**RC25594 REVISED (WAT1609-061) September 21, 2016**
**Mathematics**

# IBM Research Report

## Nonlinear Chance-Constrained Problems with Applications to Hydro Scheduling

**Andrea Lodi[1], Enrico Malaguti[2], Giacomo Nannicini[3], Dimitri Thomopulos[2]**

[1]Polytechnique Montréal, Canada

[2]Università di Bologna, Italy

[3]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

# NONLINEAR CHANCE-CONSTRAINED PROBLEMS WITH APPLICATIONS TO HYDRO SCHEDULING*

ANDREA LODI†, ENRICO MALAGUTI‡, GIACOMO NANNICINI§, AND DIMITRI THOMOPULOS¶

**Abstract.** We present a Branch-and-Cut algorithm for a class of nonlinear chance-constrained mathematical optimization problems with a finite number of scenarios. Unsatisfied scenarios can enter a recovery mode. This class corresponds to problems that can be reformulated as deterministic convex mixed-integer nonlinear programming problems with indicator variables and continuous scenario variables, but the size of the reformulation is large and quickly becomes impractical as the number of scenarios grows. The Branch-and-Cut algorithm is based on an implicit Benders decomposition scheme, where we generate cutting planes as outer approximation cuts from the projection of the feasible region on suitable subspaces. The size of the master problem in our scheme is much smaller than the deterministic reformulation of the chance-constrained problem. We apply the Branch-and-Cut algorithm to the mid-term hydro scheduling problem, for which we propose a chance-constrained formulation. A computational study using data from ten hydroplants in Greece shows that the proposed methodology solves instances orders of magnitude faster than applying a general-purpose solver for convex mixed-integer nonlinear programming problems to the deterministic reformulation, and scales much better with the number of scenarios.

**Key words.** Mixed-integer nonlinear programming, Chance-constrained programming, Outer approximation, Hydro scheduling.

**AMS subject classifications.** 90C11, 90C15, 90C57, 90C90.

## 1. Introduction.

Mathematical programming is an invaluable tool for optimal decision-making that was initially developed in a deterministic setting. However, early studies on problems with probabilistic (i.e., nondeterministic) constraints have appeared since the late 50s, see, e.g., [Charnes et al., 1958, Prekopa, 1970]. In a problem with probabilistic constraints, the formulation involves a (vector-valued) random variable that parametrizes the feasible region of the problem; the decision maker specifies a probability $\alpha$, and the solution to the problem must optimize a given objective function subject to being inside the feasible region for a set of realizations of the random variable that occurs with probability at least $1 - \alpha$. The interpretation is that a solution that does not belong to the feasible region is undesirable, and we want this event to happen with a probability at most $\alpha$. This type of problem is called a *chance-constrained mathematical programming* problem in the literature [Charnes et al., 1958].

Without loss of generality, a chance-constrained mathematical program can be expressed as

$$\text{(CCP)} \qquad \max\{cx : \Pr(x \in C_x(w)) \geq 1 - \alpha, x \in X\},$$

where $w$ is a random variable, $C_x(w)$ is a set that depends on the realization of $w$ (the set of probabilistic constraints), and $X$ is a set that is described by deterministic

*Submitted to the editors September 21, 2016.

†Department of Mathematical and Industrial Engineering, Polytechnique Montréal, Montréal, Québec, Canada (andrea.lodi@polymtl.ca).

‡Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione, Università di Bologna, Bologna, Italy (enrico.malaguti@unibo.it).

§IBM T.J. Watson, Yorktown Heights, NY (nannicini@us.ibm.com).

¶Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione, Università di Bologna, Bologna, Italy (dimitri.thomopulos@unibo.it).

constraints [Prekopa, 1970]. We use the subscript $C_x$ to emphasize the fact that, given $w$, $C_x(w)$ is described in terms of the $x$ variables only; this notation will be useful in subsequent parts of the paper. A considerable simplification of the problem is that in which $C_x(w)$ is described by a set of constraints and $\Pr(x \in C_x(w))$ takes into account the violation of constraints one at a time, instead of considering the joint probability of $x \in C_x(w)$, which is more difficult. Chance-constrained mathematical programming problems find applications in many different contexts, see, e.g., [Tanner et al., 2008, Watanabe and Ellis, 1993]. The formulation (CCP) allows for two-stage problems with recourse actions, because the sets $C_x(w)$ can be the projection of higher-dimensional sets. This paper discusses the case where recourse actions are allowed and we are interested in the joint probability of $x \in C_x(w)$.

A generalization of (CCP) is that in which unsatisfied scenarios can enter a *recovery mode*: in this case, whenever $x \notin C_x(w)$, a cost that depends on the magnitude of the infeasibility has to be paid. The interpretation of such a model is that the normal mode of operation is when $x \in C_x(w)$, and we want this to happen with probability at least $1-\alpha$, but whenever we fall outside this situation we are interested in minimizing the cost associated with recovering a normal mode of operation. This problem has been studied in Liu et al. [2014], where a cost for the normal mode of operation is also considered. If we denote by $\varphi(x, w)$ the objective function contribution of satisfied scenarios, and by $\bar{\varphi}(x, w)$ the objective function contribution of unsatisfied scenarios, we obtain the following formulation:

$$\text{(CCPR-OBJ)} \qquad \begin{aligned} \max\{cx \;+\; &\Pr(x \in C_x(w))\mathbb{E}[\varphi(x,w)|x \in C_x(w)] + \\ &\Pr(x \notin C_x(w))\mathbb{E}[\bar{\varphi}(x,w)|x \notin C_x(w)] : \\ &\Pr(x \in C_x(w)) \geq 1-\alpha, x \in X\}. \end{aligned}$$

For example, in an energy scheduling problem such as the one discussed later in this paper, the recovery mode could represent the situation in which the production quotas set by the government are not met, or the user demand is not satisfied. In these cases, the system operator may have to meet the requirements buying energy from a third-party producer, which should only happen with low probability and would have an associated cost. The methodology discussed in this paper applies to Equation (CCPR-OBJ) and therefore to (CCP), which is a special case.

If uncertainty affects only the right-hand side values of the system of inequalities that defines the feasible region, under certain assumptions it is possible to derive a tractable reformulation of (CCP), e.g., [Charnes and Cooper, 1963, Lejeune, 2012]. A more general case is considered when the uncertainty can affect all parts of the system of inequalities describing $C_x(w)$. Under this more general setting, we need additional assumptions to deal with (CCPR-OBJ). In particular, assume that

(A1) the sample space, denoted as $\Omega$, is discrete and finite, and in particular $\Omega = \{w^i : i = 1, \dots, k\}$.

We should note that the assumption of discrete and finite sample space, while restrictive, includes a large number of practically relevant situations: typically, forecasts of future events cannot be too detailed and a general distribution can be truncated and discretized if necessary. Furthermore, even in the case that discretization and truncation cannot be applied, one can typically obtain good solutions and approximation bounds for a problem that requires general distributions via sample-average approximation [Luedtke and Ahmed, 2008]. From now on, we indeed assume $\Omega = \{w^i : i = 1, \dots, k\}$. The realizations $w^1, \dots, w^k$ are typically called *scenarios*. Let $p_i = \Pr(w = w^i)$. We can then introduce indicator variables $z_i$ for each set

$C_x(w^i)$, and write (CCPR-OBJ) in the following equivalent form:

$$\begin{array}{rrcl}
\max & cx + \Pr(x \in C_x(w))\mathbb{E}[\varphi(x,w)|x \in C_x(w)] & & \\
 & \Pr(x \notin C_x(w))\mathbb{E}[\bar\varphi(x,w)|x \notin C_x(w)] & & \\
\text{s.t.:} & x & \in & X \\
i = 1,\ldots,k & z_i = 0 \;\Leftrightarrow x & \in & C_x(w^i) \\
 & \sum_{i=1}^k p_i z_i & \leq & \alpha \\
i = 1,\ldots,k & z_i & \in & \{0,1\}.
\end{array}$$

To simplify this problem, we make an additional assumption:

(A2) $\varphi(x,w^i) \geq \bar\varphi(x,w^i)\ \forall x \in X, i = 1,\ldots,k$.

This implies that whenever $x \in C_x(w^i)$, the normal mode objective function contribution $\varphi(x,w^i)$ is to be preferred to the recovery mode contribution $\bar\varphi(x,w^i)$. The assumption is verified e.g., whenever $\varphi$ represents a nonegative revenue and $\bar\varphi$ represents a cost, i.e., a nonpositive value. Assumption (A2) ensures that we do not have to worry about two-stage consistency [Takriti and Ahmed, 2004]. Given (A2), we can replace $z_i = 0 \;\Leftrightarrow x \in C_x(w^i)$ with $z_i = 0 \;\Rightarrow x \in C_x(w^i)$. To bring the problem to a standard form that simplifies our exposition, let $n$ be the dimension of $x$ in (CCPR-OBJ); augment the vector $x$ with two continuous variables per scenario, say $x_{n+i}$ and $x_{n+k+i}$ for scenario $i = 1,\ldots,k$, and augment $c$ with two copies of the vector $(p_1,\ldots,p_k)$. Then, without loss of generality we can assume that $C_x(w^i)$ subsumes the constraints $x_{n+i} \leq \varphi((x_1,\ldots,x_n),w^i), x_{n+k+i} \leq 0$, and introduce the set $\bar C_x(w^i) := \{x \in \mathbb{R}^{n+2k} : x_{n+i} \leq 0, x_{n+k+i} \leq \bar\varphi((x_1,\ldots,x_n),w^i)\}$. It is then easy to see that (CCPR-OBJ) can be written as follows:

$$\text{(CCPR)}\qquad \begin{array}{rrcl}
\max & cx & & \\
\text{s.t.:} & x & \in & X \\
i = 1,\ldots,k & z_i = 0 \;\Rightarrow x & \in & C_x(w^i) \\
i = 1,\ldots,k & z_i = 1 \;\Rightarrow x & \in & \bar C_x(w^i) \\
 & \sum_{i=1}^k p_i z_i & \leq & \alpha \\
i = 1,\ldots,k & z_i & \in & \{0,1\},
\end{array}$$

where the vectors $x, c$ of (CCPR-OBJ) can be recovered as the first $n$ components of $x$ and $c$ above.

Our third and final assumption allows us to obtain a deterministic reformulation of (CCPR), using integer programming techniques. Precisely,

(A3) all the $C_x(w^i)$'s and $\bar C_x(w^i)$'s share the same recession cone.

The reformulation is accomplished by defining a problem with all the constraints of each of the $C_x(w^i)$ and $\bar C_x(w^i)$, and using a binary variable $z_i$ for each $w^i$ to activate/deactivate the corresponding constraints via big-M coefficients. Assumption (A3) is necessary because the recession cone of the deterministic equivalent formulation is the intersection of the recession cones of all $C_x(w^i), \bar C_x(w^i), i = 1,\ldots,k$, whereas the recession cone of (CCPR) is the union of the intersection of the recession cones of only some of these sets, which may not be the same unless all sets have the same recession cone (see [Jeroslow, 1987]).

Unsurprisingly, the size of the problems obtained with the indicator-variable reformulation is unmanageable in most practically relevant situations, and moreover, the relaxations of mathematical programs with this type of indicator variables tend to be very weak, leading to poor performance of solution methods (see, e.g., [Bonami et al., 2015]). However, under relatively mild assumptions it is possible to perform implicit solution of the reformulated problem. The idea is to keep the indicator variables, but avoid the classical on/off reformulation of the constraints that involve them. Then, a Branch-and-Cut algorithm [Padberg and Rinaldi, 1991] can be applied to the problem $\max_{x,z}\{cx : x \in X, z \in \{0,1\}^k, \sum_{i=1}^k p_i z_i \leq \alpha\}$. This problem is called a master problem. Whenever the solution of the master problem $\hat{x}$ does not satisfy the constraints of (CCPR), cuts are generated for the sets $C_x(w^i)$ and $\bar{C}_x(w^i)$, depending on the values of the indicator variables. The cuts are then added to the master problem. This basic idea yields an exact algorithm for (CCPR), and it has been successfully applied to different types of problem [Luedtke, 2014, Liu et al., 2014]. However, the literature mainly focuses on the case where all of the constraints are linear and all the original variables are continuous. While there are a few studies on linear problems with integer variables and certain classes of integer two-stage problems, e.g., [Song et al., 2014, Gade et al., 2014], they are limited to specific problem structures, thus, the methods proposed cannot be applied in general. The classical decomposition approach for two-stage nonlinear problems is generalized Benders decomposition [Geoffrion, 1972], but it has the drawback of requiring separability and/or knowledge of the problem structure to be practically viable; for these reasons, to the best of our knowledge it has not been embedded in an automated, general-purpose (i.e., problem-independent) decomposition scheme for this class of problems so far.

