Les Cahiers du GERAD ISSN: 0711–2440

Stochastic hydropower generator maintenance scheduling via Benders Decomposition

J.A. Rodríguez, M.F. Anjos,
P. Côté, G. Desaulniers

G–2018–32

May 2018

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2018
– Bibliothèque et Archives Canada, 2018

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2018
– Library and Archives Canada, 2018

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Stochastic hydropower generator maintenance scheduling via Benders Decomposition

Jesús A. Rodríguez [a,b]

Miguel F. Anjos [a,b]

Pascal Côté [a,c]

Guy Desaulniers [a,b]

[a] *GERAD, Montréal (Québec), Canada, H3T 2A7*

[b] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[c] *Énergie Électrique, Rio Tinto Aluminium, Saguenay (Québec) Canada, G7S 4R5*

jesus.rodriguez@gerad.ca
anjos@stanfordalumni.org
pascal.cote@riotinto.com
guy.desaulniers@gerad.ca

May 2018
Les Cahiers du GERAD
G–2018–32

Abstract: Maintenance of power generators is essential for reliable and efficient electricity production. Because generators under maintenance are typically inactive, optimal planning of maintenance activities should consider the impact of maintenance outages on the system operation. However, finding a minimum cost maintenance schedule in hydropower systems is a challenging optimization problem due to the nonlinearity of the electricity production, the uncertainty of the water inflows and the intrinsic complexity of scheduling problems. We propose the first two-stage stochastic programming formulation for the hydropower generator maintenance scheduling problem, and we implement a parallelized Benders decomposition method with several acceleration techniques for its solution, considering a large number of scenarios. We apply statistical methods for selecting the best combination of acceleration techniques for the decomposition algorithm, and we compare the computational time of the parallelized decomposition against a mixed-integer linear programming solution approach using a testbed adapted from a real hydropower system in Canada.

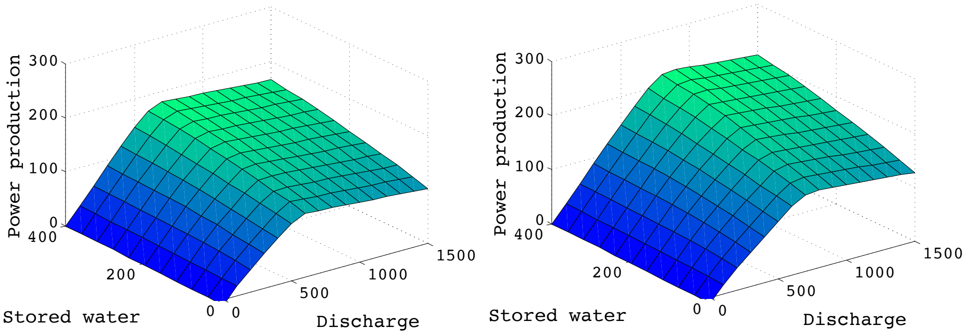
Keywords: Stochastic optimization, decomposition methods, linear approximation, parallel computing, acceleration strategies, hydroelectricity, GMSP

Acknowledgments: This research was funded by Rio Tinto and by the NSERC Engage and MITACS Accelerate programs.

1 Introduction

In order to guarantee the efficiency and reliability of electricity production, power producers carry out maintenance activities on a regular basis. As generators are usually inactive during maintenance, the economic impact of maintenance activities on the system operation must be considered. However, in hydropower systems this impact is difficult to estimate due to the nonlinearity of the hydroelectric generation, the uncertainty of the water inflows and the interdependence between multiple physical variables of the system.

A hydropower system is composed by powerhouses with turbine-generator units driven by the potential and kinetic energy of water. In each powerhouse, the hydroelectric generation is a function of the water level of the feeding reservoir or river, the discharged water through the turbines, the efficiency of the turbine-generator units and the energy loss due to the friction of the discharged water. If the turbine-generator units of a powerhouse have similar characteristics, the maximum power output p of the powerhouse with k active units can be represented by a function $p = f(s, u, k)$, where u is the water discharge and s is the stored water. We refer to this function as the Hydropower Production Function (HPF), whose nonlinearity is apparent in Figure 1.



(a) Four generators **(b) Five generators**

Figure 1: Maximum power generation p in a powerhouse, for different values of u, s and k. [28] proposed a method for computing these surfaces.

For short-term hydropower operation, the HPF has been represented with nonlinear functions [1], linear approximations [10; 6] and smoothing splines [28], among others.

The hydropower operation must also take into consideration spatial and temporal interdependencies, since water discharges can feed downstream reservoirs, and current decisions determine future costs of the system, due to the effect of the water discharges on the stored water level. Furthermore, hydroelectric generation relies on water inflows from tributary rivers, snow-melt or rainfall which tend to be difficult to predict and can exhibit large variability. Scenario trees and scenario fans (Figure 2) are some of the approaches used for representing the stochasticity of the water inflows [29].

