Mathematical Programming (2022) 191:405–444
https://doi.org/10.1007/s10107-019-01447-3

FULL LENGTH PAPER

Series B

Nonlinear chance-constrained problems with applications to hydro scheduling

Andrea Lodi[1] · **Enrico Malaguti**[2] · **Giacomo Nannicini**[3] · **Dimitri Thomopulos**[4]

Received: 10 February 2017 / Accepted: 1 November 2019 / Published online: 14 November 2019

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 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.

Keywords Chance-constraints · Outer approximation · Benders decomposition · Branch-and-Cut · Hydro scheduling

Mathematics Subject Classification 90C15 Stochastic Programming · 49M27 Decomposition Methods · 90C57 Branch-and-Cut

---

✉ Andrea Lodi
andrea.lodi@polymtl.ca

Extended author information available on the last page of the article

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 1950s, see, e.g., [1,2]. 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 α, 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 α. This type of problem is called a *chance-constrained mathematical programming* problem in the literature [1].

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

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

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 constraints [2]. 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., [3–5]. 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 operation. This problem has been studied in [6], where a cost for the normal-mode 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:

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

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 (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., [7,8]. 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 Ω, is discrete and finite, and in particular $\Omega = \{w^i : i = 1, \ldots, k\}$.

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 [9]. From now on, we indeed assume $\Omega = \{w^i : i = 1, \ldots, k\}$. The realizations $w^1, \ldots, w^k$ 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:

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

To simplify this problem, we make an additional assumption:

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

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 φ represents a nonnegative 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 [10]. 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, where the vectors x, c of (CCPR-OBJ) can be recovered as the first n components of x and c below:

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

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

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

A possible 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, and the recession cone of (CCPR), might be different otherwise. Indeed, the recession cone of (CCPR) is the union, over all feasible combinations of z_i, of the intersection of the recession cones of the sets $C_x(w^i)$, $\bar{C}_x(w^i)$ that are "active", as determined by the z_i variables. On the other hand, it is known that in the deterministic equivalent of (CCPR), formulated using big-M constraints to activate/deactivate sets, the recession cones of "inactive" sets cannot be deactivated, hence the only unbounded directions are those that belong to all the sets $C_x(w^i)$, $\bar{C}_x(w^i)$ at the same time (see [11]). This is clearly not the same as the recession cone of (CCPR), unless the sufficient condition (A3) holds. We remark that different, possibly approximate, reformulations of (CCP) or (CCPR) have been discussed in the literature, e.g., [12–14], but none of them yields an exact algorithm under the assumptions of the present paper.

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., [15]). 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 involves them. Then, a Branch-and-Cut algorithm [16] can be applied to the problem (CCPR), setting up a master problem of the form $\max_{x,z}\{cx : x \in X, z \in \{0,1\}^k, \sum_{i=1}^k p_i z_i \leq \alpha\}$. 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 problems [6,17]. 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., [18,19], 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 [20], 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 described by finitely many convex, potentially nonlinear inequalities, and propose a finitely convergent Branch-and-Cut algorithm. The cutting planes that we generate can be obtained as outer approximation cuts [21] and are therefore linear, as opposed to the generalized Benders cuts of [20], 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 [3,22–24]. 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., [25]), 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 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 [22] 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, but of course this is allowed by the algorithm that we propose, by dropping constraints and auxiliary variables related to $\bar{\varphi}(x, w^i)$, $\bar{C}_x(w^i)$ in (CCPR).

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 [6] in two ways: assumption (A2) of [6,17], imposing polyhedrality of the scenario problems, is replaced by the weaker assumption of nonlinear convex scenario problems, and assumption (B1) of [6], imposing relatively complete recourse on the recovery scenario problems, is dropped. On the other hand, assumption (B2) in [6] imposes essentially the opposite of (A2) of the present paper, and for this reason [6] has to consider a threshold policy to determine when to operate the recovery mode, see [6, Sect. 2.2.2]. Both models find their application in specific situations: we discuss this at the end of Sect. 2.1. 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 Sect. 5.

2 Decomposition algorithm for (CCPR)

Under assumptions (A1)–(A3), we discussed in Sect. 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 $X = \{x : Ax \leq b\}$, $C_x(w^i) = \{x : \exists y^i \;\; A^i x + H^i y^i \leq b^i\}$, $\bar{C}_x(w^i) = \{x : \exists \bar{y}^i \;\; \bar{A}^i x + \bar{H}^i y^i \leq \bar{b}^i\}$. 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, but our discussion refers to the case where all variables are continuous. Then, (CCPR) can be formulated as follows:

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

In this formulation, M^i and $\bar{M}^i$ are vectors of large enough constants so that for all feasible x, there exist y^i, $\bar{y}^i$ such that $A^i x + H^i y^i \le b^i + M^i$, and $\bar{A}^i x + \bar{H}^i \bar{y}^i \le \bar{b}^i + \bar{M}^i$. In other words, the big-M values deactivate sets of constraints based on the values of the variables z_i. Because of assumption (A3), we are guaranteed that there exist finite values M^i, $\bar{M}^i$ with this property. The constraint $\sum_{i=1}^{k} p_i z_i \le \alpha$ ensures that the probability associated with unsatisfied scenarios is at most α, thereby modeling the chance constraint. 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 Sect. 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 [6] for (CCPR), and [17] for (CCP).

This paper studies the case where the scenario subproblems are convex sets described as follows:

$$
\begin{aligned}
C_{x,y}(w^i) &= \{(x, y^i) : g^i_j(x, y^i) \le 0,\ j = 1, \ldots, m_i\} \\
\bar{C}_{x,y}(w^i) &= \{(x, \bar{y}^i) : \bar{g}^i_j(x, \bar{y}^i) \le 0,\ j = 1, \ldots, \bar{m}_i\} \\
C_x &= \mathrm{Proj}_x C_{x,y}(w^i) \\
\bar{C}_x &= \mathrm{Proj}_x \bar{C}_{x,y}(w^i).
\end{aligned}
$$

In the above expression and in the rest of this paper, Proj_x indicates the projection of a set onto the space of the x variables, i.e., $\mathrm{Proj}_x C_{x,y}(w^i) := \{x : \exists (x, y) \in C_{x,y}(w^i)\}$. 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 $X = \{x : Ax \le b\}$, but this does not affect our development and the generalization to the case where X is a general convex set, possibly with the addition of integrality requirements, is straightforward. If all the $C_x(w^i)$ have the same recession cone, we can write a *mixed-integer nonlinear programming* (MINLP) model for (CCP) as follows:

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

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. Using the notation introduced in the previous section, $\hat{x}$ is feasible for scenario i if $\hat{x} \in C_x(w^i)$, and is feasible for the recovery-mode scenario i if $\hat{x} \in \bar{C}_x(w^i)$. Since $C_x(w^i)$, $\bar{C}_x(w^i)$ have the same structure, from now on our discussion focuses on the sets $C_x(w^i)$, but clearly it also applies to the sets $\bar{C}_x(w^i)$.

The basic idea we exploit is to generate solutions for the master problem, 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 Sect. 2.2; the relationship with generalized Benders decomposition [20] is discussed in Sect. 2.4.

Instead of applying a pure Benders decomposition approach to (2), we use a Branch-and-Cut approach adapted from [17], 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 (to be satisfied, in the nonlinear case, with tolerance ϵ_c), combined with the variables z. A basic version of the algorithm is given by Algorithm 1.

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 steps 7 and 11 of Algorithm 1, as is done in generalized Benders decomposition [20]. However, linear inequalities have several computational advantages, and allow for an easy lifting procedure of the coefficients on the z variables following [17]. We will revisit this topic in Sect. 2.4 from a theoretical point of view, whereas a discussion of lifting on the z variables is given in Sect. 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 [26] and the Hybrid approach of [27], 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 [26,27] 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 directly 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

Algorithm 1 `Decomposition Algorithm`

Set up a master problem of the form

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

repeat
  Apply Branch and Bound on (3), i.e.: select an active node; solve continuous relaxation; if the node is not pruned, branch.
  At every node of the tree with solution $(\hat{x}, \hat{z})$, $\hat{z} \in \{0,1\}^k$, do the following:
  for $i = 1, \ldots, k$ **do**
    for $\hat{z}_i = 0$ and $\nexists \bar{x} \in C_x(w^i) : \|\bar{x} - \hat{x}\| \le \varepsilon_c$ **do**
      Separate $\hat{x}$ from $C_x(w^i)$ via an inequality $\gamma x \le \beta$.
      Add inequality $\gamma x \le \beta + M z_i$ to the master problem (3).
    end for
    for $\hat{z}_i = 1$ and $\nexists \bar{x} \in \bar{C}_x(w^i) : \|\bar{x} - \hat{x}\| \le \varepsilon_c$ **do**
      Separate $\hat{x}$ from $\bar{C}_x(w^i)$ via an inequality $\gamma x \le \beta$.
      Add inequality $\gamma x \le \beta + M(1 - z_i)$ to the master problem (3).
    end for
  end for
  If $(\hat{x}, \hat{z})$ is still feasible, update incumbent (lower bound).
until no more nodes to be explored

to obtain outer approximation cuts. It can be easily seen that the sequence of points generated by the algorithm is not necessarily the same.