In this paper we consider the case where the sets $C_x(w^i), \bar{C}_x(w^i)$ are nonlinear convex, and propose a finitely convergent Branch-and-Cut algorithm. The cutting planes that we generate can be obtained as outer approximation cuts [Duran and Grossmann, 1986] and are therefore linear, as opposed to the generalized Benders cuts of Geoffrion [1972], which can be nonlinear in general. Our cut generation algorithm is much simpler than the generalized Benders procedure: it has fewer assumptions, in particular it does not require separability of the first and second stage variables or knowledge of the gradients, and it can be automated. The main application studied in this paper is the scheduling of a hydro valley in a mid-term horizon [Baslis and Bakirtzis, 2011, Carpentier et al., 2012, Kelman, 1998]. We propose a chance-constrained quantile optimization model for this problem that is equivalent to the minimization of the Value-at-Risk (see, e.g., [McNeil et al., 2015]), and perform a case study on the scheduling of a 10-plant hydro valley in Greece, using a mix of historical and realistically-generated data. In addition, we consider a problem formulation with step price functions that involves binary variables in the sets $C_x(w^i)$, and apply the Branch-and-Cut algorithm both to solve the continuous relaxation, and to generate primal bounds as a heuristic. Computational experiments show that our approach is able to solve large instances obtained from data of Baslis and Bakirtzis [2011] very effectively, with speedups that are often of several orders of magnitude. We remark that our formulation of the hydro scheduling problem is an instance of (CCP) rather than the more general (CCPR) because we do not take into account recovery costs. However, the algorithm that we propose allows taking recovery costs into account.

This paper has therefore the following contributions. First, we propose a Branch-and-Cut algorithm for the nonlinear convex (CCPR), which is a generalization of (CCP), and show that it finitely converges under mild assumptions. Despite its conceptual simplicity, our algorithm extends the approach of Liu et al. [2014] in two ways: assumption (A2) of Liu et al. [2014] and Luedtke [2014], imposing polyhedrality of the scenario problems, is replaced by the weaker assumption of nonlinear convex scenario problems, and assumption (B2) of Liu et al. [2014], imposing relatively complete recourse on the recovery scenario problems, is dropped. On the other hand, Liu et al. [2014] do not impose (A2) of the present paper, but for this reason they have to consider a threshold policy to determine when to operate the recovery mode, see [Liu et al., 2014, Sect. 2.2.2]; their threshold policy can in principle be applied to our formulation, but our assumption (A2) allows easier treatment and is often verified in practice, when constraint violations incur a heavy cost. Second, we show that the outer approximation cuts that we use are a linearization of generalized Benders cuts from a particular choice of dual variables, but they yield several advantages over generalized Benders cuts. Third, we provide an extensive computational evaluation on an important energy scheduling problem, showing the practical effectiveness of our approach and its scalability with respect to the number of scenarios.

This paper is organized as follows. Section 2 describes the decomposition approach with the associated Branch-and-Cut algorithm, discussing separating inequalities and their properties. Section 3 formalizes a mathematical model for the hydro scheduling problem. Section 4 contains a computational evaluation of several algorithms on instances of increasing difficulty derived from our case study, and discusses the numerical results. Finally, some conclusions are drawn in Section 5.

**2. Decomposition algorithm for (CCPR).** Under assumptions (A1)-(A3), we discussed in Section 1 how to obtain a deterministic equivalent formulation for (CCPR) using binary variables. We now introduce this mathematical model for the linear case, to explain the basic ideas and notation before transitioning to the nonlinear convex case, which is the focus of this paper.

In terms of notation, we denote by $x$ the decision variables of (CCPR), by $y^i$ the recourse variables for scenario $w^i$, by $\bar{y}^i$ the recovery variables for scenario $w^i$ (i.e., variables that are used to model the set $\bar{C}_x(w^i)$), and by $z$ binary variables with the property that $z_i = 0 \Rightarrow x \in C_x(w^i), z_i = 1 \Rightarrow x \in \bar{C}_x(w^i)$. Let $X = \{x : Ax \leq b\}$, $C_x(w^i) = \{x : \exists y^i \; A^i x + H^i y^i \leq b^i\}$, $\bar{C}_x(w^i) = \{x : \exists \bar{y}^i \; \bar{A}^i x + \bar{H}^i \bar{y}^i \leq \bar{b}^i\}$. Here and throughout the paper, integrality requirements on the set $X$ can be handled in a straightforward manner within the same framework at the cost of additional computational complexity, bur our discussion refers to the case where all variables are continuous. Then, (CCPR) can be formulated as follows:

$$
\begin{array}{llllllll}
\max & cx & & & & & & \\
\text{s.t.:} & Ax & & & & & \leq & b \\
 & A^1x & + & H^1y^1 & & & \leq & b^1 + M^1 z_1 \\
 & \bar{A}^1x & + & \bar{H}^1\bar{y}^1 & & & \leq & \bar{b}^1 + \bar{M}^1(1 - z_1) \\
 & \vdots & & & \ddots & & \vdots & \\
 & A^k x & & & + & H^k y^k & \leq & b^k + M^k z_k \\
 & \bar{A}^k x & & & + & \bar{H}^k \bar{y}^k & \leq & \bar{b}^k + \bar{M}^k(1 - z_k) \\
 & & & p_1 z_1 + \ldots & + & p_k z_k & \leq & \alpha \\
 & & & z_1 \quad \ldots & & z_k, & \in & \{0, 1\}.
\end{array} \tag{1}
$$

In this formulation, $M^i$ is a vector of large enough constants that is able to deactivate the set of constraints $A^i x + H^i y^i$ whenever $z_i = 1$, and $\bar{M}^i$ is a vector of large enough constants to deactivate $\bar{A}^i x + \bar{H}^i \bar{y}^i$ whenever $z_i = 0$; the existence of such vectors is guaranteed by assumption (A2). The joint chance constraint $\sum_{i=1}^k p_i z_i \leq \alpha$ ensures that the probability associated with unsatisfied scenarios is smaller than $\alpha$. The formulation (1) is a two-stage problem with recourse where there is no objective function contribution associated with the recourse variables, therefore the second-stage problems are feasibility problems. Our discussion in Section 1 shows how (CCPR-OBJ) can be brought to this form, enlarging the vector of first-stage variables $x$ if necessary. Problem (1) is a *mixed-integer linear programming* problem (MILP) that naturally leads to a Benders decomposition algorithm, and this is the approach followed, e.g., by Liu et al. [2014] for (CCPR), and Luedtke [2014] for (CCP).

This paper studies the case where the scenario subproblems are general convex set, described as $C_x(w^i) = \{x : \exists y^i \; g^i_j(x, y^i) \leq 0, j = 1, \ldots, m_i\}$, and $\bar{C}_x(w^i) = \{x : \exists \bar{y}^i \; \bar{g}^i_j(x, \bar{y}^i) \leq 0, j = 1, \ldots, \bar{m}_i\}$. For all $i$, we write the vector functions $g^i(x, y^i) = (g^i_1(x, y^i), \ldots, g^i_{m_i}(x, y^i))^T$, $\bar{g}^i(x, y^i) = (\bar{g}^i_1(x, \bar{y}^i), \ldots, \bar{g}^i_{\bar{m}_i}(x, \bar{y}^i))^T$. For ease of notation we keep the assumption that $X = \{x : Ax \leq b\}$, but this does not affect our development and the generalization to the case where $X$ is a general convex set, possibly with integrality requirements, is straightforward. If all the $C_x(w^i)$ have the same recession cone, we can write a MINLP model for (CCP) as follows:

$$
\begin{array}{lllllll}
\max & cx & & & & & \\
\text{s.t.:} & Ax & & & & \leq & b \\
& g^1(x, & y^1) & & & \leq & M^1 z_1 \\
& \bar{g}^1(x, & \bar{y}^1) & & & \leq & \bar{M}^1(1 - z_1) \\
& \vdots & & \ddots & & \vdots & \\
& g^k(x, & & & y^k) & \leq & M^k z_k \\
& \bar{g}^k(x, & & & \bar{y}^k) & \leq & \bar{M}^k(1 - z_k) \\
& & p_1 z_1 & + \;\ldots\; + & p_k z_k & \leq & \alpha \\
& & z_1 & \ldots & z_k, & \in & \{0, 1\}.
\end{array}
\tag{2}
$$

Assuming the functions $g^i_j, \bar{g}^i_j$ are convex, (2) is a convex MINLP in the sense that it has a convex continuous relaxation.

**2.1. Overview of the approach.** Solving directly the MINLP model (2) can be impractical, therefore we follow a decomposition approach whereby we define a *master problem* with the constraints defining $x \in X$, and $2k$ scenario subproblems, one for each normal mode scenario and one for each recovery mode scenario, involving scenario-dependent constraints. Let $C_{x,y}(w^i)$ be the feasible region of a normal mode scenario, and define $C_x(w^i) = \text{Proj}_x C_{x,y}(w^i)$. So, $\hat{x}$ is feasible for scenario $i$ if $\hat{x} \in C_x(w^i)$. Similarly, we denote by $\bar{C}_{x,y}(w^i)$ the feasible region of a recovery mode scenario, and define $\bar{C}_x(w^i) = \text{Proj}_x \bar{C}_{x,y}(w^i)$. Since $C_x(w^i), \bar{C}_x(w^i)$ have the same structure, from now on our discussion focuses on the sets $C_x(w^i)$'s, but clearly it also applies to the $\bar{C}_x(w^i)$'s.

The basic idea we exploit is to generate solutions for the master, and if they are not feasible for enough scenarios to satisfy the joint chance constraint, we cut them off. This is essentially a Benders decomposition approach applied to (2). In the linear case (1), the solution to the master problem can be cut off by means of textbook Benders cuts. In the nonlinear case (2), we can use generalized Benders cuts. This paper advocates a particular choice of outer approximation cuts, that are linearizations of Benders cuts and present several advantages: this will be the subject of Section 2.2; the relationship with generalized Benders decomposition [Geoffrion, 1972] is discussed in Section 2.4.

Instead of applying a pure Benders decomposition approach to (2), we use a Branch-and-Cut approach adapted from Luedtke [2014], where the linear case is considered and therefore applies to (1) rather than (2). However, the steps of the algorithm remain the same, as this is essentially implicit Benders decomposition: we do not solve the master problem to (integral) optimality, but apply Branch and Cut and separate Benders cuts at every node with an integral solution. The algorithm uses a separation routine for the scenario subproblems, combined with the variables $z$. A basic version of the algorithm is given by Algorithm 1.

**Algorithm 1** `Decomposition Algorithm`

1: Define a master problem of the form

$$\left.\begin{array}{rrcl} \max & cx & & \\ \text{s.t.:} & Ax & \leq & b \\ & \sum_{i=1}^{k} p_i z_i & \leq & \alpha \\ & z & \in & \{0,1\}^k \end{array}\right\} \tag{3}$$

2: **repeat**
3: Perform Branch and Bound on (3)
4: At every node of the tree with solution $(\hat{x}, \hat{z}), \hat{z} \in \{0,1\}^k$, do the following:
5: **for** $i = 1, \ldots, k$ **do**
6: **for** $\hat{z}_i = 0$ and $\hat{x} \notin C_x(w^i)$ **do**
7: separate $\hat{x}$ from $C_x(w^i)$ via an inequality $\gamma x \leq \beta_i$
8: add inequality $\gamma x \leq \beta_i + M z_i$ to the master problem (3)
9: **end for**
10: **for** $\hat{z}_i = 1$ and $\hat{x} \notin \bar{C}_x(w^i)$ **do**
11: separate $\hat{x}$ from $\bar{C}_x(w^i)$ via an inequality $\gamma x \leq \beta_i$
12: add inequality $\gamma x \leq \beta_i + M(1 - z_i)$ to the master problem (3)
13: **end for**
14: **end for**
15: If $(\hat{x}, \hat{z})$ is still feasible, update incumbent (lower bound).
16: **until** no more nodes to be explored

It is not difficult to see that this algorithm can be applied even if the sets $C_x(w^i)$ are nonlinear provided that we have access to a separation routine, although termination is in general not guaranteed. We remark that we could employ a *nonlinear* separating inequality rather than a hyperplane in step 7 of Algorithm 1, as is done in generalized Benders decomposition [Geoffrion, 1972]. However, linear inequalities have several computational advantages, and allow for an easy lifting procedure of the coefficients on the $z$ variables following Luedtke [2014]. We will revisit this topic in Section 2.4 from a theoretical point of view, whereas a discussion of lifting on the $z$ variables is given in Section 4.1; notice that lifting does not affect the general scheme of the algorithm.

