Transportation Science

Shelter Location and Evacuation Route Assignment Under Uncertainty: A Benders Decomposition Approach

Vedat Bayram, Hande Yaman

Please scroll down for article—it is on subsequent pages

Shelter Location and Evacuation Route Assignment Under Uncertainty: A Benders Decomposition Approach

Vedat Bayram,[a] Hande Yaman[b]

[a] Department of Industrial Engineering, TED University, 06420 Kolej Çankaya, Ankara, Turkey; [b] Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey
Contact: vedat.bayram@tedu.edu.tr (VB); hyaman@bilkent.edu.tr, http://orcid.org/0000-0002-3392-1127 (HY)

Abstract. Shelters are safe facilities that protect a population from possible damaging effects of a disaster. For that reason, shelter location and traffic assignment decisions should be considered simultaneously for an efficient evacuation plan. In addition, as it is very difficult to anticipate the exact place, time, and scale of a disaster, one needs to take into account the uncertainty in evacuation demand, the disruption/degradation of evacuation road network structure, and the disruption in shelters. In this study, we propose an exact algorithm based on Benders decomposition to solve a scenario-based two-stage stochastic evacuation planning model that optimally locates shelters and that assigns evacuees to shelters and routes in an efficient and fair way to minimize the expected total evacuation time. The second stage of the model is a second-order cone programming problem, and we use duality results for second-order cone programming in a Benders decomposition setting. We solve practical-size problems with up to 1,000 scenarios in moderate CPU times. We investigate methods such as employing a multicut strategy, deriving Pareto-optimal cuts, and using a preemptive priority multiobjective program to enhance the proposed algorithm. We also use a cutting plane algorithm to solve the dual subproblem since it contains a constraint for each possible path. Computational results confirm the efficiency of our algorithms.

1. Introduction

Whether it is natural or man-made, disasters such as hurricanes, earthquakes, terrorist attacks, floods, tsunamis, and nuclear accidents continue to have devastating socioeconomic impacts and affect millions of people, claiming the lives of thousands and causing massive damage in infrastructure. Various traffic management problems arise during disasters; evacuation of the disaster region being one of the most important. In a report by the U.S. Federal Emergency Management Agency (FEMA), it is stated that 45–75 disasters require an evacuation annually (Transportation Research Board 2008). Hurricane Floyd in 1999 (CNN 2001) and Hurricanes Katrina and Rita in 2005 (Transportation Research Board 2008) required millions of people to evacuate, creating the largest traffic jams in U.S. history. If the evacuation is not planned and managed effectively, the surge in evacuation traffic demand can cause congestion and may leave the evacuees in harm's way, possibly resulting in further losses. Successful evacuation management does not just save lives, it contributes to the community's regaining functionality in a fast and smooth way (Perry 1979).

To protect an endangered population from the harming impact of a disaster, and to provide the evacuees with food, medical care, and accommodation, shelters are generally chosen among the existing facilities with possible modifications to meet safety standards. The evacuation planning needs to decide whether to construct additional permanent shelters that meet FEMA standards, and the locations of those shelters (Li, Jin, and Zhang 2011). The Federal Emergency Management Agency (1988, 2006, 2008) and the American Red Cross (2002) provide the basis for the selection, design, and construction of shelters against different types of disasters. In the report by the American Red Cross (2002), it is stated that existing unacceptable facilities can be enhanced to be used as hurricane evacuation shelters and minor modifications of municipal, community, or school buildings are suggested, such as the addition of hurricane shutters, while buildings are being planned. The Federal Emergency Management Agency (2006) classifies shelters as standalone shelters and internal shelters, and defines a standalone shelter as a separate building that is designed and constructed

to withstand the range of natural and man-made hazards, in neighborhoods where existing homes lack shelters.

The preparations to construct a new shelter or to retrofit an existing facility for that purpose require time and needs to be done before the disaster takes place. For that reason, the decision of where to locate the shelters is often made before a disaster occurs. Furthermore, the uncertainty about demand and the infrastructure is not resolved even after the disaster hits. Hence, one has to decide on which locations to use as shelters, thinking of different scenarios for the problem parameters. Owing to the fact that in an evacuation, the population at risk will try to reach one of those safe shelters as quickly as possible, the decision of where to locate the shelters should be considered simultaneously with evacuation traffic management decisions, as it affects the time to evacuate a disaster region drastically (Bayram, Tansel, and Yaman 2015). Addressing these two problems separately may lead to suboptimal results.

Generally, evacuation planning is done with inexact or incomplete information, as it is not easy to predict when and where disasters will occur and with how much impact. Because of the unpredictability of human behavior during disasters, specifically whether they will obey the evacuation order or not, it is not easy to estimate the evacuation demand. In 2005, the evacuation of the endangered population during Hurricane Rita caused 100-mile-long congestion on the highway, with further fatalities not caused by the hurricane itself (O'Driscoll, Wolf, and Hampson 2005). The impact of the disaster may cause road segments to be inundated in a flood or blocked by debris in a hurricane or after an earthquake, resulting in the loss of capacity or total disruption in some parts of the evacuation roadway infrastructure. Likewise, the predetermined shelter sites can be affected.

Bayram and Yaman (2015) proposed a scenario-based two-stage stochastic evacuation planning model that optimally locates shelters and assigns evacuees to the nearest shelters and the shortest paths to those shelters within a given degree of tolerance to minimize the expected total evacuation time. The nonlinear mixed-integer programming model they propose considers the uncertainty about future realizations of the evacuation demand, disruption in the road network and degraded road capacities, and disruption of the shelters. They show the significance of including uncertainty in planning for evacuations.

Our aim is to propose an exact algorithm based on Benders decomposition (BD) (Benders 1962) to solve the formulation proposed by Bayram and Yaman (2015) with a large number of scenarios. The second stage of the model is a second-order cone programming (SOCP) problem since the nonlinear objective

function is represented with SOCP. We propose an algorithm that uses duality results for SOCP to derive optimality cuts in a BD setting. We solve practical-size problems with up to 1,000 scenarios in moderate CPU times. We investigate methods such as adopting a multicut strategy, deriving Pareto-optimal cuts, using a reduced primal subproblem and preemptive priority multiobjective program to enhance the proposed algorithm. The dual subproblem contains a constraint for each path, and hence its size gets larger as the tolerance level increases. To deal with this issue, we propose a cutting plane framework to solve the dual subproblem. Computational results confirm the efficiency of our algorithm as it is considerably faster and can solve instances with a larger number of scenarios compared to solving the extended formulation (EF) with an off-the-shelf solver. Furthermore, employing a cutting plane framework enables us to solve instances with larger networks and higher tolerance levels.

The rest of the paper is organized as follows. In Section 2, we present a literature review on evacuation planning focusing on stochastic location-allocation problems and their solution methodologies. In Section 3, we describe the problem and present a two-stage stochastic mixed-integer nonlinear programming formulation (MINLP). We propose a BD approach in Section 4 and explore ways to improve it in Section 5. In Section 6, we introduce a cutting plane framework for solving the subproblem. We present the computational results of our study in Section 7 and conclude in Section 8.

2. Literature Review
2.1. Traffic Assignment Models
Evacuation planning/management models are mostly based on existing traffic assignment models. The common models are the stochastic user equilibrium (SUE), user equilibrium (UE), system optimal (SO), and nearest allocation (NA) models. The SUE approach assumes that the evacuees choose the shortest travel time path depending on their perception of the travel time. The UE approach is a special case of the SUE approach when the variance of travel time perception by the evacuees is zero. In the UE approach, it is assumed that the evacuees have perfect information about the travel times on every possible route and that they can find the optimal routes (Sheffi 1985). For both SUE and UE, the goal of the evacuees is to minimize their individual travel times, and for that reason, SUE and UE do not necessarily minimize the total evacuation time in the evacuation network. Furthermore, the perfect information assumption may not be valid in case of a disaster (Galindo and Batta 2013), since the surge in traffic demand is unusual and it is difficult for evacuees to guess the traffic congestion on a route, let alone choose the minimum-latency

routes using their past experiences (Pel, Bliemer, and Hoogendoorn 2012).

The main goal of the evacuation management authority, on the other hand, is to evacuate everyone to safety as soon as possible by minimizing the total evacuation time. This traffic assignment plan is called SO. The SO solution may allocate some evacuees to shelters much further away than they would normally choose to go, and to routes much longer than they would normally take, for the benefit of the other evacuees. Bayram, Tansel, and Yaman (2015) show in their study that such a traffic assignment may assign some evacuees to routes/shelters tens of times longer and farther away than the shortest/nearest one they would take. In a disaster, it may not be reasonable to ask evacuees to accept such distant shelters and such long routes, since they will be trying to flee from the danger zone and reach safety at a shelter as quickly as possible.

In a disaster, the information on path lengths or free-flow travel times is more accessible to evacuees compared to actual travel times. This idea is the motivation for the NA model—i.e., in the NA model, each evacuee uses a shortest path based on length (geographical distance) or free-flow travel time to reach the nearest shelter. Although this approach may be a reasonable one during an evacuation, such a traffic assignment may lead to poor system performance.

The notion of constrained system optimal (CSO) traffic assignment is first defined by Jahn et al. (2005) for a route guidance system. This model takes into consideration the individual needs through additional constraints to make sure that drivers are assigned to "acceptable" paths only. Bayram, Tansel, and Yaman (2015) propose a CSO model that considers the shelter location decisions simultaneously with efficient but fair evacuation traffic assignment decisions by assigning evacuees to the nearest shelters and to the shortest paths to those shelters—shortest and nearest within a given degree of tolerance—to minimize the total evacuation time. They show that the location and the number of shelters opened drastically affect the evacuation plan and that the SO solution may be unacceptably unfair to evacuees, whereas the NA solution may result in substantial deterioration in system performance. Their results show that the CSO model is a good compromise between efficiency and fairness for a suitable tolerance factor.

2.2. Evacuation Planning Models

There is a vast amount of literature that proposes new ideas, models, or solution methodologies to support evacuation planning/management decisions. However, despite the fact that evacuation planning is typically characterized by great uncertainties, the studies in the literature mostly rely on deterministic models that adopt a single hazard scenario such as worst case or most probable scenario. Some of these models do not consider the optimal selection of shelter sites (Yamada 1996; Cova and Johnson 2003; Kalafatas and Peeta 2009; Xie, Lin, and Waller 2010; Tüydeş 2005; Chiu et al. 2007; Stepanov and Smith 2009; Ng, Park, and Waller 2010; So and Daganzo 2010; Hamacher and Tjandra 2002; Bretschneider 2013; Bish, Sherali, and Hobeika 2013). The deterministic evacuation studies that consider the optimal selection of shelter sites and that also take into account the congestion effect are either single level or bilevel models. Single-level models (Sherali, Carter, and Hobeika 1991) decide on the shelter locations and the shelter/traffic assignments in a SO manner. The bilevel models (Kongsomsaksakul, Yang, and Chen 2005) bring together the two conflicting SO and UE ideas. They specify the locations of shelter sites in a SO manner at the upper level and assign evacuees to shelters and routes in a UE manner at the lower level.

The evacuation studies in the literature that take into consideration uncertainty mostly focus on demand uncertainty (Yao, Mandala, and Chung 2009; Huibregtse, Hoogendoorn, and Bliemer 2010; Ng and Waller 2010; Yazıcı and Özbay 2010; Kulshrestha et al. 2011) and/or capacity uncertainty (Shen et al. 2008; Ng and Waller 2010; Yazıcı and Özbay 2010).

Shen et al. (2008) develop two different scenario-based, stochastic, bilevel models that minimize the maximum UE travel time among all node shelter pairs by locating shelters at the upper level and assigning evacuees to shelters and routes in a UE manner at the lower level. The first model they propose decides on the locations of shelters and considers the distance between the demand nodes and the shelter sites as well as the demand as uncertain parameters. To solve this first model, they present a genetic algorithm–based approach. The second model is proposed to model real-time decision making during evacuations, and a simulation-based approach that uses a successive shortest path algorithm is developed to solve it. Yao, Mandala, and Chung (2009) consider the demand uncertainty in their study. They propose a cell transmission model (CTM)–based, robust linear programming (LP) model that considers the susceptibility of an area to a disaster at a particular time, minimizing the evacuees' threat exposure during evacuation. They solve their model using an off-the-shelf solver. Huibregtse, Hoogendoorn, and Bliemer (2010) propose a model that considers uncertainty in demand, the behavior of people, and the hazard in a scenario-based setting. They use an off-the-shelf solver to solve their problem. Ng and Waller (2010) consider demand and capacity uncertainty together in a scenario-based evacuation planning model. They provide a framework that

determines the amount of demand inflation and supply deflation necessary to ensure a user-specified reliability level. They solve their problem using an off-the-shelf solver for a range of nine scenarios. Yazıcı and Özbay (2010) take into consideration the uncertainty in demand and capacity simultaneously. They propose a CTM-based SO dynamic traffic assignment formulation with probabilistic constraints. They use a P-level efficient points method by Prékopa (1995) to write the deterministic equivalent formulation of the problem and solve it with an off-the-shelf solver for three scenarios. Kulshrestha et al. (2011) develop a bilevel model that minimizes the total cost to establish and operate shelters and evacuates everyone to safety in a UE manner. They focus on demand uncertainty and confine the uncertain demand to an uncertainty set. Their model is formulated as a mathematical program with complementarity constraints and is solved by a cutting plane algorithm, for a total of three (nominal, low, and high) demand scenarios. Li et al. (2012) propose a scenario-based model that chooses optimal locations of shelters, which are robust for a range of hurricane events, by considering disruption in shelter sites. At the upper level of their model, the central authority selects the shelter sites for a particular scenario. The objective of the upper-level problem is to minimize the weighted sum of the expected unmet shelter demand and the expected total network travel time. In the lower level, evacuees choose their routes in a dynamic UE manner. They develop heuristic algorithms based on Lagrangian relaxation and present a case study for the state of North Carolina for 33 hurricane scenarios.

To our knowledge, there is one study that simultaneously considers the uncertainty in disruption/degradation in road network structure, evacuation demand, and disruption in shelters. Bayram and Yaman (2015) introduce a novel scenario-based model that decides simultaneously on the locations of shelters and the allocations of evacuees to shelters and routes under uncertainty. Their model incorporates evacuees' preferences and fairness considerations by routing the evacuees on paths that are not much longer than the shortest paths to the nearest shelters. They solve practical-size problems exactly by using a SOCP approach for a range of up to 50 scenarios by using an off-the-shelf solver.