We now come back to the discussion of our assumption (A2) as compared to (B2) in [6], which, at the intersection of the normal-mode and recovery problems, assumes the opposite. We claim that both assumptions are reasonable, depending on the application. Computational experiments in [6] use a model in which the recovery problems are equal to the normal-mode problems with additional actions that penalize constraint violations. Since the feasible region of each recovery problem is a superset of the feasible region of the corresponding normal-mode problem, the objective function value of each recovery problem is at least as large as that of the normal-mode problem, for fixed first-stage variables. In this case, (B2) in [6] holds. In our paper, (A2) imposes that, at the intersection of the normal and recovery problems, the objective function value of the latter problem is not larger than that of the former, for fixed first-stage variables. This situation occurs when costs are incurred in the recovery mode, that can be avoided in the normal-mode problem. For example, in a hydro scheduling model the normal-mode problem may correspond to meeting the demand by power production, and the recovery problem may correspond to not meeting the demand by power production, and requires paying a penalty or activating a contract for buying electricity from an external supplier, incurring an additional cost. This is allowed by our assumption (A2), but not by (B2) in [6]. We remark that even if (A2) implies that we always choose to operate in normal mode if the first-stage variables x allow, this does not necessarily mean that the chance constraint is trivially satisfied: since we optimize the total profit in (CCPR), it is easy to see that it may still be profitable to violate some scenarios, thereby operating them in recovery mode, to obtain larger profits in satisfied scenarios.

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

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

where $g_j(x, y)$ is convex for all j. [Note that for scenario i, system (4) would have been $C_{x,y}(w^i) = \{(x, y^i) : g^i_j(x, y^i) \leq 0, \ j = 1, \ldots, m_i\}$, 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 [20]. 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 Sect. 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 $C_x = Proj_x C_{x,y}$ is convex, and $\hat{x} \notin C_x$. Let $(\bar{x}, \bar{y})$ be the optimal solution to (PROJ) with positive 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 ℓ_2-norm.

Proof Let $\ell^* > 0$ the optimal objective function value of (PROJ). 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., [28, 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 $\|\hat{x} - \bar{x}\|^2 = 2\ell^* > 0$ by hypothesis. To show that it is the deepest valid cut, notice that $\|\hat{x} - \bar{x}\|_x = \sqrt{2\ell^*}$. Any cut that cuts $\hat{x}$ by more than $\sqrt{2\ell^*}$ in Euclidean distance computed in the x space would cut $\bar{x}$ off, forsaking validity. □

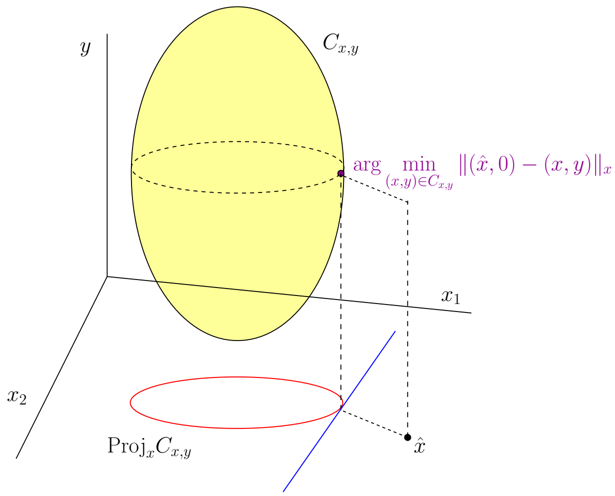

Fig. 1 Separating hyperplane

We remark that the convexity assumption about C_x in Theorem 1 is always verified whenever $C_{x,y}$ is convex. 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 [21] and has proven to be an extremely useful tool in mixed-integer convex programming [27,29,30]. 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 Proposition 1 in Sect. 2.4 for a more precise characterization of the separating inequality when constraint qualification holds) or that $C_{x,y}$ is defined by a finite number of inequalities. 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 [31, Sect. 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 $(n+1)$-

dimensional hyperplane that intersects the boundary of the epigraph of G, and does not cut its interior. As in [31], we denote such hyperplane by $r = p(x; \hat{x})$, and call it an extreme support to the graph of $G(x)$ at $\hat{x}$. Given a cost vector c, if $\|\nabla_x p(x; \hat{x})\|$ is bounded by a constant and $\hat{x}_h$ defines a sequence of points such that $c\hat{x}_h = \min\{cx | x \in X_h\}$, $h = 0, 1, \ldots,$ where $X_0 = X$ and $X_h = X_{h-1} \cap \{x | p(x, \hat{x}_{h-1}) \leq 0\}$, then the sequence $\{\hat{x}_h\}$ contains a subsequence that converges to a point ξ in X with $G(\xi) \leq 0$.

Let us define $\text{dist}(C_x(w^i), C_x(w^j)) := \min\{\|x^i - x^j\| : x^i \in C_x(w^i), x^j \in C_x(w^j)\}$ and $\text{dist}(x^i, C_x(w^j)) := \min\{\|x^i - x^j\| : x^j \in C_x(w^j)\}$. We are ready to prove the following theorem.

Theorem 2 *Consider a problem of the form (CCPR), where X is compact and convex, $C_x(w)$ and $\bar{C}_x(w)$ are closed and convex sets for all $w = w^1, \ldots, w^k$, and assumptions $(A1)-(A3)$ are satisfied. Given any $\varepsilon_c > 0$, assume that for every $i, j = 1, \ldots, k, i \neq j : C_x(w^i) \cap C_x(w^j) = \emptyset$, we have dist$(C_x(w^i), C_x(w^j)) > 2\varepsilon_c$. Similarly, if $\bar{C}_x(w^i) \cap C_x(w^j) = \emptyset$ then dist$(\bar{C}_x(w^i), C_x(w^j)) > 2\varepsilon_c$, and if $\bar{C}_x(w^i) \cap \bar{C}_x(w^j) = \emptyset$ then dist$(\bar{C}_x(w^i), \bar{C}_x(w^j)) > 2\varepsilon_c$. Assume further that we have an algorithm to solve (PROJ) to optimality. Then, the following holds:*

(i) *If (CCPR) admits an optimal solution, after a finite number of iterations Algorithm 1 returns $\hat{x}, \hat{z}$ such that:*

$$
\begin{array}{lrl}
(\hat{x}, \hat{z}) \in \arg\max & & cx \\
\qquad\qquad s.t.: & & x \in X \\
i = 1, \ldots, k \;\; z_i = 0 & \Rightarrow dist(\hat{x}, C_x(w^i)) & \leq \varepsilon_c \\
i = 1, \ldots, k \;\; z_i = 1 & \Rightarrow dist(\hat{x}, \bar{C}_x(w^i)) & \leq \varepsilon_c \\
 & \displaystyle\sum_{i=1}^{k} p_i z_i & \leq \alpha \\
i = 1, \ldots, k & z_i & \in \{0, 1\}.
\end{array}
\tag{5}
$$

(ii) *If (CCPR) is infeasible, after a finite number of iterations Algorithm 1 returns "infeasible".*

Proof It is clear that branching on the z variables does not exclude any feasible solution, and the inequalities added to the master problem (3) are valid. We therefore need to show two facts: first, that the algorithm terminates after a finite number of iterations; and second, that the algorithm correctly returns "infeasible" or an optimal solution [as defined in conditions (5)].

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. We show that for any binary assignment of the z variables, the separation routine stops generating cuts after a finite number of iterations. Note that termination of the separation algorithm is guaranteed in the setting of [17] 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 finite in number. In the context of the present paper, it must be proven.

Claim for every $\hat{z} \in \{0,1\}^k$, the separation routine of Theorem 1 requires a finite number of inequalities to converge to a point $\hat{x}$ such that for every $\hat{z}_i = 0$ we have $\mathrm{dist}(\hat{x}, C_x(w^i)) \leq \varepsilon_c$, and for every $\hat{z}_i = 1$ we have $\mathrm{dist}(\hat{x}, \bar{C}_x(w^i)) \leq \varepsilon_c$.

Proof of the claim 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 apply the convergence result of Theorem [31, Sect. 2] as follows. Let X be the set defined by the feasible region of the master problem, and define $G(x) = \min_{\tilde{x} \in C_x} \|x - \tilde{x}\|$, i.e., 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 its gradient is bounded by definition of $G(x)$. 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

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

Then, writing the last occurrence of $\hat{x}$ in the expression above as $\bar{x} + \hat{x} - \bar{x}$, and performing algebraic manipulations, we see 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 $\hat{x}$. Since the inequalities that we generate satisfy the conditions of Theorem [31, Sect. 2], we can apply that result to show that there exists a (sub)sequence of points $\hat{x}_h$ converging to a point ξ in X, $G(\xi) \leq 0$, i.e., $\xi \in C_x$. By definition of convergence, for every ε_c there exists an integer v such that after v inequalities $\|\hat{x} - \xi\| \leq \varepsilon_c$. Because this is true for any $C_x(w^i)$, $\bar{C}_x(w^i)$, it is also true for the intersection $\bigcap_{i:\hat{z}_i=0} C_x(w^i) \cap \bigcap_{i:\hat{z}_i=1} \bar{C}_x(w^i)$, after a number of iterations that is determined by the last set to satisfy the convergence condition. This concludes the proof of the claim.

