the Box-Cox transformation. We fit the following regression equation with $R^2 = 0.92$:

$$y_{ILS1}^{0.4} = 1.18 \times number\ of\ scenarios \quad \text{(equivalently } y_{ILS1} = 1.51 \times number\ of\ scenarios^{2.5}\text{)}$$

Figure 7 (scaled to be comparable with Figure 6) shows the fitted regression equation and the corresponding data points.

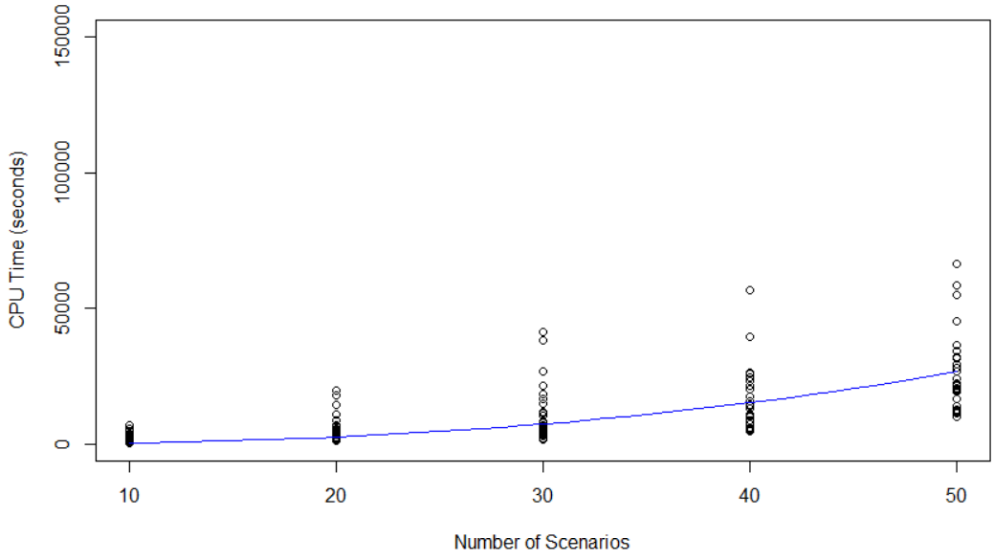

Figure 7 Relationship between the CPU time of the integer L-shaped method and the number of scenarios

5.3. Results for the Integer L-shaped Algorithm with the First Modification

Next, we discuss the effect of applying the modifications we proposed to the integer L-shaped algorithm. Let ILS II be the first modified algorithm which uses only facility location decisions for branching. Whenever, all first-stage location decisions for emergency response facilities are fixed, the model in the second step of the sequential approach is solved to determine the location of restoration equipment. Table 5 presents the average/standard deviation/minimum/maximum CPU time (in seconds) to find a solution with 1% optimality gap using ILS II (Table C.3 in Appendix C presents the CPU times for all replications). Although the second-stage is decomposed into subproblems here, this algorithm cannot exploit this structure effectively. This is mainly due to the fact that there are a large number of final nodes for many of the problem instances, and the second-step MIP is solved for each final node. Note that this second-step MIP is much easier to solve compared to the deterministic equivalent MIP as discussed in Section 3; however, this algorithm requires us to solve this MIP multiple times which may again end up in increased solution times. Table 6 together with Table 5 suggests that the solution time increases as the nodes cannot be fathomed effectively and the number of final nodes is large.

Table 5 CPU time (seconds) to solve the integrated model by ILS II

	Average CPU Time	**St. Dev CPU Time**	**Minimum CPU Time**	**Maximum CPU Time**
$\|\Omega\|$=10	745	495	291	2201
$\|\Omega\|$=20	4309	4679	1138	20627
$\|\Omega\|$=30	14247	19605	1610	92408
$\|\Omega\|$=40	20867	15928	4052	84211
$\|\Omega\|$=50	71653	44391	22052	193291

Table 6 Branch-and-bound metrics for ILS II

	Average #nodes	**Average #fathomed**	**Average #final**
$\|\Omega\|$=10	692	462	25
$\|\Omega\|$=20	1594	1063	82
$\|\Omega\|$=30	2044	1363	125
$\|\Omega\|$=40	2200	1467	125
$\|\Omega\|$=50	2908	1939	211

5.4. Results for the Integer L-shaped Algorithm with the First and Second Modifications

Our second modification, which is to impose the original binary restrictions in the master problem, focuses on improving the lower bounds obtained in the branch-and-bound tree so that non-promising nodes are detected in the

upper levels of the tree. Let us name the algorithm with both modifications as ILS III. The average/standard deviation/minimum/maximum CPU time (in seconds) to find a solution with 1% optimality gap and the corresponding branch-and-bound metrics are reported in Table 7 and Table 8, respectively. (Table C.4 in Appendix C presents the CPU times for all replications.) We can see from Table 7 that the solution times are improved by ILS III for the larger set of scenarios. As we can infer from Table 8, this algorithm effectively prunes the branch-and-bound tree. It eliminates the need to solve the second-step MIP since a solution's objective function value is proven to be within 1% of the optimal objective function before reaching the final level of the search tree.

