Constraint generation for risk averse two-stage stochastic programs

R. Mínguez[a,*], W. van Ackooij[b], R. García-Bertrand[c]

[a] *ESCISSION company, Ciudad Real, Spain*
[b] *EDF-Electricité de France - R&D, 7 Boulevard Gaspard Monge ; 91120 Palaiseau Cedex, France*
[c] *Department of Electrical Engineering, Universidad de Castilla-La Mancha, Ciudad Real, Spain*

Abstract

A significant share of stochastic optimization problems in practice can be cast as two-stage stochastic programs. If uncertainty is available through a finite set of scenarios, which frequently occurs, and we are interested in accounting for risk aversion, the expectation in the recourse cost can be replaced with a worst-case function (i.e., robust optimization) or another risk-functional, such as conditional value-at-risk. In this paper we are interested in the latter situation especially when the number of scenarios is large. For computational efficiency we suggest a (clustering and) constraint generation algorithm. We establish convergence of these two algorithms and demonstrate their effectiveness through various numerical experiments.

Keywords: Stochastic programming, Decision analysis under uncertainty, CVaR, Risk aversion
2010 MSC: 90C15

1. Introduction

Stochastic programming, e.g., Birge & Louveaux (1997); Shapiro et al. (2009); Bonnans (2019), is a framework for modelling problems that involve uncertainty. Since real world almost invariably includes uncertain parameters, the discipline of stochastic programming has grown and broadened to cover a wide range of applications. A significant share of stochastic optimization problems in practice can be cast as two-stage stochastic programs. We refer to van Ackooij et al. (2018a) for a survey covering a large number of publications within the very specific subfield of unit commitment under uncertainty to back up this claim. In this paper we are concerned with numerically solving risk averse two-stage stochastic linear programming problems. In such problems, we explicitly model the fact that a first decision needs to be taken prior to observing uncertainty and then a potential recourse action can act once this uncertainty has been revealed. With an appropriate choice of a risk functional, from an abstract viewpoint two-stage stochastic programming

*Corresponding author: rominsol@gmail.com

can be seen as a special case of non-linear non-smooth convex optimization by considering the second-stage risk measured cost as a mapping, of the first-stage variables. A popular choice for such a risk functional is Conditional Value-at-Risk (CVaR), e.g., Rockafellar & Uryasev (2000, 2002); Krokhmal et al. (2002). The work Huang et al. (2014) provides such an implementation for a two-stage stochastic unit commitment problem and Kazemzadeh et al. (2019) a comparison with a worst-case (robust) consideration.

A common assumption widely used by practitioners is to assume that uncertainty is represented by a finite set of scenarios. Under this assumption, results given by Rockafellar & Uryasev (2002) allow us to produce an extended linear programming formulation of the original risk averse two-stage stochastic program. The latter problem might still be hard to solve, especially if the number of scenarios is large. With S being the number of scenarios, the extended formulation is typically $S+1$ times the size of any underlying "deterministic base model". Therefore for reasonably realistic underlying models, it can be assumed that for large values of S, the resulting extended formulation does not solve "fast enough", if at all. In the literature one typically finds various strategies to speed up the resolution. A typical strategy is the use of decomposition (for instance Benders' like). The strategies can also be combined with:

- reducing the problem size, by reducing the number of "scenarios". Scenarios are aggregated into "alike" data.
- reducing the CPU time per iteration, by exploiting for instance partial information, employing valid, but not tight cuts.
- reducing the number of iterations, by using for instance stabilized Benders decomposition, "finding" best cuts, etc. (see e.g., Rahmaniani et al. (2017) for a recent survey on Benders' decomposition).

We will now comment these strategies before highlighting our contribution.

As a mean to solve the problem faster, a natural idea is to partition the scenario set into clusters, and approximate the cost function using one representative scenario for each cluster, which "aggregates" the information in a cluster of scenarios. The idea of using aggregation for solving stochastic programs has a long history. For instance, Birge (1985) studies aggregation techniques to obtain optimality bounds. For multistage stochastic linear programs with a general probability distribution, Wright (1994) studies aggregation and disaggregation with respect to σ-algebras of the underlying probability space. Another concept of aggregation is proposed in Rosa & Takriti (1999) for two-stage stochastic programs, which considers aggregation of constraints across different scenarios as well as constraints within each scenario. An aggregation procedure based on clustering of scenarios is employed in Jardim et al. (2001). A practical implementation of a similar idea wherein scenarios are quantized can be found in van Ackooij (2017). Important contributions on the subject of selecting smaller set of scenarios have been done by using probability metrics (Pflug, 2001; Heitsch & Römisch, 2007, 2003; Henrion et al., 2008, 2009; Heitsch & Römisch, 2009; García-Bertrand & Mínguez, 2014), in particular the Wasserstein distance (Dupačová et al., 2003). The idea is to approximate the problem by a smaller problem defined by a fixed and small partition. The partition is obtained by applying an algorithm based on probability metrics to the set of scenarios.

Morales et al. (2009) and de Oliveira et al. (2011) incorporate the second-stage costs into the distance function used to define the Wasserstein metric, and show empirically that partitions with better quality could be obtained. The idea of partitions is also exploited in Sandikçi et al. (2013) in order to derive upper and lower bounds on the optimal value of two-stage stochastic programs. However, computing these bounds essentially requires solving as many problems as the number of partitions of a given cardinality that can be found. These bounds are quite general as they do not require a fixed recourse assumption and binary variables can be considered in the second-stage problems, but can be computationally challenging to obtain. In Crainic et al. (2014), the authors combine progressive hedging and partitioning scenarios in order to solve a set of smaller two-stage problems related to the considered subsets of scenarios. However, the partition is chosen a priori, although several strategies have been exploited to help make a good choice. Further recent contributions (Song & Luedtke, 2015; van Ackooij et al., 2018b; Pay & Song, 2018) show how to dynamically adapt the partitions and also theoretically show that the size of the optimal family is independent of the number of scenarios.

A second idea to solve the problem faster is to simply compute recourse cost "inexactly". We begin here by noting that the typical algorithms will generate a sequence of candidates converging to the optimal solution. It becomes intuitively clear that computing the recourse cost with high precision for early iterates, likely far from an optimal solution, is surely a wasteful use of computing power. Accuracy will need to be integrated eventually if one is to assert optimality of the resulting solution. These ideas have been made theoretically precise in the on-demand accuracy framework (de Oliveira & Sagastizábal, 2014). The approach hinges on being able to quickly identify valid cutting planes that are not necessarily tight. This feature has been exploited in a series of publications on on-demand accuracy. Using such inexact computations combined with stabilization methods has been shown to be quite promising. One particular method of interest as stabilization is concerned is the level bundle method (Lemaréchal et al., 1995; Fábián, 2000; van Ackooij & de Oliveira, 2014). The level bundle method employing inexact oracles with on-demand accuracy has been applied to two-stage stochastic programs in de Oliveira & Sagastizábal (2014), Wolf et al. (2014), and Fábián et al. (2015). In the stochastic programming context, the level bundle method is also known as *level decomposition*, see for instance Wolf et al. (2014). The latter work presents an extensive computational study by analyzing several algorithms for two-stage stochastic linear programs, including the extended formulation, single- and multi-cut variants of the classical L-shaped method (van Slyke & Wets, 1969) and level bundle methods. Such decomposition ideas can also be extended to cover the case of risk-adjusted recourse cost functions as, for instance, discussed in Ahmed (2006); Künzi-Bay & Mayer (2006); Fábián (2008).

All the previous decomposition strategies rely on generating cutting planes for the risk adjusted recourse cost function in one way or another. In the two algorithms presented in this paper, typical iterations will increasingly add sets of constraints until the required accuracy is achieved. Both algorithms rely on a dynamic scenario aggregation technique. We are concerned with both general and fixed recourse situations, the latter implying that the duals of all second-stage linear problems share the same feasible set. Various numerical experiments demonstrate the effectiveness of the approach. We thus consider a constraint generation procedure (as in Zeng & Zhao (2013)), rather than employing cutting planes. Our contributions are as follows:

1. the suggested methods are valid for the CVaR risk functional (and in principle any convex risk function under appropriate modifications by employing their dual characterization), whereas the method by Zeng & Zhao (2013) is valid only for worst-case risk functionals. Our approaches are also valid for the risk-neutral case, and significant computational savings with respect to extended formulations are exhibited in the numerical experiments.
2. the approach suggested by Zeng & Zhao (2013) is only valid for right-hand side uncertainty, whereas both our algorithms cover the standard situation and our base algorithm can allow for uncertainty in all coefficients.

The rest of the paper is organized as follows: section 2 discusses the precise problem statement and background material. The algorithm that does not require a fixed recourse assumption is presented in section 3 and its convergence is analysed. Section 4 presents an alternative algorithm which requires fixed recourse. We also present the convergence analysis. We also discuss several computational extensions that may considerably speed up the algorithms. An extensive set of numerical experiments is performed and discussed in section 5. The paper ends with conclusions and perspectives.

2. Background material and formal problem statement

In this work, we are concerned with the following two-stage stochastic program for a given confidence level $\beta \in [0, 1]$:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\mathsf{T} x + \text{CVaR}_\beta(V(x, \xi)) \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in X, \end{aligned} \tag{1}$$

where $V : \mathbb{R}^n \times \mathbb{R}^m \to \bar{\mathbb{R}}$ is the second-stage function defined as follows:

$$V(x, z) = \min_{y \in \mathbb{R}^k} \left\{ q(z)^\mathsf{T} y \ : T(z)x + W(z)y \leq h(z), y \in Y \right\}. \tag{2}$$

In this formulation and for the time being, we will leave the dependency on z implicit, thus meaning that q, T, W and h could, in principle, be perturbed. The random vector ξ gathers these perturbations.