Applying the same arguments as those in the proof of convergence in [17], the claim then ensures finite convergence of Algorithm 1. More specifically, since X is a compact, (CCPR) is either infeasible or admits an optimum. We analyze the two cases separately:

(i) If (CCPR) is infeasible, there is no assignment of the z variables for which the corresponding active sets $C_x(w^i)$, $\bar{C}_x(w^i)$ have nonempty intersection. The case $X = \emptyset$ is trivial so assume for the sake of contradiction that $X \neq \emptyset$ and Algorithm 1 terminates returning a solution $\hat{x}, \hat{z}$. Then, $\hat{x}$ has distance $\leq \varepsilon_c$ from the feasible set of a sufficient number of scenarios, i.e., enough to satisfy the chance constraint as modeled by $\sum_{i=1}^k p_i z_i \leq \alpha$. Since (CCPR) is infeasible, there must exist indices $i, j, i \neq j$ such that $\hat{z}_i = 0$, $\hat{z}_j = 0$ and $C_x(w^i) \cap C_x(w^j) = \emptyset$ (respectively, $\hat{z}_i = 1$, $\hat{z}_j = 0$ and $\bar{C}_x(w^i) \cap C_x(w^j) = \emptyset$, or $\hat{z}_i = 1$, $\hat{z}_j = 1$ and $\bar{C}_x(w^i) \cap \bar{C}_x(w^j) = \emptyset$). However, by assumption, the sets $C_x(w^i)$, $C_x(w^j)$ [resp., $\bar{C}_x(w^i)$, $C_x(w^j)$, or $C_x(w^i)$, $C_x(w^j)$] have distance strictly greater than $2\varepsilon_c$. This implies that there exists no point with distance at most ε_c from all the satisfied scenarios. By the preceding claim, the sequence of points explored by Algorithm 1 must converge to a point with distance at most ε_c from $C_x(w^i)\ \forall i : \hat{z}_i = 0$ and $\bar{C}_x(w^i)\ \forall i : \hat{z}_i = 1$ within a finite number of iterations. But since no such point exists, the cuts added to the master problem must yield an infeasible subsystem within a finite number of iterations: this is a contradiction.

(ii) If (CCPR) admits an optimal solution, there exists an assignment $\hat{z}$ for the z variables for which the correspondings active sets $C_x(w^i)$, $\bar{C}_x(w^i)$ have nonempty intersection. Therefore, for $z = \hat{z}$ the set of points with distance at most ε_c from each of the feasible sets of active scenarios is nonempty, implying that the feasible region of (5) is nonempty. By the above claim, after a finite number of iterations we converge to a point within the feasible region of (5). Standard arguments for the correctness of Branch-and-Bound guarantee that we determine an arg max of such problem in finite time. □

Besides (A1)–(A3), Theorem 2 imposes the additional "separation assumption" that for all $i, j = 1, \dots, k : C_x(w^i) \cap C_x(w^j) = \emptyset$ then $\operatorname{dist}(C_x(w^i), C_x(w^j)) > 2\varepsilon_c$; and similarly for the sets $\bar{C}_x(w^i)$. This assumption is necessary, as can be seen from the following pathological example:

$$\begin{array}{rrl} \min & |x| & \\ \text{s.t.:} & x & \in [-100, 100] \\ z_1 = 0 \ \Rightarrow & x & \in [-100, -\varepsilon_c] \\ z_2 = 0 \ \Rightarrow & x & \in [+\varepsilon_c, 100] \\ & \frac{1}{2} z_1 + \frac{1}{2} z_2 & \leq \frac{1}{4} \\ & z_1, z_2 & \in \{0, 1\}. \end{array} \tag{6}$$

In this problem, which has the form (CCPR), both scenarios have to be satisfied. The problem is infeasible for any $\varepsilon_c > 0$, it does not satisfy the "separation assumption", and Algorithm 1 would (incorrectly) return $\hat{x} = 0$ as a solution.

We remark that the statement of Theorem 2 is somewhat weaker than showing convergence to a point with distance at most ε_c from an optimal solution of (CCPR). Indeed, in theory it is possible that Algorithm 1 returns a point with distance at most ε_c from each active scenario set, but the distance from the feasible region of (CCPR), which is the intersection of several scenario sets, is arbitrarily larger. This is mostly a theoretical issue: such a situation can happen only if the problem is extremely ill-conditioned. From a computational point of view, a small convergence tolerance ε_c typically ensures satisfactory results. To ensure stronger convergence properties, we can modify Algorithm 1 inserting, after line 13, the following additional separation condition:

if $\nexists \bar{x} \in \left(\bigcap_{i:\hat{z}_i=0} C_x(w^i) \cap \bigcap_{i:\hat{z}_i=1} \bar{C}_x(w^i) \right) : \|\bar{x} - \hat{x}\| \leq \varepsilon_c$ **then**

Separate $\hat{x}$ from $\left(\bigcap_{i:\hat{z}_i=0} C_x(w^i) \cap \bigcap_{i:\hat{z}_i=1} \bar{C}_x(w^i) \right)$ via an inequality $\gamma x \leq \beta$.

Add inequality $\gamma x \leq \beta + \sum_{i:\hat{z}_i=0} M z_i + \sum_{i:\hat{z}_i=1} M(1 - z_i)$ to the master problem (3).

end if

Notice that these additional steps only affect the algorithm in case $\hat{x}$ is close to the feasible set of each of the active scenarios, but is not close to their intersection; furthermore, the generated cuts are only active for a single realization of the z vector. It is straightforward to modify the proof of Theorem 2 using the additional steps to prove convergence to a point with distance at most ε_c from the feasible region of (CCPR). However, we implement Algorithm 1 as initially described because the extra steps are unlikely to bring any practical benefit, therefore we do not pursue this extension in detail.

2.4 Comparison with generalized Benders cuts

This section investigates the relationship between the separation approach we advocate and generalized Benders decomposition [20], 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 Sect. 1. The result in [20] 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

$$L(x,\mu) = \min_{(\tilde{x},\tilde{y})\in C_{x,y}:\tilde{x}=x} \mu^T g(\tilde{x},\tilde{y}) \tag{7}$$

can be easily computed for all $x \in X, \mu \in \mathbb{R}^m, \mu \geq 0$. 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 constraint functions are separable in x and y (see, e.g., [32,33]), 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 [20] 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 [34]. Of course, one could simply linearize a generalized Benders cut: we show that this is in fact exactly what is happening.

Proposition 1 *Assume that a constraint qualification holds, and for a given $\hat{x} \notin C_x$ let $(\bar{x}, \bar{y})$ be the optimal solution to (PROJ), $\bar{\mu}$ 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 $\bar{\mu}$.

Proof A generalized Benders cut has the form $L(x, \mu) \leq 0$, where $L(x, \mu)$ is defined as in (7). Since $\bar{\mu}$ is the vector of optimal KKT multipliers and $\hat{x} \notin C_x$, we have

$L(\hat{x}, \bar{\mu}) > 0$ and $\bar{\mu} \geq 0$; see [20]. Using KKT conditions, we see that the hyperplane $\left(\sum_{j \in I} \bar{\mu}_j \nabla g_j(\bar{x}, \bar{y})\right)^T ((x, y) - (\bar{x}, \bar{y})) = (\hat{x} - \bar{x})^T (x - \bar{x}) \leq 0$ is supporting for $\sum_{j \in I} \bar{\mu}_j g_j(x, y)$ at $(\bar{x}, \bar{y})$, so $\sum_{j \in I} \bar{\mu}_j g_j(x, y) \geq (\hat{x} - \bar{x})^T (x - \bar{x})$ for all $(x, y) \in C_{x,y}$ because the left-hand side expression is convex. Let $\ell^* > 0$ be the optimal objective function value of (PROJ). It follows that

$$\min_y \sum_{j \in I} \bar{\mu}_j g_j(\hat{x}, y) \geq (\hat{x} - \bar{x})^T (\hat{x} - \bar{x}) = 2\ell^* > 0.$$

This shows that the multipliers $\bar{\mu}$ yield a violated generalized Benders cut. Furthermore, $\sum_{j \in I} \bar{\mu}_j g_j(\bar{x}, y) \geq (\hat{x} - \bar{x})^T (\bar{x} - \bar{x}) = 0$ for all y, and $\sum_{j \in I} \bar{\mu}_j g_j(\bar{x}, \bar{y}) = 0$ by complementary slackness, hence $\bar{y} \in \arg\min_y \sum_{j \in I} \bar{\mu}_j g_j(\bar{x}, y)$. It follows that $(\hat{x} - \bar{x})^T (x - \bar{x})$ is the tangent plane to $L(x, \bar{\mu})$ 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 cut, every valid linear inequality for C_x is a linearization of a generalized Benders cut. As remarked in [26, Sect. 3.1], 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 $\bar{\mu} \geq 0$ be any vector of dual variables that gives rise to a violated generalized Benders cut, i.e., $L(x, \bar{\mu}) > 0$. Since the expression $L(x, \bar{\mu}) \leq 0$ is a convex function when taken as a function of x [it is the minimum over $\tilde{y}$ of a nonnegative linear combination of convex functions, see (7)], any tangent hyperplane is a valid inequality. The approach of [20] requires the dual variables $\bar{\mu}$ only, but in order to compute a tangent hyperplane, we additionally need a point at which the linearization is obtained. To this end, [26] proposes a hierarchy of points, where the weakest one is analogous to the ECP method [35] 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 at 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.

