Two-Stage Stochastic Programming Involving CVaR with an Application to Disaster Management

Nilay Noyan

Manufacturing Systems/Industrial Engineering Program, Sabancı University, 34956 Istanbul, Turkey,

nnoyan@sabanciuniv.edu

Abstract: Traditional two-stage stochastic programming is risk-neutral; that is, it considers the expectation as the preference criterion while comparing the random variables (e.g., total cost) to identify the best decisions. However, in the presence of variability risk measures should be incorporated into decision making problems in order to model its effects. In this study, we consider a risk-averse two-stage stochastic programming model, where we specify the conditional-value-at-risk (CVaR) as the risk measure. We construct two decomposition algorithms based on the generic Benders-decomposition approach to solve such problems. Both single-cut and multicut versions of the proposed decomposition algorithms are presented. We apply the proposed framework to disaster management, which is one of the research fields that can significantly benefit from risk-averse two-stage stochastic programming models. In particular, we consider the problem of determining the response facility locations and the inventory levels of the relief supplies at each facility in the presence of uncertainty in demand and the damage level of the disaster network. We present numerical results to discuss how incorporating a risk measure affects the optimal solutions and to demonstrate the computational efficiency of the proposed methods.

Keywords: two-stage stochastic programming; conditional-value-at-risk; decomposition; facility location; emergency supplies; disaster relief

Introduction Traditional two-stage stochastic programming considers the expectation as the preference criterion while comparing the random variables to find the best decisions; hence, it is a risk neutral approach. Two-stage stochastic programming with the expected recourse function has been applied in a wide range of applications. For different applications, we refer to Prékopa (1995), Birge and Louveaux (1997), and references therein. We can justify the optimization of the expected value by the Law of Large Numbers for situations, where the same decisions under similar conditions are made repeatedly. Thus, the typical objective of minimizing the expected total cost may yield solutions that are good in the long run, but such results may perform poorly under certain realizations of the random data (see, e.g., Ruszczyński and Shapiro (2006)). Therefore, for non-repetitive decision making problems under uncertainty a risk-averse approach that considers the effects of the variability of random outcomes, such as the random total cost, would provide more robust solutions compared to a risk-neutral approach. Tactical and/or strategic decisions such as location planning and network design constitute examples of non-repetitive decisions.

Incorporating risk measures in the objective functions in the framework of two-stage stochastic programming is a fairly recent research topic (Ahmed, 2004; 2006; Schultz and Tiedemann, 2006; Fábián, 2008; Miller and Ruszczyński, 2009). Ahmed (2004) and Ahmed (2006) introduce the two-stage stochastic programming with mean-risk objectives. Ahmed (2006) investigates several risk measures for which the mean-risk function preserves the convexity and thus, leads to computationally tractable methods. The author focuses on two classes of risk measures; the first class involves the absolute semideviation risk measure and the second one involves the quantile deviation risk measure, which is related to CVaR. Slight variants of the existing decomposition methods are applicable to solve the proposed problems, because the models do not involve any integer variables. Schultz and Tiedemann (2006) focus on two-stage mixed-integer stochastic programming involving CVaR. In their model, the integer variables appear in the second-stage problem, and therefore, straightforward decomposition methods cannot be applied due to the non-convexity of the problem. They consider the split-variable formulation and develop a solution algorithm based on the Lagrangian relaxation of nonanticipativity. Recently, Miller and Ruszczyński (2009) introduce a new type of risk-averse two-stage stochastic linear programming model, where the uncertainty is still present after the second-stage decisions are made. Due to the proposed extended structure, the authors consider compositions of conditional risk measures and develop variants of the existing decomposition methods, one of which is based on the multicut approach. We refer to Fábián (2008) for different types of two-stage stochastic programming models involving CVaR. We also note that Künzi-Bay and Mayer (2006) reformulate the single-stage CVaR minimization problem as a two-stage stochastic programming and propose a new algorithm, which is a variant of the L-shaped method and it is based on the optimality cuts proposed by Klein Haneveld and Van der Vlerk (2006) for integrated chance constraints. This study is not directly related to ours, because Künzi-Bay and Mayer (2006) consider a single-stage stochastic programming problem involving CVaR.

In this paper, we consider the risk-averse two-stage stochastic programming framework similar to those of Ahmed (2006) and Schultz and Tiedemann (2006). We characterize the inherent randomness by a finite set of scenarios and specify CVaR as the risk measure in the two-stage mean-risk analysis. Different than Schultz and Tiedemann (2006) we do not consider any integer variables in the second-stage problem and therefore, the standard decomposition (cutting plane) methods apply in our study as in Ahmed (2006). The *L-Shaped method* proposed by Van Slyke and Wets (1969) is a widely-applied Benders-decomposition approach (Benders, 1962) to solve the two-stage stochastic programming problems with the expected recourse functions for the case of a finite probability space. For a detailed discussion, we refer to Van Slyke and Wets (1969), Prékopa (1995), Birge and Louveaux (1997). We reformulate the two-stage stochastic programming problem with the CVaR measure on the total cost as a large scale linear programming problem and in order to solve it we develop an algorithm, which is a variant of the L-shaped decomposition method. As an alternate solution algorithm, the generic algorithm proposed by Ahmed (2006) is adapted to CVaR. Such decomposition algorithms allow us to handle a large number of scenarios, which is crucial in obtaining a good representation of the randomness. We extend both proposed decomposition algorithms by replacing the single-cuts by their multicut versions.

Risk-averse two-stage stochastic programming with mean-risk functions has a significant potential to be applied in different fields. Since it is a fairly recent development, it has only been applied to just a few problems; a chemical engineering problem (Schultz and Tiedemann, 2006), energy optimization problems (see, e.g., Schultz and Neise (2007)) and a transportation network protection problem (Liu et al., 2009). In this study, we apply the proposed framework to disaster management that is one of the research fields that can significantly benefit from risk-averse two-stage models. Several two-stage stochastic programming problems are proposed for disaster management (Barbarosoğlu and Arda, 2004; Balcik and Beamon, 2008; Rawls and Turnquist, 2010). However, all such existing models focus on the expected values. To the best of our knowledge, developing a risk-averse two-stage stochastic model in the disaster management literature is novel. In particular, we consider the problem of determining the response (storage) facility locations and the inventory levels of the relief supplies at each facility in the presence of uncertainty in demand and the damage level of the disaster network. We assume that when a disaster occurs the roads and facilities may be damaged and therefore, transportation capacities and the available amount of preallocated supplies may vary according to the severity of the disaster. We focus on an existing two-stage stochastic programming formulation of a disaster location-allocation model that has been recently proposed by Rawls and Turnquist (2010) and we extend it by incorporating the CVaR as the risk measure on the total cost. Our proposed model is a significant and non-trivial extension, because it incorporates a risk measure in addition to the expectation criterion. It has been quite popular in different fields to develop models involving CVaR (e.g., see (Rockafellar and Uryasev, 2002; Chen et al., 2006; Schultz and Tiedemann, 2006)). Solving the proposed problem for different risk parameters provides alternate location and allocation decisions and decision makers can evaluate those decisions with respect to costs, the quality of service and also their risk preferences.

The contributions of this study are *(i)* developing decomposition algorithms that are specific for CVaR in the two-stage mean-risk stochastic programming, *(ii)* introducing a risk-averse two-stage disaster preparedness model. The proposed general framework can be applied to other problems, and this approach would provide the decision makers with risk-averse decisions that would perform better in the presence of variability compared to a risk-neutral approach. The paper is organized as follows. In Section 1, we discuss how to model risk in a two-stage stochastic programming framework in general. In Section 2, we first develop single-cut decomposition algorithms for the proposed model and then describe their multicut versions. In Section 3, we present the existing disaster location-allocation model under the expected total cost criterion and then introduce our risk-averse model. In Section 4, we present computational results for the case study given in Rawls and Turnquist (2010) to illustrate how the location and allocation solutions change with respect to different risk preferences. In addition, numerical results for a number of problem instances are provided to demonstrate the computational efficiency of the proposed solution methods. We conclude the paper in Section 5.

1. Two-Stage Mean-Risk Stochastic Programming Framework

An abstract probability space is denoted by $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is a σ-algebra on Ω and P is a probability measure on Ω. We consider the case of a finite probability space, where $\Omega = \{\omega_1, \ldots, \omega_N\}$ with corresponding probabilities $p_1, \ldots, p_N$.

The general form of a two-stage stochastic programming problem reads:

$$f(\mathbf{x}, \omega) = \{\mathbf{c}^T\mathbf{x} + Q(\mathbf{x}, \xi(\omega))\}, \tag{1}$$

where $f(\mathbf{x}, \omega)$ is the objective function of the first-stage problem for a decision vector $\mathbf{x} \in X$ and

$$Q(\mathbf{x}, \xi^i) = \min_{\mathbf{y}_i} \ \{\mathbf{q}_i^T\mathbf{y}_i \ : T_i\mathbf{x} + W_i\mathbf{y}_i = \mathbf{h}_i, \ \mathbf{y}_i \geq 0\} \tag{2}$$

is the second-stage problem corresponding to the realization of the random data $\xi(\omega)$ for the elementary event

ω_i, denoted by $\xi^i = (\mathbf{q}_i, T_i, W_i, \mathbf{h}_i)$. Here $\mathbf{x}$ and $\mathbf{y}$ are the vectors of first-stage and second-stage decision variables, respectively. Suppose that all the matrices have the appropriate dimensions. We assume that $X \subset \mathbb{R}^n_+$ is a non-empty set of feasible decisions and $Q(\mathbf{x}, \xi(\omega)) > -\infty$ for all $\omega \in \Omega$. Note that the first-stage decisions are deterministic and the second-stage decisions are allowed to depend on the elementary events, i.e., $\mathbf{y}_i = \mathbf{y}(\omega_i),\ i = 1, \ldots, N$. Basically, the second-stage decisions represent the operational decisions, which change depending on the realized values of the random data. The objective function $Q(\mathbf{x}, \xi(\omega))$ of the second-stage problem (2), also known as the recourse (cost) function, is a random variable and therefore, the total cost function $f(\mathbf{x}, \omega)$ is a random variable. Determining the optimal decision vector $\mathbf{x}$ leads to the problem of comparing random cost variables $\{f(\mathbf{x}, \omega)\}_{\mathbf{x} \in X}$. Comparing random variables is one of the main interests of decision theory in the presence of uncertainty. Since we focus on the total cost, smaller values of $f(\mathbf{x}, \omega)$ are preferred. While comparing random variables it is crucial to consider the effect of variability, which leads to the concept of risk. The preference relations among random variables can be specified using a risk measure. One of the main approaches in the practice of decision making under risk uses mean-risk models. In these models one minimizes the mean-risk function, which involves a specified risk measure $\rho : \mathcal{Z} \to \mathbb{R}$, where ρ is a functional and $\mathcal{Z}$ is a linear space of $\mathcal{F}$-measurable functions on the probability space $(\Omega, \mathcal{F}, P)$:

$$\min_{\mathbf{x} \in X} \left\{ \mathbb{E}\left[f(\mathbf{x}, \omega)\right] + \lambda \rho \left[f(\mathbf{x}, \omega)\right] \right\}. \tag{3}$$

In this approach, λ is a nonnegative trade-off coefficient representing the exchange rate of mean cost for risk. We also refer to it as a risk coefficient, which is specified by decision makers according to their risk preferences. The well-known model by Markowitz (1952) uses the variance as the risk measure. One of the problems associated with this mean-variance formulation is that it treats under-and-over-performance equally. Usually, when typical dispersion statistics such as variance are used as risk measures, the mean-risk approach may lead to inferior solutions (see Remark 1.1). In order to remedy this drawback, models with alternative asymmetric risk measures such as downside risk measures have been proposed (see e.g., Ogryczak and Ruszczyński (2002)). Among the popular downside risk measures we focus on a recently popular and widely-applied risk measure called Conditional Value-at-Risk (CVaR). For alternate risk measures that lead to computationally tractable two-stage mean-risk models, we refer to Ahmed (2006).

In our study, we say that the decision vector $\mathbf{x}$ is efficient (in the mean-risk sense) if and only if for a given level of minimum expected cost, $f(\mathbf{x}, \omega)$ has the lowest possible CVaR, and for a given level of CVaR it has the lowest possible expected cost. One can construct the mean-risk efficient frontier by finding the efficient solutions for different risk parameters.

Definition 1.1 *Let $F_Z(\cdot)$ represent the cumulative distribution function of a random variable Z. In the financial literature, the α-quantile*

$$\inf\{\eta : \ F_Z(\eta) \geq \alpha\}$$

is called the Value at Risk (VaR) at the confidence level α and denoted by $\mathrm{VaR}_\alpha(Z)$*, $\alpha \in (0, 1]$.*

The Conditional Value-at-Risk (CVaR), also called Mean Excess Loss or Tail VaR, at level α is, in a simple way, defined as follows (Rockafellar and Uryasev, 2000):

$$\mathrm{CVaR}_\alpha(Z) = \mathbb{E}(Z \mid Z \geq \mathrm{VaR}_\alpha(Z)). \tag{4}$$

This definition provides a clear understanding of the concept of CVaR: $\mathrm{CVaR}_\alpha(Z)$ is the conditional expected value exceeding the Value at Risk at the confidence level α. In the cost minimization context, VaR_α is the α-quantile of the distribution of the cost and it provides an upper bound that is exceeded only with a small probability of $1-\alpha$. On the other hand, $\mathrm{CVaR}_\alpha(Z)$ is a measure of severity of the cost if it is more than $\mathrm{VaR}_\alpha(Z)$. The formula (4) is precise if $\mathrm{VaR}_\alpha(Z)$ is not an atom of the distribution of Z. A more precise description is given in the next definition (Rockafellar and Uryasev, 2000; 2002).

Definition 1.2 *The conditional-value-at-risk of a random variable Z at the confidence level α is given by*

$$CVaR_\alpha(Z) = \inf_{\eta\ \in\mathbb{R}} \left\{\eta + \frac{1}{1-\alpha}\mathbb{E}([Z-\eta]_+)\right\}, \tag{5}$$

where we let $[z]_+ = \max(0,z)$, $z \in \mathbb{R}$. It is well-known that the infimum in (5) is attained at a α-quantile of Z (see, e.g., Ogryczak and Ruszczyński (2002)). Our first proposed algorithm utilizes this result while calculating the CVaR value for the realized values of the recourse function, $Q(\mathbf{x},\xi^i)$, $i=1,\ldots,N$, associated with a candidate first-stage decision vector $\mathbf{x}$.