2.3. Our Contribution
We base our work on the stochastic CSO (SCSO) problem by Bayram and Yaman (2015). To be able to model a stochastic evacuation planning problem more realistically, one needs to consider a large number of scenarios. As the number of scenarios grows, the EF developed by Bayram and Yaman (2015) may not be solved within reasonable CPU times or may not be solved at all. The evacuation models in the literature that

take into account congestion by employing a nonlinear objective function are generally solved by heuristic methodologies, especially for large evacuation road networks. Against this backdrop, we propose an exact algorithm based on Benders decomposition. We test various ways of enhancing the algorithm and report the results of our computational experiments. Furthermore, we propose a cutting plane algorithm to solve the dual subproblem for instances with larger networks and tolerance levels. The results show that the algorithm can solve real problems with up to 1,000 scenarios and is faster compared to solving the EF with an off-the-shelf solver.

3. Problem Description and Model Formulation

The SCSO problem proposed by Bayram and Yaman (2015) decides on where to locate p shelters and how to assign evacuees at origins in risk zones to their destinations (shelters), including the decision of how to assign evacuees to routes so as to minimize total evacuation time. The problem is defined on a directed road network $G = (N, A)$, where N is the set of nodes and A is the set of arcs (road segments) in the network. We define $O \subset N$ as the set of origin (demand) nodes from where the population at risk are to be evacuated and $F \subset N$ as the set of destination nodes (potential shelter sites) where evacuees reach to safety, O and F being disjoint.

Since we are proposing a strategic evacuation planning model, and since no-notice evacuations are adequately represented by static models as evacuees are loaded into an evacuation network at once (Özbay et al. 2012), we use a static modeling approach. The travel time spent on a given road segment is a function of traffic flow and increases monotonically, since an increase in traffic volume decreases the travel speed because of congestion and hence increases the travel time along the road segment. To express the relationship between travel time and the volume of traffic on a road segment, we employ the Bureau of Public Roads (BPR) function, which is convex, nondecreasing, and differentiable, as in Sherali, Carter, and Hobeika (1991); Kongsomsaksakul, Yang, and Chen (2005); Ng and Waller (2010); and Li et al. (2012). The BPR function is formulated as

$$t(x) = t^0 \left(1 + \alpha \left(\frac{x}{c}\right)^{\beta}\right),$$

where $t(x)$ is the travel time at which assigned volume x can travel on the road segment, c is the practical capacity (maximum flow rate), and t^0 is the free-flow travel time at zero volume. The parameters $\alpha \geq 0$ and $\beta \geq 0$ are the parameters that reflect the road characteristics and they are taken as 0.15 and 4, respectively,

by the U.S. Department of Commerce BPR (Bureau of Public Roads 1964).

In a two-stage stochastic setting, the first stage of our evacuation problem is about where to locate the shelters, before a disaster takes place. The binary variable y_s is 1 if a shelter is located/opened at node $s \in F$, and 0 otherwise. Given the shelter location decisions from the first stage and the realization of the evacuation demand and the impact of the disaster on the road network and the shelters, the second stage assigns evacuees to shelters and to routes. We define Ω as the set of scenarios in a disaster and associate a probability $p(\omega)$ to each scenario $\omega \in \Omega$. We define $F(\omega) \subseteq F$ as the set of potential shelters that are usable (not disrupted) in scenario $\omega \in \Omega$, $c_a(\omega)$ as the (possibly degraded) capacity of road segment a in scenario $\omega \in \Omega$ with $0 \le c_a(\omega) \le c_a$, for all $a \in A$, and $A(\omega) \in A$ as the set of usable arcs—i.e., $c_a(\omega) > 0$, $a \in A(\omega)$. A shelter s is reachable from demand point r in scenario ω if this shelter is usable and if there exists a route with usable arcs from r to s in this scenario. We define $\bar{F}_r(\omega)$ as the set of reachable shelters for demand point $r \in O$ in scenario $\omega \in \Omega$. For feasibility, we assume that there exists at least one reachable shelter for every origin $r \in O$ in every scenario ω. The amount of demand at origin $r \in O$, $w_r(\omega)$, is the number of passenger vehicles that will be evacuated in scenario $\omega \in \Omega$.

Let $P_{rs}(\omega)$ be the set of alternative paths from demand point r to shelter s in scenario $\omega \in \Omega$. The tolerance of an evacuee to the length of a path he or she is offered to take is denoted by λ. Based on this tolerance level, the evacuation planning authority does not assign an evacuee to a path whose length is more than $(1 + \lambda)$ times the length of a shortest path to the closest open and usable shelter in a given scenario. This implies that some evacuees may be assigned to an open shelter within this degree of tolerance, although it might not be the nearest one. We define $P_{rs}^\lambda(\omega) = \{\pi \in P_{rs}(\omega): d^\pi \le (1 + \lambda) d_{rs}^*(\omega)\}$ as the set of acceptable and usable paths from origin r to destination s of tolerance level λ in a given scenario $\omega \in \Omega$. In this definition, d^π is the length of path π, and $d_{rs}^*(\omega)$ is the length of a shortest path from r to s in scenario $\omega \in \Omega$. This set is computed using an algorithm developed by Byers and Waterman (1984). We use geographical distances to compute the set $P_{rs}^\lambda(\omega)$. Free-flow travel times $t_a^0(\omega)$ could also be used.

For a given origin r and a scenario $\omega \in \Omega$, the set of acceptable paths is defined based on the length of a shortest path from node r to the closest open and usable shelter. As we do not know a priori which shelters will be opened, we do not know the actual set of acceptable paths; yet we know that this set is a subset of the union of $P_{rs}^\lambda(\omega)$ over all potential shelters s.

In our model, we also use the following decision variables: $v_\pi(\omega)$ is the fraction of origin r's demand

that uses path $\pi \in P_{rs}^\lambda(\omega)$ from origin $r \in O$ to shelter $s \in F(\omega)$ in scenario $\omega \in \Omega$ and $x_a(\omega)$ is the amount of traffic on arc $a \in A(\omega)$ in scenario $\omega \in \Omega$.

The EF developed by Bayram and Yaman (2015) for the SCSO problem is as follows:

$$\min \sum_{\omega \in \Omega} p(\omega) \sum_{a \in A(\omega)} t_a^0 \left(1 + \alpha \left(\frac{x_a(\omega)}{c_a(\omega)}\right)^\beta\right) x_a(\omega) \tag{1}$$

$$\text{s.t.} \sum_{s \in F} y_s = p, \tag{2}$$

$$\sum_{s \in F(\omega)} \sum_{\pi \in P_{rs}^\lambda(\omega)} v_\pi(\omega) = 1, \quad \forall r \in O, \omega \in \Omega, \tag{3}$$

$$\sum_{s \in \bar{F}_r(\omega)} y_s \ge 1, \quad \forall r \in O, \omega \in \Omega, \tag{4}$$

$$\sum_{\pi \in P_{rs}^\lambda(\omega)} v_\pi(\omega) \le y_s, \quad \forall r \in O, \omega \in \Omega, s \in F(\omega), \tag{5}$$

$$\sum_{s \in F(\omega)} \sum_{\pi \in P_{rs}^\lambda(\omega): d^\pi > (1+\lambda) d_{ri}^*(\omega)} v_\pi(\omega) + y_i \le 1,$$
$$\forall r \in O, \omega \in \Omega, i \in F(\omega), \tag{6}$$

$$x_a(\omega) = \sum_{r \in O} \sum_{s \in F(\omega)} \sum_{\pi \in P_{rs}^\lambda(\omega): a \in \pi} w_r(\omega) v_\pi(\omega),$$
$$\forall \omega \in \Omega, a \in A(\omega), \tag{7}$$

$$v_\pi(\omega) \ge 0, \quad \forall \omega \in \Omega, \pi \in \bigcup_{r \in O, s \in F(\omega)} P_{rs}^\lambda(\omega), \tag{8}$$

$$y_s \in \{0, 1\} \quad \forall s \in F. \tag{9}$$

Objective function (1) minimizes the expected total evacuation time spent by the evacuees in the network. There may be reasons such as the available budget and personnel that restrict the number of shelters that can be opened. Constraint (2) limits the number of shelters open to this prespecified number p. This constraint can be replaced with a budget constraint if the data on the associated costs are available. Constraints (3) ensure that evacuation demand from every origin r is assigned to a shelter and a route that leads to that shelter, for every scenario. We added constraints (4) to ensure that there is at least one open and reachable shelter for each demand point and each scenario. Constraints (5) prevent assigning demand to a non-open shelter. Constraints (6) ensure that if the shelter at site i is open and usable in scenario ω, then the demand at origin node r cannot be routed on any path whose length is longer than $(1 + \lambda) d_{ri}^*(\omega)$. Finally, the set of constraints (7) computes the traffic on every arc in each scenario, and constraints (8) and (9) are variable restrictions.

Bayram and Yaman (2015) point out that the SCSO problem is NP-hard even when $\alpha = 0$ and G is bipartite. They also note that SCSO generalizes the classical p-median facility location problem. SCSO also generalizes the SO and NA traffic assignment approaches. When $\lambda = 0$, we have the NA model, and when $\lambda = \infty$, we obtain a model for the SO traffic assignment.

Finally, SCSO generalizes the congested facility location (Desrochers, Marcotte, and Stan 1995; Fischetti, Ljubić, and Sinnl 2016) where the congestion costs at facilities can be modeled by splitting facility nodes into arcs with convex congestion costs.

4. Benders Decomposition Approach

To be able to model a stochastic evacuation problem more realistically, one needs to consider a large number of scenarios. As the number of scenarios increases, the EF developed by Bayram and Yaman (2015) may not be solved within reasonable CPU times or may not be solved at all. For that reason, we develop BD (Benders 1962) and generalized BD (Geoffrion 1972) based approaches to solve our problem considering a large number of scenarios and explore methodologies to accelerate our BD algorithm. This algorithm can also be considered as an "L"-shaped algorithm (LSA) developed by Van Slyke and Wets (1969) to solve stochastic programs with recourse. Our problem is well fit for the BD approach and LSA as these algorithms have been successfully implemented for location (Laporte, Louveaux, and van Hamme 1994; Contreras, Cordeau, and Laporte 2011; Noyan, Balcik, and Atakan 2016; Martins de Sá et al. 2015; Álvarez-Miranda, Fernández, and Ljubić 2015), routing (Cordeau, Soumis, and Desrosiers 2000; Cordeau et al. 2001; Laporte, Louveaux, and van Hamme 2002; Sherali and Zhu 2008; Üster and Kewcharoenwong 2011; Sherali, Bae, and Haouari 2013), location-routing (Franca and Luna 1982), large-scale (Noonan and Giglio 1977; Cordeau et al. 2001; Sherali and Zhu 2008; Contreras, Cordeau, and Laporte 2011; Sherali, Bae, and Haouari 2013; Martins de Sá et al. 2015), nonlinear (Noonan and Giglio 1977), and stochastic optimization (Franca and Luna 1982; Laporte, Louveaux, and van Hamme 1994; 2002; Sherali and Zhu 2008; Noyan, Balcik, and Atakan 2016) problems.

In the BD approach, the main idea is to project out the second-stage variables. The resulting problem is called the master problem (MP), and it contains fewer variables but a large number of constraints. These constraints are known as Benders cuts (BC), and most of them are not active at an optimal solution. Because of this fact, the most natural strategy to solve the MP is through relaxation (Geoffrion 1972). An iterative solution methodology is pursued, by solving the relaxed MP at every iteration and passing the optimal solution to subproblems that are basically duals of the original problem with shelter location decisions temporarily fixed to the values obtained from the MP and adding the violated BCs to MP until all of them are satisfied at a relaxed MP solution. Since the MP is a relaxation, its optimal value provides a lower bound on the optimal value, and an upper bound is given by the expectation of the optimal values of the subproblems.

In our model, the shelter location decisions y are taken in the presence of uncertainty. For that reason, they are called first-stage (design) variables. In the second stage, the uncertainty is revealed and recourse decisions—i.e., shelter and route assignment decisions v—are taken. However, while making decisions about where to locate the shelters in the first stage, their effect on the second-stage assignment decisions and total evacuation time is taken into account. The future effects of shelter location decisions are measured by taking the expectation of the recourse function on possible scenarios. Our model has a 0–1 first-stage problem and a nonlinear second-stage problem.

The first-stage variables in our problem—i.e., shelter location decisions—are the complicating variables, and their number is less than that of the second-stage variables. We project out the second-stage variables v and x by fixing the first-stage variables y to a value given by the MP. This results in the following primal subproblems, one for each scenario $\omega \in \Omega$.

Primal Subproblem $(PSP(\bar{y}, \omega))$

$$\min \quad \sum_{a \in A(\omega)} t_a^0 \left(1 + \alpha \left(\frac{x_a(\omega)}{c_a(\omega)}\right)^\beta \right) x_a(\omega) \tag{10}$$

$$\text{s.t.} \quad \sum_{s \in F(\omega)} \sum_{\pi \in P_{rs}^\lambda(\omega)} v_\pi(\omega) = 1, \quad \forall r \in O, \tag{11}$$

$$\sum_{\pi \in P_{rs}^\lambda(\omega)} v_\pi(\omega) \le \bar{y}_s, \quad \forall r \in O, s \in F(\omega), \tag{12}$$

$$\sum_{s \in F(\omega)} \sum_{\pi \in P_{rs}^\lambda(\omega): d^\pi > (1+\lambda) d_{ri}^*(\omega)} v_\pi(\omega) \le 1 - \bar{y}_i,$$
$$\forall r \in O, i \in F(\omega), \tag{13}$$

$$x_a(\omega) = \sum_{r \in O} \sum_{s \in F(\omega)} \sum_{\pi \in P_{rs}^\lambda(\omega): a \in \pi} w_r(\omega) v_\pi(\omega),$$
$$\forall a \in A(\omega), \tag{14}$$

$$v_\pi(\omega) \ge 0, \quad \forall \pi \in \bigcup_{r \in O, s \in F(\omega)} P_{rs}^\lambda(\omega), \tag{15}$$

where $\bar{y} \in V \subseteq Y$ is a fixed vector for the complicating variables, $V \subseteq Y$ is the set of vectors y that renders $PSP(\bar{y}, \omega)$ feasible, and Y is the set of all possible shelter location decisions y. Given the locations of the shelters, each subproblem is a nonlinear CSO shelter and traffic assignment problem.