Example 1 Consider the set $C_{x,y}$ defined by the following constraints:

$$\begin{aligned} x^2 + x + y^2 - y &\leq 0.5 \\ x^2 + xy + y^2 &\leq 1. \end{aligned}$$

The feasible region is depicted in Fig. 2. Suppose that $\hat{x} = 1$ is the value of x provided by the master problem. Note that $\hat{x} \notin C_x$: there exists no y that satisfies $x^2 + x + y^2 - y \leq 0.5$ for $\hat{x} = 1$. The method we propose solves (PROJ) to find the point

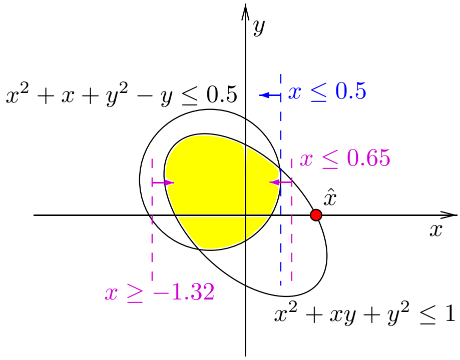

Fig. 2 Example of generalized Benders cuts, and their linearization

(0.5, 0.5) and the dual variables $\bar{\mu}_1 = 0.5, \bar{\mu}_2 = 0$. Linearizing the cut obtained with these dual variables at the point (0.5, 0.5) yields the cut $x \le 0.5$. However, a generalized Benders cut could be derived from a different vector of the dual variables, e.g., $\bar{\mu}_1 = 1, \bar{\mu}_2 = 1$. These yield the cut $L(x, \bar{\mu}) \le 0$ with expression:

$$\min_y (2x^2 + x + 2y^2 - y + xy - 1.5) \le 0.$$

Note that the minimum on the left-hand side cannot be computed independently of x, because the problem at hand is not separable. Nonetheless, for this toy problem one can easily find that the minimizer is $y = (1-x)/4$. Given this value for y, substituting into the expression above yields the nonlinear cut:

$$15x^2/8 + 5x/4 - 13/8 \le 0.$$

This cut is not only nonlinear, but it also cuts $\hat{x}$ by a smaller among than the previously computed cut: one can easily check that this generalized Benders cut is equivalent to $-1.3222 \le x \le 0.6555$ in the x space, which cuts off $\hat{x}$ by 0.3445, whereas the cut computed with the dual variables $\bar{\mu}_1 = 0.5, \bar{\mu}_2 = 0$ cuts off $\hat{x}$ by 0.5, see Fig. 2. □

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 [36]). 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 Sect. 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., [37]. More recently, methodologies that can take into account the uncertainty in the model have appeared, such as [22–24]. There has been little work on chance-constrained formulation for the mid-term hydro scheduling problem: some notable papers are [38,39]. The related unit commitment problem, widely studied in the literature, has sometimes been tackled with chance-constrained approaches, e.g., [40,41], but even in the case of unit commitment, chance-constrained optimization approaches are the least commonly used in the literature, due to their difficulty [42, 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 [25], 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 α-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 φ_x is a random variable that measures the profit, we define the loss as $L_x = -\varphi_x$. The α-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_α is the α-quantile. In other words, our choice of objective function, i.e., maximizing the α-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 $t = 1, \dots, T$), 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 $h = 1, \dots, n$. 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 program:

$$\max \sum_{t=1}^{T} \pi_t(e_t)e_t \tag{8}$$

$$\text{s.t.: } w_{(t-1)h} - x_{th} + f_{th} \geq w_{th} \quad t = 1, \ldots, T, h = 1, \ldots, n \tag{9}$$

$$0 \leq x_{th} \leq u_{th} \quad t = 1, \ldots, T, h = 1, \ldots, n \tag{10}$$

$$q_{th} \leq w_{th} \leq Q_{th} \quad t = 1, \ldots, T, h = 1, \ldots, n \tag{11}$$

$$e_{th} \leq g_h(w_{th}, x_{th}) \quad t = 1, \ldots, T, h = 1, \ldots, n \tag{12}$$

$$d_t \leq e_t \leq m_t \quad t = 1, \ldots, T \tag{13}$$

$$e_t = \sum_{h=1}^{n} e_{th} \quad t = 1, \ldots, T. \tag{14}$$

The objective function (8) maximizes the profit obtained by selling the transformed energy. Constraint (9) 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 (10) and (11) impose lower and upper bounds on the quantity of water used for transforming energy and on the water levels in the reservoirs, respectively. Constraints (12) define the relation between the released water and the obtained electrical energy at a specific plant h. Finally, (13) 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, [22] assumes 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 [22] 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 [22]. 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 full optimization 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 (9)–(13). In addition, we link the profit in each scenario to an overall measure of profit in the master problem by introducing a master variable ψ that is maximized, and defining the following additional constraints:

$$\psi \leq \sum_{t=1}^{T} \pi_t^i(e_t) e_t \quad i = 1, \ldots, k. \tag{15}$$

Hence, a specific scenario is satisfied given the decision variables in the master (energy obtained in the first time period, and measure of profit ψ) if not only constraints (9)–(13) can be satisfied for subsequent time periods, but also the total profit for the scenario is not smaller than ψ. 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 α-quantile of the profit.

Following [22], 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:

$$\max \psi \tag{16}$$

$$\text{s.t.:} \sum_{i=1,\ldots,k} z_i \leq p, \tag{17}$$

$$x_{th}^i = x_{th}^r,\ i = 1, \ldots, k-1,\ r = i+1, \ldots, k,\ t \leq \tau(i, r),\ h = 1, \ldots, n \tag{18}$$

$$0 \leq x_{th}^i \leq u_{th} \qquad t = 1, \ldots, T-1,\ i = 1, \ldots, k,\ h = 1, \ldots, n \tag{19}$$

$$z_i \in \{0, 1\} \qquad i = 1, \ldots, k. \tag{20}$$

where (17) is the joint probability constraint, constraints (18) express nonanticipativity, constraints (19) impose bounds on the quantity of water released. We remark that in practice we do not explicitly write constraints (18), 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 functions

The transformation of the water potential energy into electrical energy is described in terms of nonlinear power functions $v_h(w, \dot{x})$ 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 θ_t. Hence, we can write

$$g_h(w_{th}, x_{th}) = v_h(w_{th}, x_{th}/\theta_t)\theta_t. \tag{21}$$

Several alternatives are proposed in the literature regarding the shape of $v_h(w, \dot{x})$, see e.g., [43–47]. These alternatives depend on the characteristics of each power plant as well as the level of detail that is required by the considered application, and typically must be experimentally evaluated.

A common family of power functions consider power as a polynomial expression of the flow: $v_h = \sigma + \nu x/\theta_t + \mu(x/\theta_t)^2 + \rho(x/\theta_t)^3 + \xi(x/\theta_t)^4 + \cdots$, where the values of the coefficients $\sigma, \nu, \mu, \rho, \xi, \ldots$, 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., [48]). In our computational study we consider two families of power functions having increasing level of detail and yielding increasingly challenging optimization models.

– As a first power function we consider a fourth-degree polynomial with $\nu > 0$; $\mu, \xi < 0$; $\rho = 0$ thus obtaining a concave function. We also assume the water level w to vary between bounds that allow us to consider the function parameters as fixed, and we then disregard the dependence of the power function from water level. We then have:

$$v'_h(x) = \sigma + \nu x/\theta_t + \mu(x/\theta_t)^2 + \xi(x/\theta_t)^4. \tag{22}$$

– As a second power function we consider a quadratic expression of the flow. Since the value of parameters σ, θ, ρ is approximately linear in the water level w, we make this dependence explicit in the power function, thus obtaining:

$$v''_h(w, x) = (w + \eta)[\sigma + \nu x/\theta_t + \mu(x/\theta_t)^2], \tag{23}$$

where η is an additional parameter to be experimentally tuned. Given the value of the parameters used in the experiments, this function is neither convex nor concave.

3.3.2 Demand and price function

The electrical energy produced 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 [22]. A finer description of the market effect of a large power producer can be obtained by using nonincreasing step functions, as in [22]. 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 Sect. 4.3. We now provide more details on the two above alternatives for the cost function.

– Using a nonincreasing linear price-quantity function, the profit-quantity relation in Eq. (8) is expressed by a quadratic concave function of the energy, that is (recall that $e_t = \sum_{h=1}^{n} e_{th}$):

$$\pi_t(e_t)e_t = (\pi_{1t}e_t + \pi_{0t})e_t. \tag{24}$$

– Using a step price-quantity function with two steps, the profit-quantity relation in Eq. (8) is expressed by:

$$\pi_t(e_t)e_t \leq \pi_{1t}e_t \quad t = 1, \ldots, T \tag{25}$$
$$\pi_t(e_t)e_t \leq \pi_{2t}e_t + (\pi_{1t} - \pi_{2t})m_{1t}y_t \quad t = 1, \ldots, T \tag{26}$$
$$e_t \leq m_{1t}y_t + m_{2t}(1 - y_t) \quad t = 1, \ldots, T \tag{27}$$
$$y_t \in \{0, 1\}, \quad t = 1, \ldots, T \tag{28}$$

where m_{1t} is the maximum amount of energy that can be sold at price π_{1t} in period t, m_{2t} ($> m_{1t}$) is the maximum (overall) amount of energy that can be sold at price π_{2t} ($< \pi_{1t}$) in period t, and y_t is a binary variable indicating whether the amount of energy sold is $\leq m_{1t}$ ($y_t = 1$) 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 [22], which describes a hydro system configuration comprising 10 major hydroplants of the Greek power system, for a production capacity of 2720 MW. As in [22], 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 [22] for details. The first time period is deterministic, as previously mentioned; in [22], 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 with the Branch-and-Cut algorithm when solving decomposable chance-constrained problems. We first test the algorithm on the instances discussed in Sect. 3 by considering the concave power function (22) and the first formulation for the price function presented in Sect. 3.3.2. This yields convex subproblems and a quadratic relationship between profit and sold energy described by (24). 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 Sect. 4.3 we experiment results on the instances with the nonconvex power function (23) and the step price function formulation (25)–(28), for which we apply our approach as a heuristic 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 [49] using the interface provided by BONMIN [27]. 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 a 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 $\gamma x \leq \beta$ separating $\hat{x}$ from the feasible region of scenario j, as explained in Sect. 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 $\gamma x \leq \beta + M z_j$. We compute the value for the M coefficient as: $M = \sum_{l:\gamma_l>0} \gamma_l u_l - \beta$, where l denotes the index of the x variables in the cut and u_l is the associated upper bound in the master problem;

lifting 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 [17].