Table 7 CPU time (seconds) to solve the integrated model by ILS III

	Average CPU Time	**St. Dev CPU Time**	**Minimum CPU Time**	**Maximum CPU Time**
|Ω|=10	1180	515	506	2463
|Ω|=20	3123	1335	1048	8275
|Ω|=30	6075	3880	1978	21233
|Ω|=40	7525	2190	4615	13429
|Ω|=50	11910	3166	6825	18092

Table 8 Branch-and-bound metrics for ILS III

	Average #nodes	**Average #fathomed**	**Average #final**
|Ω|=10	6	4	0
|Ω|=20	6	4	0
|Ω|=30	9	6	0
|Ω|=40	5	3	0
|Ω|=50	5	3	0

Using the data provided in Table C.4, we repeat our regression analysis to find the relationship between the CPU time (y_{ILS3}) and the number of scenarios. We fit the following regression equation with $R^2 = 0.87$:

$$y_{ILS3} = 209.63 \times number\ of\ scenarios$$

Figure 8 (scaled to be comparable with Figure 6) shows the fitted regression equation and the corresponding data points.

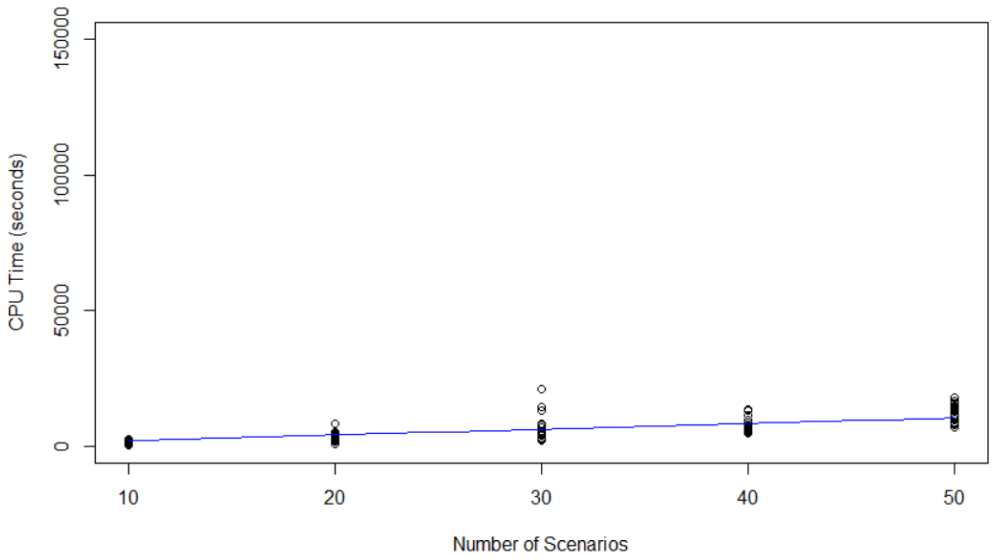

Figure 8 Relationship between the CPU time of the integer L-shaped method (with modifications) and the number of scenarios

The reason that ILS III outperforms ILS II for the larger set of scenarios is its ability to obtain tight lower bounds even in the root node of the search tree. Table 9 shows the relative gap between the lower bound at the root node and the upper bound when the algorithm terminates. In this table, we report the minimum/average/maximum relative gap when the master problem is a linear program and a mixed-integer program. As one can observe from Table 9, the lower bounds in the latter case are considerably tighter than the ones obtained in the former case.

Table 9 Minimum/average/maximum relative gap corresponding to the lower bound at the root node

	LP Master Problem			MIP Master Problem		
	Minimum	**Average**	**Maximum**	**Minimum**	**Average**	**Maximum**
$\|\Omega\|$=10	0.3%	3.8%	8.7%	0.1%	0.7%	1.5%
$\|\Omega\|$=20	2.7%	5.1%	7.3%	0.3%	0.8%	1.5%
$\|\Omega\|$=30	2.4%	5.2%	7.5%	0.2%	0.8%	1.5%
$\|\Omega\|$=40	3.2%	5.6%	7.7%	0.4%	0.8%	1.4%
$\|\Omega\|$=50	3.6%	5.7%	7.0%	0.5%	0.8%	1.2%

5.5. Results for the Two-Step Sequential Approach

Finally, we discuss the performance of the two-step sequential approach. As we mention earlier, we use the solution obtained from this heuristic procedure to start with a good upper bound on the optimal objective function value. Having a tight upper bound initially is important since it leads the branch-and-bound tree to search for more promising regions of the tree. Table 10 presents the minimum/average/maximum relative gap between the upper bound found by the two-step sequential procedure and the objective function value of the solution which is proven

to be within 1% of the optimality (Table C.5 in Appendix C presents the relative gaps for all replications). We can see from this table that this heuristic provides very good upper bounds in general.