The generator maintenance scheduling problem (GMSP) consists in determining a calendar of maintenance outages with the best performance with respect to a performance metric of the system (such as reliability, economic benefit or cost). In the GMSP, feasible schedules must satisfy constraints related to maintenance policies and resources as well as operational requirements, such as the minimum number of units available for operation. We address this problem in the context of hydropower systems, considering the aforementioned aspects, as well as the distinctive operating conditions of hydroelectric generation, such as the uncertain water inflows and the nonlinearity of electricity production. We refer to this problem as the Stochastic Generator Maintenance Scheduling Problem (SGMSP) in hydropower systems. As the optimal scheduling of generator outages can increase the electricity production [26], the impact of this problem is significant for hydropower producers.

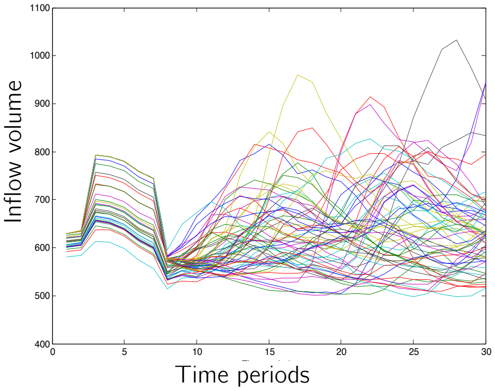


Figure 2: Scenario fan of water inflows. Each time series represents a scenario of forecasted water inflows.

Although the GMSP has been widely studied [18], the stochastic nature of the hydropower operation in combination with a realistic representation of the nonlinear hydroelectric generation has not yet been properly addressed. [13] represented the power generation with fuzzy variables but omitted important aspects of hydropower systems, such as the water storage levels and the uncertain water inflows. [16] proposed an ant colony metaheuristic for maintenance scheduling with an oversimplified model of the hydropower operation. [20] and [7] implemented a basic Benders decomposition method for the problem, without considering the nonlinearity of the electricity production and the stochastic water inflows. Recently, [26] proposed a mixed-integer linear programming (MILP) formulation for the deterministic GMSP in hydropower systems, with a convex approximation of the HPF. [26] showed that neglecting the nonlinearity of the HPF leads to significant overestimates of the electricity production and to suboptimal solutions in practice. As the resulting mathematical program is hard to solve in large instances of the problem, special solution methods that exploit its mathematical structure are necessary. Naturally, incorporating the water inflows uncertainty into the GMSP makes the problem even more challenging.

In this paper, we propose a two-stage stochastic optimization program for SGMSP in hydropower systems. This model is an extension of the deterministic MILP formulated by [26] with a linear approximation of the HPF. Using the Benders decomposition method, we partition the problem into a maintenance-only scheduling problem and scenario-wise operation subproblems. As the straightforward implementation of Benders decomposition is not a guarantee of efficient solution, we propose several enhancements to this method and we parallelize its execution. Using statistical methods, we select the best combination of the proposed acceleration techniques for the decomposition method, and we compare its performance against a MILP-based approach. For the tests, we consider a 4-powerhouse system with up to 200 inflow scenarios.

2 Mathematical programming models

In this section, we describe the optimization approach to the SGMSP, and we present its two-stage stochastic programming formulation.

2.1 Two-stage stochastic programming approach

In maintenance scheduling, the set of feasible maintenance decisions $\mathcal{Y}$ is defined by the maximum number of simultaneous outages, the time windows of maintenance activities and other relevant constraints. As the maintenance decisions $y \in \mathcal{Y}$ determine the set of available generators for electricity production in the planning horizon $\mathcal{T}$, we can compactly represent the GMSP as

$$\max_{y \in \mathcal{Y}} Q(y) - c^{\intercal} y, \tag{1}$$

where c is the cost vector of the maintenance activities, and $Q(y)$ is the operating profit during $\mathcal{T}$, corresponding to a maintenance schedule vector y. In hydropower systems, the water inflows uncertainty can be represented with a set of forecasted inflows (Figure 2), which can be used to reformulate (1) as a two-stage stochastic program with maintenance scheduling decisions in the first stage and hydropower operation decisions for each water inflow scenario in the second stage (Figure 3). As the actual scenario realization cannot be anticipated at the moment of determining the maintenance schedule y, we compute $Q(y)$ as the expected value of the profit over the set of scenarios Ω, with probability of occurrence φ_ω for scenario $\omega \in \Omega$, i.e.,

$$Q(y) = \sum_{\omega \in \Omega} \varphi_\omega Q_\omega(y),$$

where $Q_\omega(y)$ denotes the maximum cumulative operating profit corresponding to the maintenance schedule y, during $\mathcal{T}$, in scenario $\omega \in \Omega$, i.e.,

$$Q_\omega(y) = \max_{x_\omega \in \mathcal{X}(y,\xi_\omega)} \quad \Theta(x_\omega). \tag{2}$$