In the second case, for every i we first compute the minimum value β_i that makes the inequality valid for the corresponding scenario w^i, solving the optimization problem:

$$\beta_i = \max\{\gamma x | x \in X \cap C_x(w^i)\}. \tag{29}$$

Assuming the β_i values for $i = 1, \ldots, k$ 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 [17, Lemma 1]):

$$\gamma x + (\beta_i - \beta_{p+1}) z_i \leq \beta_i, \quad i = 1, \ldots, p. \tag{30}$$

From this basic set of inequalities, one could obtain stronger star inequalities (see [50]). The basic idea is that, given an ordered subset $T = \{t_1, t_2, \ldots, t_l\}$ of $\{1, \ldots, p\}$, one can derive the following star inequality, where $\beta_{t_{l+1}} = \beta_{p+1}$ (see [17] for further details):

$$\gamma x + \sum_{i=1}^{l} (\beta_{t_i} - \beta_{t_{i+1}}) z_i \leq \beta_{t_1}. \tag{31}$$

Since the star inequalities (31) 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 (30) associated with the first t_i in the ordering such that $z_{t_i} = 0$. Thus, in our implementation we add precisely the inequalities (30).

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 the (PROJ) problem 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-feasible solutions in the Branch-and-Cut search 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-feasible solutions in the Branch-and-Cut search 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:

sep1 Separation is performed at integer-feasible solutions in the Branch-and-Cut search until the first violated scenario i having associated variable $z_i = 0$ is found.
sepFrac We attempt to separate cuts at fractional solutions (i.e., LP solutions for the nodes of the Branch-and-Cut search) 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 seen in the next section, the vast majority of the CPU time is already spent in the solution of the nonlinear separation subproblems, therefore increasing in the number of separation rounds is an issue.

Concerning the large MINLP (2), we solve with BONMIN and the embedded nonlinear solver IPOPT 3.12. 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 [22] 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 $\{1, 2, 5, 7, 10\}$. In each subproblem, the number of variables is $9n + 5$ and the number of constraints is $6n + 5$, where n is the number of hydroplants. For each configuration, we can specify the robustness of the solution: we consider values of the probability α 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 Sect. 4.4 we additionally report results for $\alpha = 0.05$, but they are not included here as they do not provide further insight. Furthermore, for 10 hydroplants and all values of α, we considered four simplified scenario trees that contain 30, 48, 60 or 72 scenarios, and four larger scenario trees that contain 150, 180, 225 or 270 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.

We performed preliminary computational experiments on four variants of the Branch-and-Cut algorithm that are obtained by combining the separation procedures **sepAll** and **sepGroup** with the **bigM** and **lifting** procedures to add cuts to the master problem. According to preliminary experiments, the **sepAll-bigM** variant of our Branch-and-Cut algorithm is the fastest version on average, and we take it as our reference.

We briefly discuss its performance, especially compared to the other variants. The number of Branch-and-Bound nodes for **sepAll-bigM** is fairly small (about 500 on average on the whole set of instances), and almost all the computation time is spent solving NLPs (on average, 125,000 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 (16)–(20) 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 **lifting** cuts, we note that given a cut $\gamma x \leq \beta$ 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. In our experiments this is not the case: although the number of Branch-and-Bound nodes is slightly reduced, the average number of separation iterations is

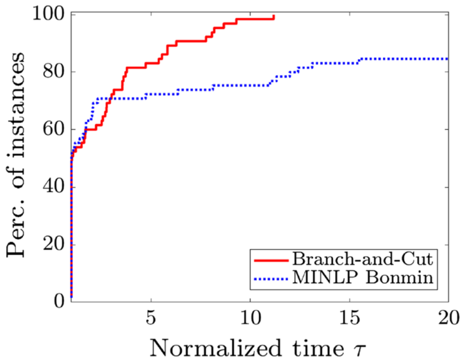

Fig. 3 Performance profiles for 65 instances (linear price function)

of similar magnitude. As a consequence, **sepAll-lifting** and **sepGroup-lifting** solve approximately 10 times 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 (29) for a given scenario i and a given hyperplane γx, only a subset of the variables with nonzero coefficient in γ 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 [51], 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), is also evaluated on all 65 problem instances. The time limit for BONMIN is set to 10 h. In Fig. 3 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 hits the time limit in 10 cases. In addition, the profiles clearly show the 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 [26] 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 1 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, as in the original application from [22]. The table reports the number of hydroplants and the level of risk α in the first two columns. Subsequent columns report the results obtained by the Branch-and-Cut algorithm, namely: the total number of Branch-and-Bound nodes (third column), number of Branch-and-Bound nodes at which separation is performed (fourth column), total computing time and

Table 1 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)

Plants	α	Branch and cut						BONMIN	
		B&B nodes	Sep. nodes	Time (s)	% time NLP	NLP solved	Added cuts	Time (s)	Nodes
1	0.1	19	3	31.7	99.9	3330	379	32.0	47
1	0.2	47	4	29.2	99.8	3187	459	28.6	35
1	0.3	143	4	26.6	99.7	2925	474	41.9	99
1	0.4	48	6	38.7	99.7	4275	593	31.8	71
1	0.5	16	3	30.2	99.7	3330	461	42.7	134
2	0.1	9	2	44.5	99.8	3998	621	76.8	65
2	0.2	181	3	24.3	99.1	2250	592	50.9	35
2	0.3	189	9	34.8	99.2	3119	636	417.6	1107
2	0.4	90	7	56.0	99.7	4954	774	265.0	700
2	0.5	25	4	51.1	99.7	4597	731	87.8	132
5	0.1	62	12	302.5	99.4	22,733	2446	177.4	65
5	0.2	96	5	238.1	99.0	17,850	2481	2953.4	3669
5	0.3	97	10	214.6	98.9	16,187	2441	131.4	53
5	0.4	172	12	180.7	98.8	13,678	2061	1986.3	2719
5	0.5	336	9	190.2	98.5	14,386	2251	2494.0	3304
7	0.1	46	4	469.6	99.0	31,542	4168	155.6	17
7	0.2	246	8	586.1	97.8	39,566	5425	6636.2	6053
7	0.3	337	16	567.7	97.4	38,283	5448	181.0	53
7	0.4	313	10	516.2	98.1	34,929	4831	204.8	71
7	0.5	229	10	356.1	97.0	23,771	4424	5510.1	5265
10	0.1	68	8	1831.9	97.2	107,332	12,697	227.0	17
10	0.2	280	10	1648.1	95.1	95,272	12,551	296.7	47
10	0.3	190	8	1707.2	94.8	98,675	13,623	292.8	53
10	0.4	957	15	1546.7	91.9	86,207	12,977	3211.2	1873
10	0.5	341	12	1317.5	92.8	73,872	13,803	3054.0	1847

fraction of time spent in the nonlinear separation subproblems (fifth and sixth column respectively), number of nonlinear programs solved (seventh column), number of cuts added to the master problem (eighth column). The last two columns report the performance of BONMIN Branch-and-Bound, indicating the total CPU time and the number of Branch-and-Bound nodes. The Branch-and-Cut algorithm solves all instances in less than 30 min, and the number of Branch-and-Bound nodes is under control, never exceeding 1000. Instances with a smaller number of hydroplants appear easier for the Branch-and-Cut algorithm: the solution time grows approximately linearly with the number of plants, while the level of risk α has little effect. Solution times with BONMIN are much more unpredictable, being faster than our Branch-and-Cut on some instances and slower on others, up to an order of magnitude.

In Table 2 we report results for instances with 10 hydroplants and a growing number of scenarios, up to 270 (as reported in the first column). Solution of the MINLP (2) via BONMIN's Branch-and-Bound algorithm is faster than our Branch-and-Cut when the number of scenarios is small, but the Branch-and-Cut scales much better. With 225 or 270 scenarios, BONMIN times out in most cases while the Branch-and-Cut decomposition approach solves all problems.