For simplicity of presentation, we will assume that problem (1) has relatively complete recourse, i.e., $\{x \in X \ : Ax \leq b\} \subseteq \mathcal{D}\text{om}(\text{CVaR}_\beta(V))$. Should this assumption not hold in practice, then one can use the unbounded ray used to establish unboundedness of problem (2), as a feasibility cut to augment the system $Ax \leq b$. The existence of such an unbounded ray is the result of Farkas' Lemma. Moreover, it is usually readily available in any solver that tried to solve problem (2). In the polyhedral setting exposed here, we may thus assume relatively complete recourse to present the main ideas.

Our further standing assumption is that ξ is known through a finite sample, $\xi_1, ..., \xi_S$ with associated probabilities $p_1, ..., p_S$. Hereafter elements of the set $\{\xi_1, ..., \xi_S\}$ are referred to as scenarios. In a way they are the realization of the random vector ξ. According to Rockafellar & Uryasev (2000), problem (1) can be recast in a risk neutral

form, upon increasing the space of variables. Indeed, if we define the auxiliary value function $\bar{V} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \bar{\mathbb{R}}$ as follows:

$$\bar{V}(x, z, \eta) = \min_{y \in \mathbb{R}^k, v \geq 0} \left\{ v \ : T(z)x + W(z)y \leq h(z), v \geq q(z)^{\mathsf{T}} y - \eta, y \in Y \right\}, \tag{3}$$

we can actually rewrite (1) as:

$$\begin{aligned} \min_{x \in \mathbb{R}^n, \eta} \quad & c^{\mathsf{T}} x + \eta + \frac{1}{1-\beta} \mathbb{E}\left(\bar{V}(x, \xi, \eta) \right) \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in X. \end{aligned} \tag{4}$$

It is also known that the optimal solution η^* to problem (4) corresponds to the value at risk, i.e., $\eta^* = \text{VaR}_\beta(V(x^*, \xi))$, at least when it is unique. Otherwise the value at risk is the smallest such optimal solution. Since ξ is known through finitely many samples, problem (4) can be written in the following extended formulation:

$$\begin{aligned} \min_{x, \eta, y_1, ..., y_S, v_1, ..., v_S} \quad & c^{\mathsf{T}} x + \eta + \frac{1}{1-\beta} \sum_{j=1}^{S} p_j v_j \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in X \\ & T_i x + W_i y_i \leq h_i, i = 1, ..., S \\ & v_i \geq q_i^{\mathsf{T}} y_i - \eta, i = 1, ..., S \\ & y_i \in Y, v_i \geq 0, i = 1, ..., S. \end{aligned} \tag{5}$$

Note that in the previous formulation, the case $\beta = 0$ corresponds to the risk-neutral formulation, i.e., $\text{CVaR}_\beta(V(x, \xi)) = \mathbb{E}\left(V(x, \xi) \right)$. Therefore, the suggested methods apply in particular to this special case. We further explore this feature in the computational experiments. In contrast, if $\beta \longrightarrow 1$ the risk functional would correspond to the worst-case setting. Since this setting is too conservative, we will not further examine it.

2.1. Dual representation of CVaR

For fixed values of $y_1, ..., y_S$ actually computing the CVaR can be achieved by solving a linear program. Indeed, it reads

$$\min_{\eta, v_1, ..., v_S \geq 0} \quad (1-\beta)\eta + \sum_{j=1}^{S} p_j v_j \tag{6}$$

$$\text{s.t} \quad v_j \geq q_j^{\mathsf{T}} y_j - \eta, j = 1, ..., S. \tag{7}$$

The latter linear program has the following dual

$$\begin{aligned} \max_{\pi_1,...,\pi_S} \quad & \sum_{j=1}^{S} \pi_j (q_j^{\mathsf{T}} y_j) \\ \text{s.t} \quad & \sum_{j=1}^{S} \pi_j = 1-\beta, \\ & 0 \leq \pi_j \leq p_j. \end{aligned}$$

Let us denote the latter dual feasible set by Δ^{β}. Now, as a consequence of weak-duality, and for any x, we obtain that for any $\pi \in \Delta^{\beta}$, it holds that

$$\frac{1}{1-\beta} \sum_{j=1}^{S} \pi_j V(x,\xi_j) \leq \text{CVaR}_{\beta}(V(x,\xi)). \tag{8}$$

In practice, there is no need to solve a linear program to compute the CVaR at $y_1, ..., y_S$ fixed and one can resort to a simple sorting mechanism. Indeed let the permutation $\sigma : \{1, ..., S\} \to \{1, ..., S\}$ be such that $q_{\sigma(1)}^{\mathsf{T}} y_{\sigma(1)} \leq ... \leq q_{\sigma(S)}^{\mathsf{T}} y_{\sigma(S)}$ and J such that $\sum_{j=1}^{J-1} p_{\sigma(j)} < \beta$, $P_J := \sum_{j=1}^{J} p_{\sigma(j)} \geq \beta$. Then

$$\hat{\pi}_{\sigma(j)} = \begin{cases} 0 & \text{for} \quad 1 \leq j < J \\ P_J - \beta & \text{for} \quad j = J \\ p_j & \text{for} \quad J < j \leq S, \end{cases} \tag{9}$$

are the optimal dual variable values, which hereafter will be considered as weights. Thus, the estimator of CVaR can be shown to asymptotically converge when $S \to \infty$ as done, for instance, in Theorem 3.3 from Pflug & Wozabal (2010).

3. Constraint generation (CG) based algorithm

In this section, we will discuss a constraint generation based algorithm for solving problem (1) more efficiently. The intuition behind the algorithm resides in the observation that the set of scenarios receiving weights (9) represents unwanted situations and should be relatively stable with respect to x. Therefore only few of the scenarios are relevant for pinning down the optimal solution x to problem (1) as in García-Bertrand & Mínguez (2014). Note that the rational behind the proposed algorithm is analogous to that given in García-Bertrand & Mínguez (2014), however, the latter requires tuning a problem dependent parameter, whose improper selection might even increase computational time with respect to extended formulations. The constraint generation algorithm overcomes that difficulty. It is as follows:

- (Initialization): Let x^0 be a given initial solution. Let $k = 0$ be set and $\varepsilon > 0$ be some stopping tolerance. Set $r^0 = -\infty$.
- (Statistics): For all scenarios, compute $V(x^k, \xi_i)$, by solving (2), $i = 1, ..., S$.

- (Stopping Criteria): If $\mathrm{CVaR}_\beta(V(x^k,\xi)) - r^k \leq \varepsilon$, then stop.
- (Scenario Weighting): Select weights $\pi^k \in \Delta^\beta$ according to rule (9).
- (MP): Solve the new master program:

$$\begin{aligned} &\min_{x\in X, r} c^\mathsf{T} x + r \\ &\text{s.t.} \quad Ax \leq b \\ &\qquad r \geq \frac{1}{1-\beta}\sum_{j=1}^{S} \pi_j^l V(x,\xi_j), l \leq k \end{aligned} \tag{10}$$

  and let (x^{k+1}, r^{k+1}) be the optimal solution.
- (Loop): Set $k = k+1$ and return to the Statistics step.

The master problem (10) admits the following extended form:

$$\begin{aligned} &\min_{x\in X, r, y} c^\mathsf{T} x + r \\ &\text{s.t.} \quad Ax \leq b \\ &\qquad r \geq \frac{1}{1-\beta}\sum_{j=1}^{S} \pi_j^l q_j^\mathsf{T} y_j, l \leq k \\ &\qquad T_j x + W_j y_j \leq h_j, y_j \in Y, j \leq S \text{ s.t. } \exists l \leq k \ : \pi_j^l > 0, \end{aligned} \tag{11}$$

wherein variables y are only added whenever the associated scenario was assigned positive weight (probability) at some past iteration.

3.1. Convergence analysis

In our analysis, the following observation is vital:

Lemma 3.1. *Consider problem* (1) *and assume that it has relatively complete recourse. For any iteration $k \geq 0$, the optimal value $\nu^{k+1} := c^\mathsf{T} x^{k+1} + r^{k+1}$ of problem* (10) *is a lower bound on the optimal value of problem* (1).

Proof. It is clear that at optimality, it holds that $r^{k+1} = \frac{1}{1-\beta}\max_{l\leq k}\sum_{j=1}^{S}\pi_j^l V(x^{k+1},\xi_j)$. Now at any iteration $l \leq k$, $\pi^l \in \Delta^\beta$ and hence as a result of (8) it follows that for those l:

$$\frac{1}{1-\beta}\sum_{j=1}^{S}\pi_j^l V(x^{k+1},\xi_j) \leq \mathrm{CVaR}_\beta(V(x^{k+1},\xi)).$$

Consequently, $r^{k+1} \leq \mathrm{CVaR}_\beta(V(x^{k+1},\xi))$. Moreover, since the actual role of x^k has not been used, the same estimate holds true for an arbitrary feasible x. □

We can now establish that the stopping condition is valid:

Lemma 3.2. *Consider problem* (1) *and assume that it has relatively complete recourse. If, at iteration k, it holds that* $\text{CVaR}_\beta(V(x^{k+1},\xi)) - \eta^{k+1} \leq \varepsilon$, *then* x^{k+1} *is* ε*-optimal for problem* (1).

Proof. As a result of Lemma 3.1, $\nu_{k+1} = c^\mathsf{T} x^{k+1} + \eta^{k+1}$ is a lower bound on the optimal value of problem (1). Since x^{k+1} is feasible, $u^{k+1} = c^\mathsf{T} x^{k+1} + \text{CVaR}_\beta(V(x^{k+1},\xi))$ is an upper bound. Now with x^* denoting an optimal solution for problem (1), it follows:

$$u^k - \varepsilon \leq \eta^k + \varepsilon \leq c^\mathsf{T} x^* + \text{CVaR}_\beta(V(x^*,\xi)),$$

which is what was to be shown. □

As an alternative stopping condition one can also look at the newly "active set of scenarios". Indeed:

Lemma 3.3. *For a given iteration k, let* $S' \subseteq \{1,...,S\}$ *be the maximal set of indices for which* $\hat{\pi}_j > 0$, $j \in S'$, *with* $\hat{\pi}$ *as in* (9). *Now should,* $S' \subseteq S^l := \left\{j = 1,...,S \,:\, \pi_j^l > 0\right\}$ *hold for some* $l < k$ *as well as* $J' = J^l$ *(where* J', J^l *refer to* J *in* (9) *in the respective iterations), then the algorithm terminates and* x^k *is the optimal solution to problem* (1).