The hydropower operation subproblem (2) determines the values of the operational variables x_ω, such as water discharges and electricity production, that maximize the profit $\Theta(x_\omega)$ during $\mathcal{T}$. The feasible set $\mathcal{X}(y,\xi_\omega)$ of the decision variables is defined by the operational constraints of the problem, such as the water balance and the generation capacity, which depend on the maintenance schedule y and the water inflow parameters ξ_ω of the corresponding scenario. Naturally, Problems (1) and (2) can be merged into a single deterministic equivalent mathematical program

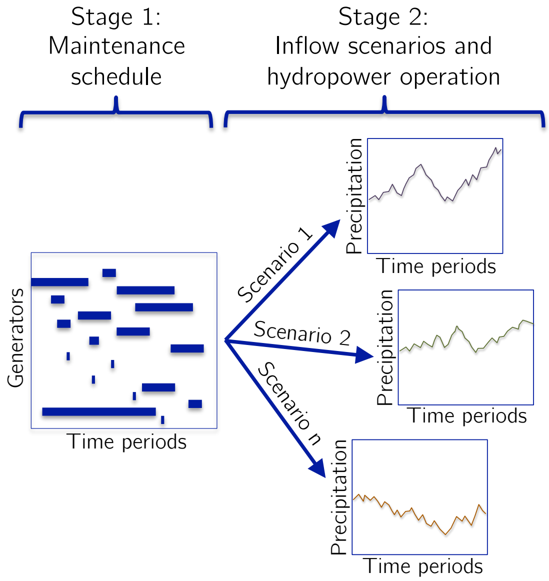


Figure 3: Generator maintenance scheduling as a two-stage stochastic problem. The maintenance schedule is defined in the first stage. Operating decisions take place in the second stage, once inflows information is revealed.

$$\max_{\substack{y \in \mathcal{Y} \\ x_\omega \in \mathcal{X}(y,\xi_\omega)}} \sum_{\omega \in \Omega} \varphi_\omega \Theta(x_\omega) \; - \; c^\intercal y. \tag{3}$$

For a background on stochastic programming, we refer the reader to [4].

The next subsection presents the deterministic equivalent (3) of the two-stage stochastic program for the SGMSP in hydropower systems. This formulation is an extension of the model proposed by [26]. Later we reformulate this problem for its solution via Benders decomposition.

2.2 Mathematical program

Consider a hydroelectric system with a set of powerhouses $\mathcal{I}$, and with a number of available generators $\bar{G}_{it}$ at each time period $t \in \mathcal{T}$ and powerhouse $i \in \mathcal{I}$. We assume that in each powerhouse the generators have similar characteristics. Let $\mathcal{M}$ be a list of generator maintenance activities to be completed within the planning horizon $\mathcal{T}$, with each activity requiring one generator outage. We define each maintenance activity m by *i)* the powerhouse where the activity must be executed, *ii)* the duration of the activity D_m, and *iii)* the time window $\mathcal{T}(m) \subseteq \mathcal{T}$ when the activity can initiate. Let $\mathcal{K}(i,t)$ be the set of numbers of generators that can be active at each time period and powerhouse. For determining the maintenance schedule, we define the binary variables $y_{mt} = 1$ if maintenance task $m \in \mathcal{M}$ starts at time period $t \in \mathcal{T}(m)$, 0 otherwise (4). We also define the binary variables $z_{itk} = 1$ if $k \in \mathcal{K}(i,t)$ generators are active in powerhouse $i \in \mathcal{I}$ at time period $t \in \mathcal{T}$, 0 otherwise (5).

$$y_{mt} \in \{0,1\}, \forall\, (m,i) \in \mathcal{M} \times \mathcal{T}(m), \tag{4}$$

$$z_{itk} \in \{0,1\}, \forall\, (i,t,k) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t). \tag{5}$$

For the SGMSP, we also define the following constraints that involve only first-stage maintenance decision variables:

$$\sum_{t \in \mathcal{T}(m)} y_{mt} = 1, \quad \forall\, m \in \mathcal{M}, \tag{6}$$

$$\sum_{\substack{m \in \mathcal{M}(i) \\ t' \in \mathcal{T}(m) \cap [\,t - D_m + 1,\, t\,]}} y_{mt'} = r_{it}, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{7}$$

$$r_{it} + \sum_{k \in \mathcal{K}(i,t)} k z_{itk} = \bar{G}_{it}, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{8}$$

$$\sum_{k \in \mathcal{K}(i,t)} z_{itk} = 1, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{9}$$

$$0 \le r_{it} \le O_{it}, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}. \tag{10}$$

Constraints (6) enforce the completion of the set of maintenance activities $\mathcal{M}$ in the planning horizon $\mathcal{T}$. Constraints (7) compute the number of maintenance outages r_{it} at each time period and powerhouse. In (7) the value of r_{it} is determined by summing the variables $y_{mt'}$ corresponding to the set of activities $\mathcal{M}(i)$ in powerhouse i that could have started at time $t' \in \mathcal{T}(m)$ and still be in execution at time $t \in \mathcal{T}$ for having started in the interval $[\,t - D_m + 1,\, t\,]$.