With the theoretical findings in the last two decades and the applicability of efficient interior point algorithms (Nesterov, Nemirovskii, and Ye 1994), SOCP has become a state-of-the-art technique in mathematical programming. We refer the reader to Ben-Tal and Nemirovski (2001) and Alizadeh and Goldfarb (2003) for an introduction to SOCP. Ben-Tal and Nemirovski (2001) state that as a well-structured convex optimization problem, solving an SOCP is no more difficult than solving an LP of a similar size. Many convex optimization problems can be modeled as SOCPs (see, e.g., Lobo et al. 1998; Ben-Tal and Nemirovski 2001).

For successful results on solving second-order mixed-integer models, see, e.g., Bonami and Lejeune (2009) on the portfolio optimization problem; Aktürk, Atamtürk, and Gürel (2009) on the machine-job assignment problem; Taylor and Hover (2012) on the power distribution system reconfiguration problem; Atamtürk, Berenguer, and Shen (2012) on the stochastic joint location inventory problem; Hijazi, Bonami, and Ouorou (2013) on the routing problem in telecommunication networks; See and Sim (2010), Natarajan, Sim, and Uichanco (2010), Ang, Lim, and Sim (2012), and Mak, Rong, and Shen (2013) on robust optimization problems; and Pınar (2013) on hedging of American contingent claims. Saito and Murota (2007) consider MIP problems with ellipsoidal uncertainty in problem data. They formulate the robust counterpart as a SOCP problem with integer constraints and propose an adaptation of the BD technique using the duality of linear programming over symmetric cones to generate feasibility cuts.

Motivated by these studies, we reformulate the $PSP(\bar{y}, \omega)$ as a SOCP model as in Bayram and Yaman (2015). By employing SOCP, the nonlinearity is transferred to the constraint set in the form of second-order quadratic constraints. This is done as follows. We first define auxiliary variables $\mu_a(\omega)$ for each $a \in A(\omega)$, $\omega \in \Omega$ and move the nonlinearity from the objective function to the constraints—i.e., the objective function of the $PSP(\bar{y}, \omega)$ becomes

$$\sum_{a \in A(\omega)} \left(t_a^0 x_a(\omega) + \frac{t_a^0 \alpha}{c_a(\omega)^\beta} \mu_a(\omega) \right),$$

and we add the constraints $x_a(\omega)^{\beta+1} \le \mu_a(\omega)$ for all $a \in A$, $\omega \in \Omega$.

Consider an inequality of the form

$$r^{2^l} \le s_1 s_2, \dots, s_{2^l} \quad for \quad r, s_1, s_2, \dots, s_{2^l} \ge 0. \quad (16)$$

An equivalent representation of the inequality (16) can be achieved by using $O(2^l)$ variables and $O(2^l)$ inequalities of the form

$$u^2 \le v_1 v_2, u, v_1, v_2 \ge 0, \quad (17)$$

which are referred to as hyperbolic inequalities since they describe half a hyperboloid (Lobo et al. 1998; Ben-Tal and Nemirovski 2001; Alizadeh and Goldfarb 2003; Günlük and Linderoth 2008). Then, each hyperbolic inequality can easily be transformed into a second-order cone inequality

$$\|2u, v_1 - v_2\| \le v_1 + v_2. \quad (18)$$

We take $\beta = 4$ and represent $x_a(\omega)^5 \le \mu_a(\omega)$ with hyperbolic inequalities of the form

$$x_a(\omega)^2 \le \theta_a(\omega) h_a(\omega),$$
$$\theta_a(\omega)^2 \le u_a(\omega) x_a(\omega),$$
$$u_a(\omega)^2 \le \mu_a(\omega) x_a(\omega),$$
$$h_a(\omega) = 1, \theta_a(\omega), u_a(\omega), x_a(\omega), \mu_a(\omega) \ge 0,$$

where $h_a(\omega), \theta_a(\omega)$, and $u_a(\omega)$ are auxiliary variables that are used to define hyperbolic inequalities.

These hyperbolic inequalities are represented by their respective quadratic cone constraints

$$\|2x_a(\omega), \theta_a(\omega) - h_a(\omega)\| \le \theta_a(\omega) + h_a(\omega), \quad (19)$$
$$\|2\theta_a(\omega), u_a(\omega) - x_a(\omega)\| \le u_a(\omega) + x_a(\omega), \quad (20)$$
$$\|2u_a(\omega), \mu_a(\omega) - x_a(\omega)\| \le \mu_{a(\omega)} + x_a(\omega), \quad (21)$$
$$h_a(\omega) = 1, \theta_a(\omega), u_a(\omega), x_a(\omega), \mu_a(\omega) \ge 0. \quad (22)$$

Hence, the subproblem is still a nonlinear programming problem, but it can be solved efficiently once represented as SOCP. Note that SOCP problems are solved with the barrier algorithm in CPLEX (IBM ILOG CPLEX 2011).

The resulting conic primal subproblem with SOCP constraints (CPSP) is given below.

Conic Primal Subproblem $(CPSP(\bar{y}, \omega))$

$$\min \sum_{a \in A(\omega)} \left(t_a^0 x_a(\omega) + \frac{t_a^0 \alpha}{c_a(\omega)^\beta} \mu_a(\omega) \right) \quad (23)$$

s.t. (11)–(14),

$$x_a'(\omega)^2 + \rho_a(\omega)^2 \le \delta_a(\omega)^2, \quad \forall a \in A(\omega), \quad (24)$$
$$\theta_a'(\omega)^2 + \sigma_a(\omega)^2 \le \phi_a(\omega)^2, \quad \forall a \in A(\omega), \quad (25)$$
$$u_a'(\omega)^2 + \gamma_a(\omega)^2 \le \eta_a(\omega)^2, \quad \forall a \in A(\omega), \quad (26)$$
$$-x_a'(\omega) + 2x_a(\omega) = 0, \quad \forall a \in A(\omega), \quad (27)$$
$$-\rho_a(\omega) + \theta_a(\omega) = 1, \quad \forall a \in A(\omega), \quad (28)$$
$$-\delta_a(\omega) + \theta_a(\omega) = -1, \quad \forall a \in A(\omega), \quad (29)$$
$$-\theta_a'(\omega) + 2\theta_a(\omega) = 0, \quad \forall a \in A(\omega), \quad (30)$$
$$-\sigma_a(\omega) + u_a(\omega) - x_a(\omega) = 0, \quad \forall a \in A(\omega), \quad (31)$$
$$-\phi_a(\omega) + u_a(\omega) + x_a(\omega) = 0, \quad \forall a \in A(\omega), \quad (32)$$
$$-u_a'(\omega) + 2u_a(\omega) = 0, \quad \forall a \in A(\omega), \quad (33)$$
$$-\gamma_a(\omega) - x_a(\omega) + \mu_a(\omega) = 0, \quad \forall a \in A(\omega), \quad (34)$$
$$-\eta_a(\omega) + \mu_a(\omega) + x_a(\omega) = 0, \quad \forall a \in A(\omega), \quad (35)$$
$$v_\pi(\omega) \ge 0, \quad \forall \pi \in \bigcup_{r \in O, s \in F(\omega)} P_{rs}^\lambda(\omega), \quad (36)$$

$$x_a(\omega), x_a'(\omega), \theta_a(\omega), \theta_a'(\omega), u_a(\omega), u_a'(\omega), \delta_a(\omega),$$
$$\eta_a(\omega), \phi_a(\omega), \mu_a(\omega) \ge 0, \quad \forall a \in A(\omega). \quad (37)$$

Objective function (23) is modified from the original objective function as defined above, and constraints (11)–(14) are the original constraints from $PSP(\bar{y}, \omega)$. Constraints (24)–(26) define the three second-order quadratic cones, and constraints (27)–(35) are generated by replacing each term (the two terms on the left-hand side inside the norms and the term on the right-hand side) of SOCP constraints (19)–(21) by a single auxiliary variable to help derive the dual of the $CPSP(\bar{y}, \omega)$. Constraints (36) and (37) are variable restrictions.

We associate the dual variables $z_r(\omega), \Gamma_{rs}(\omega), \Lambda_{ri}(\omega),$ $\psi_a(\omega), c_{1a}(\omega), c_{2a}(\omega), c_{3a}(\omega), c_{4a}(\omega), c_{5a}(\omega), c_{6a}(\omega),$

$c_{7a}(\omega)$, $c_{8a}(\omega)$, and $c_{9a}(\omega)$ for constraints (11)–(14) and (27)–(35), respectively, and the resulting dual subproblem (DSP) is formulated as follows:

Dual Subproblem $(DSP(\bar{y}, \omega))$

$$\max \left\{ \sum_{r\in O} z_r(\omega) + \sum_{r\in O}\sum_{s\in F(\omega)} \Gamma_{rs}(\omega)\bar{y}_s \right.$$
$$+ \sum_{r\in O}\sum_{i\in F(\omega)} \Lambda_{ri}(\omega)(1-\bar{y}_i) + \sum_{a\in A(\omega)} c_{2a}(\omega)$$
$$\left. - \sum_{a\in A(\omega)} c_{3a}(\omega) \right\} \quad (38)$$

$$\text{s.t. } z_r(\omega) + \Gamma_{rs}(\omega) + \sum_{i\in F(\omega):d^{\pi}(\omega)>(1+\lambda)d^*_{ri}(\omega)} \Lambda_{ri}(\omega)$$
$$- \sum_{a\in A(\omega):a\in\pi} w_r(\omega)\psi_a(\omega) \le 0,$$
$$\forall\, r\in O, s\in F(\omega), \pi\in P^{\lambda}_{rs}(\omega), \quad (39)$$

$$\psi_a(\omega) + 2c_{1a}(\omega) - c_{5a}(\omega) + c_{6a}(\omega) - c_{8a}(\omega)$$
$$+ c_{9a}(\omega) \le t_a^0, \quad \forall\, a\in A(\omega), \quad (40)$$

$$c_{8a}(\omega) + c_{9a}(\omega) \le \frac{t_a^0\alpha}{c_a(\omega)^\beta}, \quad \forall\, a\in A(\omega), \quad (41)$$

$$c_{2a}(\omega) + c_{3a}(\omega) + 2c_{4a}(\omega) \le 0, \quad \forall\, a\in A(\omega), \quad (42)$$

$$c_{5a}(\omega) + c_{6a}(\omega) + 2c_{7a}(\omega) \le 0, \quad \forall\, a\in A(\omega), \quad (43)$$

$$c_{1a}(\omega)^2 + c_{2a}(\omega)^2 \le c_{3a}(\omega)^2, \quad \forall\, a\in A(\omega), \quad (44)$$

$$c_{4a}(\omega)^2 + c_{5a}(\omega)^2 \le c_{6a}(\omega)^2, \quad \forall\, a\in A(\omega), \quad (45)$$

$$c_{7a}(\omega)^2 + c_{8a}(\omega)^2 \le c_{9a}(\omega)^2, \quad \forall\, a\in A(\omega), \quad (46)$$

$$\Gamma_{rs}(\omega) \le 0, \quad \forall\, r\in O, s\in F(\omega), \quad (47)$$

$$\Lambda_{ri}(\omega) \le 0, \quad \forall\, r\in O, i\in F(\omega), \quad (48)$$

$$c_{3a}(\omega), c_{6a}(\omega), c_{9a}(\omega) \ge 0, \quad \forall\, a\in A(\omega). \quad (49)$$

The dual subproblem is also a SOCP problem. Note that when the $CPSP(\bar{y}, \omega)$ is feasible, we can also find a point for which it is strictly feasible. Since it is also bounded, by the strong duality theorem for SOCP problems (Ben-Tal and Nemirovski 2001), the $DSP(\bar{y}, \omega)$ is feasible and bounded, and strong duality holds—i.e., $CPSP(\bar{y}, \omega)$ and $DSP(\bar{y}, \omega)$ attain the same optimal values. Since Y is a finite discrete set and we need one cut for each element of y, the generalized BD procedure generates finitely many cuts and terminates in a finite number of steps (Geoffrion 1972).

We ensure that the MP generates shelter location decisions that render every subproblem feasible. Hence, we only add optimality cuts as deemed necessary. The MP is as follows:

Master Problem (MP)

$$\min \;\; \theta \quad (50)$$

$$\text{s.t. } \sum_{s\in F} y_s = p, \quad (51)$$

$$\sum_{s\in \bar{F}_r(\omega)} y_s \ge 1, \quad \forall\, r\in O, \omega\in\Omega, \quad (52)$$

$$\theta \ge \sum_{\omega\in\Omega} p(\omega)\left(\sum_{r\in O} z_r^g(\omega) + \sum_{r\in O}\sum_{s\in F(\omega)} \Gamma_{rs}^g(\omega)y_s \right.$$
$$+ \sum_{r\in O}\sum_{i\in F(\omega)} \Lambda_{ri}^g(\omega)(1-y_i)$$
$$\left. + \sum_{a\in A(\omega)} c_{2a}^g(\omega) - \sum_{a\in A(\omega)} c_{3a}^g(\omega) \right),$$
$$\forall\, g\in G, \quad (53)$$

$$\theta \ge l, \quad (54)$$

$$y_s \in \{0,1\}, \quad \forall\, s\in F, \quad (55)$$

where $G = \bigcup_{\bar{y}\in V} G(\bar{y})$ is the set of optimal multiplier vectors and θ is the surrogate variable that represents the subproblems in the MP objective function and is a lower bound on the expected total evacuation time (Van Slyke and Wets 1969; Birge and Louveaux 1997).

The objective function (50) of MP minimizes the value of the surrogate variable. Constraint (51) limits the number of shelter sites open to a prespecified number p. By adding induced constraints (52) in the MP, we ensure that SPs are always feasible—i.e., there exists at least an open and reachable shelter for each $r\in O$, in every scenario $\omega\in\Omega$. Constraint set (53) is the optimality cuts. Constraints (54) set a lower bound l on the auxiliary variable θ. We compute such a lower bound with a very simple heuristic method. In a given scenario, we find the shortest path to the closest shelter for each origin r and compute the total travel time of the vehicles on this path using the free-flow travel time. The sum of the total travel times on these paths gives us a lower bound for that specific scenario. We take their expected value to compute the lower bound l. Imposing such a lower bound on θ in the formulation improves the lower bound during initial iterations and the overall solution time to some extent. Constraints (55) define the types of variables.