Algorithm 1 has some similarities with the LP/NLP-BB approach of Abhishek et al. [2010] and the Hybrid approach of Bonami et al. [2008], in the sense that all these methodologies involve a Branch-and-Cut algorithm where additional outer approximation inequalities are computed at nodes of the tree with integer solution. However, a fundamental difference exists: the algorithms of Bonami et al. [2008], Abhishek et al. [2010] as applied to (2) would work with a relaxation of the feasible region that includes all the decision variables, using NLP subproblems to construct outer approximation cuts fixing the integer variables. In the case of Algorithm 1, the master contains a subset of decision variables and is not aware of the recourse variables $y^i$ or the recovery variables $\bar{y}^i$. Therefore, we work on a projection of the feasible region of (2), and some integer and continuous variables ($z$ and $x$) are fixed to obtain outer approximation cuts. It can be easily seen that the sequence of points

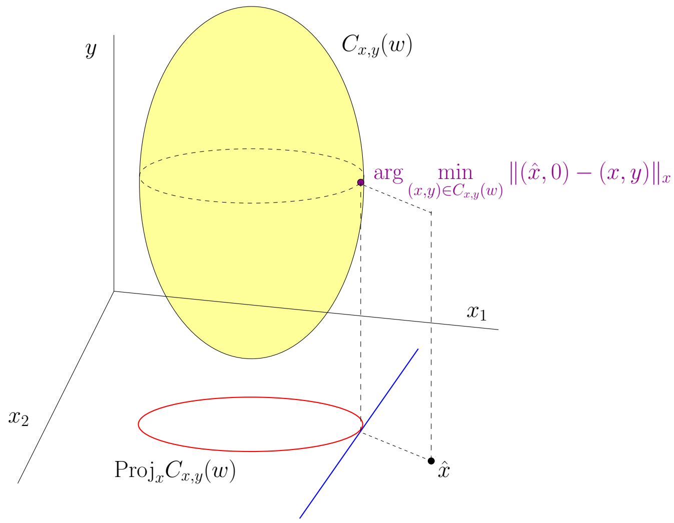


FIG. 1. *Separating hyperplane.*

generated by the algorithm is not necessarily the same.

**2.2. Separation algorithm.** In this section we provide a separation algorithm for step 7 of Algorithm 1 in the setting of this paper, i.e., convex scenario problems. The same development applies to step 11. For ease of notation, we drop the dependence on $w$ and refer to $C_{x,y}, C_x$ as the subproblems associated with a particular realization of $w$, i.e., a scenario. Therefore, for a given scenario $i$, we can write

$$C_{x,y} = \{(x,y) : g_j(x,y) \leq 0, j = 1,\dots,d\} \tag{4}$$

where $g_j(x,y)$ is convex for all $j$. (Note that for scenario $i$, system (4) would have been $C_{x,y}(w^i) = \{(x,y^i) : g^i_j(x,y^i) \leq 0, j = 1,\dots,m_i\}$, i.e., $d = m_i$.) Given a solution for the master problem $\hat{x}$, we need to answer the question: does there exist $\hat{y}$ such that $(\hat{x},\, \hat{y}) \in C_{x,y}$? If such $\hat{y}$ does not exist, we must find a separating hyperplane: this is the purpose of the separation routine.

Notice that the master problem involves the $x$ variables only. For this reason, the separation routine must find a cut in the $x$ space. One approach to do so is given by generalized Benders decomposition [Geoffrion, 1972]. Here we advocate a simpler approach that allows computation of a separating hyperplane under mild conditions; we discuss its relationship with generalized Benders decomposition in Section 2.4.

Define the problem

$$\min_{(x,y)\in C_{x,y}} \frac{1}{2}\|x-\hat{x}\|^2_x, \tag{PROJ}$$

where by $\|\cdot\|_x$ we denote the Euclidean distance in the $x$ space only. If $\hat{x} \notin C_x$, the optimal value of (PROJ) must be strictly greater than 0.

THEOREM 1. *Let $C_{x,y}$ be a closed set such that $C_x = \mathrm{Proj}_x C_{x,y}$ is convex, and $\hat{x} \notin C_x$. Let $(\bar{x},\, \bar{y})$ be the optimal solution to (PROJ), $\ell^* > 0$ the optimal objective function value. Then, the hyperplane*

$$(\hat{x} - \bar{x})^T (x - \bar{x}) \leq 0$$

*separates $\hat{x}$ from $C_x$. This hyperplane is the deepest valid cut that separates $\hat{x}$ from $C_x$, if depth is computed in $\ell_2$-norm.*

*Proof.* Proof. Because $C_{x,y}$ is closed, $C_x$ is closed, and convex by assumption. Therefore, there exists a unique vector $v$ that minimizes $\|v - \hat{x}\|$ over all $v \in C_x$. By definition of (PROJ), $v = \bar{x}$. Then, we can apply the projection theorem (see, e.g., [Bertsekas, 1999, Prop. B.11 (b)]) to obtain

$$(\hat{x} - \bar{x})^T(x - \bar{x}) \leq 0 \quad \forall x \in C_x.$$

Hence, this hyperplane is valid for $C_x$, and it separates $\hat{x}$ because $\|\hat{x} - \bar{x}\|^2 = \ell^* > 0$ by hypothesis. To show that it is the deepest valid cut, notice that $\text{dist}_x(\hat{x}, \bar{x}) = 2\ell^*$. Any cut that cuts $\hat{x}$ by more than $2\ell^*$ in Euclidean distance computed in the $x$ space would cut $\bar{x}$ off, forsaking validity. □

A sketch of the main elements of Theorem 1 can be found in Fig. 1. It is evident that the inequalities described in Theorem 1 are outer approximation cuts. Outer approximation was introduced by Duran and Grossmann [1986] and has proven to be an extremely useful tool in mixed-integer convex programming [Bonami et al., 2008, 2009, Fletcher and Leyffer, 1994]. Outer approximation is used to separate a point not belonging to a convex set from the convex set itself, and typically the point and the set live in the same space. In this paper, we apply outer approximation to separate a point from the *projection* of a set on a lower-dimensional space, and we do not have an explicit description of such projection: for this reason, to obtain the separating inequality we perform an optimization in the higher-dimensional space, and the result is the outer approximation cut that would have been obtained if we had the explicit description of the projection.

The only assumption in Theorem 1 is that $C_{x,y}$ projects to a closed convex set: we do not even require constraint qualification (see Prop. 3 for a more precise characterization of the separating inequality when constraint qualification holds). However, to find the hyperplane we must be able to solve (PROJ), which is an optimization problem over $C_{x,y}$: the difficulty of separation depends on the difficulty of optimizing over $C_{x,y}$. In particular, since we assume that $C_{x,y}$ is described as a set of (continuous) nonlinear convex constraints, the separation can be carried out in polynomial time.

**2.3. Termination of the Branch-and-Cut algorithm.** We now show that Algorithm 1, combined with the separation routine that generates the cut $(\hat{x}-\bar{x})^T(x-\bar{x}) \leq 0$ as in Theorem 1, terminates under mild assumptions.

Theorem [Kelley, 1960, Sec. 2] considers a continuous convex function $G(x)$ defined on a compact convex set $X$ such that, at every point $\hat{x} \in X$, there exists an extreme support $y = p(x, \hat{x})$ to the graph of $G(x)$ whose gradient is bounded by a constant. Given a cost vector $c$, if $\hat{x}_h$ defines a sequence of points such that $c\hat{x}_h = \min\{cx | x \in X_h\}$, $h = 0, 1, \dots,$ where $X_0 = X$ and $X_h = X_{h-1} \cup \{x | p(x, \hat{x}_{h-1}) \leq 0\}$, then the sequence $\{\hat{x}_h\}$ contains a subsequence that converges to a point $\xi$ in $X$ with $G(\xi) \leq 0$.

We are ready to prove the following theorem.

THEOREM 2. *Consider a problem of the form*

$$\text{(CCP)} \qquad \max\{cx : \Pr(x \in C_x(w)) \geq 1 - \alpha, x \in X\},$$

*where $X$ is compact, $C_x(w)$ is a closed and convex set for all $w = w^1, \dots, w^k$, and assumptions (A1)-(A3) are satisfied. Assume further that we have an algorithm to* *solve (PROJ) to optimality. Then, given any $\varepsilon_c > 0$, if an optimal solution $x^*$ to (CCP) exists, Algorithm 1 finds a solution $\tilde{x}$ with $\|x^* - \tilde{x}\| \leq \varepsilon_c$ after a finite number of iterations.*

*Proof.* Proof. Because the number of binary variables $z_i$ is $k < \infty$, a Branch-and-Bound algorithm on $z_i$ trivially processes a finite number of values for the $z$ variables. It is obvious that the LP relaxation of the master problem (3) is a relaxation of (CCPR), because all inequalities added to the problem are valid. Therefore, we must show that lower bounds are computed correctly, i.e., whenever a feasible solution is found by the algorithm, it is $\varepsilon_c$-feasible for (CCPR). Notice that a solution $(\hat{x}, \hat{z})$ is considered feasible by Algorithm 1 only if $\hat{z}$ is binary, and $\hat{x}$ is not cutoff after looping through all the scenario subproblems. Hence, we must show that after a finite number of separation rounds for a given value of $\hat{z}$, the solution to the master problem $\hat{x}$ belongs to $C_x(w^i)$ (or is $\varepsilon_c$-close) for all $i$ such that $\hat{z}_i = 0$, and belongs to $\bar{C}_x(w^i)$ for all $i$ such that $\hat{z}_i = 1$. This ensures that for every value of $\hat{z} \in \{0,1\}^k$, a finite number of iterations is necessary, therefore the overall algorithm terminates. The latter statement is true in the setting of Luedtke [2014] because $C_{x,y}(w^i)$ is a polyhedron, and the paper considers only inequalities corresponding to extreme points of the Benders cut generating problem, which are in finite number. In the context of the present paper, it must be proven.

For this, it is sufficient to show that for every $C_x(w^i)$ and every $\bar{C}_x(w^i)$ satisfying the assumptions, the separation routine of Theorem 1 requires a finite number of inequalities for $\varepsilon_c$-convergence. This ensures finite $\varepsilon_c$-convergence to the intersection of $C_x(w^i)$ for all $i$ such that $\hat{z}_i = 0$ and all $\bar{C}_x(w^i)$ for all $i$ such that $\hat{z} = 1$. For ease of notation, we drop $w^i$ and discuss a generic set $C_{x,y}$ with projection $C_x$, as the argument is the same for all $C_x(w^i)$ and $\bar{C}_x(w^i)$.

We now apply the convergence result of Theorem [Kelley, 1960, Sec. 2] as follows. Let $X$ be the set defined by feasible region of the master problem, and define $G(x) = \min_{\tilde{x} \in C_x} \|x - \tilde{x}\|$, i.e., as the distance function from the convex set $C_x$. Therefore, $G(x)$ is convex. By convexity, an extreme support of $G(x)$ exists at each point of $X$, and, by the definition of $G(x)$, its gradient is bounded. We have $G(x) = 0 \Leftrightarrow x \in C_x$, $G(x) > 0 \Leftrightarrow x \notin C_x$. Given $\hat{x} \in X$, $\hat{x} \notin C_x$, define $\bar{x} = \arg\min_{\tilde{x} \in C_x} \|\hat{x} - \tilde{x}\|$, so that $G(\hat{x}) = \|\hat{x} - \bar{x}\|$. An extreme support $y = p(x, \hat{x})$ to $G(x)$ at $\hat{x}$ is

$$y = G(\hat{x}) + \nabla^T G(\hat{x})(x - \hat{x}) = \|\hat{x} - \bar{x}\| + \frac{(\hat{x} - \bar{x})}{\|\hat{x} - \bar{x}\|}(x - \hat{x}).$$

Then, since $\hat{x} = \bar{x} + \hat{x} - \bar{x}$, the expression $p(x, \hat{x}) \leq 0$ reads as

$$(\hat{x} - \bar{x})(x - \bar{x}) \leq 0,$$

which is exactly the condition we use to (iteratively) separate $\hat{x}$. By Theorem [Kelley, 1960, Sec. 2], we can define a sequence of $\hat{x}_h$ converging to a point $\xi$ in $X$, $G(\xi) \leq 0$, i.e., $\xi \in C_x$. By definition of convergence, for every $\varepsilon_c$, there exists an integer $v$ such that after $v$ inequalities, $\|\hat{x} - \xi\| \leq \varepsilon_c$. This concludes the proof. ∎

**2.4. Comparison with generalized Benders cuts.** This section investigates the relationship between the separation approach we advocate and generalized Benders decomposition [Geoffrion, 1972], which applies to the same class of problems studied in this paper, namely those that can be formulated as (2). Here we only discuss the case where the second-stage problems are feasibility problems, following our formulation in Section 1. The result in Geoffrion [1972] assumes that a "dual adequate" algorithm to solve the scenario subproblems is available, that is, if the problem is infeasible a dual certificate of infeasibility can be computed. In its computational considerations it remarks that "it appears necessary" to assume additional properties on the structure of the problem, namely, that the function

$$L(x,\lambda) = \min_{y \in C_{x,y}} \lambda^T g(x,y)$$

can be easily computed for all $x \in X, \lambda \in \mathbb{R}^m, \lambda \geq 0$. In particular this means that we should be able to find an analytical expression for such function. This can be done in some specific situations, for example if the nonlinear functions are separable in $x$ and $y$ (see, e.g., [Bloom, 1983, França and Luna, 1982]), but may be difficult in general if the solution to the minimization problem over $y$ depends on $x$. Even when that is the case, one issue remains: in the approach of Geoffrion [1972] these functions are the Benders cut added to the master problem, and they have the form of the constraints $g(x,y)$. If the $g(x,y)$ are nonlinear, we are left in the unfortunate situation of possibly adding nonlinear constraints to the master problem. The nonlinear cuts could be stronger than linear inequalities, but are computationally less attractive and would not allow us to use the existing well-developed machinery for linear inequalities, such as mixing techniques [Günlük and Pochet, 2001]. Of course, one could simply linearize a generalized Benders cut: we show that this is in fact exactly what is happening.