Table 10 Relative gap corresponding to the upper bound obtained from the heuristic

	Minimum Relative Gap	Average Relative Gap	Maximum Relative Gap
$\|\Omega\|$=10	0.00%	0.20%	1.80%
$\|\Omega\|$=20	0.00%	0.40%	1.60%
$\|\Omega\|$=30	0.00%	0.35%	2.00%
$\|\Omega\|$=40	0.00%	0.36%	1.80%
$\|\Omega\|$=50	0.00%	0.54%	2.00%

Table 11 provides the minimum/average/maximum CPU time to solve the stochastic programming models in the first and second steps of the two-step sequential procedure as well as the minimum/average/maximum total CPU time of the procedure. As we can observe from this table, the run time of this procedure is fairly short even for $|\Omega| = 50$. Table 10 and Table 11 together suggest that the two-step sequential approach generally provides a tight upper bound in a reasonably short time.

Table 11 Minimum/average/maximum CPU time of the two-step approach

	First Step CPU Time (sec)			Second Step CPU Time (sec)			Two-Step Total CPU Time (sec)		
	Min	**Avg**	**Max**	**Min**	**Avg**	**Max**	**Min**	**Avg**	**Max**
$\|\Omega\|$=10	2	9	21	6	10	15	12	19	29
$\|\Omega\|$=20	15	39	95	15	26	135	36	64	163
$\|\Omega\|$=30	43	79	135	27	89	983	77	168	1037
$\|\Omega\|$=40	77	161	312	46	163	932	136	324	1086
$\|\Omega\|$=50	113	276	523	61	212	976	220	488	1131

We use the upper bound obtained from the two-step sequential approach to initialize the integer L-shaped algorithm with and without our modifications. However, we do not use this information in the deterministic equivalent MIP solved by CPLEX. To see if this heuristic solution helps to improve the solution time of CPLEX, we add a constraint ensuring that the objective function value does not exceed the objective value of this solution. Figure 9 illustrates the CPU time of CPLEX before and after the addition of the constraint. Note that the latter includes the CPU time of the two-step sequential procedure as well. This figure suggests that the upper bound information does not have a significant effect on the CPU time of CPLEX.

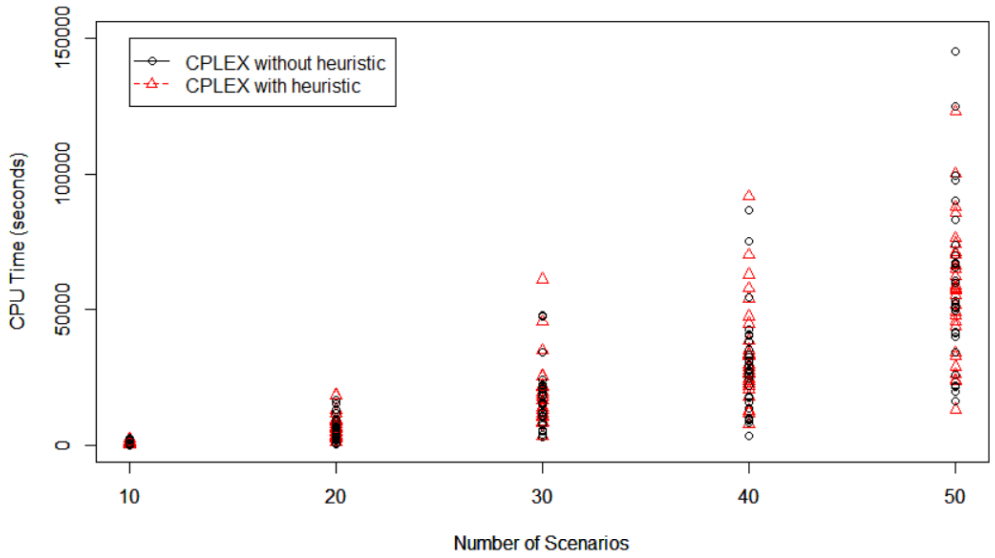

Figure 9 Scatter plot for CPU time of CPLEX with/without heuristic solution vs. the number of scenarios

6. SUMMARY

In this paper, we propose an exact solution approach to solve the integrated location and network restoration model. This solution approach is based on the integer L-shaped method developed by Laporte and Louveaux (1993) to solve two-stage stochastic programs with binary first-stage decision variables and mixed-integer second-stage decision variables. We implement the improved version of the algorithm (Angulo et al., 2016) with two modifications to better exploit the structure of our integrated model. Our computational results show that our modified integer L-shaped algorithm outperforms the CPLEX solver and the integer L-shaped algorithm without our modifications in terms of the solution time. We empirically show that the solution time of our algorithm increases only linearly as the number of scenarios increases.

Incorporating chance constraints and considering the deprivation costs in the objective function are increasingly becoming popular in disaster relief management literature. An interesting future research direction might be to approach the integrated location and network restoration problem using these frameworks and develop efficient solution algorithms. Another intriguing direction might be to investigate the performance of our modified integer L-shaped algorithm to solve similar two-stage stochastic programming models integrating two problems in the first and second stages.