5. Improving the Performance of the BD Algorithm

Since the BD has been introduced (Benders 1962), many researchers have investigated methods to improve its performance. Geoffrion and Graves (1974) propose a branch-and-bound framework in which they solve the MP in an ϵ-optimal fashion instead of solving it to optimality at every iteration. McDaniel and Devine (1977) present a methodology that solves the LP relaxation of the integer subproblem for some initial number of iterations to generate Benders cuts to reduce the computational burden. Magnanti and Wong (1981) propose to accelerate the BD algorithm by generating strong (Pareto-optimal) optimality cuts from the alternate optima of the Benders subproblem. Papadakos (2008), Fischetti, Salvagnin, and Zanette (2010), Saharidis and Ierapetritou (2010), and Sherali and Lunday (2013) propose alternative methodologies on deriving strong

nondominated Benders cuts. Van Roy (1986) proposes a cross-decomposition method that unifies Benders decomposition and Lagrangian relaxation into a single framework. Saharidis, Minoux, and Ierapetritou (2010) demonstrate the effectiveness of a new strategy, which they refer to as covering cut bundle generation. The method they propose is based on the idea of generating multiple cuts that involve as many complicating variables as possible or directly generating a high-density Pareto cut by lifting the Pareto-optimal cuts (Tang, Jiang, and Saharidis 2013). Saharidis, Boile, and Theofanis (2011) and Tang, Jiang, and Saharidis (2013) work on deriving valid inequalities to improve the lower bounds obtained by the MP. Naoum-Sawaya and Elhedhli (2013) present an interior-point branch-and-cut algorithm based on BD and the analytic center cutting plane method (ACCPM), and show that the ACCPM-based Benders cuts are both Pareto-optimal and valid for any node of the branch-and-bound tree. Fischetti and Lodi (2003) and Rei et al. (2009) show how local branching can be used to accelerate the classical BD algorithm.

5.1. Multicut Strategy

In our initial experiments, we observed that generating a single cut aggregated from the optimal multiplier vectors of each subproblem results in slow convergence of the BD algorithm. By adding disaggregate cuts, more detailed information is given to the first stage, which often results in fewer iterations compared to the single-cut method (Birge and Louveaux 1997). Hence, we employ a multicut strategy—i.e., we add an optimality cut for every subproblem related to a scenario in case a violation is identified. Therefore, for any optimal solution of $DSP(\bar{y}, \omega)$, the Benders optimality cuts are redefined as follows:

$$
\begin{aligned}
\theta(\omega) \geq p(\omega)\Bigg(& \sum_{r \in O} z_r^g(\omega) + \sum_{r \in O} \sum_{s \in F(\omega)} \Gamma_{rs}^g(\omega) y_s \\
& + \sum_{r \in O} \sum_{i \in F(\omega)} \Lambda_{ri}^g(\omega)(1 - y_i) \\
& + \sum_{a \in A(\omega)} c_{2a}^g(\omega) - \sum_{a \in A(\omega)} c_{3a}^g(\omega) \Bigg), \\
& \forall \omega \in \Omega, g \in G.
\end{aligned}
$$

The objective function of the MP is modified as $\sum_{\omega \in \Omega} \theta(\omega)$, where $\theta(\omega)$ is a surrogate variable that represents a subproblem related to a specific scenario $\omega \in \Omega$, and we set a lower bound for each subproblem—i.e., we modify constraint (54) as $\theta(\omega) \geq l(\omega)$ for all $\omega \in \Omega$.

5.2. Implementing a Callback Routine

In the classical implementation of BD, the current relaxed MP is solved to optimality at every iteration of the algorithm, and for that reason a search tree is generated from scratch every time the relaxed MP is solved.

Alternatively, the reformulation can be solved with a branch-and-cut algorithm where the Benders cuts are generated as cutting planes. In this approach, a single integer problem is solved using a single search tree. Each time an integer solution is found, it is either certified as feasible or a Benders cut violated by this candidate solution is added. This can be done using the lazy constraint callback routines available in the commercial solvers.

We refer to the version of the BD algorithm in which we employ the multicut strategy and the callback routine as *BD*.

5.3. Defining Strong (Pareto-Optimal) Cuts

When the primal subproblem is a network optimization problem such as the facility location on networks, shortest route, and transhipment, it is common to get degenerate solutions, which leads to multiple optimal solutions for the dual subproblem (Magnanti and Wong 1981). Because of this fact, cuts of different strength can be generated. Although any of these are valid optimality cuts, defining strong (Pareto-optimal) ones at every iteration of the algorithm may significantly decrease the number of iterations and hence improve the overall solution time. A cut is said to be Pareto-optimal if it is not dominated by any other cut. Let y^0 be a point in the relative interior of the convex hull of feasible location vectors, i.e., $y^0 \in ri(CH(V))$ and $v(DSP(\bar{y}, \omega))$ be the optimal value of $DSP(\bar{y}, \omega)$. To generate a Pareto-optimal cut, we solve the following auxiliary problem:

Magnanti–Wong (MW) *Problem*

$$
\begin{aligned}
\max \Bigg\{ & \sum_{r \in O} z_r(\omega) + \sum_{r \in O} \sum_{s \in F(\omega)} \Gamma_{rs}(\omega) y_s^0 \\
& + \sum_{r \in O} \sum_{i \in F(\omega)} \Lambda_{ri}(\omega)(1 - y_i^0) \\
& + \sum_{a \in A(\omega)} c_{2a}(\omega) - \sum_{a \in A(\omega)} c_{3a}(\omega) \Bigg\} \quad (56)
\end{aligned}
$$

s.t. (39)–(49),

$$
\begin{aligned}
& \sum_{r \in O} z_r(\omega) + \sum_{r \in O} \sum_{s \in F(\omega)} \Gamma_{rs}(\omega) \bar{y}_s \\
& + \sum_{r \in O} \sum_{i \in F(\omega)} \Lambda_{ri}(\omega)(1 - \bar{y}_i) + \sum_{a \in A(\omega)} c_{2a}(\omega) \\
& - \sum_{a \in A(\omega)} c_{3a}(\omega) = v(DSP(\bar{y}, \omega)). \quad (57)
\end{aligned}
$$

Constraint (57) in the dual auxiliary problem MW ensures that one chooses an optimal multiplier vector from among alternative ones; and the objective function of the MW problem chooses from among these multiplier vectors the one that generates the strongest cut to be added to the MP.

Papadakos (2008) points out that the dependency of the MW method on the Benders subproblem and on an MP core point may sometimes decrease the performance of the algorithm, and that it may not always be easy to find a readily available core point. Another issue

with the MW problem they discuss is the numerical unboundedness caused by constraint (57), which they show can be eliminated from the MW problem to generate a Pareto-optimal cut. That way, an enhanced MW (EMW) method independent of the subproblem, which enables adding a useful cut before solving the MP, is proposed.

Papadakos (2008) prove that y^0 does not have to be a core point or even a point of V. Furthermore, since y^0 only modifies the objective function and does not alter the feasible region of the MW problem, choosing a y^0 which is not in ri($CH(V)$) still generates a valid optimality cut, although it may not be Pareto-optimal. In our implementation, we start this algorithm with $y^0 = 1$ and update this point at every iteration k using the equation $y_i^{0,k} = \frac{1}{2} y_i^{0,k-1} + \frac{1}{2} \bar{y}_i^k$.

Both algorithms with MW and EMW methodologies solve an auxiliary dual subproblem to generate the Pareto-optimal cuts. The main drawback of these two algorithms is that one has to solve the dual subproblem and the MW auxiliary problem at every iteration for every scenario, which may result in long CPU times. Our preliminary experiments show that for both of these algorithms, the number of iterations and the total number of optimality cuts generated generally decrease compared to the BD. However, the CPU times worsen as a result of solving the dual subproblem and the MW auxiliary problem, at every iteration and for every subproblem when there is a violation. We also experimented on using a reduced primal subproblem (RPSP) in combination with the MW or EMW procedure that resulted in long CPU times as well. However, since the EMW problem is independent of the subproblem, we can take advantage of adding initial cuts to the MP before we begin solving it, and continue as in BD. We denote this algorithm as BD_IC.

Sherali and Lunday (2013) propose a procedure that generates maximal nondominated Benders cuts. Instead of solving an auxiliary MW problem at each iteration, which brings a computational burden and increases the CPU times, the authors solve a preemptive priority multiple objective program. The aim is to solve the original dual subproblem optimally with the first priority, and among alternative optimal solutions to this problem, maximize (56). They point out that there exists a $\zeta > 0$ small enough such that this preemptive priority multiple objective program can be equivalently represented as the following weighted-sum problem:

Modified Sherali and Lunday Dual Subproblem

$$
\max \ (38) + \zeta \Bigg(\sum_{r \in O} z_r(\omega) + \sum_{r \in O} \sum_{s \in F(\omega)} \Gamma_{rs}(\omega) y_s^0
$$
$$
+ \sum_{r \in O} \sum_{i \in F(\omega)} \Lambda_{ri}(\omega)(1 - y_i^0)
$$
$$
+ \sum_{a \in A(\omega)} c_{2a}(\omega) - \sum_{a \in A(\omega)} c_{3a}(\omega) \Bigg) \quad (58)
$$
$$
\text{s.t. } (39)\text{–}(49),
$$

where y_s^0 is a positive-weight vector—i.e., a positive core point solution as we defined previously. We begin with a core point and update it as we described in the EMW method. We take $\zeta = 10^{-11}$. We denote this algorithm as BD_SL.

Fischetti, Salvagnin, and Zanette (2010) propose a new selection criterion for Benders cuts—in particular, when both violations of feasibility and optimality cuts exist. They represent the primal subproblem as a pure feasibility problem. Preliminary computational studies showed that this methodology could not find a solution in good CPU times for our problem.

We also investigated two alternative algorithms based on the generalized BD (GBD) method introduced by Geoffrion (1972) to solve the MINLP. While the first one is the original GBD algorithm proposed by Geoffrion (1972), the second algorithm uses an acceleration strategy called "variable fixing" proposed by Fischetti, Ljubić, and Sinnl (2016). We observed that BD algorithms that employ SOCP duality perform much better compared to the algorithms that use Lagrangian duality in the GBD setting. We believe that this is partly because solving the dual subproblem is faster than solving the primal subproblem. The GBD algorithms perform better than EF only in a small number of instances. Their advantage over the EF is that they do not encounter memory problems for the instances with a large number of scenarios.

6. Solving the Subproblems with a Cutting Plane Approach

To solve the DSP, we pregenerate all possible feasible paths using an algorithm by Byers and Waterman (1984). As the network size and the tolerance level get larger, we may encounter memory problems. Incorporating uncertainty with a large number of scenarios certainly makes the problem more difficult. In this case, it may be advantageous not to work with large models that involve variables for all possible paths and to generate these variables when required within a column-generation framework to solve the subproblem. As we solve the dual of the subproblem, we employ a cutting plane approach to generate the constraints related to these paths when needed.

For a given scenario $\omega \in \Omega$, we begin solving the DSP with a subset of constraints (39). We use the constraints associated with the set of shortest paths between every demand node and functioning shelters in the scenario ω. At every iteration, we determine whether there exists a path for which the respective constraint in the DSP is violated. If such paths are detected, we add the respective constraints to the DSP and repeat the procedure until there is no violated constraint.

For any given scenario $\omega \in \Omega$ and demand point-shelter pairs $r \in O, s \in F(\omega)$, finding a path that violates

constraint (39) requires solving the following separation problem:

Separation Problem

$$\Phi_{rs}(\omega) = \max \left\{ z_r(\omega) + \Gamma_{rs}(\omega) + \sum_{i \in F(\omega)} \Lambda_{ri}(\omega) f_i \right.$$

$$\left. - \sum_{a \in A(\omega)} w_r(\omega) \psi_a(\omega) \chi_a \right\} \quad (59)$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} \chi_a - \sum_{a \in \delta^-(i)} \chi_a$$

$$= \begin{cases} 1 & i = r, \\ 0 & \forall i \in N \setminus (r,s), \\ -1 & i = s, \end{cases} \quad (60)$$

$$\sum_{a \in A(\omega)} l_a \chi_a \le (1+\lambda) d_{ri}^*(\omega) + M f_i,$$

$$\forall i \in F(\omega), \quad (61)$$

$$\sum_{a \in A(\omega)} l_a \chi_a \le (1+\lambda) d_r^*(\omega), \quad (62)$$

$$\chi_a \in \{0,1\}, \quad \forall a \in A(\omega), \quad (63)$$

$$f_i \in \{0,1\}, \quad \forall i \in F(\omega), \quad (64)$$

where $\delta^-(i)$ and $\delta^+(i)$ denote the sets of incoming and outgoing arcs of node $i \in N$, χ_a is 1 if arc a is used in an optimal solution and 0 otherwise, M is a very big number, f_i is 1 if $\sum_{a \in A(\omega)} l_a \chi_a > (1+\lambda) d_{ri}^*(\omega)$ and 0 otherwise, and $d_r^*(\omega)$ is the shortest path distance to the closest open shelter from origin r in scenario $\omega \in \Omega$. If $\Phi_{rs}(\omega) > 0$, then the constraint related to the path defined by an optimal vector χ is added to the DSP, and if $\Phi_{rs}(\omega) \le 0$ for all $r \in O, s \in F(\omega)$, then there is no violated constraint to be added to the DSP in scenario $\omega \in \Omega$—i.e., the $DSP(\bar{y}, \omega)$ is solved to optimality.

Considering that the separation problem is solved in every iteration and for every subproblem of the BD algorithm, our early experiments showed that solving the separation problem as it is is not effective in terms of the CPU times. For that reason, we propose the following methodology to solve the separation problem.

For a given $r \in O$, we compute $d_{ri}^*(\omega)$ for all $i \in F(\omega)$ in scenario $\omega \in \Omega$ and sort them in ascending order. Then, for a given scenario $\omega \in \Omega$ and demand point-shelter pair $r \in O, s \in F(\omega)$, we create the scalar vectors $f = \{0,0,0,0,0,\dots,0\}$, $f = \{1,0,0,0,0,\dots,0\},\dots, f = \{1,1,1,1,0,\dots,0\}$, where the final 1 in the sequence is determined by the index of $\max_i\{d_{ri}^*(\omega)\} < d_r^*(\omega)$. The scalar vector f replaces the binary variable f in the separation problem. We modify constraint (61) as $\sum_{a \in A(\omega)} l_a \chi_a \le (1+\lambda) d_{ri'}^*(\omega)$, where $i' = \text{argmin}\{d_{ri}^*: f_i = 0\}$. Then, the separation problem requires solving a number of single resource-constrained shortest path problems (CSP) over a graph with arc costs equal to $w_r(\omega)\psi_a(\omega)$ and arc resources as the geographical distances. The CSP is either solved for every vector f in sequence as defined

above or it terminates as soon as a path with the associated constraint is detected. We employ the algorithms by Santos, Coutinho-Rodrigues, and Current (2007) and Yen (1971) to solve the CSP.

