New algorithmic framework for conditional value at risk: application to stochastic fixed-charge transportation

Elena Fernández[a], Yolanda Hinojosa[b], Justo Puerto[c], Francisco Saldanha-da-Gama[d]

[a]*Departamento d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, Barcelona, Spain*
[b]*IMUS and Departamento de Economía Aplicada I, Universidad de Sevilla, Spain*
[c]*IMUS and Departamento Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Sevilla, Spain*
[d]*Departamento de Estatística e Investigação operacional and Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Portugal*

Abstract

This paper introduces a new algorithmic scheme for two-stage stochastic mixed-integer programming assuming a risk averse decision maker. The focus is the minimization of the conditional value at risk for a hard combinatorial optimization problem. Some properties of a mixed-integer non-linear programming formulation for conditional value at risk are studied as well as their algorithmic implications. This yields to a procedure for obtaining lower and upper bounds on the optimal value of the problem that may lead to an optimal solution. The new developments are applied to a fixed-charge transportation problem with stochastic demand, and they are computationally tested. The corresponding results are thoroughly presented and discussed.

Keywords: Transportation, Stochastic mixed-integer programming, CVaR

1. Introduction

In this work we propose an algorithmic scheme for two-stage stochastic mixed-integer programming assuming a risk averse decision maker.

As recently pointed out by Escudero et al. (2017), many risk measures can be found in the literature, among which the most popular ones are scenario immunization, semi-deviation, min-risk, stochastic dominance, value-at-risk, and the conditional value-at-risk (CVaR) (see e.g., Chelst and Canbolat, 2011, for further details on these measures). The two latter measures are possibly the most widely used in the literature. The risk measure adopted in this work is the CVaR (Uryasev and Rockafellar, 2001).

Stochastic Programming has become a very important tool to hedge against uncertainty when compared to approaches derived from simplified (deterministic) counterpart models (Birge and Louveaux 2011, Wallace and Ziemba 2005). Due to applications arising in many relevant fields (e.g., logistics), an increasing amount of work is being done on two-stage, and multi-stage, stochastic mixed-integer linear programming (e.g., Santoso et al., 2005; Hinojosa et al., 2014; Nickel et al., 2012).

Initially, stochastic programming literature focused predominantly on risk-neutral decision makers, i.e., the future assets are accounted for using a mean-type objective function. More recently this trend has changed and there exists an increasing interest on models capturing a risk-averse attitude. To the best of our knowledge, the first work considering a mean-risk objective function in two-stage stochastic programming is Ahmed (2006), where the complexity of mean-risk stochastic programming considering different risk measures was investigated. Fábián (2008) studied two-stage CVaR minimization problems (as well as CVaR-constrained problems) and devised decomposition and solution schemes based on the linear programming representation of CVaR proposed by Künzi-Bay and Mayer (2006). Also in the context of two-stage stochastic linear programming problems, Miller and Ruszczyński (2011) consider an objective function combining several conditional risk measures.

For the case of two-stage stochastic mixed-integer linear programming problems, Schultz and Tiedemann (2006) presented an explicit mixed-integer linear programming formulation for the CVaR, when the probability distribution is discrete and finite. For the same problem Carøe and Schultz (1999) devised a dual decomposition approach by relaxing the non-anticipativity constraints.

The use of risk measures in two-stage stochastic mixed-integer linear programming problems has been considered in several applications. Noyan (2012) proposed a CVaR mean-risk model in the context of disaster relief, which was solved with a variation of the L-Shaped method (Van Slyke and Wets, 1969). In the context of retrofitting infrastructures in a transport network to protect against disaster events (e.g., earthquake), Liu et al. (2009) studied a mean-risk two-stage stochastic programming model in which risk is handled via a semi-deviation measure.

Recently, Homem-de-Mello and Pagnoncelli (2016) discussed risk measures in multi-stage stochastic programming and potential applications. Also, Escudero et al. (2017) considered a mean-risk model using a so-called time stochastic dominance measure and proposed a scenario cluster Lagrangean decomposition approach. Finally, we refer to the work by Alonso-Ayuso et al. (2016) who presented a mean-risk multistage stochastic mixed-integer linear programming model for a forestry management problem.

In this work we focus on the minimization of the CVaR for a hard combinatorial optimization problem. We investigate a two-stage stochastic programming formulation, derived from the direct definition of CVaR, as an alternative to previous linear representations for the case when the random vector has a discrete distribution (Rockafellar and Uryasev, 2002; Schultz and Tiedemann, 2006). In particular, we consider a mixed-integer non-linear programming formulation and study some of its properties as well as their algorithmic implications. Given the combinatorial nature of the considered problem, the evaluation of each scenario requires solving a hard optimization problem, even if the number of scenarios is discrete. However, the properties of the proposed formulation can be exploited to derive a solution framework, as an alternative option to schemes based on linear representations.

The new algorithmic framework has been applied to a risk-averse stochastic fixed-charge transportation problem, which generalizes the risk-neutral objective studied in Hinojosa et al. (2014). This problem incorporates three main features to a standard transportation problem. First, it is assumed that a strategic decision has to be made concerning the distribution channels that will be activated for shipping demand between origin/destination pairs. Similarly to Balinski (1961), activating a distribution channel incurs a fixed cost. Second, variable costs associated with flows, both for handling them at the origins and for sending them through the activated network, are considered as well. This extends the fixed-charge transportation problem of Balinski (1961), which does not consider handling costs at the origins. Third, we assume that demand is stochastic (reflecting volatility of current markets leading to continuously changing demands), and consider the problem as a two-stage stochastic problem with recourse. The first stage is given by the strategic decision concerning the distribution channels and the recourse action refers to the tactical decision corresponding to the transportation pattern.

Examples of potential applications of such a problem are listed in Hinojosa et al. (2014) and arise in logistics systems in which previous contracts are needed in order to activate distribution channels (see, for instance, Xu and Nozick, 2009). Other examples can be found in telecommunications where traffic can be sent through some channel only if such channel has been previously made physically available (see Ahuja et al., 1993). Applications can also be found in production planning when a set of machines is available during a certain period of time for producing a set of products and one has to decide which machines will produce which products and in what quantities (see, for instance, Grieco et al., 2001, and the references therein). Finally, one potential area where the findings of this work can be applied concerns fleet management. In fact, most of the distribution companies must decide in advance about the fleet portfolio to use (type and number of vehicles) before knowing the demand. Some work in this direction includes List et al. (2003) and Meng et al. (2012).

It is also possible to find some work in the literature that investigates the use of CVaR in the context of logistics/transportation systems. Hemmati et al. (2016) highlights the usage of CVaR as a risk measure in a context of energy transportation. The authors propose a two-stage stochastic mixed integer linear programming. Uncertainty is associated with wind resources, i.e., it is associated with supply. The decisions are related with scheduling storage units. Toumazis and Kwon (2016) consider CVaR for routing decisions in the context of hazmat materials transportation. In this case, uncertainty is associated with the occurring accidents and their consequences. The concept of worst case CVaR is introduced as a way to hedge against data inaccuracy. A robust optimization modeling framework is adopted.

In this paper we develop an algorithmic framework for a two-stage stochastic combinatorial optimization problem, with a finite set of scenarios, assuming a CVaR objective. The solution algorithm exploits some properties of a mixed-integer non-linear programming formulation, derived from the direct definition of CVaR. Alternative solution strategies are applied and computationally tested for a fixed-charge stochastic transportation problem that generalizes the problem studied in Hinojosa et al. (2014). The obtained numerical results are thoroughly discussed and analyzed. These results show that our proposal outperforms alternative methods for the largest tested instances.

The remainder of this paper is organized as follows. Section 2 recalls the concepts of CVaR on risk-averse decision making as well as its computation via the mathematical programming representation developed in Uryasev (2000), Rockafellar and Uryasev (2002) and Schultz and Tiedemann (2006). In Section 3 we introduce the mixed integer non-linear formulation for CVaR that we use, we study some of its properties, and present the proposed solution framework. Section 4 focuses on its application to the CVaR stochastic fixed-charge transportation problem. The numerical results of the computational tests executed with our solution proposal are presented and analyzed in Section 5. The paper ends in Section 6 with some conclusions and guidelines for future research.

2. Models for risk-averse decision making

Consider a two-stage stochastic problem, where stochasticity is associated with a random vector $\boldsymbol{\xi}$ and is expressed via a finite set of possible scenarios Ω. Let π_ω, $\omega \in \Omega$, denote the probabilities of the different scenarios. In two-stage stochastic problems an *a priori* (first-stage) solution must be defined before the actual realization of $\boldsymbol{\xi}$ is known. We denote by Y the domain for *a priori* solutions. The second-stage *recourse action* associated with a given first-stage decision $\mathbf{y} \in Y$ represents the set of decisions to be made for each possible realization of the uncertain vector $\boldsymbol{\xi}$. Let $R(\mathbf{y}, \boldsymbol{\xi}_\omega)$ denote the cost of the recourse action under scenario $\omega \in \Omega$. Computing $R(\mathbf{y}, \boldsymbol{\xi}_\omega)$ requires to solve some associated optimization problem on the set of second-level variables. Moreover, when an *a priori* solution $\mathbf{y} \in Y$ is considered, it is not possible to compute the actual cost of the recourse action. The reason is that when the *a priori* solution $\mathbf{y} \in Y$ is chosen, the specific scenario $\omega \in \Omega$ that will actually occur in the realization of $\boldsymbol{\xi}$ is unknown. In this context it is natural to consider some objective function that incorporates joint information of all possible realizations. One such objective for the case of risk-averse decision making is the *value-at-risk* (VaR) that we revisit below.