Equations (8) map the number of maintenance outages r_{it} into the binary variables z_{itk} that represent the number of active generators k at time period t and powerhouse i. By (9) and (5), only one z_{itk} variable is equal to one for each powerhouse and time period. Constraints (10) define the non-negativity of r_{it} and limit it to the maximum number of outages O_{it} at each time period and each powerhouse.

In addition, for the hydropower operation problem the following constraints are defined for each water inflow scenario $\omega \in \Omega$ and time period $t \in \mathcal{T}$.

$$0 \le v_{it\omega}, \quad \forall\, (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{11}$$

$$0 \le u_{it\omega} \le \bar{U}_{it}, \quad \forall\, (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{12}$$

$$\underline{S}_{it} \le s_{it\omega} \le \bar{S}_{it}, \quad \forall\, (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{13}$$

$$0 \le q^{+}_{t\omega} \le \bar{W}^{+}_{t}, \quad \forall\, (t,\omega) \in \mathcal{T} \times \Omega, \tag{14}$$

$$0 \le q^{-}_{t\omega} \le \bar{W}^{-}_{t}, \quad \forall\, (t,\omega) \in \mathcal{T} \times \Omega, \tag{15}$$

$$s_{it\omega} - s_{i(t-1)\omega} = \Bigg(\xi_{it\omega} + \sum_{g \in \mathcal{U}(i)} (u_{gt\omega} + v_{gt\omega}) - u_{it\omega} - v_{it\omega}\Bigg) F,$$

$$\forall\ (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{16}$$

$$p_{itk\omega} \leq \beta_h^0 + \beta_h^u u_{it\omega} + \beta_h^s s_{it\omega}, \quad \forall\, (i,t,k,h,\omega) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t) \times \mathcal{H}(i,k) \times \Omega, \tag{17}$$

$$0 \leq p_{itk\omega} \leq z_{itk} \bar{P}_{ik}, \quad \forall\, (i,t,k,\omega) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t) \times \Omega, \tag{18}$$

$$\sum_{k \in \mathcal{K}(i,t)} p_{itk\omega} = p_{it\omega}, \quad \forall\ (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{19}$$

$$\sum_{i \in \mathcal{I}} p_{it\omega} + q_{t\omega}^- = A_t + q_{t\omega}^+, \quad \forall\ (t,\omega) \in \mathcal{T} \times \Omega. \tag{20}$$

Constraints (11)–(15) specify the bounds of the hydropower operation decision variables: water spill $v_{it\omega}$, water discharge $u_{it\omega}$, stored water in reservoirs $s_{it\omega}$, electricity purchase $q_{t\omega}^-$ and electricity sale $q_{t\omega}^+$, respectively. Constraints (16) ensure the mass balance at each time period $t \in \mathcal{T}$ and reservoir $i \in \mathcal{I}$, considering the inflows from upstream reservoirs $g \in \mathcal{U}(i)$, as well as the uncertain water inflows $\xi_{it\omega}$ of the respective scenario $\omega \in \Omega$. In (16), F is a scalar that converts the flow units (typically m^3/s) to the suitable units for its left hand side term, $s_{it\omega} - s_{i(t-1)\omega}$, i.e., the difference in stored water between consecutive periods (such as hm^3/day). Also in (16), the consistency with the initial stored water is ensured by defining $s_{i(t-1)} = S_{i0}$ for $t = 1$.

In (17), for given values of water discharge $u_{it\omega}$ and stored water level $s_{it\omega}$, the set of hyperplanes $\mathcal{H}(i,k)$ with parameters β_h^0, β_h^u and β_h^s, approximates the power production $p_{itk\omega}$ corresponding to $k \in \mathcal{K}(i,t)$ active generators in powerhouse $i \in \mathcal{I}$. Constraints (18) restrict the generation capacity according to the number k of active generators, which is indicated by the binary variable z_{itk}. Thus, when the number of active generators is not equal to $\bar{k}$ ($z_{it\bar{k}} = 0$), the power production for this number of generators is set to zero ($p_{it\bar{k}\omega} = 0$). Constraints (19) compute the power generation $p_{it\omega}$ in each powerhouse, time period and scenario by summing the power production $p_{itk\omega}$ over the set of numbers of active generators $\mathcal{K}(i,t)$.

At each time period and scenario, the power balance is enforced by (20). In this balance, the total power injections into the system equal the power withdrawals. The injections correspond to the sum of the hydroelectric generation $p_{it\omega}$ and the electricity purchase $q_{t\omega}^-$. The power withdrawals are the electricity load A_t and the electricity sales $q_{t\omega}^+$.