Proposition 1.1 *For the case of a finite probability space, where $\Omega = \{\omega_1,\ldots,\omega_N\}$ with corresponding probabilities $p_1,\ldots,p_N$, we can equivalently reformulate the mean-risk problem*

$$\min_{\mathbf{x}\in X} \{\mathbb{E}[f(\mathbf{x},\omega)] + \lambda \mathrm{CVaR}_\alpha(f(\mathbf{x},\omega))\} \tag{6}$$

as the following linear programming problem

$$\begin{aligned}
\min \quad & (1+\lambda)\mathbf{c}^T\mathbf{x} + \sum_{i=1}^N p_i q_i^T \mathbf{y}_i + \lambda(\eta + \frac{1}{1-\alpha}\sum_{i=1}^N p_i v_i) \\
\textit{subject to} \quad & W_i\mathbf{y}_i = h_i - T_i\mathbf{x}, \quad i=1,\ldots,N, \\
& \mathbf{x}\in X, \\
& \mathbf{y}_i \geq 0,\ i=1,\ldots,N, \\
& v_i \geq \mathbf{q}_i^T\mathbf{y}_i - \eta, \quad i=1,\ldots,N, \\
& \eta\in\mathbb{R}, v_i \geq 0,\ i=1,\ldots,N.
\end{aligned}$$

Proof. It is easy to show that (see, eg., (G.Pflug, 2000; Birbil et al., 2009)) CVaR_α is translation invariant, i.e.,

$$\mathrm{CVaR}_\alpha(Z+a) = \mathrm{CVaR}_\alpha(Z) + a$$

for $a\in\mathbb{R}$ and $Z\in\mathcal{Z}$. Then, we have

$$\mathrm{CVaR}_\alpha(f(\mathbf{x},\omega)) = \mathbf{c}^T\mathbf{x} + \mathrm{CVaR}_\alpha(Q(\mathbf{x},\xi(\omega)))$$

and

$$\begin{aligned}
\mathbb{E}[f(\mathbf{x},\omega)] + \lambda\mathrm{CVaR}_\alpha(f(\mathbf{x},\omega)) &= \mathbf{c}^T\mathbf{x} + \mathbb{E}[Q(\mathbf{x},\xi(\omega))] + \lambda\left(\mathbf{c}^T\mathbf{x} + \mathrm{CVaR}_\alpha(Q(\mathbf{x},\xi(\omega)))\right) \\
&= (1+\lambda)\mathbf{c}^T\mathbf{x} + \mathbb{E}[Q(\mathbf{x},\xi(\omega))] + \lambda\mathrm{CVaR}_\alpha(Q(\mathbf{x},\xi(\omega))).
\end{aligned} \tag{7}$$

For the given finitely many realizations of the random data $\xi(\omega)$, we have the realizations of the recourse function as $Q(\mathbf{x},\xi^1) = q_1^T\mathbf{y}_1,\ldots,Q(\mathbf{x},\xi^N) = q_N^T\mathbf{y}_N$. Then by the description of the second-stage problem (2) and by Definition (1.2), the assertion follows. Note that we can interpret the variable η as a first-stage variable and the excess variables, $v_i, i = 1,\ldots,N$, as second-stage variables. □

For a finite probability space, reformulating a risk-neutral linear two-stage stochastic programming problem without integer restrictions as a large-scale linear programming problem is well-known (see, e.g., (Prékopa, 1995; Birge and Louveaux, 1997)). Proposition 1.1 is significant to show that the two-stage stochastic programming problem involving CVaR can also be formulated as a linear programming problem. A similar formulation is also presented in Schultz and Tiedemann (2006).

Remark 1.1 ***Consistency with the second order stochastic dominance (*SSD*).*** *The stochastic dominance relations are fundamental and widely-applied concepts in comparing random variables. For a review on stochastic dominance rules and other types of measures to compare random variables, we refer the reader to Shaked and Shanthikumar (1994), Müller and Stoyan (2002) and the references therein. We say that the mean-risk model* $(E[Z],\rho[Z])$ *is consistent with* SSD *if the following relation holds:*

$$Z_1 \text{ dominates } Z_2 \text{ in the second order } \Rightarrow E[Z_1] \leq E[Z_2] \textit{ and } \rho[Z_1] \leq \rho[Z_2].$$

The mean-risk model is in general not consistent with stochastic dominance rules. For example, when the variance is used as a risk measure the mean-risk model is not consistent with SSD. Ogryczak and Ruszczyński (1999; 2002) have studied the consistency of mean-risk models with SSD relation and they have proved that the mean-risk model with CVaR is consistent with SSD. This result emphasizes the importance of the mean-risk models involving CVaR as a risk measure.

It is easy to show that $Q(\mathbf{x},\xi(\omega))$ is convex in $\mathbf{x}$ for all $\omega \in \Omega$, and so is $f(\mathbf{x},\omega)$. Moreover, for a given $\alpha \in (0,1)$ the mean-risk function in (6) is convexity preserving for all $\lambda \geq 0$ (see Ahmed (2006)). For the second-stage problems the duality theory can be used to obtain outer approximations of the mean-risk function of the recourse cost in (7). Constructing the outer approximations of the recourse function is basically the one of the main ideas of the *L-Shaped method* proposed by Van Slyke and Wets (1969). In this study we develop two variants of the L-shaped method to solve the two-stage stochastic programming problem (6) involving the CVaR measure on the random total cost.

2. Decomposition Algorithms

In this section we first present two decomposition algorithms to solve problem (6). Both algorithms are variants of the L-Shaped method; the optimality (objective) cuts are generated to approximate the optimal mean-risk function of the recourse cost, instead of just the expected recourse cost. Initially, we describe the algorithms based on the single-cut method, which aggregates the dual information of the second-stage problem over all the realizations of the random data to generate optimality cuts. In the first algorithm, we introduce additional variables to construct the optimality cuts, while in the second algorithm we avoid introducing new variables by using the subgradients of the recourse function. We extend both of the proposed decomposition algorithms by replacing the single-cuts by their multicut versions. The multicut approach disaggregates the optimality cuts to avoid the loss of the dual information associated with each realization of the second-stage problem. Since CVaR is only involved in the objective function and second-stage decision variables are all continuous, the feasibility cuts used for the expectation case directly apply here as well.

2.1 Basic Decomposition Algorithm for CVaR In this algorithm, we generate the tth optimality cut associated with CVaR of the recourse cost by introducing new variables η_t and $\nu_{it},\ i = 1, \ldots, N$. At an iteration of the algorithm, if the second-stage problems for the current first-stage solution and all the realizations of $\xi(\omega)$ are feasible, we solve all the second-stage problems to optimality. Suppose that at the current iteration, we construct the tth optimality cut and $\mathbf{u}_{it}$ is the dual vector associated with the optimal basis of the second-stage problem for realization ξ^i. Then, $\mathbf{u}_{it}^T(h_i - T_i\mathbf{x})$ is the optimal recourse function value for the realization ξ^i and the CVaR value of the recourse cost, is given by

$$\text{CVaR}_\alpha(Q(\mathbf{x}, \xi(\omega))) = \inf_{\eta_t\ \in \mathbb{R}} \Big\{\eta_t + \frac{1}{1-\alpha}\mathbb{E}[\max\{\mathbf{u}_{it}^T(h_i - T_i\mathbf{x}) - \eta_t, 0\}]\Big\}.$$

In the master problem of the decomposition algorithm, the new variable η_t is introduced to represent the α-quantile of the recourse cost, while the new variables $\nu_{it},\ i = 1, \ldots, N$, are introduced to calculate $\max\{\mathbf{u}_{it}^T(h_i - T_i\mathbf{x}) - \eta_t, 0\}$. By considering the CVaR values of the recourse cost associated with the first-stage solutions obtained so far, we can iteratively improve the lower bound on the optimal mean-risk function of the total cost. The optimality cuts are constructed by aggregating the dual information of the second-stage problem over all the realizations of the random data. Thus, the master problem of this decomposition algorithm becomes

$$\begin{aligned}
&\min && (1+\lambda)\mathbf{c}^T\mathbf{x} + \theta_1 + \lambda\theta_2 && (8)\\
&\text{subject to} && \mathbf{x} \in X, && (9)\\
&\text{optimality cuts} && \theta_1 \geq \sum_{i=1}^{N} p_i \mathbf{u}_{it}^T(h_i - T_i\mathbf{x}), \quad t = 1, \ldots, \tau, && (10)\\
&\text{optimality cuts} && \theta_2 \geq \eta_t + \frac{1}{1-\alpha}\sum_{i=1}^{N} p_i \nu_{it}, \quad t = 1, \ldots, \tau, && (11)\\
& && \nu_{it} \geq \mathbf{u}_{it}^T(h_i - T_i\mathbf{x}) - \eta_t, \quad i = 1, \ldots, N,\ t = 1, \ldots, \tau, && (12)\\
& && \nu_{it} \geq 0, \quad i = 1, \ldots, N,\ t = 1, \ldots, \tau, && (13)\\
& && \eta_t \in \mathbb{R}, \quad t = 1, \ldots, \tau, && (14)\\
&\text{feasibility cuts} && (T_i^T\sigma^{s_i}\mathbf{x} - \mathbf{h}_i^T\sigma^{s_i}) \geq 0, \quad i = 1, \ldots, N,\ s_i = 1, \ldots, S_i. && (15)
\end{aligned}$$

Here τ and S_i denote the number of optimality and feasibility cuts constructed up to the current iteration, respectively. σ^{s_i} denotes the dual vector corresponding to the optimal basis of the feasibility problem of the second-stage problem for realization ξ^i and the s_ith feasibility cut. For the details of the feasibility problem and the feasibility cuts we refer to Birge and Louveaux (1997) and Prékopa (1995). Notice that we solve the master problem $\sum\limits_{i=1}^{N} S_i + \tau$ times to obtain the formulation above.

In the formulation of the master problem, we approximate the expected recourse function and the associated CVaR value separately. Thus, the aggregate dual information is used to generate two single-cuts; one for the expectation term and one for the CVaR term. Alternatively, we can approximate

$\mathbb{E}[Q(\mathbf{x},\xi(\omega))] + \lambda \mathrm{CVaR}_\alpha\left(Q(\mathbf{x},\xi(\omega))\right)$ by introducing $\hat{\theta} = \theta_1 + \lambda\theta_2$ and the combined optimality cuts

$$\hat{\theta} \geq \sum_{i=1}^{N} p_i \mathbf{u}_{it}^T (h_i - T_i\mathbf{x}) + \lambda(\eta_t + \frac{1}{1-\alpha}\sum_{i=1}^{N} p_i \nu_{it}), \quad t = 1,\ldots,\tau. \tag{16}$$

In this case, a single-cut is generated for the combination of the expectation and the CVaR terms. However, as we observed in our preliminary experiments, it is in general computationally more efficient to consider the separate optimality cuts presented in (10) and (11).

The description of this decomposition algorithm is presented in Algorithm 1. The correctness and finiteness of the algorithm can be proved similar to those of the traditional L-shaped algorithm (Van Slyke and Wets, 1969; Birge and Louveaux, 1997; Prékopa, 1995).

Algorithm 1 Decomposition Algorithm 1

```
1: Initialize $\tau = 0,\ S_i = 0,\ i = 1,\ldots,N$.
2: Solve the master problem (8)-(15). Let $(\bar{\mathbf{x}}, \bar{\theta}_1, \bar{\theta}_2)$ be the optimal solution and $\bar{\theta} = \bar{\theta}_1 + \lambda\bar{\theta}_2$. (When $\tau = 0$ ignore
   $\theta_1$, $\theta_2$, and the optimality cuts. When $S_i = 0$ ignore the feasibility cuts.)
3: Check whether second-stage problems are all feasible for $\bar{\mathbf{x}}$ by solving the following feasibility problem:
   $$\{\min \sum_j (v_j^+ + v_j^-) : W_i\mathbf{y}_i + \mathbf{v}^+ - \mathbf{v}^- = h_i - T_i\bar{\mathbf{x}},\ \mathbf{y}_i \geq 0, \mathbf{v}^+ \geq 0, \mathbf{v}^- \geq 0\}. \tag{17}$$
   If the optimal objective function value of (17) is zero for all $i = 1,\ldots,N$, then all the second-stage problems
   are feasible and go to Step 4. Otherwise pick an infeasible problem, say $i$th problem. Then let $S_i = S_i + 1$ and
   introduce the feasibility cut for the dual vector $\sigma^{S_i}$ associated with the optimal basis in problem (17).
   Repeat Steps 2 and 3 until all second-stage problems are feasible.
4: Solve all second-stage problems:
   $$\{\min \mathbf{q}_i^T\mathbf{y}_i : W_i\mathbf{y}_i = \mathbf{h}_i - T_i\bar{\mathbf{x}},\ \mathbf{y}_i \geq 0\}. \tag{18}$$
   Let $\tau = \tau + 1$ and $\mathbf{u}_{i\tau}$ denote the dual vector corresponding to the optimal basis in problem (18).
5: Observe that $\mathbf{u}_{i\tau}^T(\mathbf{h}_i - T_i\bar{\mathbf{x}}),\ i = 1,\ldots,N$, are the realizations of $Q(\bar{\mathbf{x}},\omega)$.
   Find the $\alpha$-quantile of $Q(\bar{\mathbf{x}},\omega)$, denoted by $\eta_\alpha$, and calculate $\mathrm{CVaR}_\alpha\left(Q(\bar{\mathbf{x}},\omega)\right)$:
   $$\mathrm{CVaR}_\alpha\left(Q(\bar{\mathbf{x}},\omega)\right) = \eta_\alpha + \frac{1}{1-\alpha}\left(\sum_{i=1}^{N} p_i * \max\{Q(\bar{\mathbf{x}},\omega_i) - \eta_\alpha, 0\}\right).$$
6: Calculate the mean-risk function value of the recourse cost at the current solution:
   $$\theta^* = \sum_{i=1}^{N} p_i \mathbf{u}_{i\tau}^T(\mathbf{h}_i - T_i\bar{\mathbf{x}}) + \lambda\mathrm{CVaR}_\alpha\left(Q(\bar{\mathbf{x}},\omega)\right).$$
7: if $\bar{\theta} \geq \theta^*$ then
8:     Stop. $\bar{\mathbf{x}}$ is the optimal first-stage decision vector.
9: else
10:    Introduce the optimality cuts for $\mathbf{u}_{i\tau}$ and go to Step 2.
11: end if
```