Let $\mathcal{R}(\mathbf{y};\boldsymbol{\xi}) = f(\mathbf{y}) + R(\mathbf{y},\boldsymbol{\xi})$ be the cost associated with an *a priori* solution $\mathbf{y} \in Y$, where $f(\mathbf{y})$ represents the here-and-now cost, and $R(\mathbf{y},\boldsymbol{\xi})$ the value of the recourse function. The latter is a random variable representing the cost of the optimal second-stage decision for the *a priori* solution and thus, $\mathcal{R}(\mathbf{y};\boldsymbol{\xi})$ is also a random variable. For a given confidence level α, the VaR is defined as the α-quantile of the probability distribution of $\mathcal{R}(\mathbf{y};\boldsymbol{\xi})$. That is, the VaR of an *a priori* solution $\mathbf{y} \in Y$, denoted by $\eta_\alpha(\mathbf{y})$, is the value of the minimum threshold η for which $\mathbb{P}\left[\mathcal{R}(\mathbf{y};\boldsymbol{\xi}) \leq \eta\right]$ is at least α. More formally,

$$\eta_\alpha(\mathbf{y}) = \min\left\{\eta \mid \mathbb{P}\left[\mathcal{R}(\mathbf{y};\boldsymbol{\xi}) \leq \eta\right] \geq \alpha\right\}.$$

Using as objective the minimization of the VaR for a given confidence level α, that is, $\min_{y\in Y} \eta_\alpha(\mathbf{y})$, the decision maker aims at minimizing the threshold for the risk level $1-\alpha$. Still, for the solution $\mathbf{y}$ obtained with this objective, the value of $\mathcal{R}(\mathbf{y};\boldsymbol{\xi})$ may be too high, for some of the scenarios whose cost exceeds this minimum threshold, even if their joint probability is very low (smaller than or equal to $1-\alpha$). An alternative that takes into account this consideration is to minimize the expected cost of the scenarios whose cost exceeds the threshold for a fixed confidence level α. This leads to the minimization of the $\alpha-$*Conditional Value at Risk* (α-CVaR), which is precisely the objective that we consider in this work and we formally define below.

The α-CVaR of an *a priori* solution $\mathbf{y} \in Y$, $\Psi_\alpha(\mathbf{y})$, is the expected value of the total cost in the $(1-\alpha)\times 100\%$ worse scenarios for the *a priori* solution $\mathbf{y}$, i.e., it is the expected cost conditional to the scenarios whose value exceeds $\eta_\alpha(\mathbf{y})$. Note that the α-CVaR generalizes the expected cost objective, which is a particular case when all the scenarios come into play. Thus, when α is suitably chosen the α-CVaR of an *a priori* solution $\mathbf{y}$ is just the expected value of $\mathcal{R}(\mathbf{y};\boldsymbol{\xi})$ (see Puerto et al., 2017).

For a given solution $\mathbf{y} \in Y$, it is difficult to express $\Psi_\alpha(\mathbf{y})$, unless an analytical representation for $\eta_\alpha(\mathbf{y})$ is known, because the latter is involved in the definition of the former. One possibility (see Uryasev, 2000; Rockafellar and Uryasev, 2002, for further details) is to consider the following function:

$$\Phi_\alpha(\mathbf{y},\eta) = \eta + \frac{1}{1-\alpha}\mathbb{E}\left[(\mathcal{R}(\mathbf{y};\boldsymbol{\xi}) - \eta) \mid \mathcal{R}(\mathbf{y};\boldsymbol{\xi}) > \eta\right],$$

which, when uncertainty is expressed by a finite set of scenarios reduces to

$$\Phi_\alpha(\mathbf{y},\eta) = \eta + \frac{1}{1-\alpha}\sum_{\omega\in\Omega}\left(\mathcal{R}(\mathbf{y};\boldsymbol{\xi}_\omega) - \eta\right)^+\pi_\omega, \tag{1}$$

with $a^+ = \max\{0, a\}$. Then, the α-CVaR of $\mathbf{y} \in Y$ can be computed as:

$$\Psi_\alpha(\mathbf{y}) = \Phi_\alpha(\mathbf{y},\eta(\mathbf{y})) = \min_\eta \Phi_\alpha(\mathbf{y},\eta).$$

Thus, the problem of finding a feasible vector $\mathbf{y} \in Y$ with the smallest α-CVaR reduces to:

$$\mathrm{CVaR}_\alpha \qquad \min_{\mathbf{y}\in Y} \Psi_\alpha(\mathbf{y}) = \min_{\mathbf{y}\in Y} \Phi_\alpha(\mathbf{y}, \eta(\mathbf{y})) = \min_{\mathbf{y}\in Y,\, \eta}, \Phi_\alpha(\mathbf{y}, \eta), \tag{2}$$

which, taking into account (1), can be expressed as:

$$\mathrm{CVaR}_\alpha \qquad \text{minimize} \quad \eta + \frac{1}{1-\alpha} \sum_{\omega\in\Omega} \pi_\omega \psi_\omega \tag{3}$$

$$\text{subject to} \quad \mathbf{y} \in Y, \tag{4}$$

$$\psi_\omega \geq \mathcal{R}(\mathbf{y}; \boldsymbol{\xi}_\omega) - \eta, \quad \omega \in \Omega, \tag{5}$$

$$\psi_\omega \geq 0, \quad \omega \in \Omega. \tag{6}$$

Note that the meaning of the ψ_ω variables in optimal solutions to the above formulation is precisely $\psi_\omega = (\mathcal{R}(\mathbf{y}; \boldsymbol{\xi}_\omega) - \eta)^+$, $\omega \in \Omega$. When the domain Y can be expressed by means of a set of linear constraints, the above formulation is a valid mathematical programming formulation, linear on the decision variables $\mathbf{y}$.

It is possible to derive an alternative formulation for CVaR by observing that when $\mathbf{y} \in Y$ is fixed, its α-CVaR, $\Psi_\alpha(\mathbf{y})$, can be expressed just in terms of $\mathbf{y}$ (Rockafellar and Uryasev, 2002). In order to make the paper self-contained we include the reformulation in which we base our algorithmic framework.

For a given *a priori* solution $\mathbf{y} \in Y$, let us solve the associated second-stage problem for each realization, $\boldsymbol{\xi}_\omega$ $\omega \in \Omega$, of the uncertain vector $\boldsymbol{\xi}$, then sort the scenarios by non-increasing values of $\mathcal{R}(\mathbf{y}; \boldsymbol{\xi}_\omega)$, and finally index them accordingly. To alleviate notation, if $\omega_r(\mathbf{y})$ is the index of the r-th sorted scenario we will write $\boldsymbol{\xi}_{r(\mathbf{y})}$ and $\pi_{r(\mathbf{y})}$ instead of $\boldsymbol{\xi}_{\omega_r(\mathbf{y})}$ and $\pi_{\omega_r(\mathbf{y})}$, respectively. Hence,

$$\mathcal{R}(\mathbf{y}; \boldsymbol{\xi}_{1(\mathbf{y})}) \geq \ldots \geq \mathcal{R}(\mathbf{y}; \boldsymbol{\xi}_{|\Omega|(\mathbf{y})}),$$

with probabilities $\pi_{1(\mathbf{y})}, \ldots, \pi_{|\Omega|(\mathbf{y})}$, respectively. Let $\ell(\mathbf{y})$ be the index such that

$$\pi_{1(\mathbf{y})} + \ldots + \pi_{\ell(\mathbf{y})-1} \leq 1 - \alpha \tag{7}$$

$$\pi_{1(\mathbf{y})} + \ldots + \pi_{\ell(\mathbf{y})} > 1 - \alpha. \tag{8}$$

Then, according to Rockafellar and Uryasev (2002), $\eta_\alpha(\mathbf{y}) = \mathcal{R}(\mathbf{y}; \boldsymbol{\xi}_{\ell(\mathbf{y})})$.

Let $\Omega(\mathbf{y}) = \{\omega_r(\mathbf{y}) \in \Omega \mid 1 \leq r \leq \ell(\mathbf{y})\}$ and let $\boldsymbol{\beta}(\mathbf{y})$ be the weight vector induced by

$\Omega(\mathbf{y})$, whose r-th component is given by

$$\beta_{r(\mathbf{y})} = \begin{cases} \frac{\pi_{r(\mathbf{y})}}{1-\alpha} & \text{if} \quad \omega_r(\mathbf{y}) \in \Omega(\mathbf{y}) \setminus \{\omega_{\ell(\mathbf{y})}\} \\ 1 - \frac{\sum_{\omega \in \Omega(\mathbf{y}) \setminus \{\omega_{\ell(\mathbf{y})}\}} \pi_\omega}{1-\alpha} & \text{if} \quad \omega_r(\mathbf{y}) = \omega_{\ell(\mathbf{y})} \\ 0 & \text{if} \quad \omega_r(\mathbf{y}) \in \Omega \setminus \Omega(\mathbf{y}). \end{cases} \tag{9}$$

Since the non-zero components of $\boldsymbol{\beta}(\mathbf{y})$ determine the conditional probabilities for the scenarios in $\Omega(\mathbf{y})$, the α-CVaR of $\mathbf{y}$ can be expressed as:

$$\vartheta(\boldsymbol{\beta}(\mathbf{y}), \mathbf{y}) := f(\mathbf{y}) + \sum_{r=1}^{|\Omega|} \beta_{r(\mathbf{y})} R(\mathbf{y}, \boldsymbol{\xi}_{r(\mathbf{y})}). \tag{10}$$

As pointed out by Fábián (2008) the above $\boldsymbol{\beta}(\mathbf{y})$ vectors correspond to the dual optimal solution of the linear programming problem emerging from (3)–(6) when $\mathbf{y} \in Y$ is fixed. Such weights can also be looked at as the optimal solution of the linear relaxation of a knapsack problem. This is an aspect that has been explored in the context of k-sum optimization by Bertsimas and Sim (2003, 2004); Puerto et al. (2017); and Punen (1992).