PROPOSITION 3. *Assume that constraint qualification conditions are met, and let $(\bar{x},\bar{y})$ be the optimal solution to (PROJ), $\mu$ be the corresponding KKT multipliers. Then, the cut*

$$(\hat{x}-\bar{x})^T(x-\bar{x}) \leq 0$$

*is the linearization of a generalized Benders cut obtained from $\hat{x}$ with multipliers $\mu$.*

*Proof.* Proof. A generalized Benders cut has the form $L(x,\lambda) \leq 0$, where $L(x,\lambda) = \min_{y \in C_{x,y}} \lambda^T g(x,y)$ and $\lambda$ is a nonegative vector such that $\min_{y \in C_{x,y}} \lambda^T g(\hat{x},y) > 0$; see [Geoffrion, 1972]. By construction, the hyperplane $\left(\sum_{j\in I} \mu_j \nabla g_j(\bar{x},\bar{y})\right)((x,y) - (\bar{x},\bar{y})) = (\hat{x}-\bar{x})^T(x-\bar{x}) \leq 0$ is supporting for $\sum_{j\in I}\mu_j g_j(x,y)$ at $(\bar{x},\bar{y})$, so $\sum_{j\in I}\mu_j g_j(x,y) \geq (\hat{x}-\bar{x})^T(x-\bar{x})$ for all $(x,y) \in C_{x,y}$ because the left-hand side expression is convex. It follows that

$$\min_y \sum_{j\in I} \mu_j g_j(\hat{x},y) \geq (\hat{x}-\bar{x})^T(\hat{x}-\bar{x}) = \ell^* > 0.$$

This shows that the multipliers $\mu$ yield a violated generalized Benders cut. Furthermore, $\sum_{j\in I}\mu_j g_j(\bar{x},y) \geq (\hat{x}-\bar{x})^T(\bar{x}-\bar{x}) = 0$ for all $y$, and $\sum_{j\in I}\mu_j g_j(\bar{x},\bar{y}) = 0$ by complementary slackness, hence $\bar{y} = \arg\min_y \sum_{j\in I}\mu_j g_j(\bar{x},y)$. It follows that $(\hat{x}-\bar{x})^T(x-\bar{x})$ is the tangent plane to $L(x,\mu)$ at the point $\bar{x}$. ∎

The fact that outer approximation cuts are linearizations of generalized Benders cuts is well known: since every nonnegative combination of the constraints $g_j$ can be considered a generalized Benders cuts, every valid linear inequality for $C_x$ is a linearization of a generalized Benders cuts. [Abhishek et al., 2010, Sect. 3.1] remarks that aggregating linearizations to the constraints using optimal dual multipliers simplifies the cut, and the unfixed variables disappear from the cut expression.

It is important to remark that our way of generating cuts is conceptually simpler than applying generalized Benders decomposition, and it has some clear advantages. In fact, let $\lambda \geq 0$ be any vector of dual variables that gives rise to a violated generalized Benders cut, i.e., $\min_{y \in C_{x,y}} \lambda^T g(\hat{x}, y) > 0$. Since the expression $\min_{y \in C_{x,y}} \lambda^T g(\hat{x}, y) \leq 0$ is convex, any tangent hyperplane is a valid inequality. The approach of Geoffrion [1972] requires the dual variables $\lambda$ only, but in order to compute a tangent hyperplane, we additionally need a point about which the linearization is obtained. To this end, Abhishek et al. [2010] propose a hierarchy of points, where the weakest one is analogous to the ECP method [Westerlund et al., 1998] and does not require solving a subproblem, while the strongest one obtains the point by solving the NLP relaxation of the current node. Notice that in our context, because no value for $y$ is initially known, it seems that solving an NLP subproblem to generate the point is a better approach. Furthermore, if the point about which the linearization is generated does not belong to $C_{x,y}$ the tangent hyperplane may not be supporting for $C_x$, hence it would be dominated by some other valid inequality.

In principle, our projection approach to generate a separating inequality can also be applied in the case where $C_{x,y}$ is a polyhedron, and it yields violated Benders cuts from a particular choice of dual variables. The most commonly approach used in the literature is instead to obtain the dual variables by minimizing the largest constraint violation, which corresponds to a specific truncation of the unbounded dual rays (see Fischetti et al. [2010]). The standard approach guarantees that all the inequalities are generated from extreme points of the dual polyhedron, whereas our projection approach may construct a Benders cut from dual variables that are not extreme, in which case the cut would not be extreme either, i.e., it could be obtained as a combination of extreme Benders cuts.

**3. (CCP) for mid-term hydro scheduling.** We apply the decomposition algorithm for nonlinear chance-constrained problem of Section 2 to the hydro scheduling problem that we describe next. Our formulation is an instance of (CCP), which is a special case of (CCPR).

A central problem in power generation systems is that of optimally planning resource utilization in the mid and long term and in the presence of uncertainty. Hydro power production networks usually consist of several reservoir systems, often interconnected, which are operated on a yearly basis: it is common to have seasonal cycles for demand and inflows, which can be out of phase by a few months, i.e., inflow peaks typically precede demand peaks.

The mid-term hydro scheduling problem refers to the problem of planning production over a period of several months. To be effective, such planning must take into account uncertainty affecting rainfall, energy price and demand, as well as the complex and nonlinear power production functions. A commonly used approach in practice is to rely on deterministic optimization tools and on the experience of domain experts to deal with the uncertainty, because of the sheer difficulty of incorporating uncertainty into the model. Many deterministic approaches can be found in the literature, e.g., [Carneiro et al., 1990]. More recently, methodologies that can take into account the uncertainty in the model have appeared, such as [Baslis and Bakirtzis, 2011, Carpentier et al., 2012, Kelman, 1998]. We are not aware of previous work that employs a chance-constrained formulation for the mid-term hydro scheduling problem, although there has been work on the related unit commitment problem, e.g., [van Ackooij, 2014, Wang et al., 2012]. Even in the case of unit commitment, chance-constrained optimization approaches are the least commonly used in the literature, due to their difficulty [Tahanan et al., 2015, Sect. 4.4].

The problem studied in this paper can be described as follows: there are $n$ hydroplants, each one associated with a reservoir. The water in each reservoir can be used to obtain energy through the power plant. Our goal is to define a mid-term production plan, that is, how much water to release in each period from each reservoir, over a time horizon of several months, in order to maximize a profit function. The profit depends on the amount of energy obtained and on the market price, assuming that the amount of energy sold influences the final price. In each time period, the total quantity of water in the reservoirs must satisfy some lower and upper bounds. All the water that is not released in period $t$ is available at $t+1$, in addition to the natural water inflow from rivers, precipitations and seasonal snow melting. The definition of a production plan faces two sources of uncertainty, namely: the natural water inflow, and the energy price on the market.

**3.1. Choice of the objective function.** When the problem takes into account a long time span, the decision maker is typically interested in the optimal present-time (i.e., first stage) decisions: future decisions can be adjusted depending on the evolution of the market and the context. Consequently, we consider a problem formulation with recourse, where in our case, the recourse actions are simply all the decision taken at time periods $t > 1$.

It is important to remark that the profit for the generating company is a function of the first-stage decisions and the scenario, i.e., the realization of $w$. Thus, in order to formulate the objective function of the problem, we must decide what measure of profit we are interested in. Widely used choices when optimizing an uncertain profit are the expected profit and the worst-case profit. Our approach draws from the financial risk management literature: we use a measure of profit related to the well-known Value-at-Risk [McNeil et al., 2015], which allows the decision maker to determine the trade-off between risk and returns. In particular, given $0 \leq \alpha < 1$, our objective function is the maximization of the $\alpha$-quantile of the profit. We now show how this relates to Value-at-Risk.

Let $\varphi(x, w^i)$ be the profit that can be obtained in scenario $w^i$ with first-stage decision variables $x$; notice that given $x$ and $w^i$, the value of $\varphi(x, w^i)$ can be computed by solving a deterministic optimization problem. Define the random variable $\varphi_x : \Omega \to \mathbb{R}$, $\varphi_x(w) = \varphi(x, w)$. Since $\varphi_x$ is a random variable that measures the profit, we define the loss as $L_x = -\varphi_x$. The $\alpha$-Value-at-Risk is defined as

$$\mathrm{VaR}_\alpha(L_x) = \inf\{\ell \in \mathbb{R} : \Pr(L_x > \ell) \leq 1 - \alpha\}.$$

It is easy to show via algebraic manipulations that

$$\min_x \mathrm{VaR}_{1-\alpha}(L_x) = \max_x \sup\{q \in \mathbb{R} : \Pr(\varphi_x \geq q) \geq 1 - \alpha\} = \max_x Q_\alpha(\varphi_x),$$

where $Q_\alpha$ is the $\alpha$-quantile. In other words, our choice of objective function, i.e., maximizing the $\alpha$-quantile of the profit, is equivalent to minimizing the $(1-\alpha)$-Value-at-Risk of the loss. We remark that our decomposition scheme can also be applied to the case in which the objective function contains a penalization for not satisfying some of the scenario constraints (e.g., not meeting a production quota), but we did not pursue further study of this type of objective function.

**3.2. Optimization model.** We consider a multi-period planning problem with $T$ periods (indexed by $t = 1, \ldots, T$), where all information regarding period 1 is deterministically known, while the remaining periods are subject to uncertainty. We consider uncertainty with respect to inflows and energy market prices, and we model the uncertainty by defining a finite number of inflow and energy market scenarios, each one with an associated probability of realization. Our objective in a deterministic setting would be to maximize the profit obtained by selling energy on the energy market. Electrical energy is obtained by transforming the potential energy of the water when, during each period, the water is released from the reservoirs. There are $n$ reservoirs in total, indexed by $h = 1, \ldots, n$. We denote by $x_{th}$ the amount of water released in period $t$ from reservoir $h$, and by $w_{th}$ the water level of reservoir $h$ at the end of the period ($w_{0h}$ is a parameter denoting the initial water level). Parameter $f_{th}$ denotes the natural water inflow in period $t$ at reservoir $h$. The water released from reservoir $h$ is transformed into an amount of energy that depends on a nonlinear function $g_h(w, x)$. Energy obtained this way, denoted as $e_{th}$ for period $t$ and reservoir $h$, is sold on the market; since hydro power production has in general a large capacity, we assume to influence the market price, according to a price function $\pi_t(\cdot)$ that depends on the total amount of electrical energy we sell at period $t$, namely, $e_t = \sum_{h=1}^{n} e_{th}$. In the deterministic setting, the hydro scheduling problem described above is modeled by the following nonlinear programming problem:

$$\max \sum_{t=1}^{T} \pi_t(e_t) e_t \tag{5a}$$

$$\text{s.t.:} w_{(t-1)h} - x_{th} + f_{th} \geq w_{th} \qquad t = 1, \ldots, T, h = 1, \ldots, n \tag{5b}$$

$$0 \leq x_{th} \leq u_{th} \qquad t = 1, \ldots, T, h = 1, \ldots, n \tag{5c}$$

$$q_{th} \leq w_{th} \leq Q_{th} \qquad t = 1, \ldots, T, h = 1, \ldots, n \tag{5d}$$

$$e_{th} \leq g_h(w_{th}, x_{th}) \qquad t = 1, \ldots, T, h = 1, \ldots, n \tag{5e}$$

$$d_t \leq e_t \leq m_t \qquad t = 1, \ldots, T \tag{5f}$$

$$e_t = \sum_{h=1}^{n} e_{th} \qquad t = 1, \ldots, T. \tag{5g}$$

The objective function (5a) maximizes the profit obtained by selling the transformed energy. Constraint (5b) is an inventory constraint that defines the water balance between consecutive periods: since water can be released without obtaining energy (spillage), we have an inequality. Constraints (5c) and (5d) impose lower and upper bounds on the quantity of water used for transforming energy and on the water levels in the reservoirs, respectively. Constraints (5e) define the relation between the released water and the obtained electrical energy at a specific plant $h$. Finally, (5f) defines lower and upper bounds on the amount of obtained electrical energy. Notice that the above problem is convex assuming that $g_h$ is concave.

To model uncertainty, Baslis and Bakirtzis [2011] assume that forecasts for aggregated demand and precipitations are available as discrete random variables. The optimization occurs over a relatively long period of time (i.e., twelve months), therefore it would be unrealistic to assume temporal independence of demand and precipitations, and the assumption in [Baslis and Bakirtzis, 2011] is that the realization of the random variables at any time period depends on the realization in the previous time period. We follow the approach of Baslis and Bakirtzis [2011]. This yields a scenario tree, where a scenario is a realization of the random parameters over the entire time period, i.e., a sample path. A scenario tree starts from the root node at the first period and, for each possible realization of the random parameters, branches into a node at the next period. The branching continues up to the leaves of the tree, whose number corresponds to the number of scenarios $k$.

**3.3. Decomposition.** We decompose the problem into a master problem and $k$ scenario subproblems. Each scenario subproblem $i$ includes decision variables $x^i_{ht}$, and has a feasible region defined by (5b) – (5f). In addition, we link the profit in each scenario to an overall measure of profit in the master problem by introducing a master variable $\psi$ that is maximized, and defining the following additional constraints:

$$\psi \leq \sum_{t=1}^{T} \pi^i_t(e_t)e_t \quad i = 1, \ldots, k. \tag{6}$$

Hence, a specific scenario is satisfied given the decision variables in the master (energy obtained in the first time period, and measure of profit $\psi$) if not only constraints (5b) – (5f) can be satisfied for subsequent time periods, but also the total profit for the scenario is not smaller than $\psi$. Since the master maximizes the profit that can be obtained by satisfying a subset of scenarios having associated probability not smaller than $1 - \alpha$, this is equivalent to optimizing the $\alpha$-quantile of the profit.