Remark 2.1 *We relax the stopping condition $\bar{\theta} \geq \theta^*$ (Step 7) and terminate the algorithm with a positive optimality gap. Let $(\bar{\mathbf{x}}, \bar{\theta})$ be the optimal solution and $\bar{f}$ be the optimal objective function value of the* *master problem at an iteration of the algorithm. The decision vector* $\bar{\mathbf{x}}$ *is a feasible solution of the original problem and the associated objective function value is equal to* $\hat{f} = \bar{f} - \bar{\theta} + \theta^*$, *where* θ^* *is the value of* $\{\mathbb{E}\left[Q(\bar{\mathbf{x}},\omega)\right] + \lambda \mathrm{CVaR}_\alpha\left(Q(\bar{\mathbf{x}},\omega)\right)\}$. *It is easy to see that* $\hat{f}$ *is the best objective function value found so far and* $\bar{f}$ *is the best known lower bound on the optimal objective function value. Then an upper bound on the relative optimality can be estimated as follows:*

$$\mathrm{UBROG} = \frac{\hat{f}-\bar{f}}{\bar{f}} = \frac{\theta^* - \theta}{\bar{f}}. \tag{19}$$

We stop the algorithm when UBROG $\leq \epsilon$, *where* ϵ *is a specified stopping tolerance value.*

2.2 Subgradient-based Decomposition Algorithm The first proposed algorithm introduces new variables, which may be computationally expensive if a very large number of scenarios is used to represent the randomness. As an alternative we consider another algorithm, where we approximate the CVaR value of the recourse function by constructing the subgradient inequalities which replace the optimality cuts (11)-(14) used in Algorithm 1. This algorithm has been adapted from the general algorithm proposed by Ahmed (2006) for the class of risk measures involving the quantile deviation risk measure.

Let $\upsilon(\mathbf{x},\eta) = E([Q(\mathbf{x},\xi(\omega)) - \eta]_+)$, where $\eta \in \mathbb{R}$ denotes the α-quantile of the recourse cost. Then, we have $\mathrm{CVaR}_\alpha(Q(\mathbf{x},\xi(\omega))) = \inf_{\eta\in\mathbb{R}}\{\eta + \frac{1}{1-\alpha}\upsilon(\mathbf{x},\eta)\}$. Since $E[\max(Z,0)]$ is a convex non-decreasing function in $Z \in \mathcal{Z}$ and $Q(\mathbf{x},\xi(\omega)) - \eta$ is convex in $\mathbf{x}$ and η, the composite function $\upsilon(\mathbf{x},\eta)$ is also convex in $\mathbf{x}$ and η. Therefore, it can be approximated by the subgradient inequalities. Using the chain rule of subdifferentiation we can calculate a subgradient of $\upsilon(\mathbf{x},\eta)$ for given subgradients of $Q(\mathbf{x},\xi(\omega)) - \eta$. Next, we present the master problem based on the subgradients of $\upsilon(\mathbf{x},\eta)$:

$$\begin{aligned}
&\min && (1+\lambda)\mathbf{c}^T\mathbf{x} + \theta_1 + \lambda(\eta + \frac{1}{1-\alpha}\theta_2) && (20)\\
&\text{subject to} && \mathbf{x} \in X, && (21)\\
&\text{optimality cuts} && \theta_1 \geq \sum_{i=1}^N p_i \mathbf{u}_{it}^T(h_i - T_i\mathbf{x}), \quad t = 1,\ldots,\tau, && (22)\\
&\text{optimality cuts} && \theta_2 \geq \upsilon(\mathbf{x}^t,\eta^t) + (\upsilon_{\mathbf{x}}^t)^T(\mathbf{x}-\mathbf{x}^t) + (\upsilon_\eta^t)^T(\eta - \eta^t), \quad t = 1,\ldots,\tau, && (23)\\
& && \eta \in \mathbb{R}, && (24)\\
&\text{feasibility cuts} && (T_i^T\sigma^{s_i}\mathbf{x} - \mathbf{h}_i^T\sigma^{s_i}) \geq 0, \quad i = 1,\ldots,N,\ s_i = 1,\ldots,S_i. && (25)
\end{aligned}$$

Observe that (23)-(24) replace the constraints (11)-(14) and the objective function representation is modified so that θ_2 represents $E([Q(\mathbf{x},\xi(\omega)) - \eta]_+)$ instead of $\mathrm{CVaR}_\alpha(Q(\mathbf{x},\xi(\omega)))$. In this formulation, $(\mathbf{x}^t,\eta^t,\theta_1^t,\theta_2^t)$ denotes the optimal solution of the master problem at the iteration in which the tth optimality cut is constructed. $\upsilon_{\mathbf{x}}^t, \upsilon_\eta^t$ denote a subgradient of $\upsilon(\mathbf{x},\eta)$ with respect to $\mathbf{x}$ and η, respectively. In particular, the subgradients used in the algorithm are given by

$$\upsilon_{\mathbf{x}}^t = \sum_{i=1}^N p_i\pi_i^t\varrho_i^t, \quad \text{and } \upsilon_\eta^t = -\sum_{i=1}^N p_i\varrho_i^t, \tag{26}$$

where, for all $i = 1,\ldots,N$, $\pi_i^t = -T_i^T\mathbf{u}_{it}$, $\varrho_i^t = 1$ if $Q(\mathbf{x}^t,\xi^i) - \eta^t > 0$, and 0 otherwise.

As discussed for Algorithm 1, we can approximate $\mathbb{E}[Q(\mathbf{x},\xi(\omega))] + \lambda\mathrm{CVaR}_\alpha\left(Q(\mathbf{x},\xi(\omega))\right)$ alternatively by introducing $\hat{\theta} = \theta_1 + \lambda(\eta + \frac{1}{1-\alpha}\theta_2)$ and the combined optimality cuts:

$$\hat{\theta} \geq \sum_{i=1}^{N} p_i \mathbf{u}_{it}^T (h_i - T_i \mathbf{x}) + \lambda \left(\eta_t + \frac{1}{1-\alpha} \left(\upsilon(\mathbf{x}^t, \eta^t) + (\upsilon_{\mathbf{x}}^t)^T (\mathbf{x} - \mathbf{x}^t) + (\upsilon_{\eta}^t)^T (\eta - \eta^t) \right) \right), \quad t = 1, \ldots, \tau. \tag{27}$$

The description of this subgradient based decomposition algorithm is presented in Algorithm 2. As in Algorithm 1, we consider a stopping criterion based on a specified tolerance value ϵ. The first algorithm stops after generating a smaller number of optimality cuts with respect to Algorithm 2, as presented in Section 4.2. However, one might expect that the subgradient approach takes generally less time even if it requires a larger number of optimality cuts, since it involves a significantly less number of variables. We provide comparative results on the computational performance of the proposed algorithms in Section 4.2.

Algorithm 2 Decomposition Algorithm 2

```
1:  Step 1 of Algorithm 1.
2:  Solve the master problem (20)-(25).
    Let $(\bar{\mathbf{x}}, \bar{\eta}, \bar{\theta}_1, \bar{\theta}_2)$ be the optimal solution and $\bar{\theta} = \bar{\theta}_1 + \lambda(\bar{\eta} + \frac{1}{1-\alpha}\bar{\theta}_2)$.
3:  Step 3 of Algorithm 1.
4:  Step 4 of Algorithm 1.
5:  Let $(\mathbf{x}^\tau, \eta^\tau) = (\bar{\mathbf{x}}, \bar{\eta})$.
6:  Calculate $\upsilon(\mathbf{x}^\tau, \eta^\tau) = E([Q(\mathbf{x}^\tau, \xi^i) - \eta^\tau]_+) = \sum_{i=1}^{N} p_i(\max(\mathbf{u}_{i\tau}^T(\mathbf{h}_i - T_i\mathbf{x}^\tau) - \eta^\tau, 0))$.
7:  Calculate the mean-risk function value of the recourse cost at the current solution:
        $$\theta^* = \sum_{i=1}^{N} p_i \mathbf{u}_{i\tau}^T (\mathbf{h}_i - T_i \mathbf{x}^\tau) + \lambda(\eta^\tau + \frac{1}{1-\alpha} \upsilon(\mathbf{x}^\tau, \eta^\tau)).$$
8:  if $\bar{\theta} \geq \theta^*$ then
9:      Stop. $\mathbf{x}^\tau$ is the optimal first-stage decision vector.
10: else
11:     Introduce the optimality cuts (22)- (23) for $\mathbf{u}_{i\tau}$ and $(\mathbf{x}^\tau, \eta^\tau)$.
        For the second type of optimality cuts (23), calculate the subgradients as in (26). Go to Step 2.
12: end if
```

2.3 The Multicut Versions of Decomposition Algorithms In the decomposition algorithms presented in Sections 2.1 and 2.2, when the second-stage problem is feasible for all the realizations of the random data at an iteration, the second-stage problems are solved and the information represented by the optimal dual variables is aggregated to generate optimality cuts. At each such iteration of Algorithms 1 and 2 we generate a single-cut (10) or (22), respectively, for approximating the expectation term and a single-cut (11) or (23), respectively, for the CVaR term. Alternatively, in Algorithms 1 and 2, a single combined optimality cut (16) or (27), respectively, is generated to approximate the mean-risk function of the recourse cost. In a multicut method, instead of these aggregated cuts, one cut per realization is generated to keep more information about the dual solutions of the second-stage problems.

In Sections 2.1 and 2.2, we use the variables θ_1, θ_2 and $\hat{\theta}$ to define the optimality cuts. In the multicut method, we introduce the variables $\hat{\theta}_i$, θ_{1i} and θ_{2i}, $i = 1, \ldots, N$, to generate the optimality cuts associated with each realization of the random data. Next, we specify the multicut versions of the optimality cuts introduced in Sections 2.1 and 2.2.

- For Algorithm 1
  - The multicut version of the optimality cuts (10):

$$\theta_{1i} \geq \mathbf{u}_{it}^T(h_i - T_i\mathbf{x}), \quad t = 1,\dots,\tau,\ i = 1,\dots,N. \tag{28}$$

  - The multicut version of the optimality cuts (11):

$$\theta_{2i} \geq \eta_t + \frac{1}{1-\alpha}\nu_{it}, \quad t = 1,\dots,\tau,\ i = 1,\dots,N. \tag{29}$$

  - The multicut version of the combined optimality cut (16):

$$\hat{\theta}_i \geq \mathbf{u}_{it}^T(h_i - T_i\mathbf{x}) + \lambda(\eta_t + \frac{1}{1-\alpha}\nu_{it}), \quad t = 1,\dots,\tau\ i = 1,\dots,N. \tag{30}$$

- For Algorithm 2
  - The multicut version of the optimality cuts (22):

$$\theta_{1i} \geq \mathbf{u}_{it}^T(h_i - T_i\mathbf{x}), \quad t = 1,\dots,\tau,\ i = 1,\dots,N. \tag{31}$$

  - The multicut version of the optimality cuts (23):

$$\theta_{2i} \geq \upsilon_i(\mathbf{x}^t,\eta^t) + (\upsilon_{\mathbf{x},i}^t)^T(\mathbf{x}-\mathbf{x}^t) + (\upsilon_{\eta,i}^t)^T(\eta-\eta^t), \quad t = 1,\dots,\tau,\ i = 1,\dots,N, \tag{32}$$

    where $\upsilon_i(\mathbf{x}^t,\eta^t) = [Q(\mathbf{x},\xi^i)-\eta]_+$, $\upsilon_{\mathbf{x},i}^t = -T_i^T\mathbf{u}_{it}\varrho_i^t$ and $\upsilon_{\eta,i}^t = -\varrho_i^t$, and $\varrho_i^t = 1$ if $Q(\mathbf{x}^t,\xi^i)-\eta^t > 0$, and 0 otherwise.

  - The multicut version of the combined optimality cut (27):

$$\hat{\theta}_i \geq \mathbf{u}_{it}^T(h_i - T_i\mathbf{x}) + \lambda\left(\eta_t + \frac{1}{1-\alpha}\left(\upsilon_i(\mathbf{x}^t,\eta^t) + (\upsilon_{\mathbf{x},i}^t)^T(\mathbf{x}-\mathbf{x}^t) + (\upsilon_{\eta,i}^t)^T(\eta-\eta^t)\right)\right), \tag{33}$$

$$t = 1,\dots,\tau,\ i = 1,\dots,N.$$

In addition to the multicut versions of the optimality cuts, we replace θ_1 by $\sum\limits_{i=1}^N p_i\theta_{1i}$, θ_2 by $\sum\limits_{i=1}^N p_i\theta_{2i}$, and $\hat{\theta}$ by $\sum\limits_{i=1}^N p_i\hat{\theta}_i$ in the objective functions of the master problems. Note that in the risk-neutral linear two-stage stochastic programming we only have optimality cuts of type (28). We extend the existing multicut method for incorporating the risk term.

Using the disaggregate cuts, as listed above, is generally expected to improve the outer approximations and reduce the number of iterations with respect to the single-cut method. However, the size of the master problem may cause computational difficulties, since we add a large number of optimality cuts at once. Note that in the single-cut versions of the decomposition algorithms, the number of optimality cuts is equal to 2τ. In the multicut version of Algorithm 1 with the combined optimality cuts (30), the number of optimality cuts is equal to $N*\tau$. When one type of cuts ((28) or (29)) and both types of cuts (28) and (29) are introduced, the number of optimality cuts generated in Algorithm 1 is equal to $N*\tau+\tau$ and $2N*\tau$, respectively. Similarly, the number of optimality cuts generated in Algorithm 2 with (33), (31) or (32), (31) and (32) are $N*\tau$, $N*\tau+\tau$, and $2N*\tau$, respectively.