3. Algorithmic framework for minimizing CVaR

In this section we exploit the results of the above section to derive our algorithmic framework for the CVaR. Note that the definition of the CVaR associated with a given *a priori* solution $\mathbf{y} \in Y$ produces the following non-linear formulation for CVaR_α:

$$\text{CVaR}_\alpha \qquad \min \ \vartheta(\boldsymbol{\beta}(\mathbf{y}), \mathbf{y}) := f(\mathbf{y}) + \sum_{r=1}^{|\Omega|} \beta_{r(\mathbf{y})} R(\mathbf{y}, \boldsymbol{\xi}_{r(\mathbf{y})}) \tag{11}$$

$$\text{s. t.} \qquad \mathbf{y} \in Y. \tag{12}$$

One difficulty of this formulation, which is independent of the representation of the domain Y, is the non-linearity of the objective function. This can be partially overcome by using a fixed weight vector $\boldsymbol{\beta} \in \mathbb{R}^{|\Omega|}$. The resulting subproblem is:

$$\mathcal{H}(\boldsymbol{\beta}) = \min \ f(\mathbf{y}) + \sum_{\omega \in \Omega} \beta_\omega R(\mathbf{y}, \boldsymbol{\xi}_\omega) \tag{13}$$

$$\text{s. t.} \qquad \mathbf{y} \in Y. \tag{14}$$

Of course, there is no guarantee that when $\boldsymbol{\beta} \in \mathbb{R}^{|\Omega|}$ is fixed, $\mathcal{H}(\boldsymbol{\beta})$ produces an optimal solution to CVaR_α. However, as we will see, when $\boldsymbol{\beta} \in \mathbb{R}^{|\Omega|}$ is suitably chosen, from an optimal solution to $\mathcal{H}(\boldsymbol{\beta})$ we can derive both a lower and an upper bound on the optimal value of CVaR_α. In this context, a key issue is how to select appropriate weight vectors $\boldsymbol{\beta} \in \mathbb{R}^{|\Omega|}$ that provide useful bounds. For this we will choose suitable indices $\ell \in \{1, \ldots |\Omega|\}$ and scenario sets $\Omega^\ell \subseteq \Omega$ with $\omega_\ell \in \Omega^\ell$ that jointly satisfy

$$\sum_{\omega \in \Omega^\ell \setminus \{\omega_\ell\}} \pi_\omega \leq 1 - \alpha \tag{15}$$

$$\sum_{\omega \in \Omega^\ell} \pi_\omega \quad > 1 - \alpha. \tag{16}$$

In the following such subsets Ω^ℓ *are referred to as ℓ-tails for α.* Any ℓ-tail for α naturally induces the following vector $\boldsymbol{\beta}$ that we refer to as *ℓ-weight vector*:

$$\beta_\omega = \begin{cases} \frac{\pi_\omega}{1-\alpha} & \omega \in \Omega^\ell \setminus \{\omega_\ell\} \\ 1 - \frac{\sum_{\omega \in \Omega^\ell \setminus \{\omega_\ell\}} \pi_\omega}{1-\alpha} & \omega = \omega_\ell \\ 0 & \omega \in \Omega \setminus \Omega^\ell. \end{cases} \tag{17}$$

The particular case $\Omega^\ell = \Omega(\mathbf{y})$ is referred to as *scenario subset* induced by $\mathbf{y} \in Y$, and the ℓ-weight vector $\boldsymbol{\beta} = \boldsymbol{\beta}(\mathbf{y})$ is called *scenario vector* induced by $\Omega(\mathbf{y})$.

In principle, any ℓ-weight vector induced by a subset of scenarios defining an ℓ-tail, or any other vector in the convex hull of such vectors, can be useful choices. In our framework we will indistinctively use both weight vectors induced by generic ℓ-tails as well as scenario vectors.

Remark 1. *The expected conditional cost of an* a priori *solution* $\mathbf{y}$ *relative to any subset of scenarios defining an ℓ-tail, $\Omega^\ell \neq \Omega(\mathbf{y})$, cannot exceed the α-CVaR of* $\mathbf{y}$*. This includes the scenario subsets $\Omega(\mathbf{y}')$ induced by different* a priori *solutions $\mathbf{y}' \in Y$, $\mathbf{y}' \neq \mathbf{y}$. This is due to the sorting used to define the scenario subset $\Omega(\mathbf{y})$, which guarantees that this subset is precisely the set of* worse *scenarios for the α-quantile for* $\mathbf{y}$*, and to the fact that $\boldsymbol{\beta}(\mathbf{y})$ defines the conditional probabilities for this set of scenarios.*

In what follows we will denote by $\mathbf{u}$ the set of the second-stage variables. We further denote by $Q(\mathbf{y})$ the set of feasible solutions $\mathbf{u}$ associated with a given $\mathbf{y} \in Y$, and by $Q = \{(\mathbf{y}, \mathbf{u}) \mid \mathbf{u} \in Q(\mathbf{y}),\ \mathbf{y} \in Y\}$ the feasible domain for joint vectors. We assume that the problem we are dealing with has relatively complete recourse, i.e., $Q(\mathbf{y}) \neq \emptyset$, for all $\mathbf{y} \in Y$. This is motivated by the application we present in Section 4. When $\boldsymbol{\beta} = \boldsymbol{\beta}(\mathbf{y})$ is the scenario vector induced by $\Omega(\mathbf{y})$ for some $\mathbf{y} \in Y$, $\mathcal{H}(\boldsymbol{\beta})$ can be interpreted as the problem of finding the best solution $\mathbf{q} = (\mathbf{y}', \mathbf{u}) \in Q$ for the scenario set induced by $\mathbf{y}$. Since $\mathbf{y}'$ does not necessarily coincide with $\mathbf{y}$, this perspective is complementary to that of the computation of the scenario vector $\boldsymbol{\beta}(\mathbf{y})$, which, is to find the set of scenarios that define the actual CVaR_α for a given solution $\mathbf{y}$.

Next we analyze some of the properties of subproblems $\mathcal{H}(\boldsymbol{\beta})$. First we note that if $\mathbf{q} = (\mathbf{y}, \mathbf{u})$ is an optimal solution to $\mathcal{H}(\boldsymbol{\beta})$, then $\mathbf{q} \in Q$, since otherwise it would not be optimal. Let us denote by $v(\boldsymbol{\beta}, \mathbf{q})$ the objective function value (13) for solution $\mathbf{q} = (\mathbf{y}, \mathbf{u}) \in Q$ for $\mathcal{H}(\boldsymbol{\beta})$. Note that $v(\boldsymbol{\beta}(\mathbf{y}), \mathbf{q}) = \vartheta(\boldsymbol{\beta}(\mathbf{y}), \mathbf{y})$.

Remark 2.

(*i*) *The domain of $\mathcal{H}(\boldsymbol{\beta})$ is exactly the same as that of CVaR$_\alpha$, and only their objective functions are different. Thus, any feasible solution to $\mathcal{H}(\boldsymbol{\beta})$ produces a valid upper bound to CVaR$_\alpha$, provided that it is evaluated with respect to the function $\vartheta(\boldsymbol{\beta}(\mathbf{y}), \mathbf{y})$. Therefore, for any $\mathbf{q} = (\mathbf{y}, \mathbf{u}) \in Q$, $v(\boldsymbol{\beta}(\mathbf{y}), \mathbf{q})$ is an upper bound on the optimal value to CVaR$_\alpha$.*

(*ii*) *Let $\mathbf{q} = (\mathbf{y}, \mathbf{u}) \in Q$, with associated scenario vector $\boldsymbol{\beta}(\mathbf{y})$. Then, $v\left(\boldsymbol{\beta}', \mathbf{q}\right) \leq v(\boldsymbol{\beta}(\mathbf{y}), \mathbf{q})$ for any ℓ-weight vector $\boldsymbol{\beta}' \neq \beta(\mathbf{y})$. This property is just a restatement of Remark 1, taking into account that if $\boldsymbol{\beta}' \neq \boldsymbol{\beta}(\mathbf{y})$, then the ℓ-tail that induces $\boldsymbol{\beta}'$ will also be different from $\Omega(\mathbf{y})$.*

Below we prove that when $\boldsymbol{\beta}$ is an ℓ-weight vector the optimal value to $\mathcal{H}(\boldsymbol{\beta})$ gives a valid lower bound for CVaR$_\alpha$.

Proposition 1. *Let v^* be the optimal value to CVaR$_\alpha$. Let also $\bar{\mathbf{q}} = (\bar{\mathbf{y}}, \bar{\mathbf{u}}) \in Q$ be an optimal solution to $\mathcal{H}(\boldsymbol{\beta})$ for an ℓ-weight vector $\boldsymbol{\beta}$. Then, $v(\boldsymbol{\beta}, \bar{\mathbf{q}}) \leq v^*$.*

Proof. Let $\mathbf{q}^* = (\mathbf{y}^*, \mathbf{u}^*) \in Q$ denote an optimal solution to CVaR$_\alpha$ with associated scenario vector $\boldsymbol{\beta}^* = \boldsymbol{\beta}(\mathbf{y}^*)$. Consider also an optimal solution to $\mathcal{H}(\boldsymbol{\beta})$, $\bar{\mathbf{q}} = (\bar{\mathbf{y}}, \bar{\mathbf{u}}) \in Q$, with value $v(\boldsymbol{\beta}, \bar{\mathbf{q}})$. Then

$$v(\boldsymbol{\beta}, \bar{\mathbf{q}}) \leq v(\boldsymbol{\beta}, \mathbf{q}^*) \leq v(\boldsymbol{\beta}^*, \mathbf{q}^*) = v^*,$$

where the first inequality follows from the optimality of $\bar{\mathbf{q}}$ for $\mathcal{H}(\boldsymbol{\beta})$ and the second one from the definition of conditional value at risk of solution $\mathbf{q}^*$ and Remark 2.(ii).