ACKNOWLEDGMENTS

This work was funded by a variety of internal University of Michigan funding sources.

REFERENCES

Ahmadi, M., Seifi, A., and Tootooni, B. (2015). A humanitarian logistics model for disaster relief operation considering network failure and standard relief time: A case study on San Francisco district. *Transportation Research Part E*, 75, 145-163.

Ahmed, S. (2010). Two-stage stochastic integer programming: A brief introduction. In Cochran et al. (eds.), *Wiley Encyclopedia of Operations Research and Management Science*. Wiley.

Ahmed, S., Tawarmalani, M., and Sahinidis, N.V. (2004). A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, 100, 355-377.

Alem, D., Clark, A. and Moreno, A. (2016). Stochastic network models for logistics planning in disaster relief. *European Journal of Operational Research*, 255, 187-206.

Angulo, G., Ahmed, S., and Dey, S.S. (2016). Improving the Integer L-shaped Method. *INFORMS Journal on Computing*, 28(3), 483-499.

Aslan, E. and Çelik, M. (2018). Pre-positioning of relief items under road/facility vulnerability with concurrent restoration and relief transportation. *IISE Transactions*, 51(8), 847-868.

Balcik, B. and Beamon, B.M. (2008). Facility location in humanitarian relief. *International Journal of Logistics: Research and Applications*, 11(2), 101-121.

Caroe, C.C. and Tind, J. (1998). L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83, 451-464.

Caroe, C.C. and Schultz, R. (1999). Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24, 37-45.

Çelik, M., Ergun, O., Johnson, B., Keskinocak, P., Lorca, A., Pekgun, P., and Swann, J. (2014). Humanitarian Logistics. In *INFORMS TutORials in Operations Research*. Published online: 14 Oct 2014; 18-49. https://dx.doi.org/10.1287/educ.1120.0100

Çelik, M., Ergun, Ö., and Keskinocak, P. (2015). The post-disaster debris clearance problem under incomplete information. *Operations Research*, 63(1), 65-85.

Döyen, A., Aras, N., and Barbarosoğlu G. (2012). A two-echelon stochastic facility location model for humanitarian relief logistics. *Optimization Letters*, 6, 1123-1145.

Duran, S., Gutierrez, M.A., and Keskinocak, P. (2011). Pre-positioning of emergency items for CARE International. *Interfaces*, 41(3), 223-237.

Elçi, Ö. and Noyan, N. (2018). A chance-constrained two-stage stochastic programming model for humanitarian relief network design. *Transportation Research Part B*, 108, 55-83.

Gade, D., Kucukyavuz, S., and Sen, S. (2014). Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, 144, 39-64.

Gendreau, M., Laporte, G., and Seguin, R. (1995). An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, 29(2), 143-155.

Grass, E. and Fischer, K. (2016). Two-stage stochastic programming in disaster management: A literature survey. *Surveys in Operations Research and Management Science*, 21, 85-100.

JICA (2002). The study on a disaster prevention/mitigation basic plan in Istanbul including seismic microzonation in the Republic of Turkey. Final Report. Japan International Cooperation Agency.

Holguin-Veras, J., Perez, N., Jaller, M., Van Wassenhove, L., and Aros-Vera, F. (2013). On the appropriate objective function for post-disaster humanitarian logistics models. *Journal of Operations Management*, 31, 262-280.

Hong, X., Lejeune, M.A., and Noyan, N. (2015). Stochastic network design for disaster preparedness. *IIE Transactions*, 47(4), 329-357.

Hu, S., Han, C., Dong, Z.S., and Meng, L. (2019). A multi-stage stochastic programming model for relief distribution considering the state of road network. *Transportation Research Part B*, 123, 64-87.

Kara, B.Y. and Savaşer, S. (2017). Humanitarian Logistics. In *INFORMS TutORials in Operations Research*. Published online: 03 Oct 2017; 263-303. https://doi.org/10.1287/educ.2017.0171

Kovacs, G. and Moshtari, M. (2019). A roadmap for higher research quality in humanitarian operations: A methodological perspective. *European Journal of Operational Research*, 276, 395-408.

Kucukyavuz, S. and Sen, S. (2017). An introduction to two-stage stochastic mixed-integer programming. In *INFORMS TutORials in Operations Research*. Published online: 03 Oct 2017; 1-27. https://doi.org/10.1287/educ.2017.0171

Laporte, G. and Louveaux, F.V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13, 133-142.

Laporte, G., Louveaux, F.V., Mercure, H. (1994). A priori optimization of the probabilistic traveling salesman problem. *Operations Research*, 42(3), 543-549.

Laporte, G., Louveaux, F.V., van Hamme, L. (1994). Exact solution to a location problem with stochastic demands. *Transportation Science*, 28(2), 95-103.

Laporte, G., Louveaux, F.V., van Hamme, L. (2002). An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3), 415-425.