Before we start the BD algorithm, we find five shortest paths for every scenario $\omega \in \Omega, r \in O, s \in F(\omega)$ using the approach proposed by Yen (1971). In the separation procedure, we first check if there exists a violated constraint for any of these paths. If such a path is detected, then the corresponding constraint is added to the DSP. If no such path is detected, then the separation problem is solved exactly.

7. Computational Study

In our early experiments, we observed that the solution times of the classical BD algorithm where the MP is solved to optimality as an integer problem at each iteration is much worse than our implementation in which we employ a lazy constraint callback. Likewise, the versions of our BD algorithm that employ the aggregate cut rather than a multicut strategy and/or solving the primal subproblem instead of the dual subproblem to generate the optimality cuts are not promising in terms of CPU times either. Hence, we do not report our computational studies for these algorithms.

Among all of the versions on which we experimented, there are four algorithms that solve our problem quickly. In all versions of our BD algorithm, we solve the dual subproblem to generate the optimality cuts implementing the lazy constraint callback feature of ILOG CPLEX, and we adopt a multicut strategy. In the first three algorithms, the subproblem is solved by pregenerating all possible paths, and the last algorithm uses the cutting plane framework to solve the dual subproblem. Below, we summarize these algorithms.

Algorithm BD. This is the basic version of our BD algorithm, in which we employ the dual subproblem, lazy constraint callback, and a multicut strategy.

Algorithm BD_IC. In this algorithm, we take the advantage of the EMW problem that can be solved independently from the dual subproblem and solve the EMW problem only once before we begin solving the MP to generate an initial set of valid cuts. We set the core point $y^0 = 1$ to generate the initial set of cuts.

Algorithm BD_SL. In this version, the idea is to solve the original dual subproblem optimally with the first priority and, among alternative optimal solutions to this problem, to maximize (56). To achieve this, we use a weighted sum of (38) and (56) with a weight vector of $(1, \zeta)$ as our modified objective function of the dual subproblem. We begin with a core point $y^0 = 1$ and update it at every iteration k using the equation, $y_i^{0,k} = \frac{1}{2} y_i^{0,k-1} + \frac{1}{2} \bar{y}_i^k$. We take $\zeta = 10^{-11}$.

Algorithm BD_CP. This is the basic version of our BD algorithm, in which we employ the dual subproblem, lazy constraint callback, and a multicut strategy, but where we solve the dual subproblem by implementing a cutting plane approach instead of pregenerating all constraints for all possible paths.

7.1. Scenario Generation

We generated two of our instances—i.e., Istanbul Anatolian and Istanbul European—using real data from a disaster prevention and mitigation study conducted by the Istanbul Metropolitan Municipality (IMM) and the Japan International Cooperation Agency (JICA) (IMM-JICA 2002) for earthquake preparedness and response planning for an impending major earthquake in Istanbul, Turkey, as in Bayram and Yaman (2015). Turkey is among the countries where tsunamis, landslides, and fires have been observed as secondary disasters following a major earthquake (Marano, Wald, and Allen 2010). One of the findings in that report is that there does not exist an emergency evacuation system in Istanbul and that it is imperative that an evacuation system be established to mitigate human casualties due to second or third aftershocks and secondary disasters following the earthquake. For that reason, we assume an emergency evacuation in the sense that we are evacuating people to protect them from the impact of aftershocks and the secondary disasters.

The IMM–JICA report gives details of four scenarios for the earthquake, scenarios A–D, where scenario A is referred to as the most probable scenario and scenario C is the worst case scenario. We take scenario A as the base (nominal) scenario since the evacuation demand specific to each origin node is only given for that scenario in the report. The demand for other scenarios is generated by comparing the length of the fault lines expected to be broken and the expected magnitude of the earthquake in these scenarios to those of the base scenario A.

In the report, Istanbul City is divided into smaller disaster regions using a spatial risk analysis. For each scenario, five different risk zones are determined. We assign disruption/degradation probabilities for each zone in the base scenario and find these probabilities for risk zones of the other scenarios by a similar comparison stated above. The highest probability of disruption is assigned to risk zone 1, as it is closest to the fault line, and the lowest one to risk zone 5. The road segments and shelters in the network are classified into different sets based on the risk zone in which they are located. Given a particular scenario, we randomly determine whether an arc (road segment) is damaged by considering the risk zone in which it is located and the probability of disruption/degradation assigned to this zone. If the arc is damaged, we again randomly specify the amount of capacity it lost by the number of lanes. If there is partial damage, then this arc is degraded; and if

Table 1. Specifics of the Instances Used in the Computational Study

Instance	$|N|$	$|A|$	$|O|$	$|F|$	Total demand	$|O-F|$
Istanbul Anatolian	50	146	13	17	83,133	221
Istanbul European	80	238	25	32	272,900	800
P-median1	100	396	85	15	123,388	1,275

it has lost all of its lanes, then this road segment is disrupted and unpassable. We follow a similar approach for the disruption of the shelters. We generate the scenarios in a random fashion using the characteristics of the four main scenarios. Please see Bayram and Yaman (2015) for more detail on how we generate the scenarios.

We downloaded the P-median1 instance from Beasley (1990). We created the demand for each origin node randomly between 1,000 and 2,000 (vehicles). We also generated potential shelter sites randomly on the network for that instance. We assign the arcs and shelters into risk zones assuming there are four basic scenarios as in the Istanbul instances.

We generate instances in such a way that for every origin, there exists at least one reachable shelter—i.e., everyone can be evacuated.

7.2. Computational Testing

The specifics of the instances used in the computational study are shown in Table 1. Here, $|O-F|$ is the number of origin–destination pairs that are connected with a directed path in the original undisrupted graph. We perform our computational tests on a workstation with 2 Xeon E5-2609 4C 2.4 GHz CPUs and 96 GB RAM by using Java ILOG CPLEX version 12.5.1.

In Tables 2–6, we compare the computational efficiencies of the three BD algorithms and the EF for different values of p, λ, and $|\Omega|$ (number of scenarios). For each instance, we report the number of iterations the BD algorithm performed, the number of optimality cuts added, and the solution times. Here, the number of iterations refers to the number of times an incumbent solution is passed to subproblems—i.e., the number of times subproblems are solved. We set a time limit of five hours for our experiments. If the problem is not solved to optimality within the time limit, then we report the remaining gap in parentheses in the column of solution times. If a solution cannot be obtained within the time limit, we report such a situation as "No Solution (NS)." We are able to solve the Istanbul Anatolian instances with up to 1,000 scenarios without any memory problems using any of the BD algorithms. On the other hand, using the EF with CPLEX does not generate a solution within five hours for the instances with 1,000 scenarios. All of the BD algorithms perform much better compared to the EF in terms of the CPU times. The *BD* algorithm performs at least 1.85 times better than the EF; this rate increases up to 19.16 and marks 5.53 on

Table 2. Comparison of Different Algorithms With Respect to Computational Effectiveness (Istanbul Anatolian Instances)

			BD			BD_SL			BD_IC			EF(SCSO)
$|\Omega|$	p	λ	No. of iter.	No. of cuts	Sol. time	No. of iter.	No. of cuts	Sol. time	No. of iter.	No. of cuts	Sol. time	Sol. time
50	5	0	94	4,012	242	93	3,964	224	88	3,851	226	984
50	8	0	86	3,634	182	87	3,680	183	76	3,277	162	493
50	10	0	92	3,844	229	91	3,840	214	96	3,674	236	527
50	12	0	53	2,150	111	53	2,150	106	50	2,118	106	476
50	15	0	21	745	54	21	745	49	20	703	47	170
50	5	0.05	79	3,282	214	86	3,289	232	91	4,025	255	1,415
50	8	0.05	110	4,661	238	110	4,661	239	96	4,261	220	789
50	10	0.05	116	4,457	302	116	4,457	290	94	3,799	247	616
50	12	0.05	79	2,950	160	79	2,950	172	65	2,586	136	527
50	15	0.05	20	693	55	20	693	48	15	565	38	129
50	5	0.1	92	4,171	271	92	4,171	269	94	4,321	270	1,367
50	8	0.1	151	6,381	339	180	7,517	414	149	6,463	348	972
50	10	0.1	97	4,190	261	97	4,193	256	86	3,890	245	856
50	12	0.1	72	2,845	158	72	2,845	166	61	2,593	137	468
50	15	0.1	21	713	58	21	713	53	15	559	41	108
50	5	0.15	93	4,030	309	64	2,795	206	71	3,144	220	1,277
50	8	0.15	90	3,645	209	90	3,645	209	94	3,969	230	674
50	10	0.15	99	3,824	264	99	3,824	259	93	3,663	255	1,019
50	12	0.15	54	1,776	120	54	1,776	119	38	1,418	91	714
50	15	0.15	24	777	67	24	777	61	14	533	40	144
50	5	0.2	104	4,514	368	104	4,514	364	97	4,481	356	1,862
50	8	0.2	143	6,210	374	146	6,299	377	122	5,542	323	1,034
50	10	0.2	95	4,075	270	96	4,125	266	90	3,891	258	1,027
50	12	0.2	57	2,329	137	57	2,329	138	58	2,371	140	777
50	15	0.2	16	642	50	16	642	43	16	636	45	150
Average			78	3,222	202	79	3,224	198	72	3,053	187	743
100	5	0	62	4,985	325	62	4,985	307	66	5,172	335	3,409
100	8	0	74	5,904	323	72	5,926	302	65	5,114	280	1,838
100	10	0	87	6,095	413	87	6,095	398	91	6,864	459	1,881
100	12	0	69	5,678	285	69	5,678	278	82	5,753	344	1,635
100	15	0	24	1,775	124	24	1,775	111	22	1,757	105	603
100	5	0.05	79	6,575	421	79	6,575	414	66	5,383	370	4,114
100	8	0.05	50	4,174	222	50	4,174	218	70	5,658	324	2,657
100	10	0.05	87	7,112	466	87	7,112	434	87	6,864	465	1,786
100	12	0.05	65	4,926	272	65	4,924	269	77	5,454	343	2,455
100	15	0.05	30	2,269	159	30	2,269	147	31	2,275	169	687
100	5	0.1	98	8,258	575	98	8,258	565	69	6,476	427	11,015
100	8	0.1	119	10,065	568	119	10,065	580	108	9,254	528	3,631
100	10	0.1	131	10,589	688	130	10,484	664	120	9,371	662	2,023
100	12	0.1	73	6,021	341	73	6,021	325	96	7,579	445	2,075
100	15	0.1	23	1,856	128	23	1,856	116	22	1,908	125	657
100	5	0.15	81	7,585	521	80	7,381	501	96	8,473	607	7,469
100	8	0.15	120	10,034	595	122	10,234	603	118	10,702	600	3,624
100	10	0.15	93	7,660	518	95	7,496	534	124	9,895	688	2,995
100	12	0.15	73	5,910	346	73	5,910	341	77	5,985	366	2,474
100	15	0.15	25	1,730	147	25	1,730	130	19	1,533	105	698
100	5	0.2	66	5,883	487	66	5,883	472	68	5,685	519	5,354
100	8	0.2	91	7,938	494	91	7,938	491	112	9,477	615	3,269
100	10	0.2	90	7,994	545	90	7,994	503	96	8,001	603	2,891
100	12	0.2	71	5,802	350	69	5,904	348	75	5,910	383	2,273
100	15	0.2	9	799	62	9	799	50	11	791	63	585
Average			71	5,905	375	72	5,899	364	75	6,053	397	2,884

average, not including the actual solution times of the EF for the instances with 1,000 scenarios since these instances hit the time limit. The BD_SL algorithm performs even better, with 2.03, 19.49, and 5.77 values as the minimum, maximum, and average rates, respectively.

These numbers for the BD_IC algorithm are 2.23, 25.80, and 5.66, respectively. The BD_SL algorithm generally performs better than the BD algorithm for the Istanbul Anatolian case. In 53 of the 75 total instances for the Istanbul Anatolian network, BD_SL has smaller CPU

Table 3. Comparison of Different Algorithms With Respect to Computational Effectiveness (Istanbul Anatolian Instances)

			BD			BD_SL			BD_IC			EF(SCSO)
$|\Omega|$	p	λ	No. of iter.	No. of cuts	Sol. time	No. of iter.	No. of cuts	Sol. time	No. of iter.	No. of cuts	Sol. time	Sol. time
1,000	5	0	69	58,667	3,719	57	48,741	2,919	54	45,825	3,026	(NS)
1,000	8	0	43	35,799	1,823	43	35,799	1,890	56	43,696	2,618	(NS)
1,000	10	0	96	71,622	4,944	96	71,693	4,801	89	65,506	4,909	(NS)
1,000	12	0	62	46,228	2,565	60	45,949	2,555	98	68,007	4,323	(NS)
1,000	15	0	36	20,521	1,864	36	20,521	1,750	36	21,752	1,889	(NS)
1,000	5	0.05	67	54,698	4,045	61	51,723	3,438	78	63,416	4,886	(NS)
1,000	8	0.05	98	76,155	4,708	87	72,632	4,137	87	73,616	4,306	(NS)
1,000	10	0.05	80	66,505	4,542	105	77,038	5,876	74	58,588	4,244	(NS)
1,000	12	0.05	57	40,046	2,629	57	40,047	2,629	71	54,449	3,363	(NS)
1,000	15	0.05	25	19,557	1,394	25	19,557	1,355	18	13,498	986	(NS)
1,000	5	0.1	66	57,680	4,358	72	62,446	4,594	56	47,557	3,630	(NS)
1,000	8	0.1	106	84,512	5,514	93	74,668	4,914	83	68,413	4,503	(NS)
1,000	10	0.1	74	66,560	4,457	82	71,534	4,894	132	106,110	8,413	(NS)
1,000	12	0.1	92	67,120	4,469	73	53,905	3,513	74	53,754	3,898	(NS)
1,000	15	0.1	32	23,556	1,849	48	23,573	2,625	45	30,612	2,685	(NS)
1,000	5	0.15	75	64,640	5,224	71	61,675	4,936	71	59,705	5,196	(NS)
1,000	8	0.15	95	82,014	5,253	113	73,448	5,976	82	70,475	4,710	(NS)
1,000	10	0.15	102	83,721	6,602	135	85,520	8,129	119	95,995	7,745	(NS)
1,000	12	0.15	53	43,363	2,708	84	43,422	4,208	55	42,220	2,895	(NS)
1,000	15	0.15	30	19,234	1,819	30	19,235	1,772	31	19,860	1,928	(NS)
1,000	5	0.2	76	62,679	5,986	75	61,703	5,620	67	55,737	5,313	(NS)
1,000	8	0.2	82	73,335	4,805	74	66,407	4,303	99	74,126	5,932	(NS)
1,000	10	0.2	101	91,607	6,726	84	74,192	5,504	117	94,250	7,583	(NS)
1,000	12	0.2	66	54,250	3,406	71	55,335	3,664	73	60,058	3,997	(NS)
1,000	15	0.2	18	13,669	1,148	18	13,669	1,014	16	13,628	974	(NS)
Average			68	55,110	3,862	70	52,977	3,881	71	56,034	4,158	18,000[*]

[*]The average solution time is 18,000 or more.

times and generally ends in a smaller number of iterations adding a smaller number of optimality cuts. This rate is 41 of 75 total instances for the *BD_IC* algorithm. The *BD_IC* algorithm performs better than the *BD_SL* algorithm in 34 of 75 instances in terms of CPU times.