4.3 Computational performance with step price function

We now study the performance of the proposed solution methodology when applied as a heuristic to the problem with the nonconvex generation function (23) and the step price function modeled in (25)–(28), which requires binary variables. Our approach consists of two parts. First, we apply the Branch-and-Cut algorithm to solve the continuous relaxation of the chance-constrained problem in a heuristic manner. 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 converges to an integer solution, although not necessarily an optimal one. More specifically, there are two reasons why this approach may generate cuts that are not valid and therefore return a suboptimal solution. The first reason is that the continuous relaxation of (PROJ) is nonconvex when using generation function (23), precisely because of Eq. (23). "Appendix A" shows that the projections $C_x(w^i)$ of the feasible regions of the scenario subproblems are convex for all i, therefore the first assumption of Theorem 2 is satisfied; however, the last assumption is not, because we attempt to solve the nonconvex problem (PROJ) with IPOPT, which may not yield a globally optimal solution. Whenever the solution returned by IPOPT is not the global optimum, the corresponding outer approximation cuts may not be valid. The second reason why the cuts may not be valid is that in the second phase we enforce integrality for the binary variables of the scenario subproblems when solving (PROJ) to generate a cut.

In our experiments, this approach always yields matching (heuristic) lower and upper bounds. We remark that this is not guaranteed in general, but the following features of our application may explain why this is the case:

- 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),

Table 2 Comparison between **sepAll-bigM** and BONMIN Branch and Bound on configurations with 10 hydroplants and 30, 48, 72, 90, 150, 180, 225, and 270 scenarios (linear price function)

# Scen.	α	Branch and cut						BONMIN	
		B&B nodes	Sep. nodes	Time (s)	% time NLP	NLP solved	Added cuts	Time (s)	Nodes
30	0.1	1	1	297.3	98.2	17,815	3995	26.6	5
30	0.2	11	3	284.4	97.5	16,853	4010	32.8	11
30	0.3	1	1	267.2	98.0	16,052	3103	32.6	17
30	0.4	16	2	194.5	98.6	11,927	2735	33.3	26
30	0.5	9	2	226.0	97.2	13,530	3534	29.1	19
48	0.1	15	3	577.2	97.5	34,054	6595	152.3	95
48	0.2	44	8	819.0	94.8	46,983	9353	317.4	209
48	0.3	110	7	543.7	92.6	30,417	7200	196.0	95
48	0.4	216	10	617.4	94.6	34,820	7361	570.5	323
48	0.5	1	1	92.3	99.1	5639	1691	61.2	1
60	0.1	50	7	1168.8	97.7	68,855	8384	125.7	615
60	0.2	89	5	786.3	96.0	45,850	8598	125.2	891
60	0.3	88	8	788.3	96.3	45,806	7773	285.5	746
60	0.4	61	9	776.2	96.6	45,319	8211	214.4	769
60	0.5	1	1	86.3	99.4	5280	1216	94.0	329
72	0.1	34	4	924.9	97.1	54,121	8822	342.5	143
72	0.2	33	6	1371.1	96.4	79,969	11,403	368.1	143
72	0.3	159	10	1089.6	95.1	63,084	9918	8843.0	5186
72	0.4	151	10	1339.0	94.3	76,634	13,323	8511.7	5681
72	0.5	268	12	1365.0	92.5	77,095	14,932	809.0	631
90	0.1	68	8	1831.9	97.2	107,332	12,697	227.0	17
90	0.2	280	10	1648.1	95.1	95,272	12,551	296.7	47

Table 2 continued

# Scen.	α	Branch and cut						BONMIN	
		B&B nodes	Sep. nodes	Time (s)	% time NLP	NLP solved	Added cuts	Time (s)	Nodes
90	0.3	190	8	1707.2	94.8	98,675	13,623	292.8	53
90	0.4	957	15	1546.7	91.9	86,207	12,977	3211.2	1873
90	0.5	341	12	1317.5	92.8	73,872	13,803	3054.0	1847
150	0.1	170	9	2716.2	97.9	160,852	10,549	761.6	29
150	0.2	203	10	3907.4	98.2	233,591	13,096	825.4	59
150	0.3	976	12	3304.1	94.2	188,631	14,696	928.1	89
150	0.4	1765	22	3941.2	88.4	210,057	23,777	t.l.	6178
150	0.5	1863	39	3465.7	81.4	170,897	23,228	t.l.	14,244
180	0.1	80	6	8334.2	93.7	219,686	16,113	1545.4	35
180	0.2	444	12	3766.2	95.2	163,180	16,413	1683.5	71
180	0.3	885	19	3817.7	93.8	216,272	21,161	t.l.	9707
180	0.4	1440	24	4891.2	92.2	275,392	22,100	9745.0	3485
180	0.5	841	33	3896.1	89.7	211,315	28,547	1330.0	129
225	0.1	473	17	6479.8	97.6	382,444	18,374	7668.2	933
225	0.2	812	28	12,735.0	95.3	672,487	35,216	t.l.	6310
225	0.3	2432	28	7513.1	83.5	366,806	31,810	t.l.	5961
225	0.4	1917	41	7600.3	90.4	415,639	33,955	t.l.	6178
225	0.5	1475	57	6981.1	91.5	385,489	27,993	t.l.	5469
270	0.1	199	7	5352.0	96.7	310,269	18,161	t.l.	5093
270	0.2	1178	45	14,154.4	92.4	786,042	41,227	t.l.	4904
270	0.3	7492	53	13,535.6	72.6	571,031	38,575	t.l.	6103
270	0.4	2413	48	10,988.5	88.0	580,425	40,640	19,073.5	5202
270	0.5	302	8	1637.3	94.0	92,821	12,957	3336.9	268

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 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.

Results for the integer case and a comparison with the BONMIN Branch-and-Bound performance are reported in Table 3. Notice that here BONMIN is used as a heuristic, using the recommended "B-BB" setting for nonconvex problems [52]. We tried the global solver Couenne as well [53], but its performance was worse than BONMIN, always reaching the time limit and not yielding upper bound improvements.

The columns associated with the Branch-and-Cut algorithm include the (heuristic) lower and upper bound computation. On average, computing the upper 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. Our decomposition approach always finds solutions of comparable quality on instances on which BONMIN terminates (the profit of the solution found by Branch-and-Cut are is at most 0.83% lower than BONMIN). For the other instances, Branch-and-Cut takes less than 15 min to find solutions that are up to 5.5% better than those found by BONMIN in 10 h.

4.4 The effect of α 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 generation function (22) and 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 α, 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 α corresponds to the minimum profit that can be achieved with probability $1 - \alpha$, but the solution may be infeasible with probability α. In this section, $\alpha = 0.05$ is included in the comparison besides the α values tested above.

Once the problem is optimally solved for a specific level of risk α, 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 (15) are not]. Figure 5 depicts the inverse distribution function of the

Table 3 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)

# Scen.	α	Branch and cut							MINLP		
		B&B nodes	Sep. nodes	Time (s)	% Time (MI)NLP	(MI)NLP solved	Added cuts	Sol. value	Time (s)	Nodes	Sol. value
30	0.1	2	2	38.4	100	1788	419	674.7	272.2	195	675.8
30	0.2	10	4	131.1	99.8	3993	1065	815.9	218.2	268	815.9
30	0.3	21	6	168.8	99.9	4392	1038	838.9	2139.50	3602	842.6
30	0.4	18	4	248.3	99.7	8543	1541	872.0	2713.20	4137	872.8
30	0.5	8	6	109	99.9	3300	865	1061.9	228.1	229	1062.4
48	0.1	31	5	494.9	99.7	14,482	2319	618.2	10,709.10	6724	622.3
48	0.2	42	6	396.7	99.6	17,485	2625	666.3	t.l.	16,838	657.4
48	0.3	60	11	679.4	99.5	29,203	3747	705.9	t.l.	23,453	715.0
48	0.4	73	11	422.8	99.5	21,396	2753	730.5	t.l.	21,517	730.8
48	0.5	8	3	105	99.8	3143	897	990.9	508.7	372	999.1
60	0.1	4	3	220.2	99.9	10,424	1789	556.6	4508.20	3000	557.8
60	0.2	39	7	475.3	99.7	22,994	3588	638.3	t.l.	23,961	627.3
60	0.3	93	10	708.1	99.4	27,992	4096	682.5	t.l.	23,582	682.5
60	0.4	165	17	660.2	99.3	27,661	3696	727.7	t.l.	20,994	715.0
60	0.5	78	8	132.5	99.8	8515	1631	810.3	t.l.	23,648	792.0
72	0.1	39	7	311.9	99.8	16,260	2884	556.9	t.l.	9584	547.2
72	0.2	118	11	663.9	99.4	34,044	4450	614.1	t.l.	8920	595.3
72	0.3	137	12	814.7	99.4	27,280	3881	661.0	t.l.	10,680	644.1
72	0.4	135	12	701.5	99.5	26,291	3466	686.6	t.l.	12,087	677.2
72	0.5	179	15	655.2	99.3	25,632	3844	734.7	t.l.	20,070	730.8
90	0.1	19	4	266	99.8	12,165	2144	509.3	5890.20	2226	510.0
90	0.2	56	7	476	99.7	20,477	2796	554.8	t.l.	14,775	524.2

Table 3 continued

# Scen.	α	Branch and cut							MINLP		
		B&B nodes	Sep. nodes	Time (s)	% Time (MI)NLP	(MI)NLP solved	Added cuts	Sol. value	Time (s)	Nodes	Sol. value
90	0.3	113	10	622.2	99.3	29,469	4041	618.2	t.l.	14,027	612.9
90	0.4	340	20	903.3	99.2	35,838	4793	671.5	t.l.	13,922	666.2
90	0.5	106	12	647.9	99.3	25,136	3933	727.7	t.l.	15,833	715.6