Liberatore, F., Ortuno, M.T., Tirado, G., Vitoriano, B., and Scaparra M.P. (2014). A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in Humanitarian Logistics. Computers & Operations Research, 42, 3-13.

Mete, H.O. and Zabinsky, Z.B. (2010). Stochastic optimization of medical supply location and distribution in disaster management. *International Journal of Production Economics*, 126, 76-84.

Moreno, A., Alem, D., Ferreira, D., and Clark, A. (2018). An effective two-stage stochastic multi-trip location-transportation model with social concerns in relief supply chains. *European Journal of Operational Research*, 269, 1050-1071.

Miller-Hooks, E., Zhang, X., and Faturechi, R. (2012). Measuring and maximizing resilience of freight transportation networks. *Computers & Operations Research*, 39, 1633-1643.

Noyan, N., Balcik, B., and Atakan, S. (2016). A stochastic optimization model for designing last mile relief networks. *Transportation Science*, 50(3), 1092-1113.

Noyan, N. and Kahvecioglu, G. (2018). Stochastic last mile relief network design with recourse reallocation. *OR Spectrum*, 40, 187-231.

Paul, J.A. and MacDonald, L. (2016). Location and capacity allocations decisions to mitigate the impacts of unexpected disasters. *European Journal of Operational Research*, 251, 252-263.

Paul, J.A. and Wang, X.J. (2019). Robust location-allocation network design for earthquake preparedness. *Transportation Research Part B*, 119, 139-155.

Paul, J.A. and Zhang, M. (2019). Supply location and transportation planning for hurricanes: A two-stage stochastic programming framework. *European Journal of Operational Research*, 274, 108-125.

Ransikarbum, K. and Mason S.J. (2016). Multiple-objective analysis of integrated relief supply and network restoration in humanitarian logistics operations. *International Journal of Production Research*, 54(1), 49-68.

Rawls, C.G. and Turnquist, M.A. (2010). Pre-positioning of emergency supplies for disaster response. *Transportation Research Part B*, 44, 521-534.

Rosing, K.E. and ReVelle, C.S. (1997). Heuristic concentration: two-stage solution construction. *European Journal of Operational Research*, 97, 75-86.

Salman, F.S. and Yücel, E. (2015). Emergency facility location under random network damage: Insights from the Istanbul case. *Computers & Operations Research*, 62, 266-281.

Salmeron, J. and Apte, A. (2010). Stochastic optimization for natural disaster asset prepositioning. *Production and Operations Management*, 19(5), 561-574.

Sanci, E. and Daskin, M.S. (2019). Integrating location and network restoration decisions in relief networks under uncertainty. *European Journal of Operational Research*, 279, 335-350.

Schultz, R. (1995). On structure and stability in stochastic programs with random technology matrix and complete integer recourse. *Mathematical Programming*, 70(1), 73-89.

Sen, S. (2005). Algorithms for stochastic mixed-integer programming models. In Aardal et al. (eds.), *Handbook in OR & MS: Discrete Optimization*. Elsevier.

Sen, S. and Higle, J. (2005). The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104, 1-20.

Sen, S. and Sherali, H.D. (2006). Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106, 203-223.

Sherali, H.D. and Fraticelli, B.M.P. (2002). A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22, 319-342.

Sherali, H.D. and Zhu, X. (2006). On solving discrete two-stage stochastic programs having mixed-integer first- and second-stage variables. *Mathematical Programming*, 108, 597-616.

Van Slyke, R.M. and Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4), 638-663.

Zhang, M. and Kucukyavuz, S. (2014). Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. SIAM Journal on Optimization, 24(4), 1933-1951.

APPENDIX A

Sets	First-Stage Decision Variables
• N: set of nodes. • $N(i)$: set of neighbor nodes of node i.	• Z_{il}= 1 if a facility is located at node i with level l and 0 otherwise.

- N^F: set of candidate nodes for facilities.
- N^E: set of candidate nodes for equipment.
- L^F: set of levels for facilities.
- L^E: set of levels for equipment.
- A: set of arcs.
- T: set of time periods.

Deterministic Parameters

- f_l: fixed cost of opening a facility at level l.
- q_l: capacity of the facility at level l.
- e_l: cost of acquiring restoration equipment at level l.
- w_l: pieces of restoration equipment at level l.
- c_a: travel cost per unit relief item flowing on arc a.
- c'_a: travel cost per unit equipment flowing on arc a.
- b: cost per unit unmet demand per time period.

Uncertain Parameters

- d_i^ξ: demand at node i in realization ξ.
- ρ_i^ξ: damage ratio at node i in realization ξ.
- p_a^ξ: repair time of arc a in realization ξ.

- V_{il}= 1 if restoration equipment is located at node i with level l and 0 otherwise.
- Z and V denote the decision vectors for facility locations and restoration equipment, respectively.