□

The next result allows to determine if a given solution $\bar{\mathbf{q}} = (\bar{\mathbf{y}}, \bar{\mathbf{u}}) \in Q$ is optimal for CVaR$_\alpha$.

Proposition 2. *Let $\bar{\mathbf{q}} = (\bar{\mathbf{y}}, \bar{\mathbf{u}}) \in Q$ be an optimal solution to $\mathcal{H}(\bar{\boldsymbol{\beta}})$ for an ℓ-weight vector $\bar{\boldsymbol{\beta}}$. If the scenario vector induced by $\Omega(\bar{\mathbf{y}})$ coincides with $\bar{\boldsymbol{\beta}}$, i.e. $\boldsymbol{\beta}(\bar{\mathbf{y}}) = \bar{\boldsymbol{\beta}}$, then $\bar{\mathbf{q}}$ is an optimal solution to CVaR$_\alpha$.*

Proof. Let $\bar{\mathbf{q}} = (\bar{\mathbf{y}}, \bar{\mathbf{u}}) \in Q$ be an optimal solution to $\mathcal{H}(\bar{\boldsymbol{\beta}})$ for the ℓ-weight vector $\bar{\boldsymbol{\beta}}$. By Proposition 1, $v\left(\bar{\boldsymbol{\beta}}, \bar{\mathbf{q}}\right) \leq v^*$. On the other hand, by Remark 2.(i), $v^* \leq v(\boldsymbol{\beta}(\bar{\mathbf{y}}), \bar{\mathbf{q}})$. Finally, since $\boldsymbol{\beta}(\bar{\mathbf{y}}) = \bar{\boldsymbol{\beta}}$ we have,

$$v(\boldsymbol{\beta}(\bar{\mathbf{y}}), \bar{\mathbf{q}}) = v\left(\bar{\boldsymbol{\beta}}, \bar{\mathbf{q}}\right) \leq v^* \leq v(\boldsymbol{\beta}(\bar{\mathbf{y}}), \bar{\mathbf{q}}),$$

and the result follows.

□

The above results allow us to propose a solution algorithm that solves a sequence of subproblems $\mathcal{H}(\boldsymbol{\beta^k})$ for a series of weight vectors $\boldsymbol{\beta^k}$. As we have already pointed out there is no guarantee that when $\boldsymbol{\beta} \in \mathbb{R}^{|\Omega|}$ is fixed, $\mathcal{H}(\boldsymbol{\beta})$ produces an optimal solution to CVaR_α. However, as we have seen above, from an optimal solution to $\mathcal{H}(\boldsymbol{\beta})$ we can derive both a lower and an upper bound on the optimal value of CVaR_α. Moreover, Proposition 2 gives a termination criterion for such an iterative algorithm, certifying optimality. A pseudocode of the above algorithmic scheme is given in Algorithm 1.

Algorithm 1 Algorithmic scheme for CVaR_α.

```
// Initialization
y^0 ← Any feasible vector y;
β^0 ← β(y^0)
UB ← ∞;    LB ← 0
k ← 1
repeat
  Determine y^k solving (13)–(14)
  Update UB and LB
  β^k ← β(y^k)
  k ← k + 1
until stopping criterion met
```

The new algorithmic framework may be particularly suitable for problems of combinatorial nature in which an appropriate representation of the domain Y is not available, so the evaluation of $\mathcal{H}(\boldsymbol{\beta^k})$ remains a hard optimization problem. This is the approach that we propose and we apply for the CVaR Transportation Problem that we study in the following section.

4. The Conditional Value at Risk Transportation Problem

In this section we define the Conditional Value at Risk Transportation Problem ($\text{CVaR}_\alpha\text{TP}$) which extends the single-commodity variant of the expected cost stochastic transportation problem studied in Hinojosa et al. (2014).

As we show below, by considering the minimization of CVaR we face a more involved mathematical model than when considering an expected cost objective. This is explained by the fact that the expected value to be computed now involves only a subset of scenarios that, in turn, depends on the first-stage decision to make. Therefore, in addition to the first-stage decisions, already considered by Hinojosa et al. (2014), we now need to identify the $(1-\alpha) \times 100\%$ worst scenarios which calls for sorting the second-stage costs. This implies that the methodology and algorithms applied in that paper are no longer valid for our current problem.

Next we introduce some notation and describe the feasible domain for the $\text{CVaR}_\alpha\text{TP}$. Then, we adapt formulations (3)–(6) and (11)–(12) to it.

Let I and J denote some given index sets for origins and destinations, respectively. In order to satisfy some existing demand located at the destinations, some product can be shipped from the origins to the destinations. Each origin has a limited capacity, that represents the maximum amount of product that it can distribute to demand points.

Sending flow from an origin to a destination incurs two types of costs: a fixed set-up cost for activating the connection (link) and a variable cost, which depends on the amount of flow that is sent through the link. Activating any link starting at a given origin $i \in I$ incurs an additional fixed cost due to the set-up of handling operations at the origin. It is possible that the flows that are routed do not meet the demand at some destination point, incurring in such a case some additional cost (e.g. loss opportunity cost).

The following deterministic parameters are known:

c_{ij}	Set-up cost for link (i,j), $i \in I$, $j \in J$.
d_{ij}	Unit cost for the flow through link (i,j), $i \in I$, $j \in J$.
k_i	Maximum amount that can be distributed from (or be handled at) origin $i \in I$.
h_i	Fixed handling cost at origin $i \in I$. We assume that not all these values can be equal to 0.
p_j	Unit penalty for unmet demand at destination $j \in J$ (e.g., loss opportunity cost).

In the CVaR$_\alpha$TP stochasticity is associated with demand, and expressed by a finite set of possible scenarios Ω with probabilities π_ω, $\omega \in \Omega$. We denote by D_j the demand of customer $j \in J$ and by $\boldsymbol{\xi} = [D_j]_{j \in J}$ the associated random vector.

A priori solutions for the CVaR$_\alpha$TP are given by a set of distribution links that are activated. The cost of an *a priori* solution is the overall set-up cost for the selected links. After demand is revealed, the recourse action is to identify the origins that will actually operate, and to decide the quantity of product to ship from each such origin to each demand destination, using the links activated in the *a priori* solution. Hence, broadly speaking, feasible solutions to the CVaR$_\alpha$TP consist of (i) the links that are be activated in the *a priori* solution plus, (ii) the origins that will operate, and (iii) the flows that will be sent via the activated links, in the second-stage.

A priori decisions can be represented by means of the following set of decision variables:

$$y_{ij} = \begin{cases} 1 & \text{if link } (i,j) \text{ is activated} \\ 0 & \text{otherwise} \end{cases} \qquad i \in I,\ j \in J.$$

According to the notation used in Section 2 the domain of the *a priori* solutions is $Y = \left\{\mathbf{y} \in \{0,1\}^{|I| \times |J|}\right\}$ and the cost of *a priori* solutions can be expressed in terms of the decision variables as $f(\mathbf{y}) = \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij}$.

Given a first-stage decision $\mathbf{y}$, the problem of finding a minimum cost second-stage decision for each possible scenario $\omega \in \Omega$ (equivalently, for each possible realization of the demand $\boldsymbol{\xi}$), can be formulated using the following sets of decision variables:

$$z_{i\omega} = \begin{cases} 1 & \text{if origin } i \text{ is used under scenario } \omega \\ 0 & \text{otherwise} \end{cases} \qquad i \in I,\, \omega \in \Omega,$$

$$x_{ij\omega} = \text{Flow to be shipped from } i \in I \text{ to } j \in J \text{ under scenario } \omega \in \Omega,$$

$$s_{j\omega} = \text{Amount missing at destination } j \in J \text{ under scenario } \omega \in \Omega.$$