Finally, the objective function of the complete problem is the sum of the expected profit of the electricity trade minus the costs of maintenance activities,

$$\underset{\substack{q^+,q^-,u,v,s,\\ r,p,y,z}}{\text{maximize}} \quad \sum_{\substack{t \in \mathcal{T}\\ \omega \in \Omega}} \varphi_\omega \left(B_t^+ q_{t\omega}^+ - B_t^- q_{t\omega}^- \right) - \sum_{\substack{m \,\in\, \mathcal{M}\\ t \,\in\, \mathcal{T}(m)}} C_{mt} y_{mt}, \tag{21}$$

where C_{mt} is the cost of maintenance activity m starting at time t, and B_t^-, B_t^+ are the electricity prices of purchase and sale, respectively, at period t. Therefore, the two-stage stochastic program for the SGMSP is

$$\text{maximize (21) subject to (4)} - \text{(20).} \tag{SGMSP}$$

To reduce the number of variables in (5) and the number of constraints in (17), (18) we define the set $\mathcal{K}(i,t)$ using the time windows of the maintenance activities (see Appendix A.1).

3 Solution strategy

Because hydrological predictions are typically subject to uncertainty, the forecasted inflows can exhibit large differences (Figure 2), so solving the SGMSP with a small number of scenarios can significantly reduce the quality of the information, which results in suboptimal decisions in practice. Therefore, a sufficiently large number of representative scenarios should be included into the model, in order to find solutions with the best average performance on the wide spectrum of inflows. However, as this number of scenarios can lead to a very large problem, we use Benders decomposition for its solution.

3.1 The Benders decomposition method

Benders decomposition [3] is a solution procedure based on the idea of partitioning a mathematical program into a relaxed master problem and a convex subproblem. The decomposition algorithm solves the master problem, fixes its solution into the subproblem, solves the subproblem and uses its dual information to generate cuts that approximate the cost function or the feasible space of the subproblem into the master problem. For a formal presentation of this method, consider the mathematical program

$$\begin{aligned} \underset{x,y}{\text{maximize}}\ & c^\intercal x + f(y) \\ \text{s.t.}\ & Ax + F(y) \leq b, \\ & x \geq 0, \\ & y \in S, \end{aligned} \tag{P}$$

where S is a posssibly nonconvex feasible set. In this problem, y and x are vectors of decision variables, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ are constant vectors, $A \in \mathbb{R}^{n\times m}$ is a constant matrix, and $F(y)$, $f(y)$ are, respectively, m-component and scalar functions on y. By fixing the so-called complicating variables $\bar{y} \in S$, the resulting subproblem

$$\begin{aligned} Q(y) = \underset{x}{\text{maximize}}\ & c^\intercal x \\ \text{s.t.}\ & Ax \leq b - F(y) \\ & x \geq 0, \end{aligned} \tag{SP}$$

is convex, and thus much easier to solve. [3] showed that with the extreme dual solutions of SP, the original problem P can be rewritten as

$$\begin{aligned} \underset{z^{SP},y}{\text{maximize}}\ & z^{SP} + f(y) \\ \text{s.t.}\ & z^{SP} \leq [b - F(y)]^\intercal \pi^p,\ \forall\, p \in \mathcal{P}, \\ & y \in S, \end{aligned} \tag{MP}$$

which is the Benders Master Problem (MP). In this problem, $\mathcal{P}$ is the set of extreme solutions, π^p is the dual solution of SP corresponding to the extreme point $p \in \mathcal{P}$ and z^{SP} is the minimum value of the dual problem. In MP the constraints

$$z^{SP} \leq [b - F(y)]^\intercal \pi^p,\ \forall\, p \in \mathcal{P}, \tag{22}$$

are referred to as optimality cuts as they remove non-optimal solutions from the master problem. In cases where the master problem solution can result in an infeasible subproblem, feasibility cuts can also be included to remove master problem solutions that are infeasible for the complete problem. These cuts can be computed from the extreme rays of the dual subproblem.

Given that the set of extreme dual solutions $\mathcal{P}$ is potentially large, the Benders decomposition method relaxes the master problem by including only a subset $\mathcal{P}^R \subset \mathcal{P}$ of extreme solutions, which is empty in the initialization of the algorithm. We refer to this problem as the Relaxed Master Problem (RMP). At each iteration, the algorithm solves the RMP, fixes its solution into the subproblem, and solves the subproblem to obtain a new dual extreme point π^p that corresponds to a violated optimality cut. This cut is then included into the RMP for the next iteration. The procedure continues until reaching a specified gap between the upper bound UB^P and the lower bound LB^P on the optimal value Z^{P*} of the complete problem P. The upper bound UB^P is the optimal value of the RMP at the current iteration, and the lower bound LB^P is the objective value of the incumbent solution. At each iteration j, the Benders algorithm computes the corresponding objective value $Z^P(\bar{y}_j)$ of the master problem solution $\bar{y}_j$, using the optimal value $Q(\bar{y}_j)$ of the subproblem, i.e.,

$$Z^P(\bar{y}_j) = Q(\bar{y}_j) + f(\bar{y}_j). \tag{23}$$

Then, the lower bound on Z^{P*} at the Jth iteration is

$$LB^P = \max_{0 \leq j \leq J} \{Z^P(\bar{y}_j)\}. \tag{24}$$

3.2 Benders reformulation of the SGMSP

In order to implement the decomposition algorithm, we derive the subproblem SP and the relaxed master problem RMP for the SGMSP. In this reformulation, SP is the hydropower production problem, and RMP is the maintenance scheduling problem with the binary variables defined in (4), (5), which we compactly denote y, z. According to this partitioning of the problem, (18) are the linking constraints, i.e., the subproblem constraints where the decision variables of the master problem are fixed.