For the Istanbul European network, EF runs into memory problems for the instances with $|\Omega| = 50$, $\lambda = 0.2$, $|\Omega| = 100$, $\lambda \geq 0.15$, and $|\Omega| = 1,000$, $\lambda \geq 0.1$. When the number of scenarios is 100, for $\lambda = 0.05$ and $p = 10$, the EF can obtain a solution with a gap of 99.5% within the five-hour time limit, and seven out of 25 instances with $|\Omega| = 100$ are of this sort. When $\lambda = 0.1$ and $p = 10$, EF cannot obtain a solution at the root node within the time limit, and three out of 25 instances with $|\Omega| = 100$ encounter this problem. For all of the instances that the EF does not encounter the memory issue when the number of scenarios is 1,000, there is no solution obtained at the root node within the time limit. BD algorithms run into memory problems for the instances with $|\Omega| = 100, \lambda = 0.2$ and $|\Omega| = 1,000$, $\lambda \geq 0.15$. Excluding the instances with memory problems, the *BD*, *BD_SL*, and *BD_IC* algorithms solve all of the instances to optimality except for six, five, and six instances, respectively, with $|\Omega| = 1,000$ where they run into the time limit, the majority of the gaps being small. Each of the BD algorithms outperforms the EF in terms of the memory problems and the CPU times.

Except for the three instances with $|\Omega| = 50$ where EF performs better when p is relatively large and excluding the instances with memory problems and the ones that hit the time limit, the *BD* algorithm performs at least 1.09 times better than the EF, this rate increases up to 9.22 and marks 3.97 on average. For the *BD_SL* algorithm, these numbers are 1.14, 8.61, and 3.99 as the minimum, maximum, and average rates, respectively, and this algorithm outperforms EF in 57 of 60 instances. These numbers are 1.07, 8.07, and 3.54, respectively, for *BD_IC*, taking into account 58 of 60 instances where *BD_IC* has better CPU times. In 27 of the 60 instances, the *BD_SL* algorithm performs relatively better compared to *BD* in terms of CPU times. This rate is 21–60 for the *BD_IC* algorithm; and *BD_IC* performs better than *BD_SL* in 21 of 60 instances in terms of CPU times. The algorithms outperform each other for different instances in terms of the number of iterations. The *BD* algorithm adds generally a smaller number of optimality cuts with a small number of scenarios, but starting with larger λ values when the number of scenarios is 100, and for a larger number of scenarios, *BD_SL* adds generally a lower number of optimality cuts.

We observe similar results for the P-median1 network instances and report them in Tables 5 and 6. In more than two-thirds of the instances with 50 scenarios, EF hits the time limit with no solution at the

Table 4. Comparison of Different Algorithms With Respect to Computational Effectiveness (Istanbul European Instances)

$|\Omega|$	p	λ	BD No. of iter.	BD No. of cuts	BD Sol. time	BD_SL No. of iter.	BD_SL No. of cuts	BD_SL Sol. time	BD_IC No. of iter.	BD_IC No. of cuts	BD_IC Sol. time	EF(SCSO) Sol. time
50	10	0	114	4,930	603	114	4,930	601	136	5,584	743	2,997
50	15	0	131	5,553	559	133	5,646	583	211	8,942	918	1,273
50	20	0	184	7,630	789	174	7,279	748	207	8,041	906	5,796
50	25	0	210	8,823	1,076	224	9,006	1,119	189	7,478	954	1,464
50	30	0	50	1,843	249	50	1,843	241	46	1,793	226	208
50	10	0.05	108	4,916	612	119	5,324	655	121	5,297	700	5,643
50	15	0.05	119	5,084	539	127	5,237	570	136	5,614	630	1,502
50	20	0.05	167	6,692	747	164	6,451	734	189	7,285	858	2,445
50	25	0.05	158	5,895	808	153	5,874	784	158	6,447	834	1,248
50	30	0.05	20	716	106	20	716	99	21	668	106	157
50	10	0.1	145	6,661	1,107	151	6,904	1,168	156	7,052	1,233	5,927
50	15	0.1	92	4,140	586	94	4,233	586	158	6,221	1,032	1,189
50	20	0.1	168	6,782	1,071	148	6,228	904	133	5,262	823	6,268
50	25	0.1	144	6,064	1,039	144	6,064	1,025	126	5,639	931	1,988
50	30	0.1	24	781	179	24	781	171	16	648	117	195
50	10	0.15	139	6,367	2,569	141	6,406	2,625	129	5,926	2,447	9,220
50	15	0.15	100	4,433	1,439	102	4,531	1,472	164	6,660	2,414	2,571
50	20	0.15	153	6,208	2,206	135	5,567	1,926	186	7,406	2,683	6,832
50	25	0.15	131	4,921	2,337	111	3,628	1,955	120	4,353	2,172	1,623
50	30	0.15	14	546	279	14	546	252	9	342	160	220
50	10	0.2	144	6,781	10,906	157	7,464	12,025	193	9,002	14,719	
50	15	0.2	99	4,380	6,439	102	4,534	6,626	179	8,073	11,659	
50	20	0.2	156	6,464	10,433	152	6,659	9,834	183	7,675	11,684	Out of memory
50	25	0.2	87	3,520	6,375	106	4,074	7,782	134	5,126	10,676	
50	30	0.2	6	199	443	6	199	451	7	195	506	
Average			115	4,813	2,140	115	4,805	2,197	132	5,469	2,805	
100	10	0	91	7,805	1,006	111	9,412	1,175	160	13,607	1,841	(NS)
100	15	0	191	16,170	2,108	193	16,273	2,246	221	19,102	2,294	12,816
100	20	0	158	12,832	1,418	135	11,646	1,212	191	15,857	1,743	(0.57)
100	25	0	134	10,893	1,379	135	10,999	1,359	160	11,789	1,650	(0.003)
100	30	0	20	1,578	218	20	1,578	200	41	2,538	419	1,409
100	10	0.05	136	12,387	1,655	143	12,705	1,703	152	13,750	1,870	(0.995)
100	15	0.05	152	14,269	1,488	234	17,653	2,295	216	18,579	2,210	(0.964)
100	20	0.05	203	17,413	1,950	265	19,587	2,481	188	15,304	1,769	(NS)
100	25	0.05	263	19,549	2,843	262	19,346	2,759	181	13,368	1,976	(0.004)
100	30	0.05	32	2,569	343	36	2,679	365	24	1,958	262	1,499
100	10	0.1	167	15,733	2,684	173	14,097	2,782	214	18,999	3,480	(NS)
100	15	0.1	165	14,447	2,138	188	14,749	2,452	257	21,988	3,466	(0.967)
100	20	0.1	231	19,758	2,945	260	18,690	3,315	213	17,332	2,724	(0.996)
100	25	0.1	159	13,144	2,299	173	13,167	2,480	176	14,332	2,551	18,048
100	30	0.1	27	2,117	402	27	2,117	391	25	1,912	358	1,433
100	10	0.15	235	21,003	8,586	211	19,249	7,802	229	21,258	8,666	
100	15	0.15	241	21,431	6,993	242	21,292	7,216	322	29,351	11,165	
100	20	0.15	279	24,841	7,719	291	25,919	8,752	296	26,291	8,604	Out of memory
100	25	0.15	200	16,515	6,804	204	16,444	6,873	233	18,121	8,263	
100	30	0.15	29	2,211	1,028	29	2,211	980	30	2,422	1,063	
Average			156	13,333	2,800	167	13,491	2,942	176	14,893	3,319	
1,000	10	0	80	67,617	9,614	80	68,714	9,434	68	54,556	8,040	(NS)
1,000	15	0	94	79,366	9,051	101	85,869	9,910	107	91,222	10,324	(NS)
1,000	20	0	151	115,105	14,930	127	104,369	12,887	106	87,794	10,213	(NS)
1,000	25	0	163	115,143	(0.0001)	148	117,292	16,594	124	87,597	13,565	(NS)
1,000	30	0	26	19,508	2,931	26	19,508	2,755	49	30,516	5,268	(NS)
1,000	10	0.05	88	74,661	11,590	75	62,676	9,624	139	109,263	17,982	(NS)
1,000	15	0.05	85	74,357	8,718	86	75,401	8,644	175	138,924	(0.092)	(NS)
1,000	20	0.05	155	126,851	15,679	179	111,376	(0.001)	178	152,052	(0.015)	(NS)
1,000	25	0.05	159	125,186	(0.0025)	161	89,657	(0.04)	150	112,432	17,624	(NS)
1,000	30	0.05	40	27,770	4,472	72	27,818	7,777	39	26,908	4,370	(NS)
1,000	10	0.1	81	70,752	(0.54)	67	60,844	15,151	90	71,645	(0.45)	
1,000	15	0.1	87	74,448	(0.077)	84	74,524	(0.081)	86	71,963	(0.496)	
1,000	20	0.1	90	70,089	(0.146)	105	46,305	(0.891)	90	76,162	(0.123)	Out of memory
1,000	25	0.1	85	74,706	(0.03)	89	69,677	(0.05)	86	72,163	(0.04)	
1,000	30	0.1	42	27,961	8,287	59	27,978	10,752	69	38,991	13,465	
Average			95	76,235	9,475	97	69,467	10,353	104	81,479	11,206	

Table 5. Comparison of Different Algorithms With Respect to Computational Effectiveness (P-median1 Instances)

$|\Omega|$	p	λ	BD No. of iter.	BD No. of cuts	BD Sol. time	BD_SL No. of iter.	BD_SL No. of cuts	BD_SL Sol. time	BD_IC No. of iter.	BD_IC No. of cuts	BD_IC Sol. time	EF(SCSO) Sol. time
50	5	0	187	8,841	1,468	187	8,841	1,462	208	9,736	1,694	(NS)
50	7	0	152	7,244	982	152	7,244	985	177	8,261	1,170	(0.902)
50	8	0	126	5,960	937	127	6,010	929	152	7,086	1,138	(0.892)
50	11	0	68	2,691	403	68	2,691	398	68	2,729	419	(0.205)
50	12	0	59	2,180	408	59	2,180	400	44	1,745	310	8,510
50	5	0.05	238	11,320	1,988	236	11,224	1,930	191	8,891	1,615	(NS)
50	7	0.05	260	12,339	1,790	250	11,676	1,710	247	11,656	1,730	(0.706)
50	8	0.05	181	8,823	1,437	180	8,772	1,444	171	8,259	1,390	(0.774)
50	11	0.05	63	2,636	430	63	2,636	420	54	2,309	379	14,163
50	12	0.05	38	1,557	308	38	1,557	299	41	1,595	332	9,897
50	5	0.1	283	13,477	2,518	282	13,426	2,488	289	13,775	2,586	(NS)
50	7	0.1	300	14,284	2,202	316	15,021	2,345	327	15,425	2,444	(NS)
50	8	0.1	227	10,966	1,950	231	11,173	1,956	228	10,920	1,978	(0.565)
50	11	0.1	94	4,060	690	92	3,967	677	96	4,123	720	(0.071)
50	12	0.1	43	1,912	371	43	1,912	363	47	2,044	415	13,587
50	5	0.15	449	21,889	4,208	448	21,819	4,187	470	22,641	4,484	(NS)
50	7	0.15	531	25,493	4,175	494	23,786	3,834	506	24,419	3,910	(0.569)
50	8	0.15	385	18,855	3,424	329	16,176	2,881	339	16,243	3,056	(0.394)
50	11	0.15	121	5,497	897	120	5,458	888	102	4,650	762	16,416
50	12	0.15	57	2,593	493	57	2,593	490	56	2,580	500	9,729
50	5	0.2	546	26,286	5,431	555	26,716	5,515	531	25,462	5,365	(NS)
50	7	0.2	621	29,989	5,036	647	31,234	5,234	628	30,194	5,091	(NS)
50	8	0.2	480	23,369	4,523	476	23,199	4,472	470	22,716	4,455	(0.272)
50	11	0.2	191	8,740	1,454	183	8,395	1,393	194	8,887	1,505	13,172
50	12	0.2	106	4,825	938	119	5,386	1,055	132	5,869	1,198	11,512
Average			232	11,033	1,938	230	10,924	1,910	231	10,889	1,946	
100	5	0	171	16,333	2,762	171	16,333	2,695	173	15,933	2,872	(NS)
100	7	0	221	20,507	2,923	210	19,821	2,771	216	20,309	2,915	(NS)
100	8	0	117	11,539	1,781	119	11,747	1,784	168	15,184	2,622	(NS)
100	11	0	73	6,537	885	73	6,537	893	79	6,361	978	(NS)
100	12	0	47	3,655	689	47	3,655	645	41	3,058	591	(NS)
100	5	0.05	181	16,765	3,131	185	16,949	3,174	197	18,223	3,443	(NS)
100	7	0.05	240	23,025	3,397	256	24,385	3,634	214	20,162	3,115	(NS)
100	8	0.05	191	18,546	3,176	195	18,933	3,215	209	19,387	3,550	(NS)
100	11	0.05	90	7,558	1,250	84	7,261	1,156	74	6,127	1,049	(NS)
100	12	0.05	33	2,577	562	33	2,577	531	31	2,577	529	(NS)
100	5	0.1	249	23,705	4,541	253	24,111	4,592	283	26,579	5,248	(NS)
100	7	0.1	339	32,583	5,149	335	31,878	5,028	379	36,040	5,806	(NS)
100	8	0.1	354	33,936	6,204	333	32,511	5,868	333	31,604	6,034	(NS)
100	11	0.1	111	9,961	1,646	106	9,552	1,562	90	8,090	1,370	(NS)
100	12	0.1	49	3,953	854	49	3,953	848	52	4,204	934	(NS)
100	5	0.15	409	38,968	7,844	383	36,644	7,259	364	34,893	7,030	(NS)
100	7	0.15	512	49,409	8,210	470	45,482	7,486	502	48,281	7,990	(NS)
100	8	0.15	385	36,880	7,076	391	37,847	7,085	428	40,821	7,914	(NS)
100	11	0.15	116	10,760	1,735	119	10,984	1,765	117	10,771	1,787	(NS)
100	12	0.15	56	5,036	990	56	5,036	991	56	5,125	1,030	(NS)
100	5	0.2	413	39,601	8,359	420	40,291	8,485	465	44,233	9,581	(NS)
100	7	0.2	512	49,513	8,405	544	52,455	8,900	512	49,429	8,531	(NS)
100	8	0.2	514	49,155	9,869	509	48,812	9,860	521	50,433	10,308	(NS)
100	11	0.2	192	17,730	2,959	193	17,839	2,974	209	19,082	3,306	(NS)
100	12	0.2	102	9,435	1,881	98	9,057	1,816	122	10,626	2,272	(NS)
Average			197	18,612	3,240	193	18,310	3,149	200	18,686	3,340	(NS)

root node or with a big gap. When the number of scenarios is 100 and 1,000, EF hits the time limit of five hours for every instance with no solution at the root node. For the instances that EF does not hit the time limit, BD algorithms perform around nine (minimum), 37 (maximum), and 23 (average) times better than EF with respect to CPU times. BD algorithms outperform each other at different instances and hit the time limit for some of the instances when the number of scenarios is 1,000.