Second-Stage Decision Variables

- a and $\bar{a}$ denote $i \to j$ and $j \to i$ directions.
- $Y_{akt}^\xi, Y_{\bar{a}kt}^\xi$= fraction of demand at node k flowing on arc a at time t in realization ξ.
- W_{ikt}^ξ= fraction of demand at node k served by the facility at node i at time t in realization ξ.
- U_{kt}^ξ= fraction of unmet demand at node k at time t in realization ξ.
- $YR_{at}^\xi, YR_{\bar{a}t}^\xi$= number of pieces of restoration equipment flowing on arc a at time t in realization ξ.
- WR_{it}^ξ= number of pieces of restoration equipment available at node i at time t in realization ξ.
- X_{at}^ξ= 1 if arc a is operational at time t in realization ξ and 0 otherwise.
- $H_{at}^\xi, H_{\bar{a}t}^\xi$= number of pieces of restoration equipment repairing arc a at time t in realization ξ.

$$min \quad \sum_{i\in N^F}\sum_{l\in L^F} f_l Z_{il} + \sum_{i\in N^E}\sum_{l\in L^E} e_l V_{il} + \mathbb{E}[Q(Z,V,\boldsymbol{\xi})]$$

s.t.

$$\begin{aligned}
&\textstyle\sum_{l\in L^F} Z_{il} \le 1 && \forall i \in N^F \\
&\textstyle\sum_{l\in L^E} V_{il} \le 1 && \forall i \in N^E \\
&Z_{il} \in \{0,1\} && \forall i \in N^F, l \in L^F \\
&V_{il} \in \{0,1\} && \forall i \in N^E, l \in L^E
\end{aligned}$$

$$Q(Z,V,\xi) = min \sum_{t\in T}\left(\sum_{a\in A}\sum_{k\in N} c_a d_k^\xi (Y_{akt}^\xi + Y_{\bar{a}kt}^\xi) + \sum_{a\in A} c'_a\,(YR_{at}^\xi + YR_{\bar{a}t}^\xi) + b\sum_{k\in N} d_k^\xi U_{kt}^\xi\right)$$

s.t.

$$\begin{aligned}
&\textstyle\sum_{j\in N(i)} Y_{\bar{a}it}^\xi + W_{iit}^\xi + U_{it}^\xi = \sum_{j\in N(i)} Y_{ait}^\xi + 1, && \forall i \in N, t = 1 \\
&\textstyle\sum_{j\in N(i)} Y_{\bar{a}it}^\xi + U_{it}^\xi = \sum_{j\in N(i)} Y_{ait}^\xi + U_{i,t-1}^\xi, && \forall i \in N, \forall\, t \in T\backslash\{1\} \\
&\textstyle\sum_{j\in N(i)} Y_{\bar{a}kt}^\xi + W_{ikt}^\xi = \sum_{j\in N(i)} Y_{akt}^\xi, && \forall i \in N, \forall k \in N\backslash\{i\}, \forall\, t \in T
\end{aligned}$$

$$\sum_{k\in N}\sum_{t\in T} d_k^{\xi} W_{ikt}^{\xi} \le (1-\rho_i^{\xi})\sum_{l\in L^F} q_l Z_{il}, \quad \forall i \in N^F$$

$$W_{ikt}^{\xi} \le \sum_{l\in L^F} Z_{il}, \quad \forall i \in N^F, \forall k \in N, \forall\, t \in T$$

$$W_{ikt}^{\xi} = 0, \quad \forall i \in N\backslash N^F, \forall k \in N, \forall\, t \in T$$

$$Y_{akt}^{\xi} \le X_{at}^{\xi}, \quad \forall a \in A, \forall k \in N, \forall\, t \in T$$

$$Y_{\bar{a}kt}^{\xi} \le X_{at}^{\xi}, \quad \forall a \in A, \forall k \in N, \forall\, t \in T$$

$$\sum_{j\in N(i)} YR_{\bar{a}t}^{\xi} + \sum_{l\in L^E} w_l V_{il} = \sum_{j\in N(i)} YR_{at}^{\xi} + WR_{it}^{\xi}, \quad \forall i \in N^E, \forall\, t \in T$$

$$\sum_{j\in N(i)} YR_{\bar{a}t}^{\xi} = \sum_{j\in N(i)} YR_{at}^{\xi} + WR_{it}^{\xi}, \quad \forall i \in N\backslash N^E, \forall\, t \in T$$

$$YR_{at}^{\xi} \le MX_{at}^{\xi}, \quad \forall a \in A, \forall t \in T$$

$$YR_{\bar{a}t}^{\xi} \le MX_{at}^{\xi}, \quad \forall a \in A, \forall t \in T$$

$$\sum_{j\in N(i)} H_{at}^{\xi} \le WR_{it}^{\xi}, \quad \forall i \in N, \forall\, t \in T$$

$$p_a^{\xi} X_{at}^{\xi} = 0, \quad \forall a \in A, t = 1$$

$$p_a^{\xi} X_{at}^{\xi} \le \sum_{t'=1}^{t-1}(H_{at'}^{\xi} + H_{\bar{a}t'}^{\xi}), \quad \forall a \in A, \forall\, t \in T\backslash\{1\}$$