The motivation for considering the s-variables is the need to compute the penalties for unmet demand. Nevertheless, the use of such variables can be avoided if shortages are considered as shipments made from a fictitious origin with enough capacity and a set-up cost equal to 0. In this case, the unit shipment costs from such fictitious origin must be defined as the unit penalty cost at destination. Hence, in the following we consider an extended set of origins $I \cup \{0\}$ where origin 0 is the fictitious one and has a (scenario-dependent) capacity given by $k_{0\omega} = \max\{\sum_{j \in J} D_{j\omega} - \sum_{i \in I} k_i;\ 0\}$, $\omega \in \Omega$. The associated set-up and routing costs are defined as $h_0 = 0$ and $d_{0j} = p_j$, $j \in J$, respectively. For ease of presentation, hereafter we use I to denote the extended set of origins, including the fictitious one. When the *new* set I contains the fictitious origin, part of the flow that arrives to a demand point may correspond to some some missing amount, as just explained. Now, the correct evaluation of the missing amount at a destination requires to impose that the overall flow arriving at each destination $j \in J$ is at least its demand D_j. Then, a minimum cost recourse action for the *a priory* solution $\mathbf{y}$ when the realization of the stochastic demand $\boldsymbol{\xi}$ corresponds to scenario $\omega \in \Omega$ can be formulated as follows:

$$R(\mathbf{y}, \boldsymbol{\xi}_\omega) = \quad \text{minimize} \quad \sum_{i \in I} h_i z_{i\omega} + \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij\omega}, \tag{18}$$

$$\text{subject to} \quad x_{ij\omega} \le D_{j\omega} y_{ij}, \qquad i \in I,\ j \in J, \tag{19}$$

$$\sum_{i \in I} x_{ij\omega} \ge D_{j\omega}, \qquad j \in J, \tag{20}$$

$$\sum_{j \in J} x_{ij\omega} \le k_i z_{i\omega}, \qquad i \in I, \tag{21}$$

$$x_{ij\omega} \ge 0 \qquad i \in I,\ j \in J, \tag{22}$$

$$z_{i\omega} \in \{0,1\}, \qquad i \in I. \tag{23}$$

Constraints (19) ensure that flows are shipped using distribution links previously activated. Constraints (20) guarantee that a sufficient amount of flow is shipped to each demand destination. The identification of the origins that are activated and their capacity limitations are imposed in (21). We somehow abuse notation by not making explicit the above-mentioned scenario-dependent capacity of the fictitious origin in its associated constraint (21). The domain of the decision variables is defined in (22)–(23). The objective (18) computes the cost of a recourse action for the *a priory* solution $\mathbf{y}$ under scenario $\omega \in \Omega$, as the set-up costs of the handling operations at the origins that are used plus the overall routing costs, which include the penalty costs for unmet demand.

Note that constraints (19) can be reinforced to the set of tighter constraints:

$$x_{ij\omega} \leq \overline{D}_{ij\omega} y_{ij}, \qquad i \in I,\ j \in J,\ \omega \in \Omega \tag{24}$$

where $\overline{D}_{ij\omega} = \min\{k_i, D_{j\omega}\}$. In the following we assume that this reinforcement is applied together with the following valid inequalities, which impose that the overall capacity of the activated origins must be enough to supply the total demand in each scenario $\omega \in \Omega$:

$$\sum_{i \in I} k_i z_{i\omega} \geq \sum_{j \in J} D_{j\omega}, \quad \omega \in \Omega. \tag{25}$$

Using again the notation of Section 3 we assume that the domain of the recourse function for a given *a priori* solution $\mathbf{y}$ is expressed by

$Q(\mathbf{y}) = \{\mathbf{u} = (\mathbf{z}, \mathbf{x}) \mid (18) - (25) \text{ are satisfied for the } \textit{a priori solution } \mathbf{y}\}$
and the domain for feasible joint vectors for the CVaR$_\alpha$TP is $Q = \{\mathbf{q} = (\mathbf{y}, \mathbf{z}, \mathbf{x})$ with $\mathbf{y} \in Y, (\mathbf{z}, \mathbf{x}) \in Q(\mathbf{y})\}$.

The adaptation of the CVaR$_\alpha$ formulation (3)–(6) to the CVaR$_\alpha$TP is:

$$\text{CVaR}_\alpha\text{TP} \qquad \text{minimize} \quad \eta + \frac{1}{1-\alpha} \sum_{\omega \in \Omega} \pi_\omega \psi_\omega, \tag{26}$$

$$\text{subject to} \quad \psi_\omega \geq \left(\sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} + \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij\omega} + \sum_{i \in I} h_i z_{i\omega} \right) - \eta, \quad \omega \in \Omega, \tag{27}$$

$$\psi_\omega \geq 0, \quad \omega \in \Omega, \tag{28}$$

$$(\mathbf{y}, \mathbf{z}, \mathbf{x}) \in Q \tag{29}$$

Since the domain Q is now represented by sets of linear constraints, the above formulation is a mixed integer linear program.

On the other hand, the adaptation of (11)–(12) to the CVaR$_\alpha$TP becomes:

$$\text{CVaR}_\alpha\text{TP} \quad \min \sum_{i\in I}\sum_{j\in J} c_{ij}y_{ij} + \sum_{r=1}^{|\Omega|} \beta_r(\mathbf{y}) \left(\sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij\omega_r} + \sum_{i\in I} h_i z_{i\omega_r} \right) \tag{30}$$

$$\text{subject to} \quad (\mathbf{y}, \mathbf{z}, \mathbf{x}) \in Q. \tag{31}$$

Due to the non-linearity of the objective function, formulation (30)–(31) is a mixed integer non-linear program. Furthermore, we have no closed expression for the weight vector $\boldsymbol{\beta}(\mathbf{y})$, which depends on $\mathbf{y}$. Therefore, one possibility to partially overcome such difficulty is to focus on the subproblem $\mathcal{H}(\boldsymbol{\beta})$ that arises when the weights are fixed to some constant vector $\boldsymbol{\beta} \in \mathbb{R}^{|\Omega|}$ suitably selected. Now $\mathcal{H}(\boldsymbol{\beta})$ becomes:

$$\mathcal{H}(\boldsymbol{\beta}) = \quad \min \sum_{i\in I}\sum_{j\in J} c_{ij}y_{ij} + \sum_{\omega\in\Omega} \beta_\omega \left(\sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij\omega} + \sum_{i\in I} h_i z_{i\omega} \right) \tag{32}$$

$$\text{subject to} \quad x_{ij\omega} \le \overline{D}_{j\omega} y_{ij}, \qquad i \in I,\ j \in J,\ \omega \in \Omega, \tag{33}$$

$$\sum_{i\in I} x_{ij\omega} \ge D_{j\omega}, \qquad j \in J,\ \omega \in \Omega, \tag{34}$$

$$\sum_{j\in J} x_{ij\omega} \le k_i z_{i\omega}, \qquad i \in I,\ \omega \in \Omega, \tag{35}$$

$$\sum_{i\in I} k_i z_{i\omega} \ge \sum_{j\in J} D_{j\omega}, \quad \omega \in \Omega, \tag{36}$$

$$y_{ij} \in \{0,1\}, \qquad i \in I,\ j \in J, \tag{37}$$

$$x_{ij\omega} \ge 0 \qquad i \in I,\ j \in J,\ \omega \in \Omega, \tag{38}$$

$$z_{i\omega} \in \{0,1\}, \qquad i \in I,\ \omega \in \Omega, \tag{39}$$

In the following, we specify the elements of Algorithm 1 in its application to the CVaR$_\alpha$TP. The initialization phase starts sorting the scenarios by decreasing values of their total demand $\sum_{j\in J} D_{j\omega}$. Thus, the initial constant vector $\boldsymbol{\beta}^0$ is computed as the ℓ-weight vector induced by the ℓ-tail obtained according to this sorting. In each iteration $k \ge 1$ we solve the problem $\mathcal{H}(\boldsymbol{\beta}^{\boldsymbol{k-1}})$ and we calculate a lower bound as $LB^k = v(\boldsymbol{\beta}^{\boldsymbol{k-1}}, \mathbf{q}^k)$ being $\mathbf{q}^k = (\mathbf{y}^k, \mathbf{z}^k, \mathbf{x}^k) \in Q$ an optimal solution of $\mathcal{H}(\boldsymbol{\beta}^{\boldsymbol{k-1}})$. Then, we update the scenario vector as $\boldsymbol{\beta}^{\boldsymbol{k}} = \boldsymbol{\beta}(\mathbf{y}^k)$ and we calculate an upper bound as $UB^k = v(\boldsymbol{\beta}^{\boldsymbol{k}}, \mathbf{q}^k)$. If the updated scenario vector $\boldsymbol{\beta}^{\boldsymbol{k}}$ coincides with $\boldsymbol{\beta}^{\boldsymbol{k'}}$ for some $k' < k-1$, we define the new weight vector as $\frac{\boldsymbol{\beta}^{\boldsymbol{k}} + \boldsymbol{\beta}^{\boldsymbol{k-1}}}{2}$. Finally, the algorithm terminates either when an optimal solution of the CVaR$_\alpha$TP is found, which happens when $\boldsymbol{\beta}^{\boldsymbol{k}} = \boldsymbol{\beta}^{\boldsymbol{k-1}}$, or after a maximum number of iterations previously specified. In this last case we obtain a lower and an upper bound of the optimal solution of the CVaR$_\alpha$TP.

5. Computational tests

In this section we report on the results of a series of computational tests performed to evaluate the methodological developments discussed in the previous sections.

For the computational study we considered test data introduced by Hinojosa et al. (2014) namely, the set of instances involving a single commodity, since this is the case in the current work. In particular, we have $|I| \in \{10, 20, 50\}$, $|J| \in \{20, 50, 100\}$, and $|\Omega| \in \{8, 12, 20, 30\}$. For each combination of $|I|$ and $|J|$ such that $|I| \leq |J|$, five instances were retrieved from Hinojosa et al. (2014) for each considered value of $|\Omega|$. The reader is referred to that work for details concerning the generation of the instances. Overall, we are considering 160 instances.

The computational tests were performed on a PC Intel Core i7-6700K CPU @ 4.00 CHz 4.01 CHz with 32 GB of RAM. The solution algorithms were implemented using Visual Studio C++ 2017 integrated with ILOG CPLEX Studio 12.7.1 Concert Technology routines. In particular, all the MIP formulations were solved using that solver. Apart from the MIP gap, default parameters have been used.

With the purpose of comparing the CVaR$_\alpha$TP formulation, given by (26)–(29), and the proposed algorithmic framework based on formulation (30)–(31), we tested four ways for tackling the specific problem we are investigating:

- GEN: solve formulation (26)–(29).
- ALG1: execute one single iteration of Algorithm 1.
- GEN-2,5%: solve formulation (26)–(29), using as a stopping criteria to attain a MIP gap smaller than or equal to 2,5%.
- ALG1-2,5%: execute Algorithm 1 as follows. A MIP gap of 2,5% is set for the formulations (32)-(39) solved at the different iterations. The execution of Algorithm 1 terminates either when the number of iterations reaches $|\Omega|$, or when the deviation gap between the current lower and upper bounds is smaller than or equal to 2,5%.