Table 6. Comparison of Different Algorithms With Respect to Computational Effectiveness (P-median1 Instances)

			BD			BD_SL			BD_IC			EF(SCSO)
$\|\Omega\|$	p	λ	No. of iter.	No. of cuts	Sol. time	No. of iter.	No. of cuts	Sol. time	No. of iter.	No. of cuts	Sol. time	Sol. time
1,000	5	0	106	95,538	(0.94)	107	99,711	(0.93)	102	97,485	(0.93)	(NS)
1,000	7	0	131	126,418	(0.844)	131	126,438	(0.842)	126	115,339	(0.833)	(NS)
1,000	8	0	114	105,536	(0.74)	114	106,677	(0.76)	110	95,456	(0.76)	(NS)
1,000	11	0	69	57,008	8,977	85	56,050	11,030	63	54,130	8,354	(NS)
1,000	12	0	43	35,423	6,748	43	35,423	6,422	51	38,104	7,877	(NS)
1,000	5	0.05	103	94,581	(0.90)	102	91,572	(0.85)	98	90,580	(0.86)	(NS)
1,000	7	0.05	125	115,522	(0.725)	123	114,530	(0.728)	118	110,492	(0.696)	(NS)
1,000	8	0.05	105	101,459	(0.63)	105	100,524	(0.65)	101	89,618	(0.64)	(NS)
1,000	11	0.05	88	70,131	12,912	91	73,820	13,169	53	48,062	7,909	(NS)
1,000	12	0.05	29	25,558	5,254	29	25,500	4,941	33	25,845	5,794	(NS)
1,000	5	0.1	96	82,540	(0.90)	96	59,999	(0.88)	93	84,506	(0.88)	(NS)
1,000	7	0.1	115	105,405	(0.721)	114	104,415	(0.721)	113	103,449	(0.647)	(NS)
1,000	8	0.1	98	94,691	(0.58)	100	96,658	(0.58)	95	88,532	(0.63)	(NS)
1,000	11	0.1	108	91,185	16,644	107	91,667	16,739	93	83,599	14,777	(NS)
1,000	12	0.1	45	37,479	8,432	45	37,479	8,132	46	38,462	8,631	(NS)
1,000	5	0.15	88	76,436	(0.76)	89	80,668	(0.84)	88	85,528	(0.84)	(NS)
1,000	7	0.15	112	106,416	(0.619)	111	105,363	(0.619)	108	102,596	(0.604)	(NS)
1,000	8	0.15	94	86,544	(0.46)	95	85,581	(0.46)	93	87,536	(0.64)	(NS)
1,000	11	0.15	116	106,023	(0.308)	116	104,133	(0.279)	114	104,296	(0.241)	(NS)
1,000	12	0.15	61	51,500	11,345	62	52,465	11,125	55	45,690	10,282	(NS)
1,000	5	0.2	80	68,481	(0.76)	86	72,391	(0.72)	81	78,411	(0.82)	(NS)
1,000	7	0.2	106	100,540	(0.519)	105	100,463	(0.52)	101	94,492	(0.613)	(NS)
1,000	8	0.2	82	79,596	(0.45)	89	86,679	(0.45)	87	81,536	(0.54)	(NS)
1,000	11	0.2	111	104,501	(0.285)	112	103,804	(0.291)	107	98,237	(0.319)	(NS)
1,000	12	0.2	87	79,860	(0.21)	95	87,321	(0.21)	91	78,484	(0.21)	(NS)
Average			92	82,565	15,762	93	81,364	14,544	86	77,577	14,571	(NS)

In Figure 1, we illustrate how the upper (UB) and lower (LB) bounds are updated across iterations for the *BD*, *BD_SL*, and *BD_IC* algorithms when the number of scenarios is 1,000, $\lambda = 0.2$, $p = 10$ and $\lambda = 0.05$, $p = 30$ for the Istanbul Anatolian and Istanbul European networks, respectively.

By solving the dual subproblem with a cutting plane approach, we aim to overcome the memory problems we encounter when the size of the network or the tolerance level gets larger. For small tolerance levels— i.e., when the number of feasible paths is small— pregenerating the feasible paths and constructing the constraints related to them generally brings an advantage in terms of the CPU times. As the tolerance level increases, the CPU times worsen and, at some point, memory issues are encountered. For that reason, in
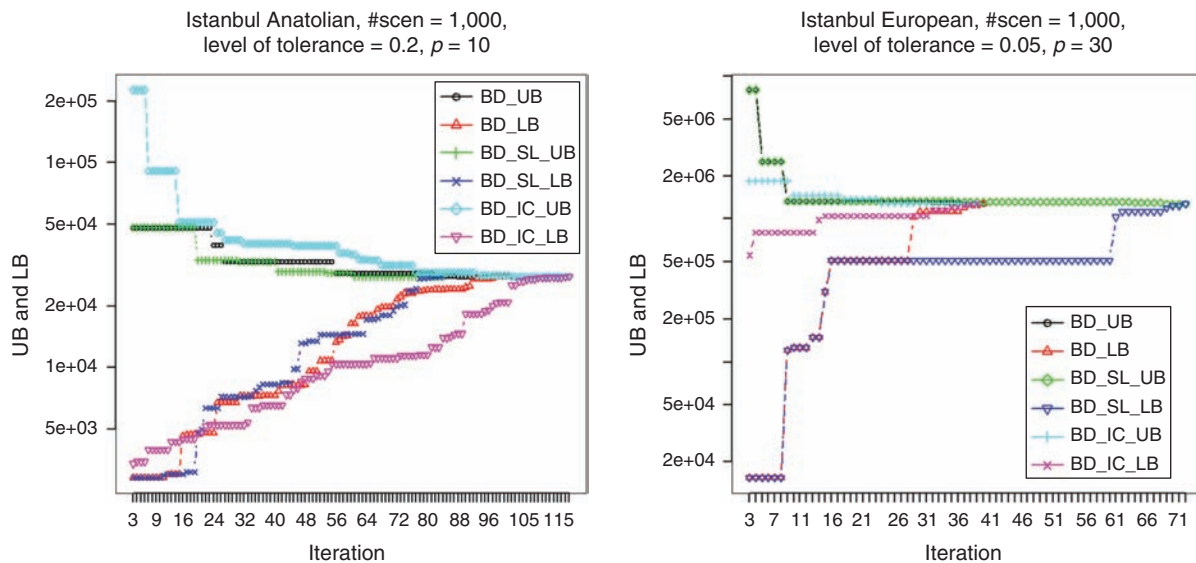
Figure 1. (Color online) Iteration vs. UB and LB for Istanbul Anatolian and European Networks

Table 7. The Results for the *BD_CP* Algorithm (Istanbul European)

$\mid\Omega\mid$	p	λ	No. of iter.	No. of cuts	Sol. time
100	20	0.2	144	13,156	15,362
100	25	0.2	164	12,475	5,361
100	30	0.2	56	3,503	1,265
1,000	20	0.15	151	110,795	(0.25)
1,000	25	0.15	92	73,843	36,011
1,000	30	0.15	47	29,501	15,254
1,000	20	0.2	69	55,536	(0.86)
1,000	25	0.2	170	118,537	71,511
1,000	30	0.2	38	27,395	12,969

Table 7, we report the results for the algorithm *BD_CP* for the instances that any one of the previously reported algorithms cannot solve because of memory issues. We report the results for $\lambda = 0.2$, $p = 20$, $p = 25$, $p = 30$ with $\mid\Omega\mid = 100$ and for $\lambda = 0.15$, $\lambda = 0.2$, $p = 20$, $p = 25$, $p = 30$ with $\mid\Omega\mid = 1,000$. All of these instances are solved to optimality within a 24-hour time limit, except for the instances with $\lambda = 0.15$, $\lambda = 0.2$, $p = 20$ with $\mid\Omega\mid = 1,000$. The gap that remains when the 24-hour time limit is reached is reported for these instances.

In summary, the BD algorithms perform better than each other on different instances; yet on average, the *BD_SL* algorithm performs better, especially for instances with a large number of scenarios, for the Istanbul Anatolian and P-median1 cases. On the other hand, *BD* performs better for the Istanbul European instances on average.

7.3. Capacitated Shelters

Since the nodes representing potential shelter sites in our network are not particular buildings but areas, capacities are not well defined in general. If the shelters have fixed capacities, then our model and solution approaches can be modified to handle capacity constraints. We add the capacity constraints

$$\sum_{r \in O} \sum_{\pi \in P_{rs}^{\lambda}(\omega)} w_r(\omega) v_{\pi}(\omega) \le K_s y_s, \quad \forall \omega \in \Omega, s \in F(\omega)$$

to our formulation, where K_s is the capacity of shelter s. Constraints (52) are not sufficient to guarantee feasibility. We change them to $\sum_{s \in \bar{F}_r(\omega)} K_s y_s \ge w_r(\omega)$, $\forall r \in O$, $\omega \in \Omega$ and add constraints $\sum_{s \in \bar{F}(\omega)} K_s y_s \ge \sum_{r \in O} w_r(\omega)$, $\forall \omega \in \Omega$ in the MP to decrease the number of infeasibilities, where $\bar{F}(\omega) = \bigcup_{r \in O} \bar{F}_r(\omega)$. We also need to add feasibility cuts in our decomposition algorithm.

In Table 8, we present the results for the Istanbul Anatolian instances when shelters are capacitated. We only report the instances for which we obtain a feasible solution. For each instance, we report the number of iterations of the BD algorithm performed, the number of optimality and feasibility cuts added, and the solution times. Clearly, the instances with small p and

Table 8. Results for the *BD* Algorithm When Shelters Are Capacitated (Istanbul Anatolian)

Ω	p	λ	No. of iter.	Opt. cuts	Feas. cuts	Sol. time
50	15	0	18	649	0	36.85
50	15	0.05	21	643	0	42.78
50	15	0.1	18	663	0	40.36
50	10	0.15	73	2,699	9	165.53
50	12	0.15	60	2,100	1	132.94
50	15	0.15	21	618	0	48.54
50	10	0.2	89	4,111	5	224.26
50	12	0.2	49	2,034	0	124.88
50	15	0.2	11	431	0	24.58
100	15	0.05	39	2,494	0	167.83
100	15	0.1	38	2,502	0	164.16
100	15	0.15	25	1,780	0	117.65
100	10	0.2	91	7,407	17	472.71
100	12	0.2	67	5,547	1	334.77
100	15	0.2	10	900	0	48.63
1,000	15	0.2	15	9,859	0	752.25

small tolerance levels turn out to be infeasible. As the number of scenarios increases, the number of feasible instances decreases. In 13 of the 16 instances that are feasible, the number of iterations, the number of optimality cuts added, and the solution time are smaller in the *BD* algorithm with capacitated shelters compared to the uncapacitated case.

8. Conclusion

For more realistic evacuation planning, one has to take into account the uncertainties regarding the evacuation demand, road network structure, and the possible disruption in shelters, and consider as many scenarios as needed.

In this study, we proposed an exact algorithm based on a BD of a formulation that aims to generate a fair and efficient evacuation plan. We employed duality results for SOCP in a BD setting. We developed different BD algorithms that can solve practical-size problems with up to 1,000 scenarios in moderate CPU times. We investigated methods such as adopting a multicut strategy, using a lazy constraint callback feature, deriving Pareto-optimal cuts, and using a preemptive priority multiobjective program to enhance the proposed algorithm. To avoid memory problems for large evacuation networks and larger tolerance levels, we also proposed an alternative algorithm in which we solve the dual subproblem in a cutting plane framework rather than generating all possible feasible paths. Computational results confirm the efficiency of our algorithm as it is considerably faster and can solve instances with a larger number of scenarios compared to solving the extended formulation with an off-the-shelf solver.

We proposed a strategic planning methodology that locates shelters to evacuate the population in danger as

quickly as possible. Our method also determines the assignment of evacuees to shelters and routes. Clearly, once the disaster hits, these assignments can be modified by the evacuation management authority by using real-time information and dynamic evacuation management models.

In our work, we assume that all of the evacuees comply with the guidance the central authority gives; yet this may not always be the case, and there may not be full compliance. The evacuees that do not comply with the evacuation orders may try to minimize their individual travel time while the evacuation authority aims to minimize the total evacuation time. An interesting extension would be a game-theoretic approach where some percentage of the evacuees in the network do not comply with the evacuation orders, and the central manager tries to find the best strategy to evacuate the disaster region as soon as possible.

Acknowledgments

References

Aktürk MS, Atamtürk A, Gürel S (2009) A strong conic quadratic reformulation for machine-job assignment with controllable processing times. *Oper. Res. Lett.* 37(3):187–191.