Proof. Observe that the cardinality of the sets S^l resulting of rule (9) does not change with the iterations. Hence, actually $S' = S^l$. Since index J' (J in (9)) is also identical with J^l, we also have $\hat{\pi} = \pi^l$. Consequently with $\hat{\pi} = \pi^l$ being dual optimal as a result of (9) we have:

$$\text{CVaR}_\beta(V(x^k,\xi)) = \frac{1}{1-\beta}\sum_{i=1}^{S} \pi_i^l V(x^k,\xi_i) \leq \frac{1}{1-\beta}\max_{l\leq k}\sum_{i=1}^{S} \pi_i^l V(x^k,\xi_i) \leq r^k,$$

i.e., the stopping condition holds (even if $\varepsilon = 0$). The latter fact also implies optimality for problem (1) of x^k, through Lemma 3.2. □

We are now capable of showing finite convergence for our Algorithm:

Proposition 3.1. *Consider problem* (1) *and assume that it has relatively complete recourse. The algorithm terminates after finitely many iterations.*

Proof. The algorithm picks a certain subset S' of $\{1,...,S\}$ to which it assigns positive weight according to rule (9). Since one can pick only finitely many such sets, the condition of Lemma 3.3 involving it will thus hold. There are also at most S choices for picking index J in rule (9) and therefore in total after finitely many iterations the condition of Lemma 3.3 will hold and the algorithm will terminate. □

4. Clustering and constraint generation (CCG) based algorithm

Still the set of scenarios that has positive weight in (9) might be quite large. In order to mitigate this effect we propose a modification of the constraint generation algorithm previously presented that relies on a progressive scenario aggregation procedure so as

to add 2, 3, and so on, sets of constraints until the required accuracy is achieved. To do so, our underlying assumption is that of fixed recourse, i.e., both q and W do not depend on z. The immediately repercussion of that assumption is that the dual feasible set to problem (2) does not depend on ξ. If the set Y is convex , when both T and h are uncertain, V is convex in x and in ξ, but not necessarily jointly in both arguments. If T is assumed to be fixed, i.e., in the situation wherein we are only dealing with right-hand side uncertainty, V is convex in both arguments. This latter situation is the one considered in Zeng & Zhao (2013) in the risk-neutral setting. We will not require T to be fixed.

The clustering and constraint generation algorithm is as follows:

- (Initialization): Let x^0 be a given initial solution. Let $k = 0$ be given, set the initial number of clusters to $n_c = n_c^0$ and let $\varepsilon > 0$ be some stopping tolerance. Set $r^0 = -\infty$ and $\Omega^{-1} = \emptyset$.
- (Statistics): For all scenarios, compute $V(x^k, \xi_i)$, by solving (2), $i = 1, ..., S$.
- (Stopping Criteria): If $\text{CVaR}_\beta(V(x^k, \xi)) - r^k \leq \varepsilon$, then stop.
- (Scenario Weighting): Select weights $\pi^k \in \Delta^\beta$ according to rule (9).
- (Scenario Aggregation and Weighting): Form n_c clusters of the scenarios $\xi_1, ..., \xi_S$ which have received positive weight π^k, i.e., $C_1, ..., C_{n_c}$ is a partition of $\left\{j = 1, ..., S \ : \pi_j^k > 0\right\}$. Now define:

$$\left\{\tilde{\pi}_j^k, \tilde{\xi}_j^k\right\} := \left\{\sum_{i \in C_j} \pi_i^k, (\bar{T}_j, \bar{h}_j) := \frac{1}{\tilde{\pi}_j^k} \sum_{i \in C_j} \pi_i^k (T_i, h_i)\right\}, \tag{12}$$

  where the tilde refers to aggregated values. We additionally consider a cluster C_{n_c+1} gathering all scenarios with null weights, i.e., $\tilde{\pi}_{n_c+1}^k = 0$.
  Set $n_c^k = n_c$, let

$$\Omega^k = \left\{\tilde{\xi}_1^k, \ldots, \tilde{\xi}_{1+n_c}^k\right\} \bigcup \Omega^{k-1} \tag{13}$$

  and define $S^k := \left|\Omega^k\right|$. Moreover expand[1] $\tilde{\pi}^k$ to become of cardinality S^k, by setting $\tilde{\pi}^k$ to zero for any scenario different from $\left\{\tilde{\xi}_1^k, \ldots, \tilde{\xi}_{1+n_c}^k\right\}$.

- (MP): Solve the new master program:

$$\begin{aligned} &\min_{x \in X, r} c^\mathsf{T} x + r \\ &\text{s.t.} \quad Ax \leq b \\ &\qquad r \geq \frac{1}{1-\beta} \sum_{j=1}^{S^l} \tilde{\pi}_j^l V(x, \tilde{\xi}_j^l), l \leq k \end{aligned} \tag{14}$$

  and let (x^{k+1}, r^{k+1}) be the optimal solution.

[1] This expansion is obviously only of notational interest for the analysis and is not practically required

- (Scenario handling) If $c^{\mathsf{T}}x^{k+1} + r^{k+1} \leq c^{\mathsf{T}}x^k + r^k$ set $n_c = \min\left\{n_c + \Delta n_c, \bar{C}\right\}$. Here $\bar{C}$ is the cardinality of the set of positive valued probabilities in (9).
- (Loop): Set $k = k + 1$ and return to the Statistics step.

Analogously to the previous algorithm, second-stage variables and constraints are only added whenever the aggregated scenario was assigned positive weight (probability) at some past iteration.

Remark 4.1. *The performance of CCG algorithm depends on the selection of parameters n_c^0 (initial number of clusters) and Δn_c (increment in the number of clusters between iterations). Practical guidelines for selecting these parameters are given in the numerical examples.*

4.1. Convergence analysis

Let us first establish the following analogue to Lemma 3.1:

Lemma 4.1. *Consider problem* (1) *and assume that Y is convex and that the problem has fixed relatively complete recourse. For any iteration $k \geq 0$, the optimal value $\nu^{k+1} := c^{\mathsf{T}}x^{k+1} + r^{k+1}$ of problem* (14) *is a lower bound on the optimal value of problem* (1)*.*

Proof. At any iteration $l \leq k$, we recall that $\tilde{\pi}^l$ has exactly $n_c^l + 1$ positive components corresponding to the scenarios $\left\{\tilde{\xi}_1^l, \ldots, \tilde{\xi}_{1+n_c^l}^l\right\}$ resulting from the clustering. Although these need not necessarily be the first $n_c^l + 1$ scenarios in Ω^l, we may assume this without loss of generality by reorganizing indexation. Hence, we have

$$\begin{aligned}
\frac{1}{1-\beta}\sum_{j=1}^{S^l}\tilde{\pi}_j^l V(x^{k+1}, \tilde{\xi}_j^l) &= \frac{1}{1-\beta}\left[\sum_{j=1}^{n_c^l+1}\tilde{\pi}_j^l V(x^{k+1}, \tilde{\xi}_j^l)\right] \\
&= \frac{1}{1-\beta}\left[\sum_{j=1}^{n_c^l+1}\tilde{\pi}_j^l V(x^{k+1}, \frac{1}{\tilde{\pi}_j^l}\sum_{i\in C_j}\pi_i^l\xi_i)\right] \\
&\leq \frac{1}{1-\beta}\left[\sum_{j=1}^{n_c^l+1}\tilde{\pi}_j^l \frac{1}{\tilde{\pi}_j^l}\sum_{i\in C_j}\pi_i^l V(x^{k+1}, \xi_i)\right] \\
&= \frac{1}{1-\beta}\sum_{i=1}^{S}\pi_i V(x^{k+1}, \xi_i) \leq \mathrm{CVaR}_\beta(V(x^{k+1}, \xi)).
\end{aligned}$$

The first equality follows from the definition of $\tilde{\pi}^l$, the second from the definition of $\tilde{\xi}_1^l, ..., \tilde{\xi}_{n_c^l+1}^l$, the first inequality follows from convexity of V in the second argument, the subsequent equality from the fact that $C_1, ..., C_{n_c+1}$ is a partition of all scenarios with positive probability in π and the last inequality stems from (8) since $\pi \in \Delta^\beta$.

Since, it is clear that at optimality, it holds that $r^{k+1} = \frac{1}{1-\beta}\max_{l\leq k}\sum_{j=1}^{S^1}\tilde{\pi}_j^l V(x^{k+1}, \tilde{\xi}_j^l)$, it thus follows that, $r^{k+1} \leq \mathrm{CVaR}_\beta(V(x^{k+1}, \xi))$. Moreover since the actual role of x^{k+1} has not been used, the same estimate holds true for an arbitrary feasible x. □

We are now capable of showing finite convergence also for this algorithm:

Proposition 4.1. *Consider problem* (1) *and assume that it has fixed relatively complete recourse. The algorithm terminates after finitely many iterations.*

Proof. Δ^β is a polyhedral (simplex like) set and thus has finitely many different extremal points. The rule (9) can only produce finitely many different probability levels π of always fixed cardinality $\bar{C}$. Therefore for any given level $n_c = 1, ..., \bar{C}-1$, the total set of different scenarios $\tilde{\xi}$ that can be generated is finite. Should the level n_c remain fixed, then there exists an iteration index $\hat{k}$ after which the master program (14) has an unchanged set of constraints for $k \geq \hat{k}$. Hence, for $k \geq \hat{k}$, the objective function values of these master programs remain unchanged, i.e., $c^\mathsf{T} x^{k+1} + r^{k+1} = c^\mathsf{T} x^k + r^k$ for $k \geq \hat{k}$, i.e., the number of clusters must be increased. Since the argument can be repeated for $n_c + \Delta n_c$, n_c will, in the worst case, be increased until it reaches $\bar{C}$. For this latter value, the original set of scenarios is used, i.e., all clusters are singletons. Then finite convergence results from Proposition 3.1. □