Birge and Louveaux (1997) note that the multicut method is expected to be more effective when the number of realizations is not significantly larger than the number of first-stage constraints. The authors also remark that the trade-off between the number of iterations and the size of the master problem is problem (instance)-dependent. For a detailed discussion on the relative advantages and disadvantages of the single-cut and multicut methods we refer to Birge and Louveaux (1988) and Gassmann (1990). As suggested by Birge and Louveaux (1997) one may employ an approach that is somehow between the single-cut and the multicut methods. Such an approach groups all the realizations into disjoint subsets and aggregates the dual variable information over each group. In the single-cut method, there is only one group consisting of all the realizations and in the multicut method, each realization constitutes one group. In our numerical study, we provide a sample set of results for the multicut versions of Algorithm 2 in order to demonstrate how the multicut approach affects the computational performance.

In this study, we have focused on the decomposition approaches to deal with a large number of scenarios when we consider CVaR as a risk measure in the two-stage mean-risk analysis. The proposed methods may be further improved computationally using regularization and bunching methods. For a detailed discussion see Ruszczyński (1986), Ruszczyński and Świetanowski (1996), Birge and Louveaux (1997).

3. Application to Disaster Preparedness Management

We apply the proposed framework to a disaster preparedness management problem, where modeling the variability is crucial in order to reduce the damage of a possible disaster. We consider an existing disaster location-allocation model proposed by Rawls and Turnquist (2010). The aim of their model is to determine the response facility locations and the inventory levels of several commodities at each facility in the presence of uncertainty in demand, transportation capacities, and the damage level of supplies. This is a significant model, which considers different types of uncertainties. In particular, the model allows the links in the network and some or all of the supplies preallocated at a facility to be damaged when the disaster occurs. We intend to contribute to the disaster management literature by extending this existing model by incorporating a risk measure. Our proposed model is a significant and non-trivial extension, because it incorporates a risk measure in addition to the expectation criterion. After briefly presenting the existing model we describe our proposed model and finally, we discuss some details regarding the application of the decomposition algorithms.

3.1 Existing Model

The problem of determining the locations of the storage (response) facilities and the inventory levels of disaster relief supplies at each facility is crucial to efficiently manage the response operations when a disaster occurs. A limited number of studies exist in the literature, which focus on the pre-positioning of the different types of emergency supplies in the presence of uncertainty. We refer to Rawls and Turnquist (2010) and the references therein for positioning of this model in the disaster management literature.

In two-stage programming problems there is a hierarchy of decisions. In the first-stage, a set of decisions is made and then uncertainty is revealed, i.e., random variables are observed, and then a second set of decisions are determined under the predetermined first-stage decision variables and the observed uncertain quantities. This setup fits the problem of pre-positioning emergency supplies at a stochastic disaster relief network perfectly. The disaster relief supplies are preallocated at the response facilities which are distributed according to the realized demand in response to a disaster. In the proposed two-stage model, the first-stage decisions are on the types and locations of the facilities, and on the amount of each commodity to be stocked at each facility. For example, the commodities that are likely to be in demand when a hurricane occurs are prepackaged food, medical kits, blankets, etc. The second-stage decision variables are concerned with distributing the supplies to satisfy the realized demand. The second-stage problem, which is in the form of a multi-commodity network flow formulation, involves stochastic constraints to satisfy the demand. Satisfying the stochastic constraints for all the realizations of random demand values would lead to overly conservative decisions. Instead, we allow infeasibilities in the stochastic constraints, but introduce penalties on the amount of their violations. Therefore, auxiliary second-stage variables are introduced to calculate the surplus and shortage amounts to control the salvage costs and penalties on unmet demand.

We basically use almost the same notation introduced in Rawls and Turnquist (2010).

Inputs

K: set of commodities, indexed by k;

I: set of locations (nodes), indexed by i;

A: set of arcs (links) in the network;

L: size categories of a storage facility, indexed by l;

M_l: the overall capacity (e.g., square feet of available space) of a facility of size category $l, l \in L$;

F_{il}: fixed cost of opening a facility of size category l at location i, $i \in I, l \in L$;

b^k: unit space requirement for commodity k, $k \in K$;

q^k: unit acquisition (purchasing) cost for commodity k, $k \in K$;

h^k: unit salvage cost for commodity k (when there is surplus), $k \in K$;

pc^k: unit penalty cost for the shortage of commodity k, $k \in K$;

u^k: link capacity used for shipping a unit of commodity k, $k \in K$;

N: number of scenarios, indexed by s;

p_s: probability of scenario s, $s = 1, \dots, N$;

c_{ij}^{ks}: cost of shipping a unit of commodity k through the link (i,j) under scenario s, $(i,j) \in A,\ k \in K,\ s = 1, \dots, N$;

ϑ_i^{ks}: realization of demand for commodity k at location i under scenario s, $i \in I,\ k \in K,\ s = 1, \dots, N$;

γ_i^{ks}: realization of the proportion of commodity k at location i that remains usable under scenario s $(0 \leq \gamma_i^{ks} \leq 1)$, $i \in I,\ k \in K,\ s = 1, \dots, N$;

U_{ij}^s: realization of the available capacity of link (i,j) under scenario s (it may be even zero), $(i,j) \in A,\ s = 1, \dots, N$.

Decision variables

First-stage variables:

- r_{ik}= Amount of commodity k pre-positioned at location i,
- $y_{il} = \begin{cases} 1 & \text{if a facility of size category } l \text{ is located at node } i, \\ 0 & \text{otherwise.} \end{cases}$

Primary second-stage variables:

- x_{ij}^{ks}= Amount of commodity k shipped through link (i,j) under scenario s,

Auxiliary second-stage variables:

- z_i^{ks}= Amount of commodity k that is not used at location i under scenario s (surplus amount),
- w_i^{ks}= Amount of the shortage of commodity k at location i under scenario s.

The randomness in the disaster network is modeled using a set of scenarios. Note that a scenario represents the joint realization of demand for each commodity ($\vartheta_i^{ks},\ i \in I, k \in K$), the arc capacities ($U_{ij}^s,\ (i,j) \in A$) and the damage levels of supplies ($\gamma_i^{ks},\ i \in I, k \in K$). Without loss of generality we assume that I is also the set of candidate storage facilities and for a first-stage decision vector $(\mathbf{r}, \mathbf{y})$ the objective function of the first-stage problem is given next:

$$f(\mathbf{r},\mathbf{y},\omega) = \sum_{i\in I}\sum_{l\in L} F_l y_{il} + \sum_{k\in K}\sum_{i\in I} q^k r_i^k + Q(\mathbf{r},\mathbf{y},\xi(\omega)),$$

where $\omega \in \Omega$ and the vector $(\mathbf{r}, \mathbf{y})$ satisfies the following constraints

$$\sum_{k\in K} b^k r_i^k \le \sum_{l\in L} M_l y_{il} \quad i \in I, \tag{34}$$

$$\sum_{l\in L} y_{il} \le 1 \quad i \in I, \tag{35}$$

$$y_{il} \in \{0,1\} \quad i \in I,\ l \in L, \tag{36}$$

$$r_i^k \ge 0 \quad i \in I,\ k \in K. \tag{37}$$

For the following second-stage problem, $\xi^s = (\mathbf{c}^s, \gamma^s, \vartheta^s, \mathbf{U}^s)$ denotes the realization of the random data under scenario $s,\ s \in \{1,\dots,N\}$:

$$Q(\mathbf{r},\mathbf{y},\xi^s) = \min \quad \sum_{(i,j)\in A}\sum_{k\in K} c_{ij}^{ks} x_{ij}^{ks} + \sum_{i\in I}\sum_{k\in K} \left(h^k z_i^{ks} + pc^k w_i^{ks}\right) \tag{38}$$

$$\text{subject to:} \quad \sum_{(j,i)\in A} x_{ji}^{ks} + \gamma_i^{ks} r_i^k - \sum_{(i,j)\in A} x_{ij}^{ks} - \vartheta_i^{ks} = z_i^{ks} - w_i^{ks}, \quad i \in I,\ k \in K,\ s = 1,\dots,N, \tag{39}$$

$$\sum_{k\in K} u^k x_{ij}^{ks} \le U_{ij}^s \quad (i,j) \in A,\ s = 1,\dots,N, \tag{40}$$

$$x_{ij}^{ks} \ge 0 \quad (i,j) \in A,\ k \in K,\ s = 1,\dots,N, \tag{41}$$

$$z_i^{ks},\ w_i^{ks} \ge 0 \quad i \in I,\ k \in K,\ s = 1,\dots,N. \tag{42}$$

Constraints (34) guarantee that commodities are stored at the open facilities and the total space taken by the pre-allocated supplies does not exceed the capacity of the facility. The number of facilities to be located at any node is at most 1 by constraints (35). The conservation of flow at each node is represented by constraints (39). By dropping the scenario index, we can write the stochastic version of the demand satisfaction constraint as follows:

$$\sum_{(j,i)\in A} x_{ji}^k(\omega) - \sum_{(i,j)\in A} x_{ij}^k(\omega) + \gamma_i^k(\omega) r_i^k - \vartheta_i^k(\omega) = z_i^k(\omega) - w_i^k(\omega), \quad i \in I.$$

Notice that the random amount of available commodity k at node i is equal to $a^k(i) = \sum_{(j,i)\in A} x^k_{ji}(\omega) - \sum_{(i,j)\in A} x^k_{ij}(\omega) + \gamma^k_i(\omega) r^k_i$, where $\gamma^k_i(\omega) r^k_i$ is the amount of undamaged supplies that are prelocated at node i. If the realized random amount of commodity k required at node i under scenario s, ϑ^{ks}_i, is larger than the realization of $a^k(i)$ under scenario s, we have a positive shortage $w^{ks}_i > 0$, otherwise we have a positive surplus $z^{ks}_i > 0$. Constraints (40) enforce that the assigned link flow does not exceed the available capacity. The rest of the constraints are for the non-negativity and binary restrictions.

In the existing model proposed by Rawls and Turnquist (2010), the expectation is used as the preference criterion: the existing model finds the location-allocation policy vector $(\mathbf{y}, \mathbf{r})$ for which the associated random total cost $f(\mathbf{r}, \mathbf{y}, \omega)$ has the minimum expected value. Rawls and Turnquist (2010) also emphasize the significance of involving risk measures into their model. In the next section, we extend their model by considering CVaR of the total cost and present the proposed risk-averse model.

3.2 Proposed Model As discussed in Section 1, we specify CVaR as the risk measure and formulate the two-stage disaster preparedness model as a two-stage mean-risk stochastic programming problem:

$$\min_{(\mathbf{r},\mathbf{y})\in X} \left\{ \mathbb{E}\left[f(\mathbf{r},\mathbf{y},\omega)\right] + \lambda \mathrm{CVaR}_\alpha \left(f(\mathbf{r},\mathbf{y},\omega)\right) \right\}, \tag{43}$$

where the feasible set X and the second-stage problem are defined by (34)-(37) and (38)-(42), respectively.

By Proposition 1.1 we have the deterministic equivalent formulation (DEF) of the stochastic disaster location-allocation problem in the following form:

$$\min \quad (1+\lambda)\left(\sum_{i\in I}\sum_{l\in L} F_l y_{il} + \sum_{k\in K}\sum_{i\in I} q^k r^k_i\right) + \sum_{s=1}^{N} p_s \left[\sum_{(i,j)\in A}\sum_{k\in K} c^{ks}_{ij} x^{ks}_{ij} + \sum_{i\in I}\sum_{k\in K}\left(h^k z^{ks}_i + pc^k w^{ks}_i\right)\right]$$

$$+\lambda\left(\eta + \frac{1}{1-\alpha}\sum_{s=1}^{N} p_s v_s\right)$$

$$\text{subject to:} \quad \sum_{(j,i)\in A} x^{ks}_{ji} - \sum_{(i,j)\in A} x^{ks}_{ij} - z^{ks}_i + w^{ks}_i = \vartheta^{ks}_i - \gamma^{ks}_i r^k_i, \quad i\in I,\ k\in K,\ s=1,\dots,N, \tag{44}$$

$$\sum_{k\in K} u^k x^{ks}_{ij} \le U^s_{ij} \quad (i,j)\in A,\ s=1,\dots,N, \tag{45}$$

$$\sum_{k\in K} b^k r^k_i \le \sum_{l\in L} M_l y_{il} \quad i\in I,$$

$$\sum_{l\in L} y_{il} \le 1 \quad i\in I,$$

$$y_{il}\in\{0,1\} \quad i\in I,\ l\in L, \quad r^k_i \ge 0 \quad i\in I,\ k\in K,$$

$$x^{ks}_{ij} \ge 0 \quad (i,j)\in A,\ k\in K,\ s=1,\dots,N,$$

$$z^{ks}_i,\ w^{ks}_i \ge 0 \quad i\in I,\ k\in K,\ s=1,\dots,N,$$

$$v_s \ge \left[\sum_{(i,j)\in A}\sum_{k\in K} c^{ks}_{ij} x^{ks}_{ij} + \sum_{i\in I}\sum_{k\in K}\left(h^k z^{ks}_i + pc^k w^{ks}_i\right)\right] - \eta, \quad s=1,\dots,N,$$

$$\eta\in\mathbb{R}, v_s \ge 0,\ s=1,\dots,N.$$

We refer to this proposed problem with CVaR as "Disaster location and allocation problem with CVaR (DALPWithCVaR)".

By changing the trade-off coefficient λ, efficient location and allocation policies can be constructed, and this would allow the decision makers to evaluate different policies. Illustrative examples are presented in our numerical section.