$$\sum_{t'=1}^{t}(H_{at'}^{\xi} + H_{\bar{a}t'}^{\xi}) \le p_a^{\xi}, \quad \forall a \in A, \forall\, t \in T$$

$$H_{at}^{\xi} + H_{\bar{a}t}^{\xi} \le p_a^{\xi}\sum_{l\in N} U_{lt}^{\xi}, \quad \forall a \in A, \forall\, t \in T$$

$$0 \le Y_{akt}^{\xi}, Y_{\bar{a}kt}^{\xi} \le 1, \quad \forall a \in A, \forall k \in N, \forall\, t \in T$$

$$0 \le W_{ikt}^{\xi} \le 1, \quad \forall i,k \in N, \forall\, t \in T$$

$$0 \le U_{kt}^{\xi} \le 1, \quad \forall k \in N, \forall t \in T$$

$$YR_{at}^{\xi}, YR_{\bar{a}t}^{\xi} \in \mathbb{Z}^+, \quad \forall a \in A, \forall\, t \in T$$

$$WR_{it}^{\xi} \in \mathbb{Z}^+, \quad \forall i \in N, \forall\, t \in T$$

$$X_{at}^{\xi} \in \{0,1\}, \quad \forall a \in A, \forall\, t \in T$$

$$H_{at}^{\xi}, H_{\bar{a}t}^{\xi} \in \mathbb{Z}^+, \quad \forall a \in A, \forall\, t \in T$$

APPENDIX B

Figure B.1 39 nodes in the Istanbul network

APPENDIX C

Note that each cell in Table C.1 corresponds to the same problem instance of the cells in Table C.2, Table C.3, Table C.4, and Table C.5.

Table C.1 CPU time (seconds) to solve the integrated model by CPLEX

	Replications									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
|Ω|=10	858	781	2399	295	59	910	1696	88	1667	222
|Ω|=20	1067	15483	4090	3722	7166	6491	7725	9496	5348	5234
|Ω|=30	15911	22374	20999	11883	8132	15172	8516	4122	10659	47441
|Ω|=40	9136	38272	13753	27370	17886	40290	31103	21793	42811	86582
|Ω|=50	21313	124660	99408	90255	41345	60480	16209	50822	19980	53015
	Replications									
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
|Ω|=10	989	649	106	71	631	88	2100	524	288	452
|Ω|=20	13310	8969	4048	5800	1827	16862	11751	2367	4558	2151
|Ω|=30	14878	5531	8463	20706	24154	12096	5063	10832	2896	47767
|Ω|=40	17652	25361	33414	40987	10238	54538	25601	32431	26271	30916
|Ω|=50	83243	52271	50662	41651	73702	41176	49715	40049	25806	59703
	Replications									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
|Ω|=10	920	912	697	79	875	406	177	273	207	167
|Ω|=20	4490	2572	8725	2883	2383	5358	473	4999	6693	4872
|Ω|=30	15832	18103	19130	2986	22225	7602	14895	22045	18516	34256
|Ω|=40	75393	27557	9460	24426	15655	30766	8212	3297	29286	35199
|Ω|=50	70091	65470	34402	53122	22055	66633	144962	67345	97398	73804

Table C.2 CPU time (seconds) to solve the integrated model by ILS I

	Replications									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
|Ω|=10	3037	1328	1692	5523	2066	7073	3856	684	2160	1118
|Ω|=20	2719	3286	8636	4285	1853	1680	8671	14440	4611	7181
|Ω|=30	18466	14870	8328	4300	21506	10641	3479	3974	3704	6565
|Ω|=40	4642	14327	5800	15797	25898	11002	7249	56635	10002	7402
|Ω|=50	10053	31608	45201	29254	14245	16747	66442	11429	36293	12635

	Replications									
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
|Ω|=10	4649	624	4626	863	442	1787	3368	3356	596	809
|Ω|=20	17898	2702	3869	5851	1108	4971	3584	6584	2800	3585
|Ω|=30	8428	4729	3965	16560	5511	4198	1882	6440	11245	5450
|Ω|=40	10855	39650	6318	5783	7328	13964	10423	21637	26284	23190
|Ω|=50	22527	11792	20556	14058	58491	26877	22016	12364	32144	20766
	Replications									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
|Ω|=10	3052	1298	2061	2539	628	2983	2237	1039	1085	392
|Ω|=20	5331	2145	1333	1458	11066	2190	2290	19824	4700	2805
|Ω|=30	11954	6968	38260	2381	3181	8451	41192	4382	7805	26893
|Ω|=40	13037	5486	5435	17776	24768	20164	8669	5103	5618	14405
|Ω|=50	24123	19211	19151	12505	19990	19916	21670	34340	28164	55150