4.2. Discussion of the algorithms and further enhancements

The algorithms solve problem (1) by identifying, at each iteration, those scenarios which are binding for the second-stage CVaR. This allows the master problem to find new values of first-stage decision variables. The reduction in computing time of the method is closely related to master problem complexity, however, when the number of scenarios increases considerably, the time to solve the sequence of second-stage problems (2) increases linearly with respect to the number of scenarios. The following possible strategies can be used to reduce this CPU time:

1. Use of parallelization techniques. Since the subproblem sequence requires the solution of the same problem for different values of random parameters, it is possible to compute those problems in parallel, which might reduce computational times significantly depending on computational resources available.
2. In some cases, it might be convenient to compute $V(x^k, \xi_i)$ simultaneously as:

$$\begin{aligned} \sum_{i=1}^{S} V(x^k, \xi_i) = \min_{y_1,...,y_S} & \sum_{i=1}^{S} q_i^\mathsf{T} y_i \\ \text{s.t.} \quad & T_i x^k + W_i y_i \leq h_i, i = 1, ..., S \\ & y_i \in Y, i = 1, ..., S, \end{aligned} \tag{15}$$

   or using different grouping strategies, where first-stage decisions variables x^k are considered as fixed parameters. This problem is decomposable (Conejo et al., 2006) so that the result of (15) using as objective function the sum of objective functions of problems (2) provides the same optimal solutions in terms of variables y^* and $V(x^k, \xi_i)$-values.
3. As a result of fixed recourse, the dual optimal solution $\bar{\lambda}_i$ obtained for a given scenario $i \in \{1, ..., S\}$ when solving problem (2) is dual feasible for any other. Consequently, $(h_i - Tx)^\mathsf{T} \bar{\lambda}_i \leq V(x, \xi_j)$ for all $j = 1, ..., S$ which thus allows us to obtain an estimate of $V(x, \xi_j)$ for all j based on a single resolution. The latter estimate

can be used to identify the likely set of active scenarios and set weights accordingly. This methodology can be amended by using clustering kind of approaches to select "similar" scenarios which would thus effectively yield a partition of the set of scenarios. One would typically compute a single dual optimal vector per partition. The analysis in Song & Luedtke (2015); Pay & Song (2018); van Ackooij et al. (2018b) shows that in some cases the size of the optimal partition (in so much that the above procedure becomes exact) does not depend on the number of scenarios S.

4. Once a full tail of scenarios has been added to the MP, one can generate or add new probabilities $\pi \in \Delta^{\beta}$ without significantly increasing the MP, but by potentially increasing the convergence speed in view of Lemma 3.3.

In addition, the following strategies could reduce the MP computational time:

1. Replacing the set of constraints in (10) and (14) from $\forall l \leq k$ to $\forall l \in \Theta_k$, where $\Theta_k \subset \{1, 2, \ldots, k\}$ is a subset of previous iterations. Different criteria to define Θ_k might be used:
   (a) The last $n_i \geq 1$ iterations $\Theta_k \subset \{k - n_i + 1, \ldots, k - 1, k\}$.
   (b) The last iteration and all those associated with constraints which were active after the previous master problem was solved.

   In both cases, once the MP (10) or (14) is solved, it is easy to check whether the iterations included in set Θ_k were appropriate of not, depending on whether the optimal MP solution is higher than the current lower bound or not.
2. We could take advantage of current objective function lower bound when solving MP (10) or (14) by including the following constraint:

$$c^{\mathsf{T}} x + r \geq \max_{j=1,\ldots,k} \{c^{\mathsf{T}} x^j + r^j\}. \tag{16}$$

   Constraint (16) allows MIP solvers to stop the solution process if the best integer and feasible solution equals to the current lower bound, without the need to continue branching until the error criteria holds. This is especially useful if the last set of constraints does not increase the updated lower bound.

5. Numerical experiments

We will illustrate the effectiveness of the proposed algorithms on various instances from the literature. In particular:

- We will benchmark CPU time for both algorithms against:
  1. The extended version (5).
  2. A Benders decomposition approach applied to problem (5) where complicating variables correspond to first-stage binary variables.

- We test several cases for an increasing number of scenarios S and different confidence levels $\beta = 0.00$, $\beta = 0.90$, $\beta = 0.95$ and $\beta = 0.99$. Note that $\beta = 0.00$ corresponds to the risk neutral model and it is used for illustrative purposes.

- Every time we call CPLEX solver, we set out a maximum time limit of 36000 cpu seconds.
- We limit the maximum number of iterations for both algorithms to 20.

All computational experiments have been run on a Windows Server with two Intel® Xeon® E5-2698 v3 processors at 2.3 GHz and 256 GB of RAM using CPLEX 12.7 under GAMS 24.8.3. To avoid performace differences due to the computational load of the server we set out a maximum number of 8 threads for the parallel computations of the branch and bound procedure.

5.1. Illustrative example: Location-transportation problem

This illustrative example is adapted from Zeng & Zhao (2013). The problem of supplying commodities to customers requires the existence of storage at m potential facilities, which are then transported from their location to n customers. Storage facilities are characterized by the fixed cost f_i of building them at locations i and their corresponding cost associated with their possible capacities a_i. Demand for each customer corresponds to d_j and the unit transportation cost from storage i to customer j is c_{ij}. The maximum allowable capacity at any site i is K_i, and for the problem to be feasible the following condition must hold $\sum_{\forall i} K_i \geq \sum_{\forall j} d_j$. The construction of a storage facility at location i is modelled by binary variable $y_i \in \{0, 1\}$, the capacity is controlled through z_i, and x_{ijk} is the transportation amount between storage i and customer j at scenario k. Deviations from nominal demands d_j are uncertain. Assuming that we have at our disposal S different scenarios of possible deviations from nominal demands to be satisfied simultaneously with probability p_k, the stochastic formulation based on CVaR (5) becomes:

$$\underset{\eta, y_i, z_i; \forall i, x_{ijk}; \forall (i,j,k)}{\text{Minimize}} \quad \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} a_i K_i y_i + \eta + \frac{1}{1-\beta} \sum_{k=1}^{S} p_k v_k \ , \tag{17}$$

subject to

$$\sum_{j=1}^{n} x_{ijk} \leq y_i K_i; i = 1, \ldots, m; k = 1, \ldots, S \tag{18}$$

$$\sum_{i=1}^{m} x_{ijk} \geq d_j + \xi_{jk}; j = 1, \ldots, n; k = 1, \ldots, S \tag{19}$$

$$x_{ijk} \geq 0; i = 1, \ldots, m; k = 1, \ldots, S \tag{20}$$

$$y_i \in 0, 1; \quad i = 1, \ldots, m \tag{21}$$

$$v_k \geq \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ijk} - \eta; \quad k = 1, \ldots, S \tag{22}$$

$$v_k \geq 0; \quad k = 1, \ldots, S. \tag{23}$$

Let consider the same data given in Zeng & Zhao (2013) with three possible storage locations and three demands: $f_1 = 400\$$, $f_2 = 414\$$, $f_3 = 326\$$, $a_1 = 18\$/\text{ud.}$, $a_2 =$

CG based algorithm			CCG based algorithm				
β	S	n_{sub}	β	S	n_{sub}	n_c^0	Δn_c
0.00	1000	1	0.00	1000	1	100	5
0.00	10000	10	0.00	10000	10	100	10
0.00	100000	50	0.00	100000	50	250	20
0.90	1000	1	0.90	1000	1	100	5
0.90	10000	10	0.90	10000	10	100	10
0.90	100000	50	0.90	100000	50	100	10
0.95	1000	1	0.95	1000	1	25	5
0.95	10000	10	0.95	10000	10	100	10
0.95	100000	50	0.95	100000	50	100	20
0.99	1000	1	0.99	1000	1	5	1
0.99	10000	10	0.99	10000	10	100	10
0.99	100000	50	0.99	100000	50	100	10

Table 1: Location-transportation illustrative example – Selection of parameters for computational experiments.

25\$/ud., $a_3 = 20$\$/ud., $K_1 = 800$ud., $K_2 = 800$ud., $K_3 = 800$ud., $c_{1,1} = 22$\$, $c_{1,2} = 33$\$, $c_{1,3} = 24$\$, $c_{2,1} = 33$\$, $c_{2,2} = 23$\$, $c_{2,3} = 30$\$, $c_{3,1} = 20$\$, $c_{3,2} = 25$\$ and $c_{3,3} = 27$\$. Nominal demands are $d_1 = 246$ud., $d_2 = 314$ud. and $d_3 = 260$ud. Deviations from nominal values are assumed to be uncertain and normally distributed with null expected values and standard deviations equal to $\sigma_1 = \sigma_2 = \sigma_3 = 40$ud. In addition, we assume that correlations are equal to $\rho_{1,2} = 0.7$, $\rho_{1,3} = 0.6$ and $\rho_{2,3} = 0.8$.

In this particular example, second-stage problems (2) for different number of scenarios are solved using (15) according to Table 1, where n_{sub} is the number of equally sized groups. This table also shows the initial number of clusters and the rate of change between iterations for the CCG algorithm. Computational results for this case are given in Table 2. This table provides for each computational experiment at its "optimal" and/or "near-optimal" solutions, the number of iterations to reach the required accuracy, the optimal objective lower and upper bounds ν_k and u_k, respectively, optimal first-stage costs $c^{\mathsf{T}}x^*$, the number of scenarios used for each MP, the number of clusters n_c, the relative error between bounds ε, and CPU time in seconds. In this particular example, the required tolerance is set to be 0.01%.