In the first stage of the proposed disaster management model the binary variables, $y_{il},\ i \in I,\ l \in L$, to prescribe the locations of the facilities. However, the proposed framework, which has been described for continuous variables only, can still be applied under these integrality restrictions. Observe that binary variables do not appear in the second-stage problem (38)-(42) and therefore, the recourse function depends on $\mathbf{y}$ through continuous variables $\mathbf{r}$. Due to this special structure, we can just drop the binary variables from the recourse function and then we can rewrite the mean-risk function of the model as

$$f(\mathbf{r},\mathbf{y},\omega) = (1+\lambda)\left(\sum_{i\in I}\sum_{l\in L} F_l y_{il} + \sum_{k\in K}\sum_{i\in I} q^k r_i^k\right) + \mathbb{E}[Q(\mathbf{r},\xi(\omega))] + \lambda \mathrm{CVaR}_\alpha\left(Q(\mathbf{r},\xi(\omega))\right).$$

As discussed in Section 1, the mean-risk function of the recourse cost is convex in $\mathbf{r}$ and we can construct the outer approximations for it in the continuous variables $\mathbf{r}$. Thus, the decomposition algorithms described in Section 2 directly apply here. The only difference is that Step 2 of Algorithms 1 and 2 requires solving a mixed-integer programming problem. This may pose a computational challenge, since solving the master problem several times can be time consuming for large problem instances. To remedy this issue, one may incorporate a heuristic into the decomposition algorithms to solve the master problem efficiently. To this end, a heuristic based on Lagrangian relaxation has been proposed by Rawls and Turnquist (2010). Alternatively, one may utilize a branch-and-cut scheme in the master problem. Such a method called "the integer L-shaped method" has been developed by Laporte and Louveaux (1993) to solve stochastic integer problems with binary first stage variables. However, in this study we restrict our attention to the structure of the optimality cuts and design decomposition methods to deal with a large number of scenarios. For a given specific application, our framework can be customized by incorporating features that improves the efficiency of the master problem.

Next, we discuss how Algorithms 1 and 2 can be employed to solve DALPWithCVaR. In the second-stage problem, auxiliary variables guarantee the feasibility. Thus, we have a complete recourse, where the second-stage problem is always feasible. Therefore, in the master problems of the decomposition algorithms we do not need to consider the feasibility cuts. In Algorithm 1, the optimality cuts for DALPWithCVaR are generated as follows:

$$\theta_1 \geq \sum_{s=1}^{N} p_s \sum_{k\in K, i\in I}\left((\vartheta_i^{ks} - \gamma_i^{ks} r_i^k)\beta_{it}^{ks} + U_{ij}^s \mu_{ijt}\right),\ t = 1,\ldots,\tau, \tag{46}$$

$$\theta_2 \geq \eta_t + \frac{1}{1-\alpha}\sum_{s=1}^{N} p_s v_{st},\ t = 1,\ldots,\tau, \tag{47}$$

$$v_{st} \geq \sum_{k\in K, i\in I}\left((\vartheta_i^{ks} - \gamma_i^{ks} r_i^k)\beta_{it}^{ks} + U_{ij}^s \mu_{ijt}^s\right) - \eta_t \text{ and } v_{st} \geq 0 \quad s = 1,\ldots,N,\ t = 1,\ldots,\tau, \tag{48}$$

where β_{it}^{ks} and μ_{ijt}^s are the dual variables associated with the constraints (44) and (45), respectively.

In order to calculate the subgradients in Algorithm 2, we note that the component of π_s^t $(s = 1,\ldots,N)$ corresponding to the first-stage variable r_i^k is equal to $-\gamma_i^{ks}\beta_{it}^{ks}$ and all components of $\hat{\pi}_s^t$ are equal to 0, because the $\mathbf{y}$ variables do not occur in the second-stage problem.

4. Numerical Results

In this section we present numerical results for the case study constructed in Rawls and Turnquist (2010) to show how the optimal location and allocation policies change with respect to risk parameters. We also provide computational results for a number of problem instances to illustrate the computational efficiency of the proposed decomposition algorithms.

The optimization problems are modeled with the AMPL mathematical programming language (Fourer et al., 2003) running on the 11.2 CPLEX solver (ILOG, 2008). Each problem instance is solved on a single core of a 64-bit HP workstation running on Linux with 2 quad-core 1.6GHz CPU, and 16GB of RAM. All reported CPU times are in seconds. In our computational study, we terminate CPLEX when the prescribed CPU time limit of 7200 seconds is reached.

4.1 Case study

We solve the proposed problem DALPWithCVaR for the case study presented in Rawls and Turnquist (2010) considering different values of the risk parameters λ and α. Rawls and Turnquist (2010) use the expected value as the preference criterion; their model is a special case of the proposed model when $\lambda = 0$. In this section, we present comparative results to show the effect of incorporating a risk measure on the random total cost and analyze how the policies change with respect to the risk parameters.

Here we briefly discuss the key points of the case study and refer the reader to Rawls and Turnquist (2010) for further details. The authors consider a network consisting of 30 nodes and 55 links to represent the southeastern US. They focus on hurricane threats and construct 51 scenarios based on the historical characteristics of hurricanes in the southeastern region of US. In the case study, three emergency commodities are considered: water, food and medical kits. Facilities are of three possible sizes: small, medium, and large. Food is considered in units of 1000 meals-ready-to eat (MREs), and water is considered in units of 1000 gallons. It is assumed that the unit penalty cost pc^k for each commodity k is κ times the purchase price $q^k, k \in K$. They set $\kappa = 10$, we also consider $\kappa = 5$ in our study. In Table 1 we provide the values of the parameters that might be useful to interpret the numerical results. The unmet demand penalty (shortage) costs are crucial while determining the policies, therefore, note that the shortage penalty cost is the highest for the food and the lowest for the medical kits.

For commodities				For facility types			
Commodity	q^k	b^k	Transportation	Size	Description	F_l	M_l
	(\$/unit)	(ft^3/unit)	Cost (\$/unit mile)	Category		(\$)	(ft^3)
Water (1000 gals)	647.7	144.6	0.3	1	Small	19,600	36,400
Food (1000 MREs)	5420	83.33	0.04	2	Medium	188,400	408,200
Medical Kits	140	1.16	0.00058	3	Large	300,000	780,000

Table 1: Input parameters in the case study.

In our proposed framework, there are two risk parameters: α and λ. Here we discuss how these risk parameters affect the optimal policies. The specified α level represents the risk preference in percentage terms, i.e., CVaR$_\alpha$ quantifies the mean value of the worst $(1-\alpha)\%$ of the total costs. When α increases the corresponding value-at-risk increases, and CVaR$_\alpha$ accounts for the risk of larger realizations. Thus, larger α values would lead to more conservative policies, which give more weight to worse scenarios. Note also that increasing the value of λ would increase the relative importance of the risk term and so would also lead to more risk averse policies. Thus, increasing the parameter λ and/or the parameter α implies a higher level of risk aversion. Figure 1 illustrates how the optimal mean-risk function of the total cost increases as the risk parameters λ or α increase (see Figure 1 (a) and (c)). Similar to the optimal mean-risk function, CVaR also increases as α increases by the definition of CVaR, i.e., for a larger value of α we focus on larger realizations of the total cost and the conditional expectation is larger as seen in Figure 1 (b) and (d). However, CVaR decreases as λ increases. Due to the changing trade-off between the expectation and the CVaR criterion, larger λ values provide us with a higher expected total cost and a lower CVaR value. However, we cannot make such a claim for the parameter α, i.e., a larger α value does not always result in a higher expected total cost. Thus, for a fixed α it is easy to see how the cost values change with respect to the risk coefficient λ. However, for a fixed λ value, we can only claim that a larger α value leads to higher mean-risk function and CVaR values, but the expected cost value may decrease or increase. Figure 2 illustrates this observation and also shows that the recourse cost and the total positioning cost, which together constitute the expected total cost, do not change monotonically as a function of α.

The total expected cost is decomposed into the positioning cost and the expected recourse cost; the expected recourse cost is decomposed into the expected transportation, salvage and the shortage costs. These detailed cost values are presented in Tables 2 and 3. According to these results, increasing λ leads to a more risk averse policy with higher positioning costs and lower expected recourse costs in general. Thus, a more risk averse policy keeps higher inventory levels overall and results in lower expected shortage costs. This does not imply that the inventory level for each commodity increases; whether the inventory level of a commodity increases or decreases depends on the associated shortage cost. Recall that shortage costs for food and water are significantly higher with respect to the medical kits. Therefore, to avoid high shortage costs, the inventory levels of food and water increase in contrast to the inventory level of medical kits for $\kappa = 5$ as in Table 2. However, Table 3 shows that for $\kappa = 10$ all shortage costs are high enough to lead an increase in the inventory level of each commodity.

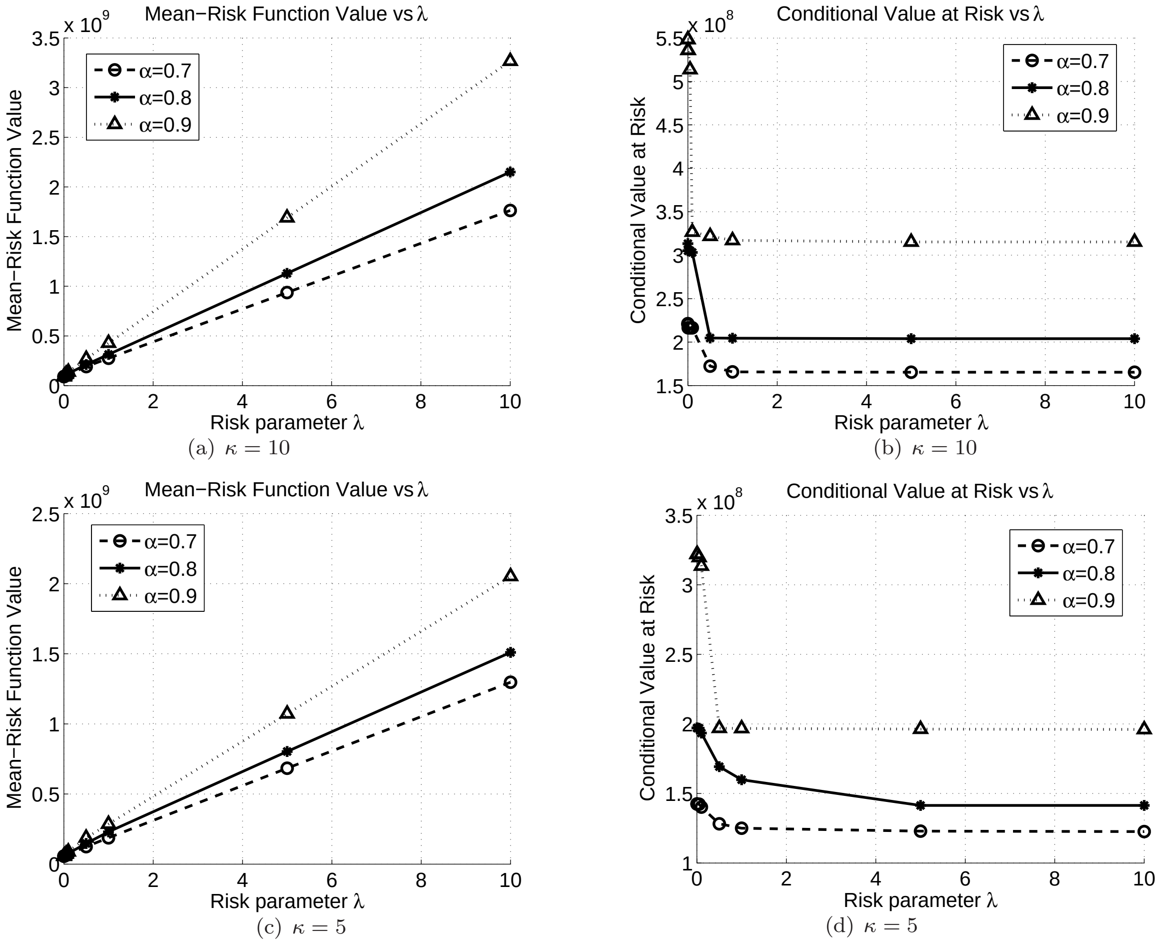
Suppose that for a given first-stage variable we solve the second-stage problem and obtain the values of the shortage variables w_i^{ks}. If the second-stage variable w_i^{ks} is strictly positive for at least one node i, we say that the demand for commodity k is not fully satisfied under scenario s and there is a (demand) shortage. Basically, the shortage probability for commodity $k \in K$ measures the probability of violating the stochastic demand satisfaction constraints:

$$\sum_{(j,i)\in A} x_{ji}^k(\omega) - \sum_{(i,j)\in A} x_{ij}^k(\omega) + \gamma_i^k(\omega) r_i^k \geq \vartheta_i^k(\omega), \quad i \in I.$$

Then for a given first-stage variable, the shortage probability for commodity k is calculated as

$$\sum_{s=1}^{N} \left\{ p_s : w_i^{ks} = \vartheta_i^{ks} - \left(\sum_{(j,i)\in A} x_{ji}^k(\omega) - \sum_{(i,j)\in A} x_{ij}^k(\omega) + \gamma_i^k(\omega) r_i^k \right) > 0 \text{ for at least one } i \in I \right\}.$$

In Tables 2 and 3 we report also the shortage probabilities for the optimal solutions. According to the results, a more risk averse policy leads to higher inventory levels and lower shortage probabilities in general. When the shortage cost for commodity k is not significantly high, the model would provide a solution for which the shortage of commodity k is more likely. This explains the higher shortage probability values for medical kits with respect to food and water. As evident from Table 3, more inventory is allocated for all commodities to obtain a lower expected shortage cost when κ increases. In this case, we also observe that the occurrence of shortage for each commodity is generally less likely. However, we cannot claim that the optimal solution of the proposed model would always perform well in terms of the shortage probabilities. The proposed model is concerned with the expected total shortage cost and not the shortage probability. Therefore, in an optimal solution the shortage probability of a commodity may be high even if the associated inventory level is high. Illustrative examples are given in the last columns of Tables 2 and 3, which show that a higher inventory level of water leads to a higher shortage probability. In addition, Figure 3 shows how the shortage probability and the inventory level for food change with respect to the risk parameters.



(a) $\kappa = 10$

(b) $\kappa = 10$

(c) $\kappa = 5$

(d) $\kappa = 5$

Figure 1: Optimal mean-risk function value of $f(\mathbf{r}, \mathbf{y}, \omega)$ and $\mathrm{CVaR}_\alpha(f(\mathbf{r}, \mathbf{y}, \omega))$.