3.2.1 Subproblem

Given a master problem solution $(\bar{y}, \bar{z})$, we set $z = \bar{z}$ in (18) to obtain the per scenario $\omega \in \Omega$ subproblems, which consist in maximizing the profit of the electricity production, subject to the operational constraints (11)–(20), i.e.,

$$Q_\omega(\bar{z}) = \underset{q^+, q^-, u, v, s}{\text{maximize}} \ \sum_{t \in \mathcal{T}} (B_t^+ q_{t\omega}^+ - B_t^- q_{t\omega}^-) \tag{25}$$

subject to

$$s_{it\omega} - s_{i(t-1)\omega} + F\left(u_{it\omega} + v_{it\omega} - \sum_{g \in \mathcal{U}(i)} (u_{gt\omega} + v_{gt\omega})\right) = F\xi_{it\omega} \perp \pi_{it\omega}, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{26}$$

$$p_{itk\omega} - \beta_h^u u_{it\omega} - \beta_h^s s_{it\omega} \le \beta_h^0 \perp \gamma_{itkh\omega}, \quad \forall\, (i,t,k,h) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t) \times \mathcal{H}(i,k), \tag{27}$$

$$0 \le p_{itk\omega} \le \bar{z}_{itk} \bar{P}_{ik} \perp \lambda_{itk\omega}, \quad \forall\, (i,t,k) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t), \tag{28}$$

$$\sum_{i \in I} p_{it\omega} + q_{t\omega}^- - q_{t\omega}^+ = A_t \perp \psi_{t\omega}, \quad \forall\, t \in \mathcal{T}, \tag{29}$$

$$\sum_{k \in \mathcal{K}(i,t)} p_{itk\omega} - p_{it\omega} = 0 \perp \theta_{it\omega}, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{30}$$

$$0 \le v_{it}, \qquad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{31}$$

$$0 \le u_{it\omega} \le \bar{U}_{it} \quad (\alpha_{it\omega}^u), \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{32}$$

$$\underline{S}_{it} \le s_{it\omega} \le \bar{S}_{it} \quad (\alpha_{it\omega}^s), \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{33}$$

$$0 \le q_{t\omega}^+ \le \bar{W}_t^+ \quad (\alpha_{t\omega}^+), \quad \forall\, t \in \mathcal{T}, \tag{34}$$

$$0 \le q_{t\omega}^- \le \bar{W}_t^- \quad (\alpha_{t\omega}^-), \quad \forall\, t \in \mathcal{T}, \tag{35}$$

where $\pi_{it\omega}$, $\gamma_{itkh\omega}$, $\lambda_{itk\omega}$, $\psi_{t\omega}$ and $\theta_{it\omega}$ denote the dual variables of constraints (26)–(30), respectively, and the symbol $\perp$ indicates the complementarity of the constraint with the corresponding dual variable.

In order to reduce the subproblem size we specify (32)–(35) as variable bounds, so that they can be treated implicitly by the linear programming (LP) solver through the bounded variable simplex method. Because (32)–(35) are not specified as general constraints, their dual variables are not explicitly defined. For each bound constraint, we denote by α (in parentheses) its dual variable which is equal to the reduced cost of the corresponding variable.

3.2.2 Master problem

The Benders Master Problem (BMP) for SGMSP maximizes the expected profit of the electricity production z^{SP} minus the maintenance cost, subject to the optimality cuts and the constraints of the original problem that involve only the maintenance decisions. Thus, the BMP is

$$\underset{y,z,z^{SP}}{\text{maximize}} \quad z^{SP} - \sum_{\substack{m \in \mathcal{M},\\ t \in \mathcal{T}(m)}} C_{mt} y_{mt} \tag{36}$$

subject to

$$\text{Eqs. } (4) - (10),$$

$$z^{SP} \leq \sum_{\omega \in \Omega} \varphi_\omega b_{\omega p}, \; \forall \, p \in \mathcal{P}^R, \tag{37}$$

$$z^{SP} \leq UB^{SP}, \tag{38}$$

where (37) are the optimality cuts corresponding to a subset $\mathcal{P}^R \subset \mathcal{P}$ of extreme solutions, and $b_{\omega p}$ is the cut term corresponding to solution $p \in \mathcal{P}^R$, in scenario $\omega \in \Omega$. At each iteration, a new solution is explored and hence the number of optimality cuts increases, unless the decomposition algorithm includes a cut removal procedure. As the Benders algorithm starts without optimality cuts, (38) prevents the unboundedness of the master problem at the first iteration. This constraint defines an initial upper bound UB^{SP} of the subproblem optimal value z^{SP}. Section 4.1.2 presents a method for computing tight values of UB^{SP}. In this master problem no feasibility cuts are necessary, due to the assumptions in Appendix A.2. The computation of (37) is described next.