From Tables 2 and 3, the following observations are in order:

1. The extended version attains the global optimal solution with null error in all cases, like the CG algorithm, thus proving optimality. The Benders approach also attains the global optimal solution, except when $S = 100000$ and $\beta = 0.90$ and $\beta = 0.95$, where it reaches the maximum CPU time limit. In contrast, the CCG algorithm stop the process once the relative error between lower and upper bounds is below the required tolerance of 0.01%. In some cases the method has reached the optimal solution, but stops without proving it (e.g., see the case $\beta = 0.90$ and $S = 1000$).
2. In terms of computational performance both iterative algorithms outperform the extended and Benders versions, and the difference becomes increasingly better as

$\text{CVaR}_\beta(V(x,\xi))$ extended formulation (5)									
β	S	k	ν_k (\$)	u_k (\$)	$c^\mathsf{T}x^*$ (\$)	MP no scenarios	n_c	ε ($\times 10^{-4}$)	CPU time (s)
0.00	1000	1	40051.88	40051.88	20714.39	1000	0	0.00	1.11
0.00	10000	1	40687.79	40687.79	21392.23	10000	0	0.00	27.58
0.00	100000	1	40967.24	40967.24	21672.76	100000	0	0.00	6132.97
0.90	1000	1	43046.88	43046.88	20920.72	1000	0	0.00	2.33
0.90	10000	1	43658.52	43658.52	21596.09	10000	0	0.00	136.51
0.90	100000	1	43917.80	43917.80	21862.04	100000	0	0.00	16799.83
0.95	1000	1	43583.42	43583.42	20748.78	1000	0	0.00	1.83
0.95	10000	1	44212.95	44212.95	21630.23	10000	0	0.00	99.03
0.95	100000	1	44434.68	44434.68	21877.95	100000	0	0.00	22039.06
0.99	1000	1	44713.25	44713.25	20982.62	1000	0	0.00	2.53
0.99	10000	1	45287.50	45287.50	21671.96	10000	0	0.00	130.00
0.99	100000	1	45439.27	45439.27	21954.31	100000	0	0.00	24862.52
$\text{CVaR}_\beta(V(x,\xi))$ extended formulation (5) through Benders decomposition									
β	S	k	ν_k (\$)	u_k (\$)	$c^\mathsf{T}x^*$ (\$)	MP no scenarios	n_c	ε ($\times 10^{-4}$)	CPU time (s)
0.00	1000	1	40051.88	40051.88	20714.39	1000	0	0.00	1.67
0.00	10000	1	40687.79	40687.79	21392.23	10000	0	0.00	35.82
0.00	100000	1	40967.24	40967.24	21672.76	100000	0	0.00	6732.73
0.90	1000	1	43046.88	43046.88	20748.78	1000	0	0.00	2.58
0.90	10000	1	43658.52	43658.52	21596.09	10000	0	0.00	399.02
0.90	100000	1	43917.80	44331.80	22276.04	100000	0	93.39	36317.29*
0.95	1000	1	43583.42	43583.42	20748.78	1000	0	0.00	2.43
0.95	10000	1	44212.95	44212.95	21630.10	10000	0	0.00	301.46
0.95	100000	1	44434.68	44848.68	22291.95	100000	0	92.31	36257.49*
0.99	1000	1	44713.25	44713.25	20982.62	1000	0	0.00	3.20
0.99	10000	1	45287.50	45287.50	21671.46	10000	0	0.00	374.65
0.99	100000	1	45439.27	45439.27	21954.31	100000	0	0.00	21600.37

* Maximum time limit reached

Table 2: Location-transportation illustrative example – Results using extended formulation (5), and extended formulation (5) through Benders decomposition for different confidence levels β and number of scenarios S.

the number of scenarios increases, being around two orders of magnitude faster for 100000 scenarios. This behaviour is shown in Figure 1, which represents the evolution of the computational time for an increasing number of scenarios in a double logarithmic scale.

3. The reduction in computational times of both iterative algorithms with respect to the extended and Benders versions is due to the reduction in the number of scenarios deployed in the MP, where CG algorithm uses around $(1-\beta)S$ scenarios, i.e., reductions about 90%, 95% and 99% for $\beta = 0.9$, $\beta = 0.95$ and $\beta = 0.99$, respectively, while the CCG algorithm uses a maximum of several hundred clusters.

$\text{CVaR}_\beta(V(x,\xi))$ solution CG based algorithm									
β	S	k	ν_k (\$)	u_k (\$)	$c^\mathsf{T}x^*$ (\$)	MP no scenarios	n_c	ε ($\times 10^{-4}$)	CPU time (s)
0.00	1000	3	40051.88	40051.88	20714.39	1000	0	0.00	1.17
0.00	10000	3	40687.79	40687.79	21392.23	10000	0	0.00	18.96
0.00	100000	3	40967.24	40967.24	21672.76	100000	0	0.00	2056.64
0.90	1000	4	43046.88	43046.88	20748.78	101	0	0.00	0.60
0.90	10000	4	43658.52	43658.52	21531.57	1001	0	0.00	6.61
0.90	100000	4	43917.80	43917.80	21861.81	10001	0	0.00	105.30
0.95	1000	4	43583.42	43583.42	20748.78	51	0	0.00	0.50
0.95	10000	4	44212.95	44212.95	21531.57	501	0	0.00	5.05
0.95	100000	4	44434.68	44434.68	21810.99	5001	0	0.00	66.11
0.99	1000	3	44713.25	44713.25	20748.78	11	0	0.00	0.41
0.99	10000	4	45287.50	45287.50	21531.57	101	0	0.00	4.33
0.99	100000	4	45439.27	45439.27	21873.17	1001	0	0.00	42.40
$\text{CVaR}_\beta(V(x,\xi))$ solution CCG algorithm									
β	S	k	ν_k (\$)	u_k (\$)	$c^\mathsf{T}x^*$ (\$)	MP no scenarios	n_c	ε ($\times 10^{-4}$)	CPU time (s)
0.00	1000	4	40049.94	40052.24	20692.52	207	110	0.57	0.72
0.00	10000	4	40687.97	40688.18	21360.53	212	120	0.05	3.78
0.00	100000	3	40964.82	40967.37	21652.87	251	270	0.62	25.32
0.90	1000	3	43042.60	43046.88	20748.78	101	100	0.99	0.44
0.90	10000	5	43658.67	43658.75	21660.33	333	130	0.02	5.00
0.90	100000	5	43914.04	43918.38	21692.90	333	130	0.99	39.60
0.95	1000	4	43583.42	43583.82	21081.02	57	35	0.09	0.55
0.95	10000	4	44212.84	44212.95	21531.57	212	120	0.02	4.06
0.95	100000	4	44433.87	44435.17	21714.18	222	140	0.29	31.50
0.99	1000	4	44713.25	44715.50	21211.24	13	7	0.50	0.48
0.99	10000	4	45287.50	45287.50	21531.57	212	110	0.00	3.83
0.99	100000	4	45437.30	45439.27	21819.71	212	120	0.43	32.35

Table 3: Location-transportation illustrative example – Results using constraint generation (CG) and clustering and constraint generation (CCG) based algorithms for different confidence levels β and number of scenarios S.

The reduction in the number of scenarios for the CCG algorithm with respect to CG does slightly reflect in terms of computational performance, especially for a large number of scenarios. The reduction becomes significant for the expected case, where CG algorithm uses all initial set of scenarios. It is worth pointing out that despite what could be expected, CG algorithm is faster with respect to the extended and Benders approaches.

4. Both CG and CCG approaches present larger reductions in computational time with respect to the extended and Benders approaches when the confidence level β increases. The reason is that the number of scenarios characterizing the tail of the distribution decreases as β increases, and the MP requires less number of scenarios.

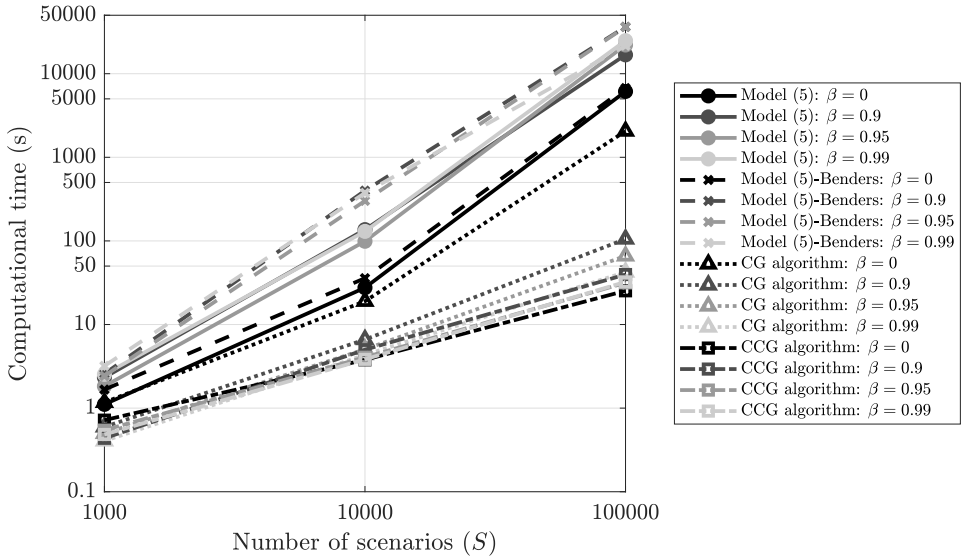

Figure 1: Evolution of the computational time for numerical experiments related to the location-transportation illustrative example.

Even though this illustrative example is relatively simple, because the number of first-stage variables is just six with only three binary variables and the rest of variables are continuous, the proposed algorithms prove to be valuable alternatives for practitioners, especially when dealing with a large number of scenarios.

Regarding the selection of parameters n_c^0 and Δn_c shown in Table 1, it has been done so that the number of scenarios used in the master plan is reduced with respect to that required according to CG algorithm. One must be careful because if the number of clusters is greater than the number of scenarios above the value-at-risk the CCG method turns out to be the CG strategy. It can be observed in Table 3 that in those cases where the number of scenarios of the MP for the CCG approach is lower than that required by CG algorithm then CCG outperforms CG method, and conversely, if the number of scenarios of the MP for the CCG approach is greater than that required by CG algorithm, CG outperforms CCG method (see, for instance, the case $\beta = 0.99$ and $S = 1000$).