For all alternatives a time limit of two hours was considered.

5.1. Results

The results obtained have been split among Tables 1–3 according to the number of origins. In these tables the information contained in each row refers to average values for 5 instances; $|J|$ and $|\Omega|$ indicate the number of customers and the number of scenarios, respectively, for each group of 5 instances; "Approach" stands for the alternative approach considered when solving CVaR$_\alpha$TP among the four detailed above; "CPU time (seconds)" is the (average) time, in seconds, required to solve the problem to optimality or to meet the stopping criterion;

"# Iter" is the (average) number of iterations required by Algorithm 1. This information is depicted only for ALG1-2,5% since this is the only approach for which this information makes sense. Finally, three columns present different gaps. "Gap (%) UB-LB" is the (average) percentage gap considering the best upper and lower bounds obtained in the corresponding approach. For each instance, it is computed as (UB − LB)/UB × 100. "Gap (%) UB" is the (average) percentage gap computed using the objective values of the best feasible solution obtained by the corresponding approach, say $\bar{v}$, and the best feasible solution found using the standard formulation (26)–(29), say $\bar{v}_{\mathrm{GEN}}$. Hence, for each instance, "Gap (%) UB" is computed as $[\bar{v} - \bar{v}_{\mathrm{GEN}}]/\bar{v}_{\mathrm{GEN}} \times 100$. We do not present such gap for approach GEN since it is trivially equal to zero. "Gap (%) LB" is the (average) percentage gap computed using the objective value of the best feasible solution obtained using the standard formulation (26)–(29)—$\bar{v}_{\mathrm{GEN}}$—and the best lower bound obtained in the corresponding approach, say $\underline{v}$. For each instance, it is obtained as $[\bar{v}_{\mathrm{GEN}} - \underline{v}]/\bar{v}_{\mathrm{GEN}} \times 100$. Again, we do not present this gap for approach GEN since in this case it coincides with Gap (%) UB-LB. We note that GEN is the "standard" existing approach. This motivates the use of the corresponding upper bound for computing two of the three gaps just described.

Despite the difficulty of drawing overall comparisons when observing the extensive information provided by Tables 1–3, there are a few aspects that can be highlighted.

First, the time limit of two hours was often reached when using either GEN or ALG1. Nevertheless, the final gap—"Gap (%) UB-LB"—is in general below 2%. When this is not the case, we often find the final gap provided by ALG1 outperforming that of GEN (e.g., $|I| = 20$, $|J| = 100$, $|\Omega| = 30$). The superiority of ALG1 over GEN seems stronger when we consider the largest instances tested ($|I|$=50; $|\Omega| = 20, 30$).

Second, observing Tables 2 and 3 we see that in several cases the average "Gap (%) UB" is negative for ALG1 and ALG1-2,5% indicating a better upper bound provided by these approaches when compared to GEN. Again, this is more evident for larger instances (Table 3).

Third, considering either GEN-2,5% or ALG1-2,5%, in most of the cases, a final gap—"Gap (%) UB-LB"—less than or equal to 2.5% was obtained in a fairly small computing time.

Finally, regarding the quality of the best lower bound (when compared to the upper bound provided by the standard approach GEN), we observe that, in general, ALG1 performs the best. In particular, one single execution of Algorithm 1 seems to be enough for obtaining a sharp lower bound.

Although true, the above observations are not enough for drawing a clear conclusion as for the superiority of our new approach when compared to the standard one. Below we present results, which provide further insight into the overall performance of the compared alternatives.

Table 1: Results for the instances with 10 origins.

$\|J\|$	$\|\Omega\|$	Approach	CPU time (seconds)	# Iter	Gap (%) UB-LB	UB	LB
20	8	GEN	2200	—	0.04	—	—
		ALG1	3299	—	0.03	0.00	0.03
		GEN-2,5%	<1	—	1.61	1.19	0.44
		ALG1-2,5%	<1	1.00	1.11	0.63	0.48
	12	GEN	1850	—	0.01	—	—
		ALG1	898	—	0.29	0.26	0.03
		GEN-2,5%	1	—	1.40	0.88	0.53
		ALG1-2,5%	1	1.00	1.43	0.96	0.48
	20	GEN	6222	—	0.29	—	—
		ALG1	5999	—	0.74	0.45	0.30
		GEN-2,5%	6	—	1.68	0.97	0.73
		ALG1-2,5%	11	2.60	1.57	0.77	0.81
	30	GEN	7203	—	0.43	—	—
		ALG1	7203	—	0.68	0.25	0.43
		GEN-2,5%	14	—	1.86	1.16	0.72
		ALG1-2,5%	12	1.00	1.58	0.86	0.73
50	8	GEN	7236	—	0.11	—	—
		ALG1	7097	—	0.11	0.00	0.11
		GEN-2,5%	<1	—	1.88	1.52	0.39
		ALG1-2,5%	<1	1.00	1.88	1.52	0.39
	12	GEN	7203	—	0.42	—	—
		ALG1	7204	—	0.69	0.24	0.45
		GEN-2,5%	4	—	1.33	0.71	0.64
		ALG1-2,5%	5	1.60	1.35	0.72	0.65
	20	GEN	7219	—	0.60	—	—
		ALG1	7203	—	1.37	0.78	0.61
		GEN-2,5%	90	—	1.45	0.76	0.71
		ALG1-2,5%	79	1.00	1.79	1.09	0.72
	30	GEN	7201	—	0.73	—	—
		ALG1	7202	—	0.95	0.19	0.76
		GEN-2,5%	262	—	1.34	0.53	0.82
		ALG1-2,5%	263	1.00	1.35	0.52	0.84

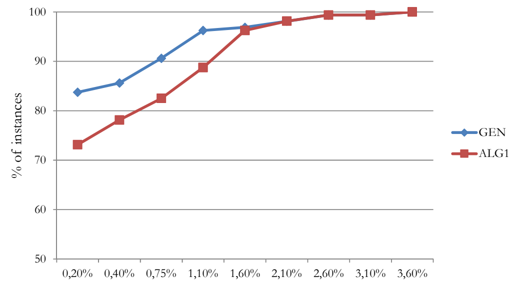
$\|J\|$	$\|\Omega\|$	Aproach	CPU time (seconds)	# Iter	Gap (%) UB-LB	UB	LB
100	8	GEN	5824	—	0.06	—	—
		ALG1	5827	—	0.05	0.00	0.05
		GEN-2,5%	<1	—	1.80	1.71	0.12
		ALG1-2,5%	1	1.00	0.94	0.82	0.12
	12	GEN	7202	—	0.20	—	—
		ALG1	7201	—	0.78	0.54	0.25
		GEN-2,5%	3	—	1.86	1.60	0.30
		ALG1-2,5%	2	1.20	1.90	1.59	0.34
	20	GEN	7232	—	0.39	—	—
		ALG1	7203	—	1.13	0.69	0.45
		GEN-2,5%	314	—	0.70	0.29	0.41
		ALG1-2,5%	229	1.80	2.14	1.70	0.48
	30	GEN	7203	—	0.50	—	—
		ALG1	7203	—	0.63	0.11	0.52
		GEN-2,5%	882	—	0.74	0.24	0.51
		ALG1-2,5%	1143	1.00	1.25	0.73	0.54

Table 2: Results for the instances with 20 origins.

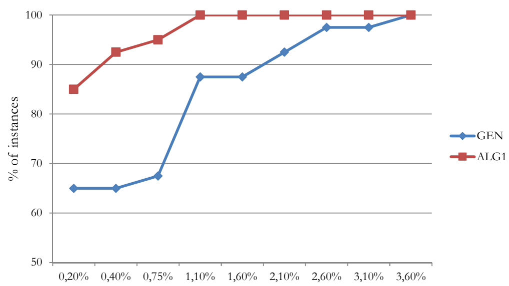
$\|J\|$	$\|\Omega\|$	Approach	CPU time (seconds)	# Iter	Gap (%) UB-LB	UB	LB	$\|J\|$	$\|\Omega\|$	Aproach	CPU time (seconds)	# Iter	Gap (%) UB-LB	UB	LB
20	8	GEN	5329	—	0.05	—	—	100	8	GEN	7230	—	0.19	—	—
		ALG1	5025	—	0.04	-0.01	0.04			ALG1	7202	—	0.17	-0.02	0.19
		GEN-2,5%	<1	—	2.26	1.93	0.37			GEN-2,5%	1	—	1.82	1.52	0.32
		ALG1-2,5%	<1	1.00	2.21	1.75	0.49			ALG1-2,5%	3	1.00	1.36	1.06	0.32
	12	GEN	5995	—	0.12	—	—		12	GEN	7271	—	0.46	—	—
		ALG1	6135	—	0.98	0.85	0.16			ALG1	7203	—	1.59	1.10	0.51
		GEN-2,5%	3	—	1.01	0.61	0.40			GEN-2,5%	325	—	1.41	0.90	0.53
		ALG1-2,5%	3	1.00	1.42	0.98	0.45			ALG1-2,5%	23	1.00	2.27	1.63	0.68
	20	GEN	7234	—	0.27	—	—		20	GEN	7219	—	1.49	—	—
		ALG1	7203	—	1.02	0.68	0.35			ALG1	7206	—	1.17	-0.34	1.51
		GEN-2,5%	9	—	1.56	0.94	0.63			GEN-2,5%	5150	—	1.86	0.37	1.50
		ALG1-2,5%	14	1.00	1.36	0.85	0.53			ALG1-2,5%	4267	1.00	2.09	0.60	1.51
	30	GEN	7203	—	0.42	—	—		30	GEN	7200	—	5.35	—	—
		ALG1	7203	—	0.57	0.12	0.45			ALG1	7212	—	3.20	-2.24	5.37
		GEN-2,5%	34	—	1.41	0.79	0.63			GEN-2,5%	7200	—	5.35	0.00	5.35
		ALG1-2,5%	55	1.20	1.14	0.60	0.55			ALG1-2,5%	7200	1.00	3.23	-2.21	5.37
50	8	GEN	7229	—	0.24	—	—								
		ALG1	7203	—	0.24	0.00	0.24								
		GEN-2,5%	<1	—	2.28	1.71	0.60								
		ALG1-2,5%	<1	1.00	2.15	1.54	0.64								
	12	GEN	7202	—	0.48	—	—								
		ALG1	7202	—	1.23	0.76	0.50								
		GEN-2,5%	50	—	1.10	0.51	0.60								
		ALG1-2,5%	63	3.20	2.30	1.72	0.63								
	20	GEN	7200	—	0.81	—	—								
		ALG1	7202	—	1.37	0.55	0.83								
		GEN-2,5%	1308	—	1.12	0.28	0.84								
		ALG1-2,5%	1239	1.60	1.79	0.94	0.87								
	30	GEN	7200	—	0.91	—	—								
		ALG1	7203	—	1.21	0.28	0.93								
		GEN-2,5%	2273	—	1.26	0.34	0.92								
		ALG1-2,5%	1285	1.20	1.70	0.75	0.96								