3.2.3 Optimality cuts

As shown in Section 3.1, the optimality cuts are calculated from the subproblem's dual solutions. Due to the definition of (32)–(35) as variable bounds, their dual variables are not explicitly available. Instead, for these bounds we use the reduced costs of the corresponding BSP primal variables to calculate their dual contribution. Thus, we compute the cut term $b_{\omega p}$ in (37) as

$$b_{\omega p} = b^1_{\omega p} + b^2_{\omega p}, \; \forall \, (\omega, p) \in \Omega \times \mathcal{P}, \tag{39}$$

where $b^1_{\omega p}$ is the dual contribution of (26)–(29), and $b^2_{\omega p}$ is the dual contribution of (32)–(35). For a given extreme solution $p \in \mathcal{P}$, we calculate $b^1_{\omega p}$ as the sum of the products between the right hand side terms of (26)–(29), and the corresponding dual variables $\pi^p_{it\omega}$, $\gamma^p_{itkh\omega}$, $\lambda^p_{itk\omega}$, $\psi^p_{t\omega}$, i.e.,

$$\begin{aligned} b^1_{\omega p} = \sum_{t \in \mathcal{T}} \Bigg(A_t \psi^p_{t\omega} + \sum_{i \in \mathcal{I}} \Bigg(& F \xi_{it\omega} \pi^p_{it\omega} \\ & + \sum_{k \in \mathcal{K}(i,t)} \Bigg(z_{itk} \bar{P}_{ik} \lambda^p_{itk\omega} + \sum_{h \in \mathcal{H}(i,k)} \beta^0_h \gamma^p_{itkh\omega} \Bigg) \Bigg) \Bigg), \; \forall \, (\omega, p) \in \Omega \times \mathcal{P}. \end{aligned} \tag{40}$$

Notice that in (40), we discarded the terms corresponding to constraints (30) because their right hand side is 0.

For $b^2_{\omega p}$, we multiply each bound by the value of the corresponding dual variable $\alpha^{pu}_{it\omega}$, $\alpha^{ps}_{it\omega}$, $\alpha^{p+}_{t\omega}$, $\alpha^{p-}_{t\omega}$ in the solution $p \in \mathcal{P}$. That is,

$$\begin{aligned} b^2_{\omega p} = \sum_{t \in \mathcal{T}} \Bigg(\bar{W}^-_t \alpha^{p-}_{t\omega} + \bar{W}^+_t \alpha^{p+}_{t\omega} + \sum_{i \in \mathcal{I}} \Bigg(& \bar{U}_{it} \alpha^{pu}_{it\omega} \\ & + \bar{S}_{it} \alpha^{ps}_{it\omega} [\alpha^{ps}_{it\omega} > 0] + \underline{S}_{it} \alpha^{ps}_{it\omega} [\alpha^{ps}_{it\omega} < 0] \Bigg) \Bigg), \; \forall \, (\omega, p) \in \Omega \times \mathcal{P}. \end{aligned} \tag{41}$$

Since the water discharge $s_{it\omega}$ has a lower bound $\underline{S}_{it}$, for the computation of $b^2_{\omega p}$ we sum either $\bar{S}_{it} \alpha^{ps}_{it\omega}$ or $\underline{S}_{it} \alpha^{ps}_{it\omega}$, depending on the sign of the corresponding dual value $\alpha^{ps}_{it\omega}$, as indicated by the Iverson brackets in (41). A positive dual value means that the upper bound is active, whereas a negative one indicates that the lower bound is binding.

4 Acceleration techniques for Benders decomposition

Although the divide and conquer principle of decomposition methods is a promising idea to reduce the computational effort, a straightforward implementation of the Benders algorithm can perform poorly due to the number of iterations required to converge, the time per iteration, and the growing size of the master problem as a result of the cuts that are included at each iteration. In response to these challenges, several ideas have been proposed to accelerate the Benders decomposition method, such as:

- Use a formulation with a tight continuous relaxation. The stronger the formulation, the faster the convergence [22].
- When the dual subproblem has multiple solutions, select the extreme point that produces the strongest cut [22; 24].
- Solve a relaxed or partially relaxed master problem in the initial iterations. The cuts obtained from these solutions are also valid for the integer master problem [11].
- In the master problem, include constraints and variables that help to approximate the original problem [27; 12; 19].
- Solve the master problem in a branch and cut approach, with Benders cuts generated each time that a feasible integer node is found in the branching tree of the master problem [17; 19; 15].
- To reduce the oscillation of the subproblem solution, use a trust region approach or a stabilization method [27; 15].
- Besides the Benders cuts, generate additional cuts (combinatorial cuts, knapsack cuts, etc.) from the explored master problem solutions [27; 15; 19].