5.2. Power system transmission expansion planning case study

Transmission network expansion planning consists in determining how to expand and reinforce the transmission network in order to meet the future demand with the available generation assets (Lumbreras & Ramos, 2016). The primary drivers of transmission network expansion planning models are uncertain demand growth and uncertainty related to available generation capacity, mostly because the growing installation of renewable-based generation assets comes into play.

The stochastic expansion planning model determines the optimal expansion plan x min-

imizing a composite objective function comprising the investment cost $c^\mathsf{T}x$ and the operating cost subject to an investment budget $c^\mathsf{T}x \leq B$ and the integrality of first-stage decisions $x \in \mathbb{N}$. These variables represent the decisions about *constructing/not constructing* each of the candidate lines which can be built to increase capacity. These decisions are taken before the realization of uncertainty, i.e., before knowing demands and power generation capacity. The second-stage reacts once uncertainty is revealed assigning how much energy is provided by each generator according to its actual capacity and distributing it through the network at minimum operating costs. A detailed description of the mathematical formulation associated with problem (5) particularized for this example is given in Appendix A.

CG algorithm			CCG algorithm				
β	S	n_{sub}	β	S	n_{sub}	n_c^0	Δn_c
0.00	1000	5	0.00	1000	5	500	5
0.00	2000	10	0.00	2000	10	500	5
0.00	5000	25	0.00	5000	25	500	5
0.90	1000	5	0.90	1000	5	25	5
0.90	2000	10	0.90	2000	10	100	5
0.90	5000	25	0.90	5000	25	100	5
0.95	1000	5	0.95	1000	5	20	5
0.95	2000	10	0.95	2000	10	50	5
0.95	5000	25	0.95	5000	25	100	5
0.99	1000	5	0.99	1000	5	5	1
0.99	2000	10	0.99	2000	10	5	1
0.99	5000	25	0.99	5000	25	25	5

Table 4: Transmission expansion planning case study – Selection of parameters for computational experiments.

Let consider a system consisting of 6 buses, 3 generating units, 5 loads, and 6 existing lines with data as described in Roldán et al. (2018a,b). Note that a \$10-million annualized investment budget is imposed and each pair of buses can be connected by three lines at most, which amounts to 39 candidate lines. Deviations from nominal demands and generation capacities are sources of uncertainty. The standard deviation of uncertain deviations associated with demand is 20% of their corresponding nominal values, for generation capacities the standard deviations related to deviations from nominal values are very low and equal to 0.2% for generators 1 and 2, and negligible for generator 3. Since we are more interested in computational performance than in reproducing accurately the physics of the problem, we assume that uncertain deviations are normally distributed.

In this particular example, second-stage problems (2) for different number of scenarios are solved using (15) according to parameters given in Table 4, where n_{sub} is the number of equally sized groups. This table also shows the initial number of clusters and the rate of change between iterations for the CCG algorithm. In this particular example, the required tolerance is set out to 0.1%. Computational results analogous to those given in Tables 2 and 3 are provided in Tables 5 and 6. From these tables, the following observations are in order:

$\text{CVaR}_\beta(V(x,\xi))$ solution extended formulation (5)									
β	S	k	ν_k ($)	u_k ($)	$c^\mathsf{T}x^*$ ($)	MP no scenarios	n_c	ε ($\times 10^{-3}$)	CPU time (s)
0.00	1000	1	205.80	206.00	5.75	1000	0	0.97	9630.03
0.00	2000	1	204.45	205.77	4.63	2000	0	6.41	36005.00*
0.00	5000	1	204.17	284.57	7.97	5000	0	282.54	36014.00*
0.90	1000	1	242.22	242.46	6.49	1000	0	0.98	9884.84
0.90	2000	1	241.52	242.70	7.07	2000	0	4.85	36006.14*
0.90	5000	1	240.19	384.83	7.97	5000	0	375.88	36009.63*
0.95	1000	1	249.86	250.10	6.49	1000	0	1.00	10600.50
0.95	2000	1	246.78	249.07	7.07	2000	0	9.18	36002.91*
0.95	5000	1	246.73	404.14	7.97	5000	0	389.50	36010.08*
0.99	1000	1	262.06	262.32	8.19	1000	0	0.99	13400.22
0.99	2000	1	260.63	262.80	7.22	2000	0	8.29	36004.95*
0.99	5000	1	261.27	440.28	7.97	5000	0	406.59	36015.06*
$\text{CVaR}_\beta(V(x,\xi))$ solution formulation (5) through Benders									
β	S	k	ν_k ($)	u_k ($)	$c^\mathsf{T}x^*$ ($)	MP no scenarios	n_c	ε ($\times 10^{-3}$)	CPU time (s)
0.00	1000	1	205.32	206.00	5.75	1000	0	3.29	36002.50*
0.00	2000	1	204.21	205.69	5.37	2000	0	7.19	36008.74*
0.00	5000	1	203.90	288.26	8.55	5000	0	292.65	36019.61*
0.90	1000	1	241.02	242.46	6.49	1000	0	5.94	36003.19*
0.90	2000	1	240.09	243.80	5.75	2000	0	15.20	36006.31*
0.90	5000	1	239.90	390.18	8.55	5000	0	385.14	36009.51*
0.95	1000	1	248.15	250.10	6.49	1000	0	7.83	36002.92*
0.95	2000	1	246.36	249.80	6.33	2000	0	13.77	36003.17*
0.95	5000	1	245.39	409.76	8.55	5000	0	401.15	36009.68*
0.99	1000	1	259.84	262.41	6.87	1000	0	9.79	36002.41*
0.99	2000	1	259.56	263.81	6.49	2000	0	16.10	36003.64*
0.99	5000	1	260.79	265.64	0.00	5000	0	18.28	36013.35*

* Maximum time limit reached

Table 5: Transmission expansion planning case study – Results using extended formulation (5)and extended formulation (5) through Benders decomposition for different confidence levels β and number of scenarios S.

1. The extended version is incapable of reaching a solution with an error lower than the tolerance within the maximum CPU time limit of 10 hours imposed for $S = 2000$ and $S = 5000$. Only for the cases with $S = 1000$ scenarios solutions within tolerance limit are reached. Using the (basic) Benders approach results are even worse. In contrast, the CG based algorithm stop the process once the relative error between lower and upper bounds is below the tolerance of 0.1% for all cases but one, the risk-neutral case with $S = 5000$. CCG based algorithm always reaches a solution within tolerance limits.
2. In terms of computational performance, the CG iterative algorithm clearly outper-

CVaR$_\beta(V(x,\xi))$ solution CG based algorithm									
β	S	k	ν_k (\$)	u_k (\$)	$c^\mathsf{T}x^*$ (\$)	MP no scenarios	n_c	ε ($\times 10^{-3}$)	CPU time (s)
0.00	1000	7	206.00	206.00	5.75	1000	0	0.00	5311.95
0.00	2000	4	205.69	205.69	5.37	2000	0	0.00	24670.33
0.00	5000	3	204.21	206.24	5.41	5000	0	9.84	36043.441*
0.90	1000	8	242.46	242.46	6.49	101	0	0.00	279.55
0.90	2000	5	242.70	242.70	7.07	201	0	0.00	2147.82
0.90	5000	4	242.76	242.76	7.07	501	0	0.00	13509.37
0.95	1000	9	250.10	250.10	6.49	51	0	0.00	109.72
0.95	2000	5	249.07	249.07	7.07	101	0	0.00	541.14
0.95	5000	4	249.50	249.50	7.07	251	0	0.00	4974.15
0.99	1000	9	262.32	262.32	8.19	11	0	0.00	32.39
0.99	2000	5	262.42	262.42	7.61	21	0	0.00	36.17
0.99	5000	4	264.44	264.49	8.19	51	0	0.20	292.53
CVaR$_\beta(V(x,\xi))$ solution CCG algorithm									
β	S	k	ν_k (\$)	u_k (\$)	$c^\mathsf{T}x^*$ (\$)	MP no scenarios	n_c	ε ($\times 10^{-3}$)	CPU time (s)
0.00	1000	7.00	205.93	206.00	5.75	218	255	0.37	58.84
0.00	2000	5.00	206.17	206.24	5.73	856	510	0.33	1897.23
0.00	5000	5.00	206.78	206.89	6.35	451	505	0.55	803.03
0.90	1000	8.00	242.22	242.46	6.49	56	35	0.99	66.80
0.90	2000	5.00	242.50	242.70	7.07	165	110	0.82	440.14
0.90	5000	4.00	242.59	242.76	7.07	196	110	0.73	913.38
0.95	1000	9.00	249.94	250.10	6.49	40	30	0.65	57.78
0.95	2000	5.00	248.99	249.07	7.07	91	60	0.32	104.39
0.95	5000	4.00	249.30	249.50	7.80	174	110	0.81	798.48
0.99	1000	9.00	262.29	262.32	8.19	11	7	0.08	31.70
0.99	2000	9.00	262.32	262.42	7.61	49	11	0.35	187.92
0.99	5000	5.00	264.38	264.49	8.19	77	40	0.41	372.88

* Maximum time limit reached

Table 6: Transmission expansion planning case study – Results using constraint generation (CG) and clustering and constraint generation (CCG) based algorithms for different confidence levels β and number of scenarios S.

forms the extended and Benders solution approaches, and the difference becomes increasingly better as the number of scenarios increases. The reason is the use of $(1-\beta)S$ scenarios for the MP, with reductions in the number of scenarios of about 90%, 95% and 99% for $\beta = 0.9$, $\beta = 0.95$ and $\beta = 0.99$, respectively. If we compare the CG and CCG algorithms, CCG shows, in general, better performance with respect to CG for confidence levels below or equal to $\beta = 0.95$, proving that the combination of clustering techniques and dynamic constraint generation seem to be the best alternative for this type of problems provided that an appropriate selection in the number of clusters is chosen. This behaviour is shown in Figure 2,

which represents the evolution of the computational time for an increasing number of scenarios with the computational time in logarithmic scale.

3. Computational times for the CG and CCG algorithms decrease with respect to that taken by the extended and Benders approaches as the confidence level β increases. The reason is that the number of scenarios characterizing the tail of the distribution decreases as β increases, and the MP requires less number of scenarios.