Alizadeh F, Goldfarb D (2003) Second-order cone programming. *Math. Programming* 95(1):3–51.

Álvarez-Miranda E, Fernández E, Ljubić I (2015) The recoverable robust facility location problem. *Transportation Res. Part B: Methodological* 79:93–120.

American Red Cross (2002) Standards for hurricane evacuation shelter selection. Technical report, ARC 4496, Appendix C.

Ang M, Lim YF, Sim M (2012) Robust storage assignment in unit-load warehouses. *Management Sci.* 58(11):2114–2130.

Atamtürk A, Berenguer G, Shen ZJ (2012) A conic integer programming approach to stochastic joint location-inventory problems. *Oper. Res.* 60(2):366–381.

Bayram V, Yaman H (2015) A stochastic programming approach for shelter location and evacuation planning. *RAIRO: Oper. Res.* Forthcoming.

Bayram V, Tansel BÇ, Yaman H (2015) Compromising system and user interests in shelter location and evacuation planning. *Transportation Res. Part B: Methodological* 72:146–163.

Beasley JE (1990) OR-Library P-median instances. http://people .brunel.ac.uk/~mastjjb/jeb/info.html.

Ben-Tal A, Nemirovski A (2001) *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS-SIAM Series Optim. (SIAM, Philadelphia).

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4(1): 238–252.

Birge JR, Louveaux F (1997) *Introduction to Stochastic Programming* (Springer, New York).

Bish DR, Sherali HD, Hobeika AG (2013) Optimal evacuation planning using staging and routing. *J. Oper. Res. Soc.* 65(1):124–140.

Bonami P, Lejeune MA (2009) An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Oper. Res.* 57(3):650–670.

Bretschneider S (2013) *Mathematical Models for Evacuation Planning in Urban Areas*, Lecture Notes Econom. Math. Systems, Vol. 659 (Springer, Berlin Heidelberg).

Bureau of Public Roads (1964) Traffic Assignment Manual. U.S. Department of Commerce, Washington, DC.

Byers TH, Waterman MS (1984) Technical note—Determining all optimal and near-optimal solutions when solving shortest path problems by dynamic programming. *Oper. Res.* 32(6):1381–1384.

Chiu YC, Zheng H, Villalobos J, Gautam B (2007) Modeling no-notice mass evacuation using a dynamic traffic flow optimization model. *IIE Trans.* 39(1):83–94.

CNN (2001) CNN live at daybreak—Escaping a hurricane. http:// transcripts.cnn.com/TRANSCRIPTS/0109/11/lad.02.html.

Contreras I, Cordeau J-F, Laporte G (2011) Benders decomposition for large-scale uncapacitated hub location. *Oper. Res.* 59(6): 1477–1490.

Cordeau J-F, Soumis F, Desrosiers J (2000) A Benders decomposition approach for the locomotive and car assignment problem. *Transportation Sci.* 34(2):133–149.

Cordeau J-F, Stojković G, Soumis F, Desrosiers J (2001) Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Sci.* 35(4):375–388.

Cova TJ, Johnson JP (2003) A network flow model for lane-based evacuation routing. *Transportation Res. Part A: Policy Practice* 37(7):579–604.

Desrochers M, Marcotte P, Stan M (1995) The congested facility location problem. *Location Sci.* 3(1):9–23.

Federal Emergency Management Agency (1988) Handbook for the seismic evaluation of buildings. Technical report, FEMA 310, Washington, DC.

Federal Emergency Management Agency (2006) Safe rooms and shelters: Protecting people against terrorist attacks. Technical report, FEMA 453, Washington, DC.

Federal Emergency Management Agency (2008) Design and construction guidance for community safe rooms. Technical report, FEMA p-361, second ed., Washington, DC.

Fischetti M, Lodi A (2003) Local branching. *Math. Programming* 98(1–3):23–47.

Fischetti M, Ljubić I, Sinnl M (2016) Benders decomposition without separability: A computational study for capacitated facility location problems. *Eur. J. Oper. Res.* 253(3):557–569.

Fischetti M, Salvagnin D, Zanette A (2010) A note on the selection of Benders cuts. *Math. Programming* 124(1–2):175–182.

Franca PM, Luna HPL (1982) Solving stochastic transportation-location problems by generalized Benders decomposition. *Transportation Sci.* 16(2):113–126.

Galindo G, Batta R (2013) Review of recent developments in OR/MS research in disaster operations management. *Eur. J. Oper. Res.* 230(2):201–211.

Geoffrion AM (1972) Generalized Benders decomposition. *J. Optim. Theory Appl.* 10(4):237–260.

Geoffrion AM, Graves GW (1974) Multicommodity distribution system design by Benders decomposition. *Management Sci.* 20(5):822–844.

Günlük O, Linderoth J (2008) Perspective relaxation of mixed integer nonlinear programs with indicator variables. Lodi A, Panconesi A, Rinaldi G, eds. *Integer Programming and Combinatorial Optimization*, Lecture Notes Comput. Sci., Vol. 5035 (Springer, Berlin Heidelberg), 1–16.

Hamacher HW, Tjandra SA (2002) Mathematical modelling of evacuation problems—A state of the art. *Pedestrian Evacuation Dynam.* 2002:227–266.

Hijazi H, Bonami P, Ouorou A (2013) Robust delay-constrained routing in telecommunications. *Ann. Oper. Res.* 206(1):163–181.

Huibregtse O, Hoogendoorn S, Bliemer MC (2010) Optimization of evacuation measures under uncertainty. *Transportation Res. Board 89th Annual Meeting, Washington, DC*, Report 10-2550.

IBM ILOG CPLEX (2011) ILOG CPLEX optimization studio 12.5.1. https://www.ibm.com/us-en/marketplace/ibm-ilog-cplex.

IMM-JICA (2002) The study on a disaster prevention/mitigation basic plan in Istanbul including seismic microzonation in the Republic of Turkey. Technical report, Istanbul Metropolitan Municipality and Japan International Cooperation Agency, Tokyo.

Jahn O, Möhring RH, Schulz AS, Stier-Moses NE (2005) System-optimal routing of traffic flows with user constraints in networks with congestion. *Oper. Res.* 53(4):600–616.

Kalafatas G, Peeta S (2009) Planning for evacuation: Insights from an efficient network design model. *J. Infrastructure Systems* 15(1):21–30.

Kongsomsaksakul S, Yang C, Chen A (2005) Shelter location-allocation model for flood evacuation planning. *J. Eastern Asia Society Transportation Stud.* 6(1):4237–4252.

Kulshrestha A, Wu D, Lou Y, Yin Y (2011) Robust shelter locations for evacuation planning with demand uncertainty. *J. Transportation Safety Security* 3(4):272–288.

Laporte G, Louveaux FV, van Hamme L (1994) Exact solution to a location problem with stochastic demands. *Transportation Sci.* 28(2):95–103.

Laporte G, Louveaux FV, van Hamme L (2002) An integer *L*-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Oper. Res.* 50(3):415–423.

Li ACY, Nozick L, Xu N, Davidson R (2012) Shelter location and transportation planning under hurricane conditions. *Transportation Res. Part E: Logist. Transportation Rev.* 48(4):715–729.

Li L, Jin M, Zhang L (2011) Sheltering network planning and management with a case in the Gulf Coast region. *Internat. J. Production Econom.* 131(2):431–440.

Lobo MS, Vandenberghe L, Boyd S, Lebret H (1998) Applications of second-order cone programming. *Linear Algebra Appl.* 284(1):193–228.

Magnanti TL, Wong RT (1981) Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Oper. Res.* 29(3):464–484.

Mak H-Y, Rong Y, Shen Z-JM (2013) Infrastructure planning for electric vehicles with battery swapping. *Management Sci.* 59(7):1557–1575.

Marano KD, Wald DJ, Allen TI (2010) Global earthquake casualties due to secondary effects: A quantitative analysis for improving rapid loss analyses. *Natl. Hazards* 52(2):319–328.

Martins de Sá E, Contreras I, Cordeau JF, Saraiva de Camargo R, de Miranda G (2015) The hub line location problem. *Transportation Sci.* 49(3):500–518.

McDaniel D, Devine M (1977) A modified Benders' partitioning algorithm for mixed integer programming. *Management Sci.* 24(3):312–319.

Naoum-Sawaya J, Elhedhli S (2013) An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Ann. Oper. Res.* 210(1):33–55.

Natarajan K, Sim M, Uichanco J (2010) Tractable robust expected utility and risk models for portfolio optimization. *Math. Finance* 20(4):695–731.

Nesterov Y, Nemirovskii AS, Ye Y (1994) *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies Applied Math., Vol. 13 (SIAM, Philadelphia).

Ng M, Waller ST (2010) Reliable evacuation planning via demand inflation and supply deflation. *Transportation Res. Part E: Logist. Transportation Rev.* 46(6):1086–1094.

Ng MW, Park J, Waller ST (2010) A hybrid bilevel model for the optimal shelter assignment in emergency evacuations. *Comput.-Aided Civil Infrastructure Engrg.* 25(8):547–556.

Noonan F, Giglio RJ (1977) Planning electric power generation: A nonlinear mixed integer model employing Benders decomposition. *Management Sci.* 23(9):946–956.

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

O'Driscoll P, Wolf R, Hampson R (2005) The evacuation worked, but created a highway horror. *USA Today* (September 26), http://usatoday30.usatoday.com/news/nation/2005-09-25-evacuation-cover_x.htm.

Özbay K, Yazıcı M, Iyer S, Li J, Ozguven E, Carnegie J (2012) Use of regional transportation planning tool for modeling emergency evacuation—Case study of northern New Jersey. *Transportation Res. Record: J. Transportation Res. Board* 2312:89–97.

Papadakos N (2008) Practical enhancements to the Magnanti–Wong method. *Oper. Res. Lett.* 36(4):444–449.

Pel AJ, Bliemer MCJ, Hoogendoorn SP (2012) A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. *Transportation* 39(1):97–123.

Perry RW (1979) Incentives for evacuation in natural disaster research based community emergency planning. *J. Amer. Planning Assoc.* 45(4):440–447.

Pınar MÇ (2013) Mixed-integer second-order cone programming for lower hedging of American contingent claims in incomplete markets. *Optim. Lett.* 7(1):63–78.

Prékopa A (1995) *Stochastic Programming*, Math. Appl., Vol. 324 (Springer, Dordrecht, Netherlands).

Rei W, Cordeau JF, Gendreau M, Soriano P (2009) Accelerating Benders decomposition by local branching. *INFORMS J. Comput.* 21(2):333–345.

Saharidis GKD, Ierapetritou MG (2010) Improving Benders decomposition using maximum feasible subsystem (MFS) cut generation strategy. *Comput. Chemical Engrg.* 34(8):1237–1245.

Saharidis GKD, Boile M, Theofanis S (2011) Initialization of the Benders master problem using valid inequalities applied to fixed-charge network problems. *Expert Systems Appl.* 38(6):6627–6636.

Saharidis GKD, Minoux M, Ierapetritou MG (2010) Accelerating Benders method using covering cut bundle generation. *Internat. Trans. Oper. Res.* 17(2):221–237.

Saito H, Murota K (2007) Benders decomposition approach to robust mixed integer programming. *Pacific J. Optim.* 3(1):99–112.

Santos L, Coutinho-Rodrigues J, Current JR (2007) An improved solution algorithm for the constrained shortest path problem. *Transportation Res. Part B: Methodological* 41(7):756–771.

See CT, Sim M (2010) Robust approximation to multiperiod inventory management. *Oper. Res.* 58(3):583–594.

Sheffi Y (1985) *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods* (Prentice-Hall, Englewood Cliffs, NJ).

Shen Z-JM, Pannala J, Rai R, Tsoi TS (2008) Modeling transportation networks during disruptions and emergency evacuations. Report, University of California Transportation Center, Berkeley.

Sherali HD, Lunday BJ (2013) On generating maximal nondominated Benders cuts. *Ann. Oper. Res.* 210(1):57–72.

Sherali HD, Zhu X (2008) Two-stage fleet assignment model considering stochastic passenger demands. *Oper. Res.* 56(2):383–399.

Sherali HD, Bae K-H, Haouari M (2013) An integrated approach for airline flight selection and timing, fleet assignment, and aircraft routing. *Transportation Sci.* 47(4):455–476.

Sherali HD, Carter TB, Hobeika AG (1991) A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. *Transportation Res. Part B: Methodological* 25(6):439–452.

So SK, Daganzo CF (2010) Managing evacuation routes. *Transportation Res. Part B: Methodological* 44(4):514–520.

Stepanov A, Smith JM (2009) Multi-objective evacuation routing in transportation networks. *Eur. J. Oper. Res.* 198(2):435–446.

Tang L, Jiang W, Saharidis GKD (2013) An improved Benders decomposition algorithm for the logistics facility location problem with capacity expansions. *Ann. Oper. Res.* 210(1):165–190.

Taylor JA, Hover FS (2012) Convex models of distribution system reconfiguration. *IEEE Trans. Power Systems* 27(3):1407–1413.

Transportation Research Board (2008) The role of transit in emergency evacuation, Special Report 294, Washington, DC, www.trb.org/main/blurbs/160047.aspx.

Tüydeş H (2005) Network traffic management under disaster conditions. Unpublished doctoral thesis, Field of Civil and Environmental Engineering, Northwestern University, Evanston, IL.

Üster H, Kewcharoenwong P (2011) Strategic design and analysis of a relay network in truckload transportation. *Transportation Sci.* 45(4):505–523.

Van Roy TJ (1986) A cross decomposition algorithm for capacitated facility location. *Oper. Res.* 34(1):145–163.

Van Slyke RM, Wets R (1969) L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.* 17(4):638–663.

Xie C, Lin D-Y, Waller ST (2010) A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. *Transportation Res. Part E: Logist. Transportation Rev.* 46(3):295–316.

Yamada T (1996) A network flow approach to a city emergency evacuation planning. *Internat. J. Systems Sci.* 27(10):931–936.

Yao T, Mandala SR, Chung BD (2009) Evacuation transportation planning under uncertainty: A robust optimization approach. *Networks Spatial Econom.* 9(2):171–189.

Yazıcı MA, Özbay K (2010) Evacuation network modeling via dynamic traffic assignment with probabilistic demand and capacity constraints. *Transportation Res. Record: J. Transportation Res. Board* 2196(1):11–20.

Yen JY (1971) Finding the *k* shortest loopless paths in a network. *Management Sci.* 17(11):712–716.