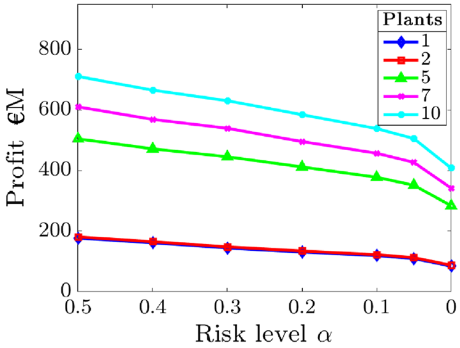

Fig. 4 Trade-off between profit in €M and level of risk: the x-axis reports the risk level α, and the y-axis the corresponding objective function value

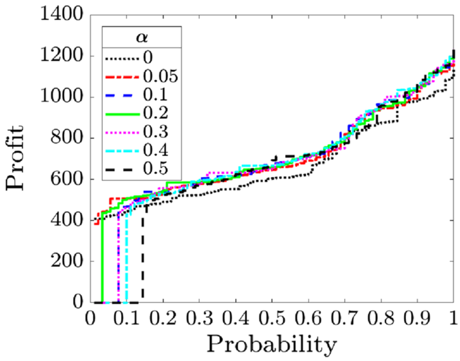

Fig. 5 Inverse distribution function of the profit

profit for the case of 10 hydroplants. We remark that here, and in the computation of expected profits below, we cannot evaluate the profit for solution that violates the water conservation constraints. In this cases we adopt a conservative approach and assume that the profit is zero; this allows us to compute a lower bound on the profit increase that can be expected relative to the case $\alpha = 0$. The solution obtained with $\alpha = 0$ (all scenarios are satisfied) achieves 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 α being optimized. It is interesting to observe 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 (left part of the graph), the solutions with $\alpha = 0.05$ and $\alpha = 0.1$ perform better than with $\alpha = 0.5$.

Table 4 reports the expected profit and the standard deviation of the solutions corresponding to the tested values of α, still under the conservative assumption that no profit is earned in infeasible scenarios. We can see that relaxing some of the constraints with small probability (≤ 0.05) yields an increase of the expected profit by 6.9% as compared to the solution with $\alpha = 0$. Allowing constraint violations with probability of 30% and higher produces infeasible solutions in a larger number of scenarios, and the corresponding lack of profit decreases the expected gain. When α is very

Table 4 Expected profit in €M (second column) and standard deviation (third column) for different values of α

α	$E[\varphi]$	σ
0.00	672.4	201.6
0.05	718.9	202.9
0.10	700.4	271.1
0.20	719.0	228.3
0.30	699.1	272.7
0.40	691.1	293.3
0.50	660.1	320.5

large ($\alpha = 0.5$), the solution obtained is infeasible for many scenarios, leading to an expected profit that could be up to 1.9% lower than the conservative solution with $\alpha = 0$, and with a much higher standard deviation.

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), can increase the expected profit by a significant amount. However, there are diminishing returns of increasing α, 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 drop and a much higher standard deviation 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 larger instances 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 an economic 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 6.9% 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 [22], to Alberto Borghetti for several helpful discussions, to the anonymous referees and the associate editor for their comments that helped significantly improve the paper. The first and fourth authors acknowledge the support of MIUR, Italy, under the Grant PRIN 2012. The second author acknowledges the support of the Air Force Office of Scientific Research under

Award Number FA9550- 17-1-0025. 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.

A Analysis of the nonconvex formulation for scenario subproblems

In this "Appendix" we analyze the set $C_x(w^i)$ when using the nonconvex generation function (23) for the scenario subproblems. More specifically, we show that this set satisfies the convexity assumption of Theorem 2.

Recall that in the language of this paper, $C_x(w^i) = \text{Proj}_x C_{x,y}(w^i)$, and $C_{x,y}(w^i)$ is defined by the constraints (8)–(14), with decision variables labeled x_{th}, w_{th}, e_{th} in the formulation. As discussed in Sect. 2, the master problem is only aware of variables x_{th}. We therefore have to show that the projection of (8)–(14) onto the space of x_{th} is a convex set [while the variables labeled w_{th}, e_{th} are the y component of $C_{x,y}(w^i)$]. It is sufficient to show that this is true for a single constraint of the form (12), because all the remaining constraints are linear, and each of the Tn constraints (12) involves a different set of variables x_{th}, w_{th}, e_{th}.

Using the generation function (23), constraint (12) can be written as follows:

$$f(x, w, e) = e + w(ax^2 - bx) + cx^2 - dx \leq 0, \tag{32}$$

where all coefficients a, b, c, d are positive in our data. Here, we dropped the subscripts from x_{th}, w_{th}, e_{th} for ease of exposition. Define $F := \{(x, w, e) : f(x, w, e) \leq 0\}$. Since the function $e + w(ax^2 - bx) + cx^2 - dx$ is not convex in its arguments x, w, e, we show convexity of $\text{Proj}_x F$ directly using the definition. Consider two points $x_1, x_2 \in \text{Proj}_x F$, and we want to show $x_3 := \lambda x_1 + (1-\lambda)x_2 \in \text{Proj}_x F$ for any $\lambda \in [0, 1]$. We will show it under the additional conditions $0 \leq x \leq \frac{b}{a}$, $w \geq 0$, both of which are satisfied by the data used in our experiments (we only consider nonnegative water levels, and the coefficients a, b in the problem data are such that $\frac{b}{a}$ is larger than the upper bound on x).

By convexity of the expression $ax^2 - bx$ for $a \geq 0$, we have $ax_3^2 - bx_3 \leq \lambda(ax_1^2 - bx_1) + (1-\lambda)(ax_2^2 - bx_2)$ and $cx_3^2 - dx_3 \leq \lambda(cx_1^2 - dx_1) + (1-\lambda)(cx_2^2 - dx_2)$ for any $\lambda \in [0, 1]$. Choose $w_3 = \max\{w_1, w_2\} \geq 0$, $e_3 = \min\{e_1, e_2\}$. Then we have:

$$\begin{aligned}
f(x_3, w_3, e_3) &= e_3 + w_3(ax_3^2 - bx_3) + cx_3^2 - dx_3 \\
&\leq e_3 + w_3[\lambda(ax_1^2 - bx_1) + (1-\lambda)(ax_2^2 - bx_2)] \\
&\quad + \lambda(cx_1^2 - dx_1) + (1-\lambda)(cx_2^2 - dx_2) \\
&= e_3 + \lambda[w_3(ax_1^2 - bx_1) + cx_1^2 - dx_1] \\
&\quad + (1-\lambda)[w_3(ax_2^2 - bx_2) + cx_2^2 - dx_2] \\
&\leq e_3 + \lambda[w_1(ax_1^2 - bx_1) + cx_1^2 - dx_1] \\
&\quad + (1-\lambda)[w_2(ax_2^2 - bx_2) + cx_2^2 - dx_2],
\end{aligned}$$

where for the last inequality we used $w_3(ax_1^2 - bx_1) \le w_1(ax_1^2 - bx_1)$ (because $w_3 \ge w_1$ and $ax_1^2 - bx_1 \le 0$, since $0 \le x \le \frac{b}{a}$), and for similar reasons, $w_3(ax_2^2 - bx_2) \le w_2(ax_2^2 - bx_2)$. Now consider the case $w_1(ax_1^2 - bx_1) + cx_1^2 - dx_1 \ge w_2(ax_2^2 - bx_1) + cx_2^2 - dx_2$ first. Then we have:

$$\begin{aligned} f(x_3, w_3, e_3) &\le e_3 + \lambda[w_1(ax_1^2 - bx_1) + cx_1^2 - dx_1] \\ &\quad + (1-\lambda)[w_2(ax_2^2 - bx_1) + cx_2^2 - dx_2] \\ &\le e_3 + w_1(ax_1^2 - bx_1) + cx_1^2 - dx_1 \\ &\le e_1 + w_1(ax_1^2 - bx_1) + cx_1^2 - dx_1 \le 0, \end{aligned}$$

where for the second inequality we used $(1-\lambda)[w_2(ax_2^2 - bx_1) + cx_2^2 - dx_2] \le (1-\lambda)[w_1(ax_1^2 - bx_1) + cx_1^2 - dx_1]$ since $1 - \lambda \ge 0$, and for the last inequality we used the fact that $e_3 = \min\{e_1, e_2\} \le e_1$.

The case $w_1(ax_1^2 - bx_1) + cx_1^2 - dx_1 < w_2(ax_2^2 - bx_1) + cx_2^2 - dx_2$ is almost identical and yields

$$\begin{aligned} f(x_3, w_3, e_3) &\le \min\{e_1, e_2\} + w_2(ax_2^2 - bx_2) + cx_2^2 - dx_2 \\ &\le e_2 + w_2(ax_2^2 - bx_2) + cx_2^2 - dx_2 \le 0. \end{aligned}$$

This shows that $x_3 \in \mathrm{Proj}_x F$ for any $\lambda \in [0, 1]$, thereby proving convexity of $\mathrm{Proj}_x F$ and of the projection of the feasible set (8)–(14) with generation function (23). As a consequence, the first assumption of Theorem 2 is satisfied.