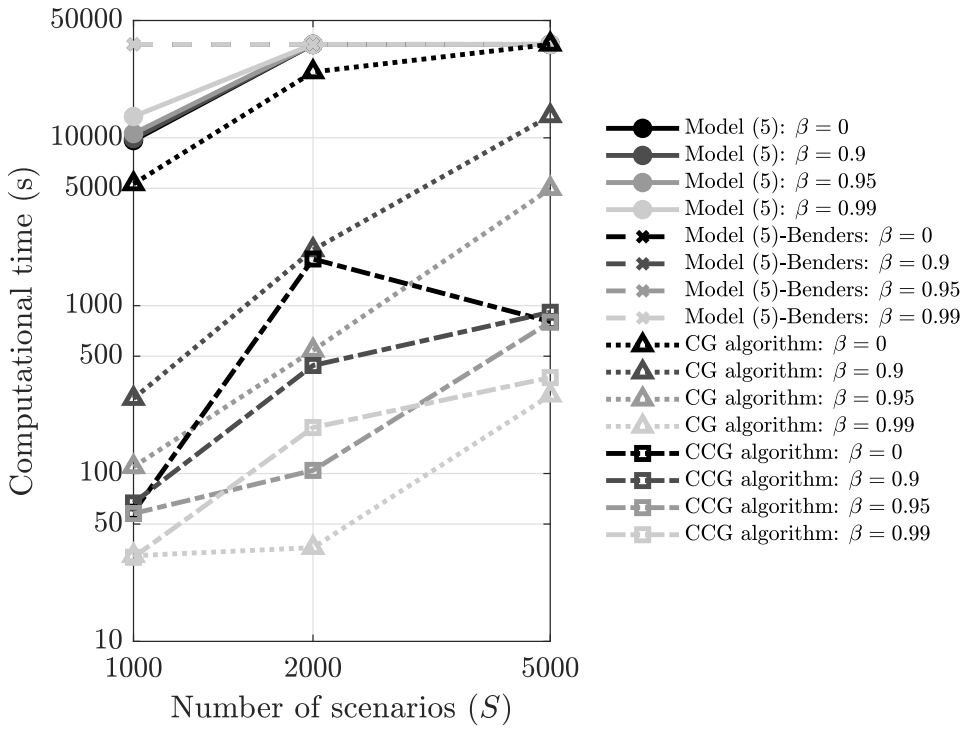

Figure 2: Evolution of the computational time for experiments related to the transmission expansion planning case study.

Computational results from this case study show that the proposed iterative algorithms clearly outperform extended and Benders approaches for large scale systems and/or a large number of scenarios. For fixed recourse problems, analogously to the previous computational example, the criteria to select between CG or CCG algorithms depends on the number of scenarios required by the master problem to achieve the desired accuracy.

6. Conclusions and perspectives

This paper presents two algorithms for solving stochastic two-stage linear programs based on scenarios and/or finite sampling using Conditional Value-at-Risk. Both algorithms are

based on a constraint generation procedure, which allows iteratively solving the problem until the relative gap between upper and lower bounds is below the required accuracy. The basic version does not require any particular problem structure, however, the version including clusterization is limited to fixed recourse problems. Numerical experiments show that these algorithms are especially suitable for solving medium- and large-scale systems and/or problems with a large number of scenarios. The proposed algorithms have the following features:

1. Convergence of both iterative algorithms in a finite number of steps is theoretically guaranteed as it should be.
2. Computational gains with respect to the extended and Benders versions are good, which allows improving the stochastic characterization of uncertain parameters by including additional scenarios.
3. Unlike methods based on scenario reduction techniques, the proposed methods allow incorporating the complete probabilistic information given by the scenario set.
4. Several additional strategies to further improve computational performance are given and could further enhance performance.
5. The algorithms are also applicable for the risk-neutral case. In fact, the CCG algorithm avoids the necessity to respond to the always difficult to answer question about how many clusters are required to solve the neutral case with a given accuracy.

The criteria to choose between the CG and CCG algorithms for fixed recourse problems should be based on the expected number of scenarios to be used in the master program, which basically consists of comparing the approximate magnitudes $(1-\beta)S$ and n_c, being n_c the number of clusters.

In summary, this paper proposes alternative methods for solving risk averse two-stage stochastic programs with many scenarios. Even though the methods proposed in this paper focus on two-stage stochastic models, they could also be of application for single-stage and/or multi-stage problems. Analysis of computational performance in these other types of stochastic problems is a subject for further research. Another avenue of research is the application of these algorithms with second-stage problems including integer variables. Integer restrictions can immediately be incorporated in the CG algorithm, but may require appropriate use of continuous relaxations to extend the CCG algorithm.

Acknowledgments

Dr. Mínguez was fully supported by the Public State Employment Service of the Ministry of Labour, Migration and Social Security of Spain. This work was partially supported by the Ministry of Science, Innovation, and Universities of Spain under Projects RTI2018-096108-A-I00 and RTI2018-098703-B-I00 (MCIU/AEI/FEDER, UE).

Appendix A. Mathematical formulation of the transmission expansion planning problem

Appendix A.1. Sets

Ω^{D} Set of indexes k of loads.

Ω_n^{D} Set of indexes k of loads connected to bus n.

Ω^{G} Set of indexes i of generating units.

Ω_n^{G} Set of indexes i of generating units connected to bus n.

Ω^{L} Set of indexes l of existing transmission lines.

Ω^{L^+} Set of indexes l of candidate transmission lines.

Ω^{N} Set of indexes n of buses.

Appendix A.2. Constants

B Annualized investment budget for new lines.

C_i^{G} Production cost coefficient of unit i.

C_l^{L} Construction cost of line l.

$fr(l)$ Sending or origin bus of line l.

$\overline{P}_l^{\mathrm{L}}$ Power flow capacity of line l.

$to(l)$ Receiving or destination bus of line l.

x_l Reactance of line l.

σ Number of hours during one year 8760.

Appendix A.3. Upper-level variables

v_l Construction status of candidate line l which is equal to 1 if the line is built, being 0 otherwise.

Appendix A.4. Uncertain or random variables

$\overline{p}_k^{\mathrm{D}}$ Nominal value of demand of load k.

$\overline{p}_i^{\mathrm{G}}$ Nominal value of generation capacity of unit i.

ξ_{kj}^{D} Uncertain deviation of demand of load k for scenario j.

ξ_{ij}^{G} Uncertain deviation of generation capacity of unit i for scenario j.

Appendix A.5. Lower-level variables

θ_{nj} Phase angle at bus n for scenario j.

p_{ij}^{G} Power output of unit i for scenario j.

p_{lj}^{L} Power flow through line l for scenario j.

Appendix A.6. Problem formulation

Based on Mínguez et al. (2018), problem (5) particularized for this example can be cast as follows:

$$\underset{v_l,\theta_{nj},p_{ij}^{\mathrm{G}},p_{lj}^{\mathrm{L}},z_j,\eta}{\text{Minimize}} \quad \sum_{\forall l\in\Omega^{\mathrm{L}^+}} C_l^{\mathrm{L}} v_l + \sigma\left(\eta + \frac{1}{1-\beta}\sum_{j=1}^{S} p_j z_j\right) \tag{A.1}$$

$$\text{subject to} \quad \sum_{\forall l\in\Omega^{\mathrm{L}^+}} C_l^{\mathrm{L}} v_l \le B \tag{A.2}$$

$$v_l \in \{0,1\}; \forall l \in \Omega^{\mathrm{L}^+} \tag{A.3}$$

$$z_j \ge \sum_{i\in\Omega^{\mathrm{G}}} C_i^{\mathrm{G}} p_{ij}^{\mathrm{G}} - \eta; \quad j = 1,\ldots,S \tag{A.4}$$

$$z_j \ge 0; \quad j = 1,\ldots,S \tag{A.5}$$

$$\sum_{i\in\Omega_n^{\mathrm{G}}} p_{ij}^{\mathrm{G}} + \sum_{l\in(\Omega^{\mathrm{L}}\cup\Omega^{\mathrm{L}^+})|to(l)=n} p_{lj}^{\mathrm{L}} - \sum_{l\in(\Omega^{\mathrm{L}}\cup\Omega^{\mathrm{L}^+})|fr(l)=n} p_{lj}^{\mathrm{L}} = \sum_{k\in\Omega_n^{\mathrm{D}}} (\overline{p}_k^{\mathrm{D}} + \xi_{kj}^{\mathrm{D}}); \forall n \in \Omega^{\mathrm{N}};\ j = 1,\ldots,S \tag{A.6}$$

$$p_{lj}^{\mathrm{L}} = \frac{1}{x_l}(\theta_{fr(l)j} - \theta_{to(l)j}); \forall l \in \Omega^{\mathrm{L}};\ j = 1,\ldots,S \tag{A.7}$$

$$p_{lj}^{\mathrm{L}} = \frac{v_l}{x_l}(\theta_{fr(l)j} - \theta_{to(l)j}); \forall l \in \Omega^{\mathrm{L}^+};\ j = 1,\ldots,S \tag{A.8}$$

$$-\overline{P}_l^{\mathrm{L}} \le p_{lj}^{\mathrm{L}} \le \overline{P}_l^{\mathrm{L}}; \forall l \in (\Omega^{\mathrm{L}} \cup \Omega^{\mathrm{L}^+});\ j = 1,\ldots,S \tag{A.9}$$

$$0 \le p_{ij}^{\mathrm{G}} \le \overline{p}_i^{\mathrm{G}} + \xi_{ij}^{\mathrm{G}}; \forall i \in \Omega^{\mathrm{G}};\ j = 1,\ldots,S. \tag{A.10}$$

Problem (A.1)–(A.10) is driven by the minimization of the construction and weighted conditional value-at-risk of operating cost with respect to the values of upper-level variables v_l and lower-level variables for given values of uncertain deviations ξ_{kj}^{D} and ξ_{ij}^{G}. Constraint (A.1) imposes investment budget for building candidate transmission lines. Constraints (A.2) define the binary nature of transmission line investment variables. Expressions (A.3)-(A.4) allow to determine the value of the auxiliary variables z_j. Expressions (A.6)–(A.9) model the effect of the network including nodal power balances (A.6), line flows through existing lines (A.7), line flows through candidate lines (A.8), and line flow limits (A.9). Finally, constraints (A.10) set the generation limits. Note that in this particular case we do not include the possibility of load-shedding, which means that the problem might be infeasible if demand is not satisfied completely.