Following Baslis and Bakirtzis [2011], we assume that all scenarios have an associated probability of $1/k$ (modifying the formulation to allow for nonuniform scenario probabilities is straightforward), and the joint chance constraints are equivalent to imposing that at least $k - p$ scenarios are satisfied, where $p = \lfloor \alpha k \rfloor$. Nonanticipativity constraints are enforced by the master, guaranteeing that for all $t$, decisions up to period $t$ are the same for all sample paths that are identical up to $t$. Given two scenario indices $i$ and $r$, define $\tau(i, r)$ as the largest time period index such that the sample path realizations of scenarios $i$ and $r$ are identical up to it. We can then write the initial master problem (before addition of outer approximation cuts) as the following MILP:

$$\max \psi \tag{7a}$$

$$\text{s.t.:} \sum_{i=1,\ldots,k} z_i \leq p, \tag{7b}$$

$$x^i_{th} = x^r_{th}, i = 1, \ldots, k-1,\ r = i+1, \ldots, k,\ t \leq \tau(i, r),\ h = 1, \ldots, n \tag{7c}$$

$$0 \leq x^i_{th} \leq u_{th} t = 1, \ldots, T-1,\ i = 1, \ldots, k,\ h = 1, \ldots, n \tag{7d}$$

$$z_i \in \{0, 1\}, i = 1, \ldots, k \tag{7e}$$

where (7b) is the joint probability constraint, constraints (7c) express nonanticipativity, constraints (7d) impose bounds on the quantity of water released. We remark that in practice we do not explicitly write constraints (7c), because we keep only one copy of the $x$ variables for all sample paths identical up to a given period, implicitly performing the substitution. This is conceptually equivalent and reduces the size of the problem.

**3.3.1. Electricity generation function.** The transformation of the water potential energy into electrical energy is described in terms of a nonlinear power function $v_h(w, \dot{x})$ that depends on the water flow and water level $w$ at reservoir $h$. We assume that the water flow and level are constant within each time period, and that the amount of electrical energy obtained during a given period is directly proportional to the length of the period $\theta_t$. Hence, we can write

$$g_h(w_{th}, x_{th}) = v_h(w_{th}, x_{th}/\theta_t)\theta_t. \tag{8}$$

Several alternatives are proposed in the literature regarding the shape of $v_h(w, \dot{x})$, see e.g., [Bacaud et al., 2001, Salam et al., 1998, Chang and Chen, 1998]. These alternatives depend on the characteristics of each power plant and typically must be experimentally evaluated.

The most common power functions consider power as a quadratic expression of the flow, as $v_h = \rho(x/\theta_t)^2 + \nu x/\theta_t + \sigma$, where the values of the coefficients $\rho$, $\nu$, and $\sigma$, when specified, accurately describe the characteristics of several real-world plants. The value of these parameters is not a constant, but it is instead read or interpolated from a table, and depends on the water level $w$ (see, e.g., [Ružić et al., 1996]). Instead of interpolating the values from a table, since the value of the parameters $\rho$, $\nu$, $\sigma$ is approximately linear in the water level $w$ [Salam et al., 1998], we define the power function as

$$v_h(w, x) = (w + \eta)(\rho(x/\theta_t)^2 + \nu x/\theta_t + \sigma), \tag{9}$$

where $\eta$ is then a fourth parameter to be experimentally tuned.

**3.3.2. Demand and price function.** Obtained electrical energy can be sold on the electricity market at the market price; since we are considering a hydro power producer with a large capacity, the producer influences the market price, i.e., the market price depends on the amount of energy that it sells. We consider two alternatives to describe the price-quantity relation: a simple relationship is obtained by linearizing the step (staircase) price-quantity functions of Baslis and Bakirtzis [2011]. A finer description of the market effect of a large power producer can be obtained by using nonincreasing step functions, as in Baslis and Bakirtzis [2011]. However, modeling a step function requires binary variables in the scenario subproblems. In this case, the decomposition method we propose can only be applied to solve the continuous relaxation of the problem, and we additionally need a way to construct primal bounds: this will be discussed in Section 4.3. We now provide more details on the two above alternatives for the cost function.

- Using a linear price-quantity function, the profit-quantity relation in equation (5a) is expressed by a quadratic function of the energy, that is (recall that $e_t = \sum_{h=1}^{n} e_{th}$)

$$\pi_t(e_t)e_t = (\pi_{1t}e_t + \pi_{0t})e_t. \tag{10}$$

- Using a step price-quantity function with two steps, the profit-quantity relation in equation (5a) is expressed by

$$\pi_t(e_t)e_t \le \pi_{1t}e_t t = 1, \ldots, T \tag{11a}$$

$$\pi_t(e_t)e_t \le \pi_{2t}e_t + (\pi_{1t} - \pi_{2t})m_{1t}y_t t = 1, \ldots, T \tag{11b}$$

$$e_t \le m_{1t}y_t + m_{2t}(1 - y_t) t = 1, \ldots, T \tag{11c}$$

$$y_t \in \{0, 1\}, t = 1, \ldots, T \tag{11d}$$

where $m_{1t}$ is the maximum amount of energy that can be sold at price $\pi_{1t}$ in period $t$, $m_{2t}\,(> m_{1t})$ is the maximum (overall) amount of energy that can be sold at price $\pi_{2t}\,(< \pi_{1t})$ in period $t$, and $y_t$ is a binary variable indicating whether the amount of sold energy is $\le m_{1t}$ $(y_t = 1)$ or $> m_{1t}$ $(y_t = 0)$.

**3.4. Data.** The computational evaluation presented in this paper considers a case study based on the data from Baslis and Bakirtzis [2011], that describe a hydro system configuration comprising 10 major hydroplants of the Greek power system, for a production capacity of 2720 MW. As in [Baslis and Bakirtzis, 2011], we consider a three period configuration covering 12 months. The choice of the time periods is based on the Greek hydrological and load demand patterns, where high inflows are observed in winter and spring, and a load peak is observed in summer: the first period is the month of October, the second period goes from November to February, and the third period from March to September. Inflows and demand curves are computed based on historical data; we refer the reader to [Baslis and Bakirtzis, 2011] for details. The first time period is deterministic, as previously mentioned; a scenario tree comprising 90 scenarios is obtained by considering 5 inflow realizations coupled with 3 demand realizations at the second time period, and 3 inflow realizations coupled with 2 demand realizations at the third time period.

**4. Computational experiments.** In this section we report on the experimental results obtained by the described Branch-and-Cut algorithm when solving decomposable chance-constrained problems, where the subproblems are continuous and convex. We first test the algorithm on the instances discussed in Sect. 3 using the first formulation for the price function presented in Section 3.3.2, yielding a quadratic relationship between profit and sold energy described by (10); since we are not aware of any specialized solution method for the class of problems that we consider, we compare the algorithm performance with the direct solution of the large MINLP (2) using a general-purpose solver for convex MINLPs. Subsequently, in Section 4.3 we discuss our computational experience on the instances with the step price function formulation (11a)-(11c), for which we apply our approach to solve the continuous relaxation of the problem and to construct feasible integer solutions. The objective of these experiments is twofold: on the one hand, they are intended to assess the algorithmic performance of the method we propose; on the other hand, they allow us to evaluate our modeling approach for mid-term hydro scheduling problems, determining the size of the instances that can successfully be dealt with, and highlighting the trade-off between profit and robustness of the solution.

**4.1. Implementation details.** We implemented the Branch-and-Cut algorithm within the IBM ILOG CPLEX 12.6 MILP solver, and solved the convex subproblems with IPOPT 3.12 using the interface provided by BONMIN. In our implementation, CPLEX manages the branching tree of the master problem, and returns the control to a user-written callback function when the solution associated with a tree node is integer feasible.

Within the callback function, we define a separation problem (PROJ) for those scenarios $i$ having associated variable $z_i = 0$, i.e., the scenarios whose constraints must be satisfied. Problems (PROJ) are then solved by IPOPT. If the optimal solution of problem (PROJ) has strictly positive value for some scenario $j$, that is, the current master solution $\hat{x}$ violates the constraints of scenario $j$, then we derive a (single) valid cut $\gamma x \leq \beta_j$ separating $\hat{x}$ from the feasible region of scenario $j$, as explained in Section 2.2.

Then, we consider adding the obtained cut to the master problem in two alternative ways:

**big M** The cut is directly added to the master problem in the form $\gamma x \leq \beta_j + M z_j$. We compute the value for the $M$ coefficient as: $M = \sum_{l:\gamma_l>0} \gamma_l u_l - \beta_j$, where $l$ denotes the index of the $x$ variables in the cut and $u_l$ is the associated upper bound in the master problem;

**lifted** The cut is lifted by computing valid coefficients for the $z_i$ variables corresponding to other scenarios, i.e., $i \neq j$, as suggested by Luedtke [2014].

In the second case, for every $i$ we first compute the coefficient $\beta_i$ making the inequality valid for the corresponding scenario $w^i$, solving the optimization problem

$$\beta_i = \max\{\gamma x | x \in X \cap C_x(w^i)\}. \tag{12}$$

Assuming the $\beta_i$ values, $i = 1, \dots, k$, are sorted by non-decreasing order, we consider the first $p+1$ scenarios (recall $p = \lfloor \alpha k \rfloor$), and we obtain the following valid inequalities (see [Luedtke, 2014, Lemma 1]):

$$\gamma x + (\beta_i - \beta_{p+1}) z_i \leq \beta_i, \; i = 1, \dots, p. \tag{13}$$

From this basic set of inequalities, one could obtain stronger star inequalities (see [Atamtürk et al., 2000]). The basic idea is that, given an ordered subset $T = \{t_1, t_2, \dots, t_l\}$ of $\{1, \dots, p\}$, one can derive the following star inequality, where $\beta_{t_{l+1}} = \beta_{p+1}$ (see [Luedtke, 2014] for further details):

$$\gamma x + \sum_{i=1}^{l} (\beta_{t_i} - \beta_{t_{i+1}})) z_i \leq \beta_{t_1}. \tag{14}$$

Since the star inequalities (14) are in exponential number, they need to be separated. Separation can be performed by solving a longest path problem in an acyclic digraph. However, since we are separating integer solutions in the $z$ variables, the most violated inequality by a solution $(\hat{x}, z)$ is exactly the inequality (13) associated with the first $t_i$ in the ordering such that $z_{t_i} = 0$. Thus, in our implementation we add precisely inequalities (13).

Notice that since – in our specific application – separation for $\bar{C}_x(w^i)$ is not necessary to ensure correctness of the Branch-and-Cut algorithm, it is sufficient to find one violated scenario $i$ having associated variable $z_i = 0$, and to add the cut obtained by solving (PROJ) to the master problem: alternatively, all scenarios having associated variable $z_i = 0$ are satisfied, and the node does not have to be processed further. In our implementation we considered the following alternatives to determine how and when to perform separation:

**sepAll** Separation is performed at integer solutions for each scenario $i$ having associated variable $z_i = 0$;

**sepGroup** Scenarios are partitioned in subsets, where each subset includes those scenarios of the scenario tree having a common ancestor at the second time period (i.e., the corresponding sample paths are equal up to that point in time). Separation is performed at integer solutions for each group, until a violated scenario $i$ in the group having associated variable $z_i = 0$ is found.

The rationale for **sepGroup** is that scenarios in the same group have common decision variables at the second time period, hence a cut for one of these scenarios might change the primal solution for all scenarios in the same group. We tested two additional strategies that turned out to have poor computational performance. Hence, we describe them briefly below, but we will not report the corresponding results. Namely:

**sep1** Separation is performed at integer solutions until the first violated scenario $i$ having associated variable $z_i = 0$ is found.

**sepFrac** We attempt to separate cuts at fractional solutions using one of the other strategies mentioned above.

Both **sep1** and **sepFrac** were ineffective for the same reason: these two strategies increase the number of separation rounds, and, as it will be shown in the next section, the vast majority of the CPU time is already spent in solving the nonlinear separation subproblems, therefore increasing the number of separation rounds is an issue.

| | Nodes | | Time | | | |
|---|---|---|---|---|---|---|
| B&C algorithm | B&B | Sep. | CPU [s] | % NLP | NLP solved | Added cuts |
| **sepAll-bigM** | 30.4 | 4.2 | 121.1 | 99.5 | 8,228.0 | 1,384.2 |
| **sepGroup-bigM** | 28.9 | 4.0 | 154.5 | 99.5 | 10,504.2 | 1,346.9 |
| **sepAll-lifted** | 10.3 | 2.7 | 1,251.1 | 100.0 | (8,309.6) 102,529.3 | 1,409.1 |
| **sepGroup-lifted** | 13.3 | 2.7 | 1,098.2 | 100.0 | (9,829.5) 87,653.6 | 1,188.3 |

Table 1

*Performance summary for the four main variants of the B&C algorithm.*

Concerning the large MINLP (2), it is tackled through BONMIN, with IPOPT 3.12 as embedded nonlinear solver. For each constraint of the MINLP formulation to be activated/deactivated by the associated $z$ variable, we compute the smallest value of the $M$ coefficient using the bounds on the $x$ variables and the maximum profit that can be obtained in the scenarios by releasing the associated water quantities.