For a recent review on Benders decomposition, we refer the reader to [25].

Furthermore, as the subproblems can be solved independently once the master problem solution is fixed, parallelization of the scenario-wise subproblems is a natural alternative for speeding up the Benders algorithm. Nevertheless, an efficient parallel computing implementation must consider particular aspects, such as the parallelization protocol and the fine-grained design of the parallel algorithm, in order to reduce the communication overhead, to improve load balance and to exploit the intrinsic parallelism of the solution method. Previous works have addressed some of these aspects in the context of stochastic programming[e.g. 23; 21].

4.1 Implemented techniques

For speeding up the Benders decomposition method, we tested the following strategies, as discussed afterwards: 1) valid inequalities [26], 2) warm start, 3) multi-phase relaxation [11], 4) special ordered sets [2], 5) combinatorial cuts [9], 6) presolve, 7) integer rounding cuts [27], 8) parallelization.

4.1.1 Valid inequalities (VI)

As tight formulations can be favorable for Benders decomposition [22], we test the effect of the valid inequalities (42)–(43) [26] and (44) on the performance of the decomposition method for the SGMSP.

$$\sum_{\substack{m\in\mathcal{M}(i)\\ t'\in\mathcal{T}(m)\cap[t-D_m+1,\,t]}} y_{mt'} + z_{itk} \leq 1 \tag{42}$$

$$\text{for } k = \bar{G}_{it}, \forall\ (i,m,t) \in \mathcal{I}\times\mathcal{M}(i)\times\mathcal{T},$$

$$\sum_{k\,\in\,\mathcal{K}(i,t)\,\setminus\{\bar{G}_{it}\}} z_{itk} \leq r_{it},\ \forall\ (i,t)\in\mathcal{I}\times\mathcal{T}, \tag{43}$$

$$r_{it} + \sum_{k\in\mathcal{K}(i,t)\,\setminus\{\underline{K}_{it}\}} (k-\underline{K}_{it})z_{itk} \leq \bar{R}_{it},\ \forall\ (i,t)\in\mathcal{I},\times\mathcal{T}, \tag{44}$$

where $\underline{K}_{it}$ and $\bar{R}_{it}$ are respectively the minimum number of active generators and the maximum number of activities simultaneously in execution at (i,t). For a derivation of (42)–(44), see Appendix A.3.

4.1.2 Warm start (WS)

In a branch and bound process, the objective value of the current best feasible solution cuts off sections of the branching tree with no potential of harboring an optimal solution. The tighter the cutoff value, the fewer the number of nodes to be explored in the tree. In MILP solvers, cutoff values can be user-defined or can be computed from user-supplied initial solutions. Even if the initial solution is infeasible, MILP solvers can apply re-optimization or heuristics to obtain a new feasible solution and a corresponding cutoff value [14]. At any iteration of the Benders algorithm, the lower bound LB^P in (24) is naturally a cutoff value for the master problem. Therefore, at each iteration we specify to the solver a cutoff value $LB^P - \epsilon$, where $\epsilon = TOL \cdot |LB^P|$, and TOL is the default relative optimality tolerance of the MILP solver. In addition, we provide the master problem solution of the previous iteration as an initial solution to the MILP solver for the new iteration.
As tightening bounds of variables can also make the search more efficient, at the first step of the algorithm we obtain an initial upper bound UB^{SP} of z^{SP} in (38) by solving a linear relaxation of the complete problem. Moreover, at each iteration we define the current solution value of z^{SP} in the master problem as the upper bound UB^{SP} for the next iteration.

4.1.3 Multi-phase relaxation (MP)

Considering that the solutions to a RMP can generate valid cuts [11], we evaluate the effect of several relaxation schemes. For the master problem (4)–(10), (36)–(38), we define four relaxation levels of the binary variables y, z (Table 1). Among the possible sequences for applying these relaxations, we consider those that start with a complete linear relaxation (relaxation level 3) and in the subsequent phases solve an integer or partially integer RMP (relaxation levels 0, 1 or 2). To ensure a feasible solution, the last phase solves the integer master problem. We compare these relaxation sequences against a standard single-phase algorithm (without a relaxation phase, defined as sequence 0 in Table 2).

Table 1: Configuration of relaxation levels.

Relaxation level	Binary variables	Linear relaxation type
0	y, z	No relaxation
1	y	Partial
2	z	Partial
3	-	Complete

Table 2: Sequences of relaxation levels for multi-phase relaxation.

Index sequence	Relaxation sequence
0	0
1	3, 0
2	3, 2, 0
3	3, 1, 0
4	3, 1, 2, 0
5	3, 2, 1, 0

The relaxation of the master problem at the initial stages helps to quickly generate optimality cuts. Nevertheless, to prevent an excessive number of cuts that can slow down the decomposition algorithm, each relaxation stage can be finished when certain conditions are met, such as the maximum number of cuts