Table C.3 CPU time (seconds) to solve the integrated model by ILS II

	Replications									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
|Ω|=10	1340	491	1922	425	477	755	1348	506	678	538
|Ω|=20	1167	3503	5971	3221	5271	1697	20627	2685	1930	2679
|Ω|=30	10126	9967	30661	4172	8808	6727	6427	1610	10963	6005
|Ω|=40	7504	21094	9530	15928	16359	9467	10908	84211	43149	32559
|Ω|=50	31153	193291	96686	35329	82864	137531	33660	28716	22052	51655
	Replications									
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
|Ω|=10	2201	535	448	523	426	480	2015	525	436	662
|Ω|=20	7274	5422	2276	6391	1138	2929	6315	2132	6266	1489
|Ω|=30	12292	10282	4482	68083	5329	4161	2587	5164	5632	10275
|Ω|=40	7427	22670	9300	17124	13053	15887	4309	24055	32174	36860
|Ω|=50	95805	65882	52794	84463	148445	32449	22922	27456	147596	68924
	Replications									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
|Ω|=10	615	681	540	601	594	507	554	616	634	291

|Ω|=20	1831	2894	3289	1770	1390	2164	1221	19734	1978	2609
|Ω|=30	8427	16278	32161	1890	8211	4594	21019	11265	7410	92408
|Ω|=40	25583	20081	7219	24282	9476	40298	16896	4052	19562	24982
|Ω|=50	48015	80134	32494	82357	45791	114045	91787	28281	50597	116414

Table C.4 CPU time (seconds) to solve the integrated model by ILS III

	Replications									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
|Ω|=10	1777	1443	2463	726	755	1774	1561	641	1880	957
|Ω|=20	2301	3329	3385	2812	2301	4493	8275	4062	2139	2415
|Ω|=30	5874	8175	4879	2799	4476	4342	5807	1978	4124	5006
|Ω|=40	7812	7468	5183	7331	6161	4861	6880	13429	9732	12983
|Ω|=50	9919	16887	13351	8595	14202	11092	6825	8283	7728	12949
	Replications									
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
|Ω|=10	2081	878	640	1799	1126	732	1209	1210	506	1535
|Ω|=20	5057	2957	2054	4825	2315	3207	3733	2969	2813	3038
|Ω|=30	5895	5266	4339	21233	4891	4417	3236	3992	4490	6851
|Ω|=40	4615	7297	6746	7686	5726	9593	4646	7932	6351	11237
|Ω|=50	14541	13515	11619	12575	16805	8121	8796	10180	10171	14489
	Replications									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
|Ω|=10	796	1195	1044	741	715	627	1021	1819	976	765
|Ω|=20	3027	2392	3905	2101	1732	2393	1048	3518	2010	3070
|Ω|=30	5402	4869	13157	3215	5520	4798	7750	5622	5204	14652
|Ω|=40	7998	8675	4935	7211	8007	7387	7125	5201	7737	7806
|Ω|=50	7996	15034	8785	18092	11013	14326	13317	8205	15633	14241

Table C.5 Relative gap corresponding to the upper bound obtained from the heuristic

	Replication									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
|Ω|=10	0.6%	0.0%	0.0%	0.3%	0.0%	0.3%	0.0%	0.0%	0.0%	0.2%
|Ω|=20	0.0%	0.0%	0.3%	0.3%	0.6%	0.0%	1.6%	0.3%	0.4%	0.1%

|Ω|=30	0.2%	0.5%	0.2%	0.2%	0.3%	0.2%	0.2%	0.2%	0.4%	0.3%
|Ω|=40	0.0%	0.0%	0.6%	0.9%	1.8%	0.5%	0.0%	1.3%	0.0%	0.0%
|Ω|=50	0.4%	0.9%	0.9%	0.1%	0.2%	0.7%	2.0%	1.6%	0.1%	0.4%
	Replication									
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
|Ω|=10	0.2%	0.0%	0.4%	0.0%	0.0%	0.0%	0.0%	0.5%	0.0%	0.0%
|Ω|=20	0.5%	0.9%	0.7%	0.3%	0.2%	0.4%	0.4%	0.2%	0.3%	0.5%
|Ω|=30	0.0%	0.3%	0.4%	0.6%	0.0%	0.0%	0.0%	0.2%	2.0%	0.2%
|Ω|=40	0.0%	0.3%	0.0%	0.1%	0.1%	0.1%	0.0%	0.2%	1.0%	0.1%
|Ω|=50	0.4%	0.1%	0.4%	0.3%	0.8%	1.1%	0.1%	0.1%	0.5%	0.7%
	Replication									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
|Ω|=10	0.0%	0.0%	0.0%	0.2%	0.0%	0.8%	1.8%	0.0%	0.8%	0.0%
|Ω|=20	0.0%	1.5%	0.0%	0.7%	0.0%	0.2%	0.5%	0.0%	0.2%	1.0%
|Ω|=30	0.1%	0.0%	1.7%	0.5%	0.0%	0.3%	0.8%	0.0%	0.1%	0.5%
|Ω|=40	0.4%	0.0%	0.5%	0.5%	0.9%	0.0%	0.5%	0.2%	0.0%	0.7%
|Ω|=50	0.0%	0.2%	0.5%	0.6%	0.3%	0.5%	0.1%	0.4%	0.3%	1.6%