Table 3: Results for the instances with 50 origins.

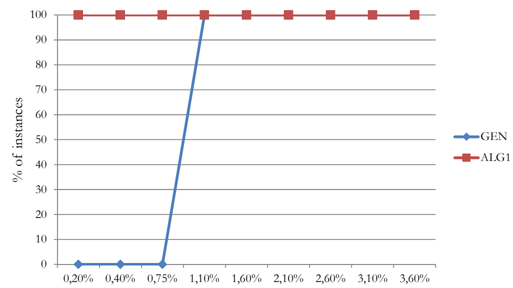
$\|J\|$	$\|\Omega\|$	Approach	CPU time (seconds)	# Iter	Gap (%) UB-LB	UB	LB	$\|J\|$	$\|\Omega\|$	Aproach	CPU time (seconds)	# Iter	Gap (%) UB-LB	UB	LB
50	8	GEN	7201	—	0.14	—	—	100	8	GEN	7200	—	0.20	—	—
		ALG1	7202	—	0.13	-0.01	0.14			ALG1	7207	—	0.20	0.01	0.20
		GEN-2,5%	2	—	1.24	0.94	0.31			GEN-2,5%	5	—	1.15	0.71	0.44
		ALG1-2,5%	3	1.00	1.45	1.15	0.31			ALG1-2,5%	11	1.00	1.34	0.90	0.44
	12	GEN	7200	—	0.29	—	—	—	12	GEN	7200	—	1.58	—	—
		ALG1	7204	—	1.49	1.20	0.31			ALG1	7214	—	1.94	0.32	1.62
		GEN-2,5%	6	—	2.12	1.60	0.55			GEN-2,5%	16	—	2.04	0.13	1.92
		ALG1-2,5%	46	1.40	2.04	1.51	0.56			ALG1-2,5%	667	4.80	2.07	0.30	1.77
	20	GEN	7200	—	2.62	—	—		20	GEN	7200	—	2.32	—	—
		ALG1	7208	—	2.28	-0.36	2.63			ALG1	7239	—	1.79	-0.54	2.32
		GEN-2,5%	5764	—	2.67	0.01	2.67			GEN-2,5%	744	—	2.43	0.00	2.43
		ALG1-2,5%	445	1.20	2.34	-0.33	2.66			ALG1-2,5%	1168	1.00	2.19	-0.21	2.39
	30	GEN	7200	—	3.22	—	—		30	GEN	7200	—	2.96	—	—
		ALG1	7219	—	2.61	-0.64	3.23			ALG1	7316	—	2.07	-0.92	2.97
		GEN-2,5%	5913	—	3.22	0.00	3.22			GEN-2,5%	7200	—	2.96	0.00	2.96
		ALG1-2,5%	18094	2.60	2.58	-0.66	3.23			ALG1-2,5%	4040	1.00	2.09	-0.92	2.99



(a) Whole set of instances.



(b) All instances with $|\Omega| = 30$.



(c) Larger instances ($|I| = 50$, $|J| = 100$, $|\Omega| = 30$).

Figure 1: Performance profile with respect to the Gap.

5.2. Performance profiles

In this section we analyze the performance profiles of the different approaches tested. We recall that a performance profile graph plots the percentage of instances solved for each value of the corresponding measure.

We start with a gap performance profile for GEN and ALG1 with a time limit of two hours. For this, we consider the gaps between the best upper bound provided by each approach and the best upper bound among the two approaches. More specifically, let $\bar{v}_{\text{GEN}}$ be the objective function value for the best feasible solution found using GEN and $\bar{v}_{\text{ALG1}}$ the corresponding value for ALG1. Define $v^* = \min\{\bar{v}_{\text{GEN}}, \bar{v}_{\text{ALG1}}\}$. For GEN and ALG1 we compute the gap according to $(\bar{v}_{\text{GEN}} - v^*)/v^* \times 100$ and $(\bar{v}_{\text{ALG1}} - v^*)/v^* \times 100$, respectively. The gap profiles are depicted in Figure 1.

As we can see in Figure 1(a), when we consider the whole set of instances the performance profile with respect to the gap of the standard formulation (26)–(29) is slightly better than the performance profile of our alternative solution approach. However for all instances with $|\Omega| = 30$ and for the larger instances —Figures 1(b) and 1(c)— the performance profile of our new approach ALG1 is superior to that of the standard formulation. In this comparison, it is worth-noting that we are considering only one iteration of Algorithm 1. A similar conclusion was already sketched in the previous section from the analysis of the extensive results presented in Tables 1–3.

Figure 2 depicts performance profiles with respect to computing times for GEN and ALG1.

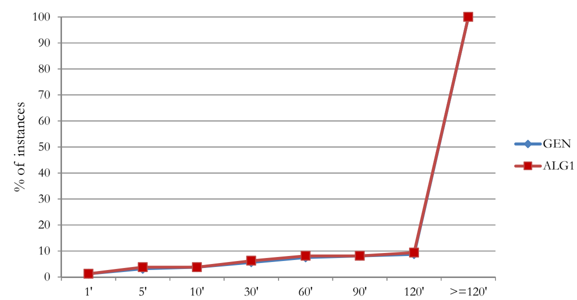


Figure 2: Performance profile with respect to computing times (all instances).

All instances were considered when drawing this profiles. As we can see, both approaches exhibit a similar behavior. In this figure, we also get a clear empirical evidence of the difficulty for solving CVaR$_\alpha$TP. As can be seen, only 10% of the instances were solved to optimality within the time limit of two hours with any of the two tested methods.

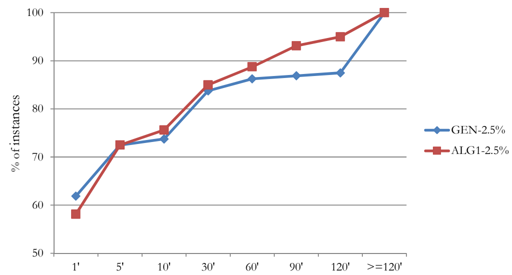
Due to the difficulty in solving CVaR$_\alpha$TP to proven optimality, we focused on the behavior of the two variants of the proposed approaches, when a $2,5\%$ MIP gap is set as a termination criterion when solving the MIP formulations. The corresponding performance profiles for the computing times are depicted in Figure 3. In this case, the superiority of the new modeling framework (ALG1-2,5%) as compared to the standard one (GEN-2,5%) becomes visible. Nonetheless, it is stronger for the largest instances tested as shown in Figure 3(c).

The superiority of ALG1-2,5% becomes more noticeable if we consider performance profiles with respect to the Gap(%) UB-LB, as depicted in Figure 4. In the particular cases of Figures 4(b) and 4(c) we see that ALG1-2,5% provides in general smaller gaps. This fact, combined with a better performance in terms of computing times, makes clear the superiority of ALG1-2,5% when compared to the standard approach GEN-2,5%.

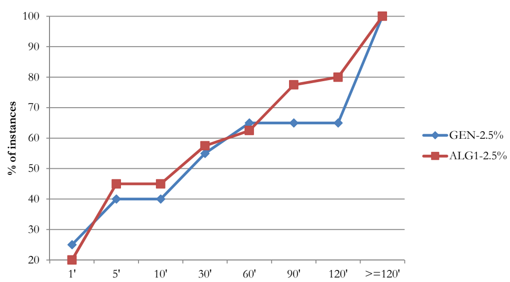
Finally, Figure 5 gives the performance profile of ALG1-2,5% with respect to the number of iterations. As we can observe, in 90% of the cases, Algorithm 1 stopped after performing just one iteration since that was enough to obtain a solution with a MIP gap smaller than or equal to $2,5\%$.