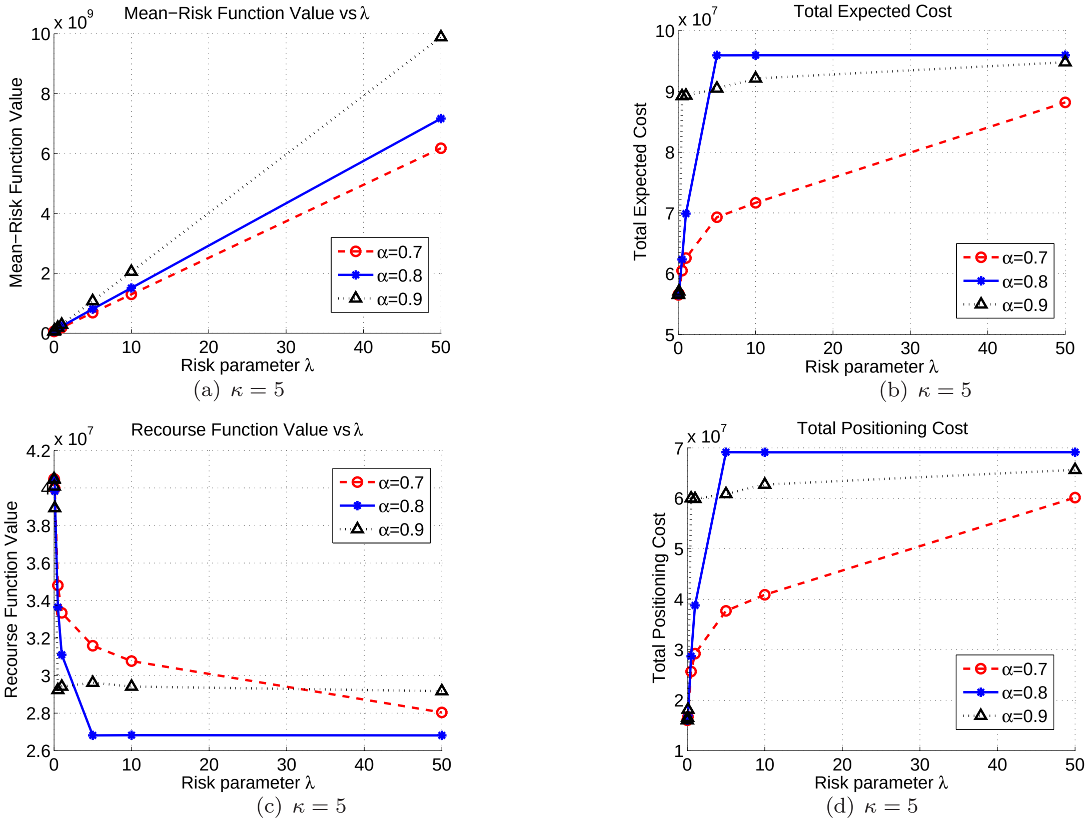
In order to demonstrate the effect of incorporating the risk measure CVaR on the total cost, we obtain the empirical cumulative distribution of the random total cost associated with the optimal solution of the proposed model. When $\lambda = 0$, we have the risk-neutral two-stage stochastic programming formulation, which we refer to as the "Base Problem". We solve the "Base Problem" and also the proposed risk-averse problem with CVaR for specified α values and obtain optimal policies. For a given optimal policy, we can calculate the realization of the total cost under each scenario and derive an empirical cumulative distribution of the random total cost. Figure 4 shows the cumulative distributions of the random total cost associated

	Base Problem, Risk coefficient $\lambda = 0$		Risk coefficient $\lambda = 0.1$		Risk coefficient $\lambda = 1$	
α	0.7	0.8	0.7	0.8	0.7	0.8
Mean-risk func.	$56.51{\cdot}10^6$	$56.51{\cdot}10^6$	$70.71{\cdot}10^6$	$76.13{\cdot}10^6$	$187.57{\cdot}10^6$	$229.74{\cdot}10^6$
CVaR_α	$142.44{\cdot}10^6$	$197.09{\cdot}10^6$	$140.18{\cdot}10^6$	$193.43{\cdot}10^6$	$125.01{\cdot}10^6$	$159.85{\cdot}10^6$
Total exp. cost	$56.51{\cdot}10^6$	$56.51{\cdot}10^6$	$56.70{\cdot}10^6$	$56.78{\cdot}10^6$	$62.56{\cdot}10^6$	$69.88{\cdot}10^6$
Positioning cost	$16.03{\cdot}10^6$	$16.03{\cdot}10^6$	$16.72{\cdot}10^6$	$16.96{\cdot}10^6$	$29.21{\cdot}10^6$	$38.77{\cdot}10^6$
Exp. recourse func.	$40.48{\cdot}10^6$	$40.48{\cdot}10^6$	$39.97{\cdot}10^6$	$39.82{\cdot}10^6$	$33.35{\cdot}10^6$	$31.11{\cdot}10^6$
Exp. trans & salvage cost	$2.02{\cdot}10^6$	$2.02{\cdot}10^6$	$2.21{\cdot}10^6$	$2.23{\cdot}10^6$	$4.8{\cdot}10^6$	$6.79{\cdot}10^6$
Exp. shortage cost	$38.46{\cdot}10^6$	$38.46{\cdot}10^6$	$37.76{\cdot}10^6$	$37.59{\cdot}10^6$	$28.55{\cdot}10^6$	$24.32{\cdot}10^6$
Water (1000 gals)	4920.1	4920.1	5659.9	5043.9	7924.9	8999
Food (1000 meals)	1830	1830	1933	2052	3921	5493.6
Medical kits	17560.1	17560.1	15049	15049	15049	16688.8
Small Facilities	14	14	5	14	8	4
Medium Facilities	1	1	2	1	3	4
Large Facilities	0	0	0	0	0	0
Shortage prob. Water	0.286	0.286	0.226	0.258	0.213	0.16
Shortage prob. Food	0.235	0.235	0.198	0.186	0.098	0.079
Shortage prob. Med. kits	0.363	0.363	0.41	0.41	0.465	0.416

Table 2: Detailed cost values, total inventory levels, total number of facilities, shortage probabilities for $\kappa = 5$.

	Base Problem, Risk coefficient $\lambda = 0$		Risk coefficient $\lambda = 0.1$		Risk coefficient $\lambda = 1$	
α	0.7	0.8	0.7	0.8	0.7	0.8
Mean-risk func.	$90.22{\cdot}10^6$	$90.22{\cdot}10^6$	$111.89{\cdot}10^6$	$120.70{\cdot}10^6$	$275.01{\cdot}10^6$	$313.93{\cdot}10^6$
CVaR_α	$221.18{\cdot}10^6$	$313.29{\cdot}10^6$	$216.43{\cdot}10^6$	$303.18{\cdot}10^6$	$165.69{\cdot}10^6$	$204.46{\cdot}10^6$
Total exp. cost	$90.22{\cdot}10^6$	$90.22{\cdot}10^6$	$90.25{\cdot}10^6$	$90.38{\cdot}10^6$	$109.32{\cdot}10^6$	$109.46{\cdot}10^6$
Positioning cost	$28.81{\cdot}10^6$	$28.81{\cdot}10^6$	$30.50{\cdot}10^6$	$31.04{\cdot}10^6$	$70.71{\cdot}10^6$	$70.89{\cdot}10^6$
Exp. recourse func.	$61.42{\cdot}10^6$	$61.42{\cdot}10^6$	$59.75{\cdot}10^6$	$59.34{\cdot}10^6$	$38.61{\cdot}10^6$	$38.57{\cdot}10^6$
Exp. trans & salvage cost	$4.76{\cdot}10^6$	$4.76{\cdot}10^6$	$5.11{\cdot}10^6$	$5.20{\cdot}10^6$	$13.01{\cdot}10^6$	$13.03{\cdot}10^6$
Exp. shortage cost	$56.66{\cdot}10^6$	$56.66{\cdot}10^6$	$54.64{\cdot}10^6$	$54.14{\cdot}10^6$	$25.60{\cdot}10^6$	$25.54{\cdot}10^6$
Water (1000 gals)	8999.0	8999.0	8999.0	9020.7	17735.7	17667.0
Food (1000 meals)	3609.0	3609.0	3921.0	4001.0	9403.9	9403.9
Medical kits	18749.0	18749.0	18753.4	19109.0	47111.7	48732.0
Small Facilities	6	6	6	8	6	6
Medium Facilities	2	2	2	2	5	5
Large Facilities	1	1	1	1	2	2
Shortage prob. Water	0.093	0.093	0.093	0.093	0.205	0.205
Shortage prob. Food	0.120	0.120	0.098	0.090	0.074	0.074
Shortage prob. Med. kits	0.215	0.215	0.215	0.210	0.199	0.199

Table 3: Detailed cost values, total inventory levels, total number of facilities, shortage probabilities for $\kappa = 10$.



(a) $\kappa = 5$ (b) $\kappa = 5$

(c) $\kappa = 5$ (d) $\kappa = 5$

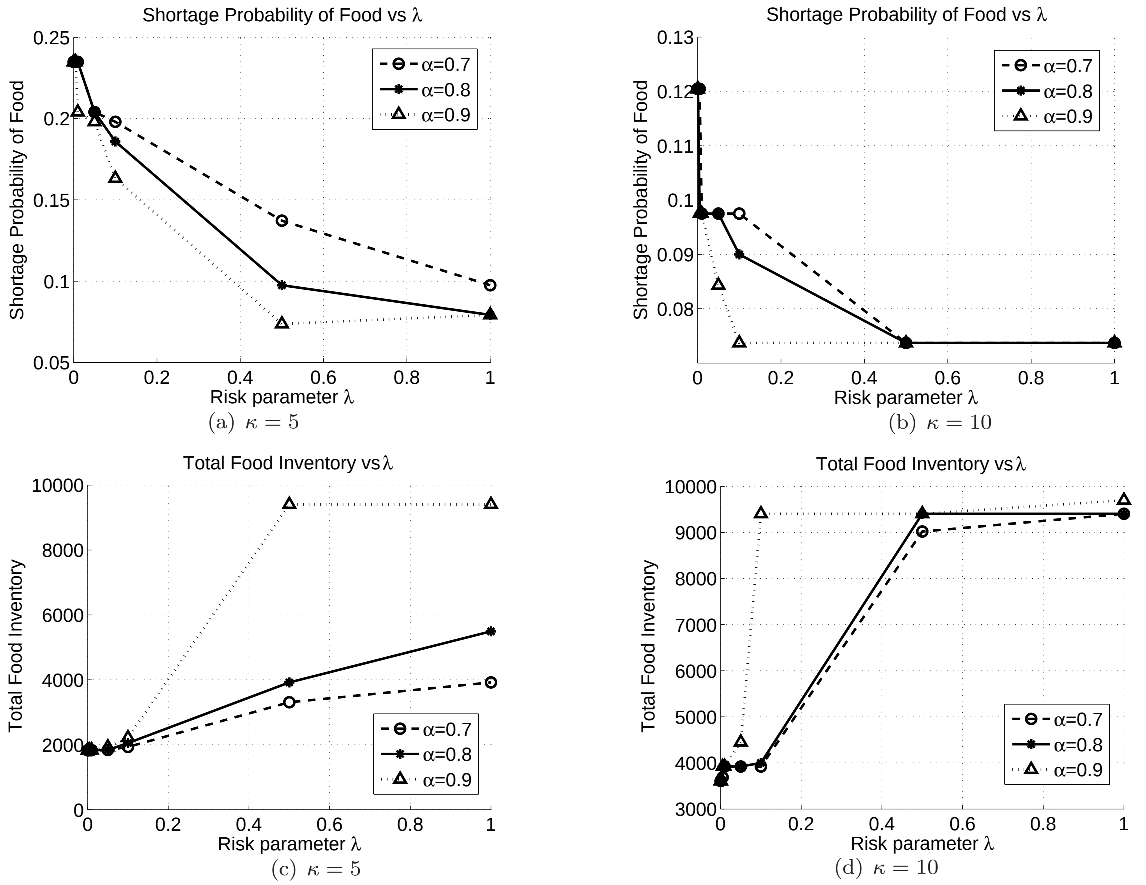
Figure 2: Cost function values for different values of κ and λ parameters.

with the "Base Problem" and the proposed problem for $\alpha = 0.7$ and $\alpha = 0.9$. Basically, the α parameter shapes the cumulative distribution according to the preferences of the decision maker. A larger α value avoids the occurrence of large realization values, and therefore leads to a shift to the left in the right tail of the cumulative distribution function. As a trade-off, the expectation increases and it implies a shift to the right in the left tail of the cumulative distribution function.

As mentioned before, the proposed mean-risk model can support the decision process by constructing policies for different risk parameters; the policy makers can evaluate those policies according to their risk preferences. This is similar to constructing an efficient frontier in finance. In this spirit, we analyze the changes in the expected total cost versus the corresponding CVaR for different values of the risk parameters α and λ. Table 4 presents the expected total cost and CVaR of the total cost at a specified confidence level α for the optimal solution of the traditional expectation-based model, i.e., for $\lambda = 0$. Tables 5 and 6 report the relative differences in the optimal expectation and CVaR values of the total cost with respect to the corresponding values obtained by the risk-neutral model.

Risk Coef.	$\alpha = 0.7$		$\alpha = 0.8$		$\alpha = 0.9$	
κ	Expected Total Cost	CVaR$_\alpha$	Expected Total Cost	CVaR$_\alpha$	Expected Total Cost	CVaR$_\alpha$
10	$90.22{\cdot}10^6$	$221.18{\cdot}10^6$	$90.22{\cdot}10^6$	$313.29{\cdot}10^6$	$90.22{\cdot}10^6$	$548.36{\cdot}10^6$
5	$56.51{\cdot}10^6$	$142.44{\cdot}10^6$	$56.51{\cdot}10^6$	$197.09{\cdot}10^6$	$56.51{\cdot}10^6$	$322.08{\cdot}10^6$

Table 4: Expected total cost and $\mathrm{CVaR}_\alpha(f(\mathbf{r}, \mathbf{y}, \omega))$ for $\lambda = 0$.