**4.2. Computational performance with linear price function.** The data from Baslis and Bakirtzis [2011] includes 10 hydroplants and a scenario tree with 90 equiprobable scenarios. From this data, we construct 5 smaller configurations with a number of plants chosen from the set $\{1, 2, 5, 7, 10\}$. For each configuration, we can specify the robustness of the solution: we consider values of the probability $\alpha$ starting from $\alpha = 0.5$ and decreasing by 0.1 down to $\alpha = 0.1$ (the computed solution must satisfy scenarios with associated probability of at least $1 - \alpha$). In the discussion about the performance of the hydroplants in Section 4.4 we additionally report results for $\alpha = 0.05$, but they are not included here as they do not provide further insight. Moreover, for 5 and 10 hydroplants and all values of $\alpha$, we considered four simplified scenario trees that contain 30, 48, 60 or 72 scenarios. We therefore obtain 65 instances of varying difficulty. All experiments are performed on a single node of a cluster containing machines equipped with an Intel Xeon E3-1220 processor clocked at 3.10 GHz and 8 GB RAM.

In Table 1 we report the main indicators to evaluate the performance of the four variants of the Branch-and-Cut algorithm that are obtained by combining the separation procedures **sepAll** and **sepGroup** with the **bigM** and **lifted** procedures to add cuts to the master problem. More specifically, the table reports average values of the total number of Branch-and-Bound nodes (second column), number of Branch-and-Bound nodes at which separation is performed (third column), total computing time and fraction of time spent in the nonlinear separation subproblems (fourth and fifth column respectively), number of nonlinear programs solved (sixth column), number of cuts added to the master problem (seventh column). For the **bigM** case, the number of NLPs solved is the same as the number of iterations of the separation procedure. For the **lifted** case, the number in brackets in the seventh column is the number of nonlinear programs solved to prove a given scenario is satisfied or derive the cut (iterations of the separation procedure), and the number of NLPs solved includes the NLPs to lift the cut.

All versions of the B&C algorithm solve all tested instances in less than 2 hours of computing time per instance. The **sepAll-bigM** variant is the fastest version on average, and we take it as our reference. Table 1 shows that the number of Branch-and-Bound nodes is very small on average (about 30), and almost all the computing time is spent in solving NLPs (on average, 8,228 NLPs per instance). Most of the separation iterations occur at the root node of the Branch-and-Cut algorithm (approximately 3/4 on average). We observe that many separation rounds are performed at each node where separation occurs. In the majority of the cases, when several mixed-integer solutions are produced at the same node each new mixed-integer solution differs from the previous one only in its continuous components. Only occasionally a new mixed-integer solution has different values for the $z$ variables, unless of course the separation is performed at different nodes of the Branch-and-Bound tree. This behavior can be explained by recalling that the master problem (7a)-(7e) is not aware of the nonlinear dynamics of the scenario subproblems, therefore a good approximation must be constructed by means of several linear cuts, even when the integer variables are fixed. Results with **sepGroup-bigM** are similar, with a small increase in the number of NLPs solved, and a corresponding increase of computing time.

Concerning the **lifted** cuts, we note that given a cut $\gamma x \leq \beta_i$ obtained for some scenario $i$, computing the lifting is computationally expensive due to the solution of several additional NLPs. This additional effort would be justified only if lifted cuts were able to significantly reduce the number of cut separation iterations with respect to **bigM** cuts. Table 1 shows that this is not the case: although the number of Branch-and-Bound nodes is reduced, the average number of separation iterations is of similar magnitude. As a consequence, **sepAll-lifted** and **sepGroup-lifted** solve many more NLPs and the CPU time increases accordingly. The ineffectiveness of lifted cuts can be explained in connection to the specific structure of the scenario tree we consider: when solving the optimization problem (12) for a given scenario $i$ and a given hyperplane $\gamma x$, only a subset of the variables with nonzero coefficient in $\gamma$ appears in nontrivial constraints (i.e., not bound constraints) for scenario $i$. Hence, the lifting procedure is rarely able to produce stronger cuts. The same observation on the weak computational performance of the mixing inequalities generated by an analogous lifting procedure is reported in [Qiu et al., 2014], where a chance-constrained formulation is studied as well.

The computational performance of BONMIN's NLP-based Branch-and-Bound algorithm, applied directly to the MINLP (2), are also evaluated on all 65 problem instances. The time limit for BONMIN is set to 10 hours. In Figure 2 we report the performance profile for the **sepAll-bigM** Branch-and-Cut algorithm and BONMIN, for the whole set of instances. The Branch-and-Cut algorithm can solve all instances, while BONMIN's Branch-and-Bound algorithm hits the time limit in 7 cases. In addition, the profiles clearly show the significantly better performance of the proposed approach compared to the direct solution of the large MINLP (2). Before reporting detailed computational results comparing the two approaches, we remark that we tried to solve the MINLP (2) with additional solvers based on other solution methods, namely, the BONMIN Outer Approximation algorithm, the BONMIN hybrid algorithm and the FilMINT Branch-and-Cut algorithm. None of the mentioned solvers could consistently handle the MINLP (2), and all solvers were plagued by severe numerical issues; as a consequence, they could correctly solve only small instances or instances with simplified nonlinear functions, and we decided to exclude them from our evaluation.

In Table 2 we report detailed results for a subset of instances of increasing complexity, comparing **sepAll-bigM** with BONMIN Branch and Bound. All instances in the table have 90 scenarios. The table reports the number of hydroplants and the level of risk $\alpha$ in the first two columns. Subsequent columns report the results obtained by the Branch-and-Cut algorithm, as in Table 1. The last two columns report the performance of BONMIN Branch-and-Bound algorithm, indicating the total CPU time and the number of Branch-and-Bound nodes. The Branch-and-Cut algorithm solves all instances in less than 10 minutes each, and in a very limited number of Branch-and-Bound nodes. Instances with a smaller number of hydroplants appear

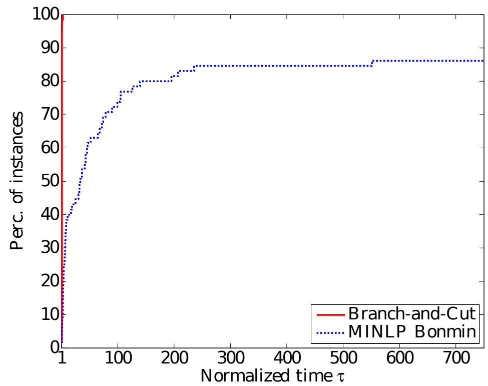


FIG. 2. *Performance profiles for 65 instances (linear price function).*

| | | Branch and Cut | | | | | | BONMIN | |
|---|---|---|---|---|---|---|---|---|---|
| Plants | $\alpha$ | B&B nodes | Sep. nodes | Time (s) | % time NLP | NLP solved | Added cuts | Time (s) | Nodes |
| 1 | 0.1 | 31 | 5 | 22.3 | 100.0 | 2,520 | 228 | 155.6 | 365 |
| 1 | 0.2 | 9 | 3 | 26.5 | 100.0 | 3,114 | 289 | 2,775.4 | 14,173 |
| 1 | 0.3 | 21 | 4 | 23.0 | 99.9 | 2,799 | 304 | 2,291.6 | 12,955 |
| 1 | 0.4 | 14 | 4 | 20.9 | 99.9 | 2,845 | 327 | 4,331.8 | 26,144 |
| 1 | 0.5 | 16 | 3 | 15.7 | 99.8 | 2,160 | 273 | 3,691.6 | 22,856 |
| 2 | 0.1 | 16 | 5 | 27.0 | 99.9 | 2,844 | 395 | 1,836.7 | 2,606 |
| 2 | 0.2 | 1 | 1 | 16.0 | 99.7 | 1,746 | 339 | 14,279.8 | 24,749 |
| 2 | 0.3 | 31 | 3 | 27.1 | 99.7 | 2,925 | 418 | 14,961.3 | 29,606 |
| 2 | 0.4 | 35 | 4 | 32.4 | 99.8 | 3,514 | 482 | 6,326.3 | 20,554 |
| 2 | 0.5 | 7 | 2 | 8.1 | 99.8 | 990 | 329 | 7,996.1 | 27,761 |
| 5 | 0.1 | 12 | 2 | 98.6 | 99.9 | 7,887 | 1,315 | 2,471.5 | 3,060 |
| 5 | 0.2 | 34 | 5 | 82.3 | 99.7 | 6,720 | 1,258 | 5,331.1 | 6,255 |
| 5 | 0.3 | 17 | 3 | 93.6 | 99.8 | 7,676 | 1,339 | 13,086.6 | 17,800 |
| 5 | 0.4 | 63 | 7 | 89.1 | 99.6 | 7,380 | 1,338 | 9,376.6 | 11,745 |
| 5 | 0.5 | 27 | 5 | 88.7 | 99.7 | 7,427 | 1,325 | 8,011.3 | 10,742 |
| 7 | 0.1 | 9 | 3 | 205.3 | 99.8 | 14,883 | 2,021 | 9,554.3 | 7,001 |
| 7 | 0.2 | 36 | 3 | 136.7 | 99.7 | 9,977 | 1,803 | 7,107.8 | 6,338 |
| 7 | 0.3 | 47 | 6 | 180.3 | 99.6 | 13,201 | 2,424 | 5,776.8 | 4,445 |
| 7 | 0.4 | 115 | 6 | 131.8 | 99.1 | 9,052 | 1,605 | 16,619.1 | 13,397 |
| 7 | 0.5 | 74 | 9 | 175.1 | 99.5 | 11,783 | 1,785 | 7,520.0 | 6,159 |
| 10 | 0.1 | 45 | 7 | 237.6 | 99.6 | 14,375 | 2,295 | 4,520.6 | 2,189 |
| 10 | 0.2 | 33 | 4 | 200.9 | 99.6 | 11,822 | 2,062 | 9,186.7 | 4,811 |
| 10 | 0.3 | 29 | 5 | 383.6 | 99.6 | 23,904 | 3,602 | 14,136.2 | 6,380 |
| 10 | 0.4 | 131 | 9 | 429.7 | 99.1 | 26,668 | 3,845 | 13,818.4 | 7,035 |
| 10 | 0.5 | 94 | 8 | 414.2 | 99.2 | 26,280 | 3,821 | T.L. | 17,077 |

TABLE 2
*Comparison between **sepAll-bigM** and BONMIN Branch and Bound on configurations of 1, 2, 5, 7, and 10 hydroplants and 90 scenarios (linear price function).*

easier for the Branch-and-Cut algorithm, while the level of risk $\alpha$ has little effect on the solution time. Solution via BONMIN Branch-and-Bound algorithm takes a much larger number of nodes and computing time (two orders of magnitude larger on average). Very few instances are solved to optimality within 1 hour of computing time.

Similar considerations can be drawn from Table 3, where all the instances are for the 10 hydroplants configuration, and different number of scenarios, as reported in the first column. In addition, the table clearly shows that reducing the number of scenarios makes the problem easier, for both the Branch-and-Cut algorithm and the BONMIN Branch-and-Bound algorithm.