5.3. Further testing—large-scale instances

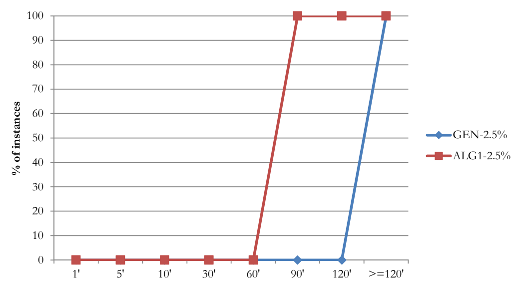
In order to give more evidence of the superiority of our proposed solution framework relative to the standard one, we have compared both of them on one additional set of 15 large-scale instances, generated as proposed in Hinojosa et al. (2014). In all cases we considered $|\Omega| = 50$ scenarios. For $|I| = 50$ we took $|J| = 100$ and 200; for $|I| = 100$ we considered $|J| = 200$. Five instances have been generate for each combination of $|I|, |J|$ and $|\Omega|$. The obtained results are summarized in Table 4. We do not report computing times, because the two hours time limit was reached in all cases. Moreover, neither the final gap at termination is reported for approaches GEN-$2,5\%$ and ALG1-$2,5\%$ with the largest instances ($|I| = 100$,



(a) Whole set of instances.

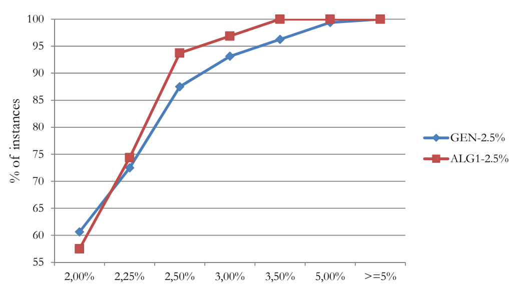


(b) All instances with $|\Omega| = 30$.

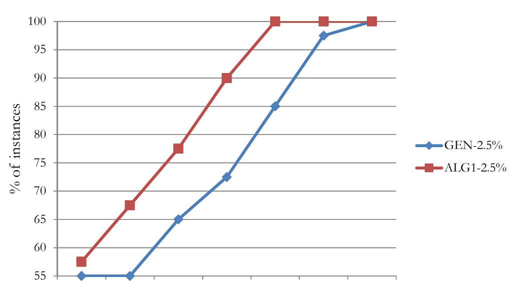


(c) Larger instances ($|I| = 50$, $|J| = 100$, $|\Omega| = 30$).

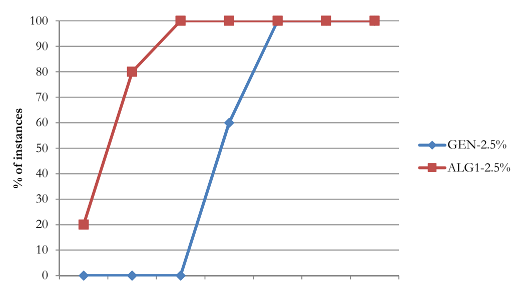
Figure 3: Performance profile with respect to computing time (MIP gap 2,5%).



(a) Whole set of instances.



(b) All instances with $|\Omega| = 30$.



(c) Larger instances ($|I| = 50$, $|J| = 100$, $|\Omega| = 30$).

Figure 4: Performance profile with respect to Gap (MIP gap 2,5%).

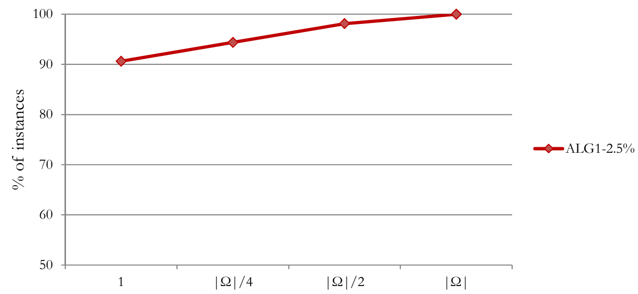


Figure 5: Performance profile with respect to the number of iterations.

$|J| = 200$), as it is always greater than $2,5\%$. For these new instances, our new approach clearly outperforms the standard one in terms of the average gap attained. In some cases, particularly for all cases with $|J| = 200$ the difference is very significant.

Table 4: Additional results.

$\|I\|$	$\|J\|$	$\|\Omega\|$	Tested Approach	Gap (%) UB-LB	# Iter
50	100	50	GEN	5,18	—
			ALG1	3,57	—
			GEN-$2,5\%$	5,18	—
			ALG1-$2,5\%$	3,56	1,2
	200	50	GEN	80,88	—
			ALG1	6,10	—
			GEN-$2,5\%$	80,88	—
			ALG1-$2,5\%$	5,36	1,4
100	200	50	GEN	22,23	—
			ALG1	3,46	—

6. Conclusions

In this paper, we proposed a new algorithm for obtaining lower and upper bounds on the optimal value of a stochastic mixed-integer programming problem considering the minimization of the conditional value at risk. The methodological developments were applied to a fixed-charge stochastic transportation problem. The results show that the new approach is competitive when compared to the standard one. This is particularly true for the largest instances tested.

One drawback of our approach is the need to solve one MIP problem in each iteration. Therefore, the competitiveness of this approach could possibly be higher if we considered a specially-tailored procedure for solving those MIPs. This depends of course on the specific application being considered. Nevertheless, this is a research direction to explore when making use of our new algorithmic scheme.

Related with the previous aspect, it is worth investigating the application of our new algorithm to other problems. This is relevant not only because the new approach is a promising alternative to the existing one but also because that would allow evaluating the robustness of our approach across different experimental settings.

Acknowledgement

The research of the first three authors was partially supported by the Spanish Ministry of Economy and Competitiveness through MINECO/FEDER grants MTM2015-63779-R and MTM2016-74983-C02-01. The research of the fourth author was partially supported by the Portuguese Science Foundation (FCT—Fundação para a Ciência e Tecnologia) under the project UID/MAT/ 04561/2013 (CMAF-CIO/FCUL).

References

References

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

Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.

Alonso-Ayuso, A., Escudero, L., Guignard, M., and Weintraub, A. (2016). Risk management for forestry planning under uncertainty in demand and prices. *Optimization Online*. http://www.optimization-online.org/DB_HTML/2016/12/5756.html.

Balinski, M. (1961). Fixed-cost transportation problems. *Naval Research Logistics Quaterly*, 8:41–54.

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98:49–71.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52:35–53.

Birge, J. and Louveaux, F. (2011). *Introduction to stochastic Programming*. Springer, Berlin-Heidelberg. Second edition.

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

Chelst, K. and Canbolat, Y. (2011). *Value-added decision making for managers*. CRC Press, Boca Raton-London-New York.

Escudero, L., Garín, M., and Unzueta, A. (2017). Scenario cluster Lagrangean decomposition for risk averse in multistage stochastic optimization. *Computers & Operations Research*, 85:1339–1351.

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

Grieco, S., Semeraro, U., and Tolio, T. (2001). A review of different approaches to the FMS loading problem. *International Journal of Flexible Manufacturing Systems*, 13(4):361–384.

Hemmati, R., Saboori, H., and Saboori, S. (2016). Stochastic risk-averse coordinated scheduling of grid integrated energy storage units in transmission constrained wind-thermal systems within a conditional value-at-risk framework. *Energy*, 113:762–775.

Hinojosa, Y., Puerto, J., and Saldanha-da-Gama, F. (2014). A two-stage stochastic transportation problem with fixed handling costs and a priori selection of the distribution channels. *TOP*, 22:1123–1147.

Homem-de-Mello, T. and Pagnoncelli, B. (2016). Risk aversion in multistage stochastic programming: A modeling and algorithmic perspective. *European Journal of Operational Research*, 249:188–199.

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

List, G. F., Wood, B., Nozick, L. K., Turnquist, M. A., Jones, D. A., Kjeldgaard, E. A., and Lawton, C. R. (2003). Robust optimization for fleet planning under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 39:209–227.

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

Meng, Q., Wang, T., and Wang, S. (2012). Short-term liner ship fleet planning with container transshipment and uncertain container shipment demand. *European Journal of Operational Research*, 223:96–105.

Miller, N. and Ruszczyński, A. (2011). Risk-averse two-stage stochastic linear programming: modeling and decomposition. *Operations Research*, 59:125–132.

Nickel, S., da Gama, F. S., and Ziegler, H.-P. (2012). A multi-stage stochastic supply network design problem with financial decisions and risk management. *Omega*, 40:511–524.

Noyan, N. (2012). Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39:541–559.

Puerto, J., Rodríguez-Chía, A., and Tamir, A. (2017). Revisiting k-sum optimization. *Mathematical Programming*, 165:579–604.

Punen, A. (1992). k-sum linear programming. *Journal of the Operational Research Society*, 43:359–363.

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

Santoso, T., Ahmed, S., Goetschalckx, M., and Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167:96–115.

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

Toumazis, I. and Kwon, C. (2016). Worst-case conditional value-at-risk minimization for hazardous materials transportation. *Transportation Science*, 50:1174–1187.

Uryasev, S. (2000). Conditional value-at-risk: optimization algorithms and applications. *Proceedings of the IEEE/IAFE/INFORMS 2000 Conference on Computational Intelligence for Financial Engineering*, pages 49–57.

Uryasev, S. and Rockafellar, R. T. (2001). *Conditional Value-at-Risk: Optimization Approach*, pages 411–435. Springer US, Boston, MA.

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

Wallace, S. and Ziemba, W. (2005). *Applications of Stochastic Programming.* SIAM, Berlin-Heidelberg. MOS-SIAM Series on Optimization.

Xu, N. and Nozick, L. (2009). Modeling supplier selection and the use of option contracts for global supply chain design. *Computers & Operations Research*, 36:2786–2800.