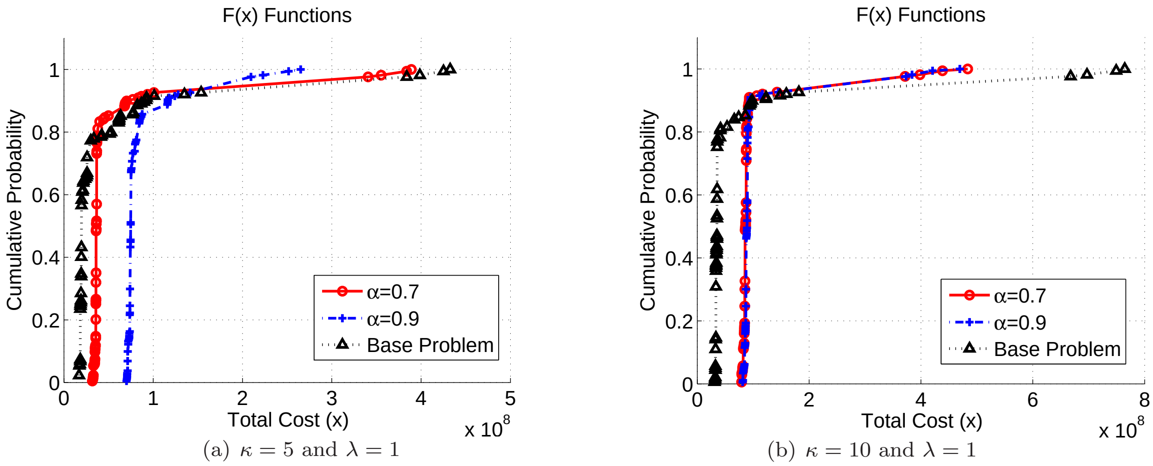
(a) $\kappa = 5$

(b) $\kappa = 10$

(c) $\kappa = 5$

(d) $\kappa = 10$

Figure 3: Shortage probability and total inventory for food.



(a) $\kappa = 5$ and $\lambda = 1$

(b) $\kappa = 10$ and $\lambda = 1$

Figure 4: Cumulative distribution functions of total cost for different values of κ and λ parameters.

	$\alpha = 0.7$		$\alpha = 0.8$		$\alpha = 0.9$	
Risk Coefficient	Relative Difference		Relative Difference		Relative Difference	
λ	Expected Total Cost	CVaR_α	Expected Total Cost	CVaR_α	Expected Total Cost	CVaR_α
0.005	0.006%	-0.530%	0.030%	-2.612%	0.030%	-2.254%
0.01	0.030%	-2.150%	0.030%	-2.612%	0.030%	-2.254%
0.05	0.030%	-2.150%	0.030%	-2.612%	1.120%	-6.314%
0.1	0.030%	-2.150%	0.173%	-3.226%	16.416%	-40.419%
0.5	15.988%	-22.093%	21.097%	-34.658%	18.723%	-41.296%
1	21.164%	-25.089%	21.322%	-34.737%	23.140%	-42.169%
5	21.916%	-25.244%	22.175%	-34.895%	26.907%	-42.513%
10	21.958%	-25.247%	22.183%	-34.895%	26.907%	-42.513%
50	21.991%	-25.248%	22.311%	-34.897%	29.075%	-42.536%
100	21.991%	-25.248%	22.399%	-34.898%	28.825%	-42.537%
1000	21.991%	-25.248%	22.404%	-34.898%	28.838%	-42.537%

Table 5: Expected total cost versus $\text{CVaR}_\alpha(f(\mathbf{r}, \mathbf{y}, \omega))$ for $\kappa = 10$.

In Table 7 we present the CPU times and the upper bounds on the relative optimality gaps of the proposed algorithms for the case study instances. “Direct CPU” refers to the CPU time required to solve the direct formulation of DALPWithCVaR (DEF) by the branch-and-bound (B&B) algorithm of CPLEX. Solving DEF by CPLEX provides us with an optimal objective function value, f^*. Let $\hat{f}$ denote the objective function value of the solution found by a decomposition algorithm when it stops within tolerance ϵ ($\epsilon = 0.015$). Then we calculate the upper bound on the relative optimality gap associated with the objective function value $\hat{f}$ as follows

$$\text{UBROG}^* = \frac{\hat{f} - f^*}{f^*}.$$

As seen from Table 7, solving the case study instances using the direct formulation might take less CPU time, especially for large λ values. The direct approach might outperform a decomposition algorithm for a small number of scenarios, but it is well-known that a decomposition algorithm would be computationally more efficient for a larger set of scenarios. In order to show some numerical results on the computational efficiency of the proposed algorithms, we randomly generate problem instances, as discussed in the following section, using parameters similar to those in the case study.

4.2 Computational performance of the proposed algorithms We generated random problem instances of different sizes to illustrate the computational performance of the proposed algorithms. We randomly generated 50 nodes on the plane and then constructed links between the nodes according to a specified threshold distance in order to obtain a disaster network consisting of 50 nodes. We generated the demand value at each node for a particular commodity under each scenario as follows: First, we randomly selected a node as the center of the disaster. Then, for a random number of nodes in the neighborhood we randomly generated demand values based on the distances from the center node; the generated demand for a node that is further away from the center node is smaller than the demand at a node that is closer to the center of the disaster. Table 8 presents the dimensional properties of the generated test problem instances.

First, we would like to emphasize that solving DEF directly using a standard mixed integer programming

	$\alpha = 0.7$		$\alpha = 0.8$		$\alpha = 0.9$	
Risk Coefficient	Relative Difference		Relative Difference		Relative Difference	
λ	Expected Total Cost	CVaR$_\alpha$	Expected Total Cost	CVaR$_\alpha$	Expected Total Cost	CVaR$_\alpha$
0.005	0.000%	0.000%	0.000%	0.000%	0.000%	-0.002%
0.01	0.001%	-0.036%	0.000%	0.000%	0.003%	-0.064%
0.05	0.007%	-0.140%	0.007%	-0.109%	0.189%	-0.733%
0.1	0.331%	-1.591%	0.484%	-1.858%	0.989%	-2.634%
0.5	7.051%	-10.013%	10.326%	-14.114%	57.927%	-38.871%
1	10.710%	-12.235%	23.670%	-18.893%	58.046%	-38.904%
5	22.620%	-13.729%	69.792%	-28.249%	60.112%	-39.058%
10	26.826%	-13.960%	69.805%	-28.249%	63.048%	-39.134%
50	56.059%	-14.517%	69.821%	-28.250%	67.767%	-39.186%
100	56.059%	-14.517%	69.821%	-28.250%	67.767%	-39.186%
1000	56.062%	-14.516%	69.821%	-28.250%	67.781%	-39.186%

Table 6: Expected total cost versus CVaR$_\alpha(f(\mathbf{r}, \mathbf{y}, \omega))$ for $\kappa = 5$.

		CPU Times			# of Iterations		UBROG*	
		Direct	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
$\lambda = 0.1$	0.7	269.32	300.07	86.23	40	52	0.44%	0.52%
	0.8	533.25	288.45	63.84	40	42	0.49%	0.41%
	0.9	503.68	135.51	72.46	33	46	0.43%	0.31%
$\lambda = 0.5$	0.7	162.81	311.37	128.32	29	71	0.29%	0.38%
	0.8	117.86	160.75	103.73	31	65	0.36%	0.49%
	0.9	102.64	405.34	132.67	45	74	0.64%	0.78%
$\lambda = 1$	0.7	201.07	266.47	256.37	23	102	0.33%	0.63%
	0.8	101.19	557.64	328.98	30	115	0.39%	0.39%
	0.9	70.72	544.46	173.93	49	83	0.52%	0.89%

Table 7: CPU times and upper bounds on the relative optimality gap.

solver such as CPLEX is hard for large problem instances. Thus, most of the generated problem instances cannot be solved for optimality within the prescribed time limit and therefore, in order to calculate the relative optimality gap we use the best known lower bound on the objective value found by the B&B algorithm of CPLEX. Let Obf$_t$ denote the best lower bound on the objective function value that is provided by the CPLEX solver, when the prescribed time limit t is reached. Obf$_t^*$ denotes the best available objective function value within the time limit, which defines an upper bound on the objective value. Then, we define an upper bound on the relative optimality gap as follows:

$$\text{UBROG}_t = \frac{\text{Obf}_t^* - \text{Obf}_t}{\text{Obf}_t}.$$

We also calculate the upper bound on the relative optimality gap for the best objective function value obtained by the proposed algorithms for the stopping tolerance $\epsilon = 0.015$:

$$\text{UBROG}_t = \frac{\hat{f} - \text{Obf}_t}{\text{Obf}_t}. \tag{49}$$

To have a fair comparison, we also set the mixed-integer optimality gap tolerance parameter ("mipgap") to 0.015 for the B&B algorithm of CPLEX. As noted in Table 9, CPLEX could not provide a lower bound on

Instance	Scenarios	Links	Continuous	Constraints	Instance	Scenarios	Links	Continuous	Constraints
Number			Vars.		Number			Vars.	
1	$N = 100$	594	208,450	74,551	11	$N = 300$	422	470,250	171,951
2		636	221,050	78,751	12		586	617,850	221,151
3		492	177,850	64,351	13		434	481,050	175,551
4		596	2090,50	74,751	14		712	731,250	258,951
5		534	190,450	68,551	15		444	490,050	178,551
6	$N = 200$	430	318,350	116,251	16	$N = 500$	422	1,059,650	378,551
7		600	420,350	150,251	17		606	783,650	286,551
8		464	338,750	123,051	18		406	759,650	278,551
9		630	438,350	156,251	19		624	1,086,650	387,551
10		438	323,150	117,851	20		458	837,650	304,551

Table 8: Dimensions of the problem instances.

the optimal objective function value for some problem instances. In this case, the maximum of the objective function values obtained by the master problems at termination of Algorithms 1 and 2 is taken as the best available lower bound. According to Table 9, the proposed (single-cut) decomposition algorithms provide solutions with very small optimality gaps in reasonable CPU times. We can also state that the (single-cut) subgradient-based decomposition algorithm requires the generation of significantly more optimality cuts and it generally performs better than the basic decomposition algorithm in terms of computation times.

Since the number of the iterations for the single-cut version of Algorithm 1 is already small, the multicut approach does generally not reduce the computation times. Therefore, we decided to report results for the multicut versions of Algorithm 2 in order to demonstrate how the multicut approach affects its computational performance. We consider three different multicut versions of Algorithm 2; the first, the second and the third one involves the optimality cuts (33), (31) and (32), and (32), respectively. In Table 10 we report the CPU times, the number of iterations and the upper bound values on the relative optimality gaps with respect to the results of Algorithm 2 presented in Table 9. According to Table 10, using multicuts (32) significantly reduces the computational times for $\alpha = 0.9$. However, the results for $\alpha = 0.7$ show that the multicut versions of the decomposition algorithms may be computationally expensive. As expected according to the discussions in Section 2.3, Table 10 illustrates that the multicut methods significantly reduce the number of iterations, but the tradeoff between the number of iterations and the size of the master problem is instance-dependent.

5. Conclusion and Future Work The typical stochastic programming objective of minimizing the expected total cost may yield solutions that are good in the long run but perform poorly under certain realizations of the random data. Therefore, for non-repetitive decision making problems under uncertainty, risk measures should be used to model the effects of the variability and the policy makers' risk aversion. In this study, we consider a two-stage mean-risk stochastic programming model, where the trade-off between the expected total cost and a risk measure on the random total cost is considered. Among the downside risk measures we focus on a recently popular and widely-applied risk measure CVaR. The main contribution of our study is developing solution algorithms for the proposed model with CVaR. As an application, we consider a disaster management problem that can significantly benefit from risk-averse two-stage stochastic