References

1. Charnes, A., Cooper, W.W., Symonds, G.H.: Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. Manage. Sci. **4**(3), 235–263 (1958)
2. Prekopa, A.: On probabilistic constrained programmming. In: Kuhn, H.W. (ed.) Proceedings of the Princeton Symposium on Mathematical Programming, pp. 113–138. Princeton University Press, Princeton, NJ (1970)
3. Dupačová, J., Gaivoronski, A., Kos, Z., Szántai, T.: Stochastic programming in water management: a case study and a comparison of solution techniques. Eur. J. Oper. Res. **52**(1), 28–44 (1991)
4. Tanner, M.W., Sattenspiel, L., Ntaimo, L.: Finding optimal vaccination strategies under parameter uncertainty using stochastic programming. Math. Biosci. **215**(2), 144–151 (2008)
5. Watanabe, T., Ellis, H.: Stochastic programming models for air quality management. Comput. Oper. Res. **20**(6), 651–663 (1993)
6. Liu, X., Küçükyavuz, S., Luedtke, J.: Decomposition algorithms for two-stage chance-constrained programs. Math. Program. **157**(1), 219–243 (2014)
7. Charnes, A., Cooper, W.W.: Deterministic equivalents for optimizing and satisficing under chance constraints. Oper. Res. **11**(1), 18–39 (1963)
8. Lejeune, M.A.: Pattern-based modeling and solution of probabilistically constrained optimization problems. Oper. Res. **60**(6), 1356–1372 (2012)
9. Luedtke, J., Ahmed, S.: A sample approximation approach for optimization with probabilistic constraints. SIAM J. Optim. **19**(2), 674–699 (2008)
10. Takriti, S., Ahmed, S.: On robust optimization of two-stage systems. Math. Program. **99**, 109–126 (2004)
11. Jeroslow, R.G.: Representability in mixed integer programming, I: characterization results. Discrete Appl. Math. **17**(3), 223–243 (1987)

12. Adam, L., Branda, M.: Nonlinear chance constrained problems: optimality conditions, regularization and solvers. J. Optim. Theory Appl. **170**(2), 419–436 (2016)
13. Branda, M., Dupačová, J.: Approximation and contamination bounds for probabilistic programs. Ann. Oper. Res. **193**(1), 3–19 (2012)
14. Lejeune, M., Margot, F.: Solving chance-constrained optimization problems with stochastic quadratic inequalities. Oper. Res. **64**(4), 939–957 (2016)
15. Bonami, P., Lodi, A., Tramontani, A., Wiese, S.: On mathematical programming with indicator constraints. Math. Program. **151**(1), 191–223 (2015)
16. Padberg, M., Rinaldi, G.: A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Rev. **33**(1), 60–100 (1991)
17. Luedtke, J.: A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. Math. Program. **146**, 219–244 (2014). https://doi.org/10.1007/s10107-013-0684-6
18. Song, Y., Luedtke, J.R., Küçükyavuz, S.: Chance-constrained binary packing problems. INFORMS J. Comput. **26**(4), 735–747 (2014)
19. Gade, D., Küçükyavuz, S., Sen, S.: Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. Math. Program. **144**(1–2), 39–64 (2014)
20. Geoffrion, A.M.: Generalized Benders decomposition. J. Optim. Theory Appl. **10**(4), 237–260 (1972)
21. Duran, M., Grossmann, I.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**, 307–339 (1986)
22. Baslis, G.C., Bakirtzis, G.A.: Mid-term stochastic scheduling of a price-maker hydro producer with pumped storage. IEEE Trans. Power Syst. **26**(4), 1856–1865 (2011)
23. Carpentier, P.L., Gendreau, M., Bastin, F.: Midterm hydro generation scheduling under uncertainty using the progressive hedging algorithm. Tech. Rep. 2012-35, CIRRELT (2012)
24. Kelman, M.P.N.C.R.: Long-term hydro scheduling based on stochastic models. EPSOM **98**, 23–25 (1998)
25. McNeil, A.J., Frey, R., Embrechts, P.: Quantitative Risk Management: Concepts, Techniques and Tools (Revised Edition). Princeton Series in Finance. Princeton University Press, Princeton, NJ (2015)
26. Abhishek, K., Leyffer, S., Linderoth, J.: FilMINT: an outer approximation-based solver for convex mixed-integer nonlinear programs. INFORMS J. Comput. **22**(4), 555–567 (2010)
27. Bonami, P., Biegler, L., Conn, A., Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: An algorithmic framework for convex Mixed Integer Nonlinear Programs. Discrete Optim. **5**, 186–204 (2008)
28. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
29. Bonami, P., Cornuéjols, G., Lodi, A., Margot, F.: A feasibility pump for Mixed Integer Nonlinear Programs. Math. Program. **119**(2), 331–352 (2009)
30. Fletcher, R., Leyffer, S.: Solving Mixed Integer Nonlinear Programs by outer approximation. Math. Program. **66**, 327–349 (1994)
31. Kelley, J.E.: The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. **8**(4), 703–712 (1960)
32. Bloom, J.A.: Solving an electricity generating capacity expansion planning problem by generalized Benders' decomposition. Oper. Res. **31**(1), 84–100 (1983)
33. França, P., Luna, H.: Solving stochastic transportation-location problems by generalized Benders decomposition. Transp. Sci. **16**(2), 113–126 (1982)
34. Günlük, O., Pochet, Y.: Mixing mixed-integer inequalities. Math. Program. **90**(3), 429–457 (2001)
35. Westerlund, T., Skrifvars, H., Harjunkoski, I., Pörn, R.: An extended cutting plane method for a class of non-convex MINLP problems. Comput. Chem. Eng. **22**(3), 357–365 (1998)
36. Fischetti, M., Salvagnin, D., Zanette, A.: A note on the selection of Benders' cuts. Math. Program. **124**(1–2), 175–182 (2010)
37. Carneiro, A., Soares, S., Bond, P.: A large scale of an optimal deterministic hydrothermal scheduling algorithm. IEEE Trans. Power Syst. **5**(1), 204–211 (1990)
38. Andrieu, L., Henrion, R., Römisch, W.: A model for dynamic chance constraints in hydro power reservoir management. Eur. J. Oper. Res. **207**(2), 579–589 (2004)
39. van Ackooij, W., Henrion, R., Möller, A., Zorgati, R.: Joint chance constrained programming for hydro reservoir management. Optim. Eng. **15**, 509–531 (2014)
40. van Ackooij, W.: Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit commitment. Math. Methods Oper. Res. **80**(3), 227–253 (2014)

41. Wang, Q., Guan, Y., Wang, J.: A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. IEEE Trans. Power Syst. **27**(1), 206–215 (2012)
42. Tahanan, M., van Ackooij, W., Frangioni, A., Lacalandra, F.: Large-scale unit commitment under uncertainty. 4OR **13**(2), 115–171 (2015)
43. Salam, M.S., Nor, K.M., Hamdan, A.R.: Hydrothermal scheduling based Lagrangian relaxation approach to hydrothermal coordination. IEEE Trans. Power Syst. **13**(1), 226–235 (1998)
44. Chang, H.C., Chen, P.H.: Hydrothermal generation scheduling package: a genetic based approach. In: IEEE Proceedings on Generation, Transmission and Distribution, vol. 145. IET (1998)
45. Bacaud, L., Lemarèchal, C., Renaud, A., Sagastizábal, C.: Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners. Comput. Optim. Appl. **20**(3), 227–244 (2001)
46. Diniz, A., Costa, F., Pimentel, A.L.G., Xavier, L.N.R., Maceira, M.E.P.: Improvement in the hydro plants production function for the mid-term operation planning model in hydrothermal systems. In: EngOpt 2008—International Conference on Engineering Optimization (2008)
47. Diniz, A.L., Maceira, M.E.P.: A four-dimensional model of hydro generation for the short-term hydrothermal dispatch problem considering head and spillage effects. IEEE Trans. Power Syst. **23**(3), 1298–1308 (2008)
48. Ružić, S., Rajaković, N., Vučković, A.: A flexible approach to short-term hydro-thermal coordination. I. Problem formulation and general solution procedure. IEEE Trans. Power Syst. **11**(3), 1564–1571 (1996)
49. Wächter, A., Biegler, L.T.: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)
50. Atamtürk, A., Nemhauser, G.L., Savelsbergh, M.W.P.: The mixed vertex packing problem. Math. Program. **89**(1), 35–53 (2000)
51. Qiu, F., Ahmed, S., Dey, S.S., Wolsey, L.A.: Covering linear programming with violations. INFORMS J. Comput. **26**(3), 531–546 (2014)
52. Bonami, P., Lee, J.: BONMIN User's Manual. Tech. rep. IBM Corporation (2007)
53. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex MINLP. Optim. Methods Softw. **24**(4–5), 597–634 (2008)

Affiliations

Andrea Lodi[1] · Enrico Malaguti[2] · Giacomo Nannicini[3] · Dimitri Thomopulos[4]

Enrico Malaguti
enrico.malaguti@unibo.it

Giacomo Nannicini
nannicini@us.ibm.com

Dimitri Thomopulos
dimitri.thomopulos@unipi.it

[1] Canada Excellence Research Chair, École Polytechnique de Montréal, Montreal, Canada

[2] DEI, Università di Bologna, Bologna, Italy

[3] IBM T.J. Watson, Yorktown Heights, NY, USA

[4] DESTEC, Università di Pisa, Pisa, Italy