References

References

Ahmed, S. (2006). Convexity and decomposition of mean-risk stochastic programs. *Mathematical Programming*, *106*, 433–446. doi:10.1007/s10107-005-0638-8.

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

Birge, J. R. (1985). Aggregation bounds in stochastic linear programming. *Mathematical Programming*, *31*, 25–41. doi:10.1007/BF02591859.

Bonnans, J. (2019). *Convex and Stochastic Optimization*. (1st ed.). Springer Nature Switzerland.

Conejo, A. J., Castillo, E., Mínguez, R., & García-Bertrand, R. (2006). *Decomposition techniques in mathematical programming. Engineering and science applications*. New York: Springer-Verlag Berlin Heidelberg.

Crainic, T., Hewitt, M., & Rei, W. (2014). Scenario grouping in a progressive heding-based meta-heuristic for stochastic network design. *Computers and Operations Research*, *43*, 90–99. doi:10.1016/j.cor.2013.08.020.

Dupačová, J., Gröwe-Kuska, N., & Römisch, W. (2003). Scenario reduction in stochastic programming: An approach using probability metrics. *Mathematical Programming*, *95*, 493–511. doi:10.1007/s10107-002-0331-0.

Fábián, C. (2000). Bundle-type methods for inexact data. In *Proceedings of the XXIV Hungarian Operations Research Conference (Veszprém, 1999)* (pp. 35–55). volume 8 (special issue, T. Csendes and T. Rapcsák, eds.).

Fábián, C., Wolf, C., Koberstein, A., & Suhl, L. (2015). Risk-averse optimization in two-stage stochastic models: computational aspects and a study. *SIAM Journal on Optimization*, *25*, 28–52. doi:10.1137/130918216.

Fábián, C. I. (2008). Handling CVaR objectives and constraints in two-stage stochastic models. *European Journal of Operational Research*, *191*, 888–911. doi:10.1016/j.ejor.2007.02.052.

García-Bertrand, R., & Mínguez, R. (2014). Iterative scenario based reduction technique for optimizing the conditional value-at-risk. *Optimization and Engineering*, *15*, 355–380. doi:10.1007/s11081-012-9201-7.

Heitsch, H., & Römisch, W. (2003). Scenario reduction algorithms in stochastic programming. *Computation Optimization and Applications*, *24*, 187–206. doi:10.1023/A:1021805924152.

Heitsch, H., & Römisch, W. (2007). A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, *35*, 731–738. doi:10.1016/j.orl.2006.12.008.

Heitsch, H., & Römisch, W. (2009). Scenario tree reduction for multistage stochastic programs. *Computational Management Science*, *6*, 117–133. doi:10.1007/s10287-008-0087-y.

Henrion, R., Küchler, C., & Römisch, W. (2008). Discrepancy distances and scenario reduction in two-stage stochastic integer programming. *Journal of Industrial and Management Optimization*, *4*, 363–384. doi:10.3934/jimo.2008.4.363.

Henrion, R., Küchler, C., & Römisch, W. (2009). Scenario reduction in stochastic programming with respect to discrepancy distances. *Computational Optimization and Applications*, *43*, 67–93. doi:10.1007/s10589-007-9123-z.

Huang, Y., Zheng, Q. P., & Wang, J. (2014). Two-stage stochastic unit commitment model includingnon-generation resources with conditional value-at-risk constraints. *Electric Power Systems Research*, *116*, 427–438.

Jardim, D., Maceira, M., & Falcao, D. (2001). Stochastic streamflow model for hydroelectric systems using clustering techniques. In *Power Tech Proceedings, 2001 IEEE Porto* (p. 6 pp. vol.3). volume 3. doi:10.1109/PTC.2001.964916.

Kazemzadeh, N., Ryan, S. M., & Hamzeei, M. (2019). Robust optimization vs. stochastic programming incorporating risk measures for unit commitment with uncertain variable renewable generation. *Energy Systems*, *10*, 517–541. doi:10.1007/s12667-017-0265-5.

Krokhmal, P., Palmquist, J., & Uryasev, S. (2002). Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of Risk*, *4*, 11–27. doi:10.21314/JOR.2002.057.

Künzi-Bay, A., & Mayer, J. (2006). Computation aspects of minimizing conditional value-at-risk. *Computational Management science*, *3*, 3–27. doi:10.1007/s10287-005-0042-0.

Lemaréchal, C., Nemirovskii, A., & Nesterov, Y. (1995). New variants of bundle methods. *Mathematical Programming*, *69*, 111–147. doi:10.1007/BF01585555.

Lumbreras, S., & Ramos, A. (2016). The new challenges to transmission expansion planning.

Survey of recent practice and literature review. *Electric Power System Research*, *134*, 19–29. doi:10.1016/j.epsr.2015.10.013.

Mínguez, R., García-Bertrand, R., Arroyo, J. M., & Alguacil, N. (2018). On the solution of large-scale robust transmission network expansion planning under uncertain demand and generation capacity. *IEEE Transactions on Power Systems*, *33*, 1242–1251. doi:10.1109/TPWRS.2017.2734562.

Morales, J., Pineda, S., Conejo, A., & Carrión, M. (2009). Scenario reduction for futures market trading in electricity markets. *IEEE Transactions on Power Systems*, *24*, 878–888. doi:10.1109/TPWRS.2009.2016072.

de Oliveira, W., & Sagastizábal, C. (2014). Level bundle methods for oracles with on demand accuracy. *Optimization Methods and Software*, *29*, 1180–1209. doi:10.1080/10556788.2013.871282.

de Oliveira, W., Sagastizábal, C., & Scheimberg, S. (2011). Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, *21*, 517–544. doi:10.1137/100808289.

Pay, B. S., & Song, Y. (2018). Partition-based decomposition algorithms for two-stage stochastic integer programs with continuous recourse. *Annals of Operations Research*, (pp. 1–22). doi:10.1007/s10479-017-2689-7.

Pflug, G., & Wozabal, N. (2010). Asymptotic distribution of law-invariant risk functional. *Finance and Stochastics*, *14*, 397–418. doi:10.1007/s00780-009-0121-0.

Pflug, G. C. (2001). Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, *89*, 251–271. doi:10.1007/PL00011398.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: a literature review. *European Journal of Operational Research*, *259*, 801–817.

Rockafellar, R., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, *2*, 21–42. doi:10.21314/JOR.2000.038.

Rockafellar, R., & Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, *26*, 1443–1471. doi:10.1016/S0378-4266(02)00271-6.

Roldán, C., Mínguez, R., García-Bertrand, R., & Arroyo, J. M. (2018a). Robust transmission network expansion planning under correlated uncertainty. *IEEE Transactions on Power Systems*, . doi:10.1109/TPWRS.2018.2889032.

Roldán, C., Mínguez, R., García-Bertrand, R., & Arroyo, J. M. (2018b). Robust transmission network expansion planning under correlated uncertainty: Data for the case studies. Available: https://drive.google.com/open?id=1jVbVrCEl2tUW2WdGys5BMkLxYioDtK0G.

Rosa, C. H., & Takriti, S. (1999). Improving aggregation bounds for two-stage stochastic programs. *Operations Research Letters*, *24*, 127–137. doi:10.1016/S0167-6377(99)00019-X.

Sandikçi, B., Kong, N., & Schaefer, A. J. (2013). A hierarchy of bounds for stochastic mixed-integer programming. *Mathematical Programming*, *138*, 253–272. doi:10.1007/s10107-012-0526-y.

Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2009). *Lectures on Stochastic Programming. Modeling and Theory* volume 9 of *MPS-SIAM series on optimization*. SIAM and MPS, Philadelphia.

van Slyke, R., & Wets, R.-B. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics*, *17*, 638–663. doi:10.1137/0117061.

Song, Y., & Luedtke, J. (2015). An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. *SIAM Journal on Optimization*, *25*, 1344–1367. doi:10.1137/140967337.

van Ackooij, W. (2017). A comparison of four approaches from stochastic programming for large-scale unit-commitment. *EURO Journal on Computational Optimization*, *5*, 119–147. doi:10.1007/s13675-015-0051-x.

van Ackooij, W., Danti Lopez, I., Frangioni, A., Lacalandra, F., & Tahanan, M. (2018a). Large-scale unit commitment under uncertainty: an updated literature survey. *Annals of Operations Research*, *271*, 11–85. doi:10.1007/s10479-018-3003-z.

van Ackooij, W., & de Oliveira, W. (2014). Level bundle methods for constrained convex optimization with various oracles. *Computation Optimization and Applications*, *57*, 555–597. doi:10.1007/s10589-013-9610-3.

van Ackooij, W., de Oliveira, W., & Song, Y. (2018b). An adaptive partition-based level decomposition for solving two-stage stochastic programs with fixed recourse. *Informs Journal on Computing*, *30*, 57–70. doi:10.1287/ijoc.2017.0765.

Wolf, C., Fábián, C. I., Koberstein, A., & Stuhl, L. (2014). Applying oracles of on-demand accuracy in two-stage stochastic programming: A computational study. *European Journal of Operational Research*, *239*, 437–448. doi:10.1016/j.ejor.2014.05.010.

Wright, S. E. (1994). Primal-dual aggregation and disaggregation for stochastic linear programs. *Mathematics of Operations Research*, *19*, 893–908. URL: `http://www.jstor.org/stable/3690318`.

Zeng, B., & Zhao, L. (2013). Solving two-stage robust optimization problems using a column-and-

constraint generation method. *Operation Research Letters*, *41*, 457–461. doi:10.1016/j.orl.2013.05.003.