Instance	CPU; (# of Iterations)		CPU	UBROG$_t$			Relative Reduction (CPU)	
Number	Algorithm 1	Algorithm 2	Direct	Alg. 1	Alg. 2	Direct	Algorithm 1	Algorithm 2
$\alpha = 0.9$								
1	164.49; (13)	234.31; (36)	1635.71	0.56%	0.80%	0.62%	89.94%	85.68%
2	119.16; (11)	537.32; (55)	1917.13	0.48%	1.12%	0.45%	93.78%	71.97%
3	106.92; (11)	363.64; (50)	898.09	0.67%	0.94%	1.00%	88.09%	59.51%
4	151.35; (11)	308.04; (43)	1378.11	0.45%	0.64%	0.45%	89.02%	77.65%
5	419.89; (14)	527.27; (56)	347.27	1.07%	0.98%	1.44%	-20.91%	-51.83%
6	508.73; (12)	468.39; (41)	3083.51	0.48%	0.60%	0.58%	83.50%	84.81%
7	161.97; (10)	657.55; (46)	4965.30	0.80%	0.67%	0.52%	96.74%	86.76%
8	905.83; (16)	553.20; (46)	**7205.62**	0.74%	0.74%	1.71%	>87.43%	>92.32%
9	517.74; (12)	1094.68; (59)	5156.86	0.43%	1.18%	0.42%	89.96%	78.77%
10	454.95; (12)	301.49; (32)	6338.66	0.71%	0.56%	0.61%	92.82%	95.24%
11	498.36; (11)	637.88; (40)	6123.73	0.84%	0.54%	0.50%	91.86%	89.58%
12	707.84; (12)	1421.39; (57)	**7219.68**	*1.25%	*1.23%	100.00%	>90.20%	>80.31%
13	1089.23; (14)	770.07; (46)	**7209.08**	0.79%	0.76%	1.72%	>84.89%	>89.32%
14	655.73; (11)	947.10; (42)	6434.75	0.43%	0.60%	0.40%	89.81%	85.28%
15	918.35; (13)	486.69; (34)	5845.66	0.66%	0.79%	0.43%	84.29%	91.67%
16	2684.76; (13)	1273.57; (46)	**7251.32**	0.84%	0.94%	10.93%	>62.98%	>82.44%
17	1114.60; (13)	3567.46; (73)	**7241.72**	*1.06%	*1.38%	100.00%	>84.61%	>50.74%
18	3539.52; (16)	1268.51; (46)	**7233.41**	*1.05%	*1.05%	100.00%	>51.07%	>82.46%
19	1426.73; (11)	1535.98; (45)	**7255.26**	*1.08%	*1.28%	100.00%	>80.34%	>78.83%
20	1723.20; (12)	1050.51; (40)	**7243.75**	*1.41%	*1.17%	100.00%	>76.21%	>85.50%
$\alpha = 0.7$								
1	264.74; (14)	194.33; (32)	1422.08	0.48%	0.65%	0.42%	81.38%	86.33%
2	186.66; (13)	180.88; (30)	1831.26	0.48%	0.58%	0.42%	89.81%	90.12%
3	302.88; (11)	306.69; (44)	1116.67	0.62%	0.92%	1.04%	72.88%	72.54%
4	129.19; (10)	255.05; (35)	1829.69	0.67%	0.92%	0.45%	92.94%	86.06%
5	548.52; (16)	306.73; (42)	566.27	1.15%	1.16%	1.47%	3.13%	45.83%
6	1581.94; (15)	352.32; (33)	5239.41	0.53%	0.68%	0.53%	69.81%	93.28%
7	678.18; (13)	684.41; (44)	6630.70	0.50%	1.09%	0.51%	89.77%	89.68%
8	1092.36; (16)	514.95; (43)	5843.85	0.64%	0.81%	1.07%	81.31%	91.19%
9	1007.16; (12)	465.90; (37)	6820.72	0.61%	0.56%	0.48%	85.23%	93.17%
10	493.28; (11)	292.05; (31)	5894.88	0.59%	0.81%	0.44%	91.63%	95.05%
11	837.19; (13)	553.43; (37)	**7218.93**	*1.08%	*1.33%	100.00%	>88.40%	>92.33%
12	1099.49; (11)	633.11; (35)	**7221.57**	*1.08%	*1.44%	100.00%	>84.77%	>91.23%
13	2927.85; (15)	670.90; (41)	**7219.78**	*1.29%	*1.21%	100.00%	>59.45%	>90.71%
14	862.53; (11)	541.13; (30)	**7219.99**	*1.04%	*1.24%	100.00%	>88.05%	>92.51%
15	1787.62; (12)	334.46; (27)	**7219.53**	0.53%	0.65%	11.45%	>75.24%	>95.37%
16	3799.04; (14)	1192.49; (42)	**7240.30**	*0.93%	*1.23%	100.00%	>47.53%	>83.53%
17	3042.06; (11)	1476.19; (44)	**7246.99**	*0.94%	*1.29%	100.00%	>58.02%	>79.63%
18	3607.01; (14)	1036.28; (40)	**7229.07**	*1.28%	*1.18%	100.00%	>50.10%	>85.67%
19	1542.06; (12)	787.91; (30)	**7246.43**	*1.27%	*1.22%	100.00%	>78.72%	>89.13%
20	2392.59; (12)	729.14; (32)	**7235.35**	*1.23%	*1.20%	100.00%	>66.93%	>89.92%
" * ": CPLEX could not provide a lower bound on the optimal objective function value. The CPU times exceeding the time limit are in boldface.								

Table 9: CPU times, number of optimality cuts, and the upper bounds on the relative optimality gap for $\lambda = 1$.

	Reduction in CPU **Wrt to (single-cut) Alg. 2**			**Reduction in # of Iterations** **Wrt to (single-cut) Alg. 2**			**Reduction in** UBROG$_t$ **Wrt to (single-cut) Alg. 2**		
Instance	**Algorithm 2 with multicuts**			**Algorithm 2 with multicuts**			**Algorithm 2 with multicuts**		
Number	(33)	(31) & (32)	(32)	(33)	(31) & (32)	(32)	(33)	(31) & (32)	(32)
				$\alpha = 0.9$					
1	51.71%	**67.32%**	51.53%	69.44%	**72.22%**	61.11%	4.00%	-0.17%	0.21%
2	55.13%	62.35%	**226.50%**	74.55%	**80.00%**	76.36%	4.21%	0.57%	0.57%
3	71.52%	41.37%	**186.94%**	78.00%	**82.00%**	76.00%	3.66%	0.04%	0.03%
4	55.26%	70.34%	**462.74%**	74.42%	**76.74%**	74.42%	5.95%	0.07%	-0.02%
5	56.86%	74.42%	**185.82%**	76.79%	**82.14%**	75.00%	7.47%	-0.39%	-0.11%
6	-74.59%	-22.16%	**77.31%**	70.73%	**73.17%**	70.73%	3.91%	0.07%	-0.03%
7	46.54%	69.73%	**183.36%**	73.91%	**78.26%**	76.09%	4.37%	0.07%	0.13%
8	21.86%	-4.14%	**50.86%**	**76.09%**	**76.09%**	67.39%	4.75%	-0.44%	0.11%
9	60.01%	77.59%	**456.21%**	79.66%	**83.05%**	81.36%	4.13%	0.71%	0.67%
10	-102.85%	11.41%	**71.92%**	62.50%	**68.75%**	65.63%	5.81%	0.06%	-0.23%
11	19.40%	44.44%	**238.06%**	72.50%	**75.00%**	72.50%	5.70%	0.00%	-0.02%
12	59.12%	71.15%	**575.67%**	80.70%	**82.46%**	**82.46%**	3.40%	0.19%	-0.07%
13	-8.95%	-50.90%	**47.62%**	73.91%	**78.26%**	71.74%	5.11%	0.03%	-0.28%
14	43.79%	69.07%	**510.69%**	71.43%	**76.19%**	**76.19%**	4.60%	-0.40%	0.05%
15	-331.03%	-387.61%	**-34.60%**	**61.76%**	**61.76%**	58.82%	5.99%	0.30%	0.28%
16	-100.91%	71.88%	**227.23%**	73.91%	**80.43%**	76.09%	6.22%	-0.13%	0.02%
17	67.71%	-25.21%	**468.50%**	**83.56%**	**83.56%**	**83.56%**	4.95%	0.40%	0.36%
18	-110.65%	24.99%	**36.80%**	71.74%	**80.43%**	76.09%	5.88%	-0.09%	-0.07%
19	53.94%	-86.64%	**648.86%**	75.56%	75.56%	**80.00%**	4.80%	0.20%	0.08%
20	-35.55%	-162.10%	**1.76%**	**72.50%**	**72.50%**	67.50%	5.09%	-0.53%	-0.15%
				$\alpha = 0.7$					
1	55.66%	**79.65%**	43.18%	65.63%	**75.00%**	62.50%	6.38%	-0.37%	0.07%
2	-50.05%	3.10%	**54.21%**	53.33%	**63.33%**	**63.33%**	4.29%	0.13%	0.02%
3	47.27%	4.62%	**58.68%**	72.73%	**75.00%**	**75.00%**	3.82%	-0.16%	0.15%
4	45.00%	45.15%	**75.78%**	68.57%	**71.43%**	**71.43%**	6.58%	0.38%	0.34%
5	**10.79%**	-407.20%	-64.02%	71.43%	**76.19%**	73.81%	5.90%	0.04%	-0.03%
6	-119.92%	-323.26%	**-50.92%**	**63.64%**	**63.64%**	60.61%	5.66%	0.22%	0.18%
7	19.76%	60.94%	**76.37%**	72.73%	**77.27%**	**77.27%**	5.23%	0.53%	0.41%
8	-17.21%	-122.27%	**1.84%**	**69.77%**	**69.77%**	67.44%	4.94%	-0.12%	0.19%
9	-19.89%	-24.89%	**9.23%**	70.27%	**72.97%**	67.57%	2.54%	0.10%	0.11%
10	-217.50%	-727.33%	**-7.85%**	58.06%	58.06%	**61.29%**	6.23%	0.35%	0.28%
11	19.47%	6.01%	**69.95%**	67.57%	**72.97%**	**72.97%**	5.85%	0.34%	0.23%
12	**40.46%**	-7.94%	18.41%	**71.43%**	**71.43%**	**71.43%**	3.87%	0.39%	0.34%
13	-11.31%	-347.82%	**13.89%**	**73.17%**	70.73%	70.73%	4.89%	0.02%	-0.05%
14	**7.42%**	-391.29%	-39.45%	**63.33%**	**63.33%**	60.00%	3.52%	0.12%	0.02%
15	**-309.06%**	**-309.99%**	-366.87%	51.85%	**59.26%**	44.44%	5.96%	-0.44%	0.17%
16	-133.51%	-150.37%	**-94.16%**	71.43%	**73.81%**	69.05%	6.72%	0.44%	0.44%
17	-135.17%	-162.89%	**21.51%**	72.73%	**77.27%**	72.73%	4.97%	0.38%	0.41%
18	-68.34%	**-5.87%**	-39.81%	70.00%	**77.50%**	75.00%	6.62%	0.16%	0.10%
19	-95.52%	-376.75%	**46.14%**	63.33%	**66.67%**	**66.67%**	4.52%	0.06%	0.06%
20	-599.69%	-422.53%	**-84.12%**	56.25%	**65.63%**	**65.63%**	6.60%	0.17%	-0.06%
				The best reduction percentages are in boldface.					

Table 10: Multicut versions: CPU times, number of optimality cuts, and the upper bounds on the relative optimality gap for $\lambda = 1$.

programming models. Our second contribution is developing a new risk-averse location and allocation model for a disaster relief network under uncertainty in demand and the damage level of the disaster network. The proposed problem can be solved for different risk parameters, and this would allow the decision makers to evaluate different location and allocation decisions based on the tradeoff between the quality of service and costs. The computational study shows that we can solve the proposed model for a reasonably large number of scenarios. In our future research, we would like to also consider decisions on transportation capacities and develop new stochastic disaster relief network design models.

Acknowledgments We are sincerely grateful to Carmen G. Rawls for providing detailed data on the case study.

References

Ahmed, S. (2004). Mean-risk objectives in stochastic programming. Technical report, Georgia Institute of Technology. E-print available at http://www.optimization-online.org.

Ahmed, S. (2006). Convexity and decomposition of mean-risk stochastic programs. *Mathematical Programming*, 106(3):433–446.

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

Barbarosoğlu, G. and Arda, Y. (2004). A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the Operational Research Society*, 55:43–53.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variable programming problems. *Numerische Mathematik*, 54:238–252.

Birbil, S. I., Frenk, J., Kaynar, B., and Noyan, N. (2009). *The VaR Implementation Handbook*, chapter Risk Measures and Their Applications in Asset Management, pages 311–337. The McGraw-Hill Companies, New York. editor: G. N. Gregoriou.

Birge, J. and Louveaux, F. (1997). *Introduction to stochastic programming.* Springer, New York.

Birge, J. R. and Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34:384–392.

Chen, G., Daskin, M., Shen, M., and S.Uryasev (2006). The alpha-reliable mean-excess regret model for stochastic facility location modeling. *Naval Research Logistics*, 53(7):617–626.

Fábián, C. I. (2008). Handling cvar objectives and constraints in two-stage stochastic models. *European Journal of Operational Research*, 191(3):888–911.

Fourer, R., Gay, D., and Kernighan, B. (2003). *AMPL: A Modeling Language for Mathematical Programming Second Edition.* Duxbury Press Brooks Cole Publishing Co.

Gassmann, H. I. (1990). Mslip: a computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, 47:407–423.

G.Pflug (2000). *Probabilistic Constrained Optimization: Methodology and Applications*, chapter Chapter

1: Some remarks on the Value-at-Risk and the conditional Value-at-Risk. Kluwer Academic Publishers, Dordrecht. editor: Uryasev, S.

ILOG (2008). *ILOG AMPL CPLEX System Version 11 User's Guide*. ILOG CPLEX Division.

Klein Haneveld, W. K. and Van der Vlerk, M. H. (2006). Integrated chance constraints: Reduced forms and an algorithm. *Computational Management Science*, 3:245–269.

Künzi-Bay, A. and Mayer, J. (2006). Computational aspects of minimizing conditional value-at-risk. *Computational Management Science*, 3:3–27.

Laporte, G. and Louveaux, F. (1993). The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142.

Liu, C., Fan, Y., and F.Ordóñez (2009). A two-stage stochastic programming model for transportation network protection. *Computers and Operations Research*, 36:1582 – 1590.

Markowitz, H., M. (1952). Portfolio selection. *Journal of Finance*, 7(1):77–91.

Miller, N. and Ruszczyński, A. (2009). Risk-averse two-stage stochastic linear programming: Modeling and decomposition. Optimization Online Digest August.

Müller, A. and Stoyan, D. (2002). *Comparison Methods for Stochastic Models and Risks*. John Wiley & Sons, Chichester.

Ogryczak, W. and Ruszczyński, A. (1999). From stochastic dominance to mean-risk models: semideviations as risk measures. *European Journal of Operational Research*, 116:33–50.

Ogryczak, W. and Ruszczyński, A. (2002). Dual stochastic dominance and related mean-risks models. *SIAM journal of optimization*, 13(2):60–78.

Prékopa, A. (1995). *Stochastic Programming*. Kluwer Academic, Dordrecht, Boston.

Rawls, C. G. and Turnquist, M. A. (2010). Pre-positioning of emergency supplies for disaster response. *Transportation Research Part B: Methodological*, 44(4):521–534.

Rockafellar, R. and Uryasev, S. (2000). Optimization of conditional value at risk. *The journal of risk*, 2(3):21–41.

Rockafellar, R. and Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26:1443–1471.

Ruszczyński, A. (1986). A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35:309–333.

Ruszczyński, A. and Shapiro, A. (2006). *Probabilistic and Randomized Methods for Design under Uncertainty*, chapter Optimization of Risk Measures, pages 119–157. Springer-Verlag, London. editors: G. Calafiore and F. Dabbene.

Ruszczyński, A. and Świetanowski, A. (1996). On the regularized decomposition method for two stage stochastic linear problems. Technical report, International Institute for Applied Systems Anlaysis, WP-96-014.

Schultz, R. and Neise, F. (2007). Algorithms for mean-risk stochastic integer programs in energy. *Revista Investigación Operational*, 28:4–16.

Schultz, R. and Tiedemann, S. (2006). Conditional value-at-risk in stochastic programs with mixed-integer recourse. *Mathematical Programming*, 105(2):365–386.

Shaked, M. and Shanthikumar, J. G. (1994). *Stochastic Orders and their Applications.* Associated Press, Boston.

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