| | | Branch and Cut | | | | | | BONMIN | |
|---|---|---|---|---|---|---|---|---|---|
| # scen. | $\alpha$ | B&B nodes | Sep. nodes | Time (s) | % time NLP | NLP solved | Added cuts | Time (s) | Nodes |
| 30 | 0.1 | 1 | 1 | 74.9 | 99.7 | 4,506 | 780 | 27.2 | 7 |
| 30 | 0.2 | 2 | 2 | 23.5 | 99.7 | 1,470 | 512 | 39.7 | 34 |
| 30 | 0.3 | 1 | 1 | 34.7 | 99.6 | 2,171 | 666 | 85.0 | 117 |
| 30 | 0.4 | 15 | 3 | 55.0 | 99.7 | 3,396 | 788 | 60.1 | 85 |
| 30 | 0.5 | 1 | 1 | 24.9 | 99.8 | 1,425 | 409 | 36.0 | 30 |
| 48 | 0.1 | 30 | 3 | 115.1 | 99.7 | 6,643 | 1,378 | 868.9 | 347 |
| 48 | 0.2 | 29 | 5 | 187.3 | 99.5 | 11,278 | 2,121 | 6,286.4 | 2,029 |
| 48 | 0.3 | 38 | 7 | 266.5 | 99.5 | 16,702 | 2,642 | 2,301.4 | 1,169 |
| 48 | 0.4 | 14 | 3 | 190.4 | 99.6 | 11,304 | 1,615 | 5,785.1 | 2,907 |
| 48 | 0.5 | 19 | 3 | 71.0 | 99.5 | 4,283 | 974 | 52.6 | 1 |
| 60 | 0.1 | 1 | 1 | 215.4 | 99.7 | 12,992 | 1,956 | 1,299.7 | 615 |
| 60 | 0.2 | 11 | 3 | 209.1 | 99.6 | 13,091 | 2,182 | 1,472.1 | 891 |
| 60 | 0.3 | 31 | 5 | 377.1 | 99.4 | 22,024 | 3,408 | 1,111.1 | 746 |
| 60 | 0.4 | 106 | 8 | 294.9 | 99.4 | 18,419 | 2,659 | 1,155.8 | 769 |
| 60 | 0.5 | 39 | 6 | 127.7 | 99.4 | 7,923 | 1,869 | 406.7 | 329 |
| 72 | 0.1 | 3 | 2 | 183.8 | 99.6 | 11,181 | 2,399 | 7,896.0 | 2,692 |
| 72 | 0.2 | 29 | 4 | 221.9 | 99.4 | 13,700 | 2,698 | T.L. | 8,575 |
| 72 | 0.3 | 48 | 7 | 284.4 | 99.5 | 17,814 | 2,942 | T.L. | 11,505 |
| 72 | 0.4 | 112 | 8 | 458.5 | 99.1 | 28,658 | 3,905 | T.L. | 10,471 |
| 72 | 0.5 | 161 | 11 | 260.1 | 99.0 | 16,522 | 2,844 | 4,421.3 | 2,860 |
| 90 | 0.1 | 45 | 7 | 237.6 | 99.6 | 14,375 | 2,295 | 4,520.6 | 2,189 |
| 90 | 0.2 | 33 | 4 | 200.9 | 99.6 | 11,822 | 2,062 | 9,186.7 | 4,811 |
| 90 | 0.3 | 29 | 5 | 383.6 | 99.6 | 23,904 | 3,602 | 14,136.2 | 6,380 |
| 90 | 0.4 | 131 | 9 | 429.7 | 99.1 | 26,668 | 3,845 | 13,818.4 | 7,035 |
| 90 | 0.5 | 94 | 8 | 414.2 | 99.2 | 26,280 | 3,821 | T.L. | 17,077 |

TABLE 3

*Comparison between **sepAll-bigM** and BONMIN Branch and Bound on configurations with 10 hydroplants and 30, 48, 60, 72, and 90 scenarios (linear price function).*

**4.3. Computational performance with step price function.** The step price function modeled in (11a)–(11d) requires binary variables, yielding sets $C_x(w^i)$ that are nonconvex (because of the integrality constraints). However, the continuous relaxation of (11a)–(11d) is convex, therefore we can apply our decomposition method to solve the continuous relaxation of the chance-constrained problem. Such a relaxation yields dual, i.e., upper, bounds. Our heuristic approach to generate primal bounds is as follows. First, we apply the Branch-and-Cut algorithm to solve the continuous relaxation of the chance-constrained problem to optimality, and obtain an exact dual bound. Then we restart the Branch-and-Cut algorithm, keeping the pool of generated cuts, and enforcing integrality requirements for scenario subproblems in the cut generation process, i.e., when solving (PROJ). This way, the Branch-and-Cut algorithm tries to converge to an integer solution, although not necessarily an optimal one. In our experiments this approach always yields a primal bound that matches the dual bound, therefore proving that we have found an optimal solution. Notice that our approach is in general not guaranteed to find primal bounds matching the dual bound or even any integer solution, but the following two features of our application may explain why we find optimal integer solutions with such ease after solving the continuous relaxation:

- In each scenario subproblem, when the quantity of energy sold in period $t$ falls in the first step of the step price function (larger price for a limited amount of energy), integrality of the associated $y_t$ variable is automatically attained because of the objective function's direction, i.e., profit maximization;
- In the master problem, the maximization of a quantile of the profit implies that the objective function value is given by the minimum profit among satisfied scenarios. The scenario attaining minimum profit is likely to involve a

| | | Branch and Cut | | | | | | MINLP | |
|---|---|---|---|---|---|---|---|---|---|
| Plants | $\alpha$ | B&B nodes | Sep. nodes | Time (s) | % time MINLP | MINLP solved | Added cuts | Time (s) | Nodes |
| 1 | 0.1 | 17 | 6 | 24.0 | 100.0 | 2529 | 208 | 184.4 | 262 |
| 1 | 0.2 | 12 | 5 | 54.1 | 100.0 | 3780 | 289 | 283.0 | 552 |
| 1 | 0.3 | 25 | 7 | 61.1 | 99.9 | 4149 | 319 | 1,863.4 | 7,202 |
| 1 | 0.4 | 30 | 7 | 32.0 | 100.0 | 3042 | 340 | 13,844.6 | 50,827 |
| 1 | 0.5 | 2 | 2 | 48.2 | 99.9 | 3510 | 315 | 1,724.7 | 6,402 |
| 2 | 0.1 | 42 | 6 | 44.7 | 99.9 | 4068 | 430 | 11,184.3 | 14,579 |
| 2 | 0.2 | 19 | 6 | 80.1 | 99.9 | 4932 | 459 | t.l. | 44,120 |
| 2 | 0.3 | 28 | 7 | 67.6 | 99.9 | 6102 | 544 | t.l. | 48,410 |
| 2 | 0.4 | 145 | 15 | 80.9 | 99.8 | 5424 | 509 | t.l. | 43,913 |
| 2 | 0.5 | 2 | 2 | 51.4 | 100.0 | 3669 | 463 | 2,144.2 | 3,537 |
| 5 | 0.1 | 10 | 4 | 203.5 | 100.0 | 11975 | 1373 | 7,272.0 | 5,097 |
| 5 | 0.2 | 17 | 6 | 171.2 | 99.9 | 9612 | 1314 | 13,161.5 | 11,400 |
| 5 | 0.3 | 33 | 7 | 206.8 | 99.9 | 12414 | 1546 | t.l. | 30,940 |
| 5 | 0.4 | 74 | 9 | 204.5 | 99.8 | 9630 | 1535 | t.l. | 23,131 |
| 5 | 0.5 | 43 | 8 | 132.0 | 99.9 | 6684 | 1350 | 3,277.8 | 3,232 |
| 7 | 0.1 | 26 | 5 | 155.7 | 99.8 | 7470 | 1633 | 8,531.0 | 4,400 |
| 7 | 0.2 | 64 | 9 | 301.8 | 99.9 | 14972 | 2076 | 23,673.6 | 12,202 |
| 7 | 0.3 | 61 | 8 | 356.8 | 99.8 | 15826 | 2232 | t.l. | 19,806 |
| 7 | 0.4 | 46 | 7 | 267.6 | 99.9 | 12491 | 2008 | 9,765.6 | 7,094 |
| 7 | 0.5 | 100 | 13 | 227.9 | 99.7 | 11050 | 1896 | 3,731.0 | 2,572 |
| 10 | 0.1 | 19 | 4 | 266.0 | 99.8 | 12165 | 2144 | 5,890.2 | 2,226 |
| 10 | 0.2 | 56 | 7 | 476.0 | 99.7 | 20477 | 2796 | t.l. | 14,775 |
| 10 | 0.3 | 113 | 10 | 622.2 | 99.3 | 29469 | 4041 | t.l. | 14,027 |
| 10 | 0.4 | 340 | 20 | 903.3 | 99.2 | 35838 | 4793 | t.l. | 13,922 |
| 10 | 0.5 | 106 | 12 | 647.9 | 99.3 | 25136 | 3933 | t.l. | 15,833 |

TABLE 4

*Comparison between* ***sepAll-bigM*** *and BONMIN Branch and Bound on configurations of 1, 2, 5, 7, and 10 hydroplants and 90 scenarios (step price function).*

limited amount of water flow, thus a limited energy production that falls in the first step of the step price function. Not only such a scenario may have an integral solution to the continuous relaxation, but we may also expect binding cuts in the master problem to be obtained from scenarios where integrality constraints are satisfied.

We remark that if the scheme described above fails to produce a feasible integer solution, such a solution for each scenario subproblem can be obtained by rounding $y_t$ to 1 whenever $y_t > 0$ in the solution to its continuous relaxation, and recomputing the corresponding profit, which will in general be lowered. However, this was never necessary in our experiments, and the suggested procedure has the advantage of producing (possibly invalid) cuts that describe the production-profit relation to better inform the master problem. Results for the integer case and a comparison with the BONMIN Branch-and-Bound performance are reported in Tables 4 and 5. The columns associated with the Branch-and-Cut algorithm include the dual and primal bounds computation. On average, computing the dual bound takes 39.5% of the computing time, and generates 78.6% of the cuts. BONMIN times out on the majority of the instances, and is up to two orders of magnitude slower than our approach on the instances that it can solve. This is evident from the performance profile in Figure 3. Summarizing, on the formulation with a step price function that includes binary variables in the scenario subproblems we are able to find provably optimal integer solutions on all instances whereas BONMIN solves less than a half within the time limit, and the solution speed is up to two orders of magnitude faster. However, it is important to note that in general our approach can only solve the continuous relaxation of the problem to optimality and may not be able to find a primal bound matching the dual bound.

| | | Branch and Cut | | | | | | MINLP | |
|---|---|---|---|---|---|---|---|---|---|
| # scen. | $\alpha$ | B&B nodes | Sep. nodes | Time (s) | % time MINLP | MINLP solved | Added cuts | Time (s) | Nodes |
| 30 | 0.1 | 2 | 2 | 38.4 | 100.0 | 1788 | 419 | 272.2 | 195 |
| 30 | 0.2 | 10 | 4 | 131.1 | 99.8 | 3993 | 1065 | 218.2 | 268 |
| 30 | 0.3 | 21 | 6 | 168.8 | 99.9 | 4392 | 1038 | 2,139.5 | 3,602 |
| 30 | 0.4 | 18 | 4 | 248.3 | 99.7 | 8543 | 1541 | 2,713.2 | 4,137 |
| 30 | 0.5 | 8 | 6 | 109.0 | 99.9 | 3300 | 865 | 228.1 | 229 |
| 48 | 0.1 | 31 | 5 | 494.9 | 99.7 | 14482 | 2319 | 10,709.1 | 6,724 |
| 48 | 0.2 | 42 | 6 | 396.7 | 99.6 | 17485 | 2625 | t.l. | 16,838 |
| 48 | 0.3 | 60 | 11 | 679.4 | 99.5 | 29203 | 3747 | t.l. | 23,453 |
| 48 | 0.4 | 73 | 11 | 422.8 | 99.5 | 21396 | 2753 | t.l. | 21,517 |
| 48 | 0.5 | 8 | 3 | 105.0 | 99.8 | 3143 | 897 | 508.7 | 372 |
| 60 | 0.1 | 4 | 3 | 220.2 | 99.9 | 10424 | 1789 | 4,508.2 | 3,000 |
| 60 | 0.2 | 39 | 7 | 475.3 | 99.7 | 22994 | 3588 | t.l. | 23,961 |
| 60 | 0.3 | 93 | 10 | 708.1 | 99.4 | 27992 | 4096 | t.l. | 23,582 |
| 60 | 0.4 | 165 | 17 | 660.2 | 99.3 | 27661 | 3696 | t.l. | 20,994 |
| 60 | 0.5 | 78 | 8 | 132.5 | 99.8 | 8515 | 1631 | t.l. | 23,648 |
| 72 | 0.1 | 39 | 7 | 311.9 | 99.8 | 16260 | 2884 | t.l. | 9,584 |
| 72 | 0.2 | 118 | 11 | 663.9 | 99.4 | 34044 | 4450 | t.l. | 8,920 |
| 72 | 0.3 | 137 | 12 | 814.7 | 99.4 | 27280 | 3881 | t.l. | 10,680 |
| 72 | 0.4 | 135 | 12 | 701.5 | 99.5 | 26291 | 3466 | t.l. | 12,087 |
| 72 | 0.5 | 179 | 15 | 655.2 | 99.3 | 25632 | 3844 | t.l. | 20,070 |
| 90 | 0.1 | 19 | 4 | 266.0 | 99.8 | 12165 | 2144 | 5,890.2 | 2,226 |
| 90 | 0.2 | 56 | 7 | 476.0 | 99.7 | 20477 | 2796 | t.l. | 14,775 |
| 90 | 0.3 | 113 | 10 | 622.2 | 99.3 | 29469 | 4041 | t.l. | 14,027 |
| 90 | 0.4 | 340 | 20 | 903.3 | 99.2 | 35838 | 4793 | t.l. | 13,922 |
| 90 | 0.5 | 106 | 12 | 647.9 | 99.3 | 25136 | 3933 | t.l. | 15,833 |

TABLE 5
*Comparison between **sepAll-bigM** and BONMIN Branch and Bound on configurations with 10 hydroplants and 30, 48, 60, 72, and 90 scenarios (step price function).*

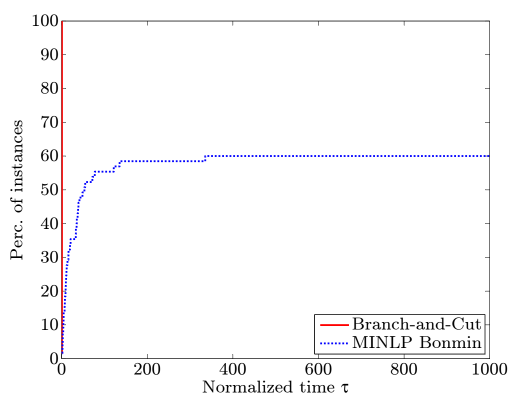


FIG. 3. *Performance profiles for 65 instances (step price function).*

**4.4. The effect of $\alpha$ on the profit.** We now discuss the trade-off between profit and risk allowed by our chance-constrained formulation for the mid-term hydro scheduling problem. The results discussed here are obtained with the linear price function, see Sect. 3.3.2. Figure 4 shows, for several configurations of the system (1 to 10 hydroplants), the objective function value (quantile of the profit) of the solutions as a function of the level of risk $\alpha$, restricted to the case of 90 scenarios. This allows the decision maker to easily evaluate not only the (minimum) profit they can obtain for a specified value of the risk, but also what profit they could expect by accepting a larger or smaller uncertainty. Of course, the objective function value obtained with a given $\alpha$ corresponds to the minimum profit that can be achieved with probability

| $\alpha$ | $E[\varphi]$ | $\sigma$ |
|---|---|---|
| 0.00 | 561.0 | 198.9 |
| 0.05 | 595.3 | 203.1 |
| 0.10 | 600.3 | 211.5 |
| 0.20 | 588.3 | 252.5 |
| 0.30 | 594.0 | 257.4 |
| 0.40 | 582.6 | 257.7 |
| 0.50 | 518.7 | 330.1 |

TABLE 6

*Expected profit in €M (second column) and standard deviation (third column) for different values of $\alpha$.*

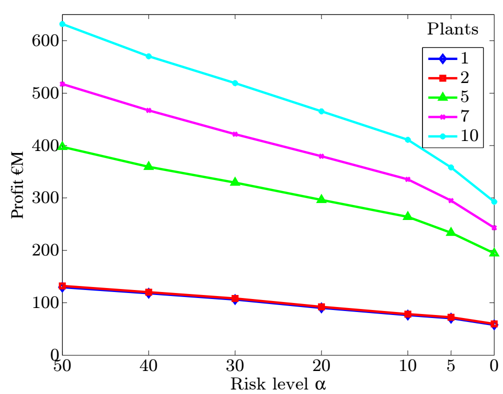


FIG. 4. *Trade-off between profit in €M and level of risk: the $x$-axis reports the risk level $\alpha$, and the $y$-axis the corresponding objective function value.*

$1-\alpha$, but the solution may be infeasible with probability $\alpha$. In this section, $\alpha = 0.05$ is included in the comparison besides the $\alpha$ values tested above.

Once the problem is optimally solved for a specific level of risk $\alpha$, the decision maker can also evaluate the distribution of the profits associated with the different scenarios. Indeed, a solution to the master problem specifies a value for the flow variables: this allows us to compute the associated profit for all satisfied scenarios, and also for those unsatisfied scenarios for which the flow variables define a physically feasible solution (i.e., those scenarios for which the water balance constraints are satisfied, but constraints (6) are not). Figure 5 depicts the inverse distribution function of the profit for the case of 10 hydroplants. We remark that here, and in the computation of expected profits below, we are assuming that the profit is zero whenever a solution violates the water conservation constraints. The solution obtained with $\alpha = 0$ (all scenarios are satisfied) achieved a profit that is consistently below the other solutions, except for scenarios when the other solutions are infeasible. As expected, there is a spike in each curve when the value on the $x$-axis corresponds to the level of risk $\alpha$ being optimized. It is interesting to note that even a risk-averse solution ($\alpha = 0.05$) achieves a profit that is relatively similar to the least risk-averse solution ($\alpha = 0.5$), although in the most favorable scenarios (right part of the graph), $\alpha = 0.5$ typically yields better profit than $\alpha = 0.05$. On the other hand, for the most unfavorable scenarios, up to a cumulative probability of almost 0.5, the solutions with $\alpha = 0.05$ and $\alpha = 0.1$ perform much better than with $\alpha = 0.5$. Solutions obtained with $\alpha \in \{0.2, 0.3, 0.4\}$ are similar to each other, and they all perform worse than $\alpha = 0.1$ for a cumulative probability of up to 0.2, as expected, but perform better

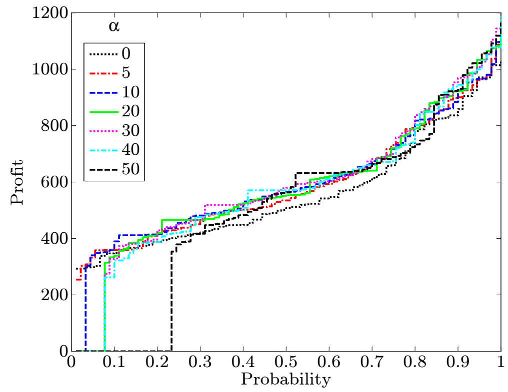


FIG. 5. *Inverse distribution function of the profit.*

in the most favorable scenarios, achieving approximately 50M higher profit in some cases. Table 6 reports the expected profit and the standard deviation of the solutions corresponding to the tested values of $\alpha$. We can see that relaxing some of the constraints with small probability ($\leq 0.05$) yields an increase of the expected profit by 6.1% as compared to the solution with $\alpha = 0$, although unsurprisingly this comes at the cost of a slightly larger standard deviation. The highest expected profit is achieved with $\alpha = 0.1$, where the increase is of 7% as compared to $\alpha = 0$. Allowing constraint violations with higher probability produces infeasible solutions in a larger number of scenarios, and the corresponding lack of profit decreases the expected gain. When $\alpha$ is very large ($\alpha = 0.5$), the solution obtained is infeasible for many scenarios, leading to an expected profit almost 10% lower than the conservative solution with $\alpha = 0$.

Summarizing, our computational experiments indicate that introducing a moderate amount of flexibility in the formulation, namely by allowing some constraints to be violated with small probability (0.05 or 0.1), can increase the expected profit by a significant amount. However, there are diminishing returns of increasing $\alpha$, and when the allowed probability of violating the constraints becomes too large, the resulting trade-off between risk and rewards seems to be unfavorable, yielding a considerable drop in the expected profit.

**5. Conclusions.** We have proposed a Branch-and-Cut algorithm for a class of nonlinear chance-constrained mathematical optimization problems with a finite number of scenarios. The algorithm is based on an implicit Benders decomposition scheme, where we generate cutting planes as outer approximation constraints from the projection of the feasible region on suitable subspaces.

The algorithm has been theoretically analyzed and computationally evaluated on a mid-term hydro scheduling problem by using data from ten hydroplants in Greece. We have shown that the proposed methodology is capable of solving instances orders of magnitude faster than applying a general-purpose solver for convex mixed-integer nonlinear programming problems to the deterministic reformulation, and scales much better with the number of scenarios.

From the economical standpoint, our numerical experiments have shown that the introduction of a small amount of flexibility in the formulation, by allowing constraints to be violated with a joint probability $\leq 5\%$, increases the expected profit by 6.1%

on our dataset.

**Acknowledgements.** The authors are extremely grateful to Costas Baslis and Anastasios Bakirtzis for sharing the data on the Greek power system discussed in [Baslis and Bakirtzis, 2011], as well as to Alberto Borghetti and Amitabh Basu for several helpful discussions. The first, second and fourth authors acknowledge the support of MIUR, Italy, under the grant PRIN 2012. Traveling support by the EU ITN 316647 "Mixed-Integer Nonlinear Optimization" is acknowledged by the third author. Part of this research was carried out at the Singapore University of Technology and Design, supported by grant SRES11012 and IDC grant IDSF1200108.

**References.**

A. Kumar, S. Leyffer, and J. Linderoth. FilMINT: An outer approximation-based solver for convex mixed-integer nonlinear programs. *INFORMS Journal on Computing*, 22(4):555–567, 2010.

A. Atamtürk, G.L. Nemhauser, and M.W.P. Savelsbergh. The mixed vertex packing problem. *Mathematical Programming*, 89(1):35–53, 2000.

L. Bacaud, C. Lemarèchal, A. Renaud, and C. Sagastizábal Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners. *Computational Optimization ans Applications*, 20(3):227–244, 2001.

G.C. Baslis and G.A. Bakirtzis. Mid-term stochastic scheduling of a price-maker hydro producer with pumped storage. *IEEE Transactions on Power Systems*, 26 (4):1856–1865, 2011.

D.P. Bertsekas. *Nonlinear Programming, 2nd Edition*. Athena Scientific, Belmont, MA, 1999.

J.A. Bloom. Solving an electricity generating capacity expansion planning problem by generalized Benders' decomposition. *Operations Research*, 31(1):84–100, 1983.

P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex Mixed Integer Nonlinear Programs. *Discrete Optimization*, 5:186–204, 2008.

P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot. A feasibility pump for Mixed Integer Nonlinear Programs. *Mathematical Programming*, 119(2):331–352, 2009.

P. Bonami, A. Lodi, A. Tramontani, and S. Wiese. On mathematical programming with indicator constraints. *Mathematical Programming*, 151(1):191–223, 2015.

A.A.F.M Carneiro, S. Soares, and P.S. Bond. A large scale of an optimal deterministic hydrothermal scheduling algorithm. *IEEE Transactions on Power Systems*, 5(1): 204–211, 1990.

P. L. Carpentier, M. Gendreau, and F. Bastin. Midterm hydro generation scheduling under uncertainty using the progressive hedging algorithm. Technical Report 2012-35, CIRRELT, 2012.

H.C. Chang and P.H. Chen. Hydrothermal generation scheduling package: a genetic based approach. *IEEE Proceedings on Generation, Transmission and Distribution*, 145(4), 1998.

A. Charnes and W. W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11(1):18–39, 1963.

A. Charnes, W. W. Cooper, and G. H. Symonds. Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil. *Management Science*, 4(3):235–263, 1958.

M. Duran and I. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

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

R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by outer approximation. *Mathematical Programming*, 66:327–349, 1994.

P.M. França and H.P.L. Luna. Solving stochastic transportation-location problems by generalized Benders decomposition. *Transportation Science*, 16(2):113–126, 1982.

D. Gade, S. Küçükyavuz, and S. Sen. Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, 144(1-2):39–64, 2014.

A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

O. Günlük and Y. Pochet. Mixing mixed-integer inequalities. *Mathematical Programming*, 90(3):429–457, 2001.

R.G. Jeroslow. Representability in mixed integer programming, I: characterization results. *Discrete Applied Mathematics*, 17(3):223–243, 1987.

J.E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society of Industrial and Applied Mathematics*, 8(4):703–712, 1960.

M.P.N.C.R. Kelman. Long-term hydro scheduling based on stochastic models. *EPSOM*, 98:23–25, 1998.

M.A. Lejeune. Pattern-based modeling and solution of probabilistically constrained optimization problems. *Operations Research*, 60(6):1356–1372, 2012.

X. Liu, S. Küçkyavuz, and J. Luedtke. Decomposition algorithms for two-stage chance-constrained programs. *Mathematical Programming*, pages 1–25, 2014.

J. Luedtke. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming*, 146:219–244, 2014.

J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

A.J. McNeil, R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts, Techniques and Tools (Revised Edition)*. Princeton Series in Finance. Princeton University Press, Princeton, NJ, 2015.

M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

A. Prekopa. On probabilistic constrained programmming. In H. W. Kuhn, editor, *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138, Princeton, NJ, 1970. Princeton University Press.

F. Qiu, S. Ahmed, S.S. Dey, and L.A. Wolsey. Covering linear programming with violations. *INFORMS Journal on Computing*, 26(3):531–546, 2014.

S. Ružić, N. Rajaković, and A. Vučković. A flexible approach to short-term hydrothermal coordination. I. Problem formulation and general solution procedure. *IEEE Transactions on Power Systems*, 11(3):1564–1571, 1996.

M.S. Salam, K.M. Nor, and A.R. Hamdan Hydrothermal scheduling based lagrangian relaxation approach to hydrothermal coordination. *IEEE Transactions on Power Systems*, 13(1):226–235, 1998.

Y. Song, J. Luedtke, and S. Küçükyavuz. Chance-constrained binary packing problems. *INFORMS Journal on Computing*, 26(4):735–747, 2014.

M. Tahanan, W. van Ackooij, A. Frangioni, and F. Lacalandra. Large-scale unit commitment under uncertainty. *4OR*, 13(2):115–171, 2015.

S. Takriti and S. Ahmed. On robust optimization of two-stage systems. *Mathematical Programming*, 99:109–126, 2004.

M.W. Tanner, L. Sattenspiel, and L. Ntaimo. Finding optimal vaccination strategies under parameter uncertainty using stochastic programming. *Mathematical Biosciences*, 215(2):144–151, 2008.

W. van Ackooij. Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit commitment. *Mathematical Methods of Operations Research*, 80(3):227–253, 2014.

Q. Wang, Y. Guan, and J. Wang. A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. *Power Systems, IEEE Transactions on*, 27(1):206–215, 2012.

T. Watanabe and H. Ellis. Stochastic programming models for air quality management. *Computers & Operations Research*, 20(6):651–663, 1993.

T. Westerlund, H. Skrifvars, I. Harjunkoski, and R. Pörn. An extended cutting plane method for a class of non-convex MINLP problems. *Computers & Chemical Engineering*, 22(3):357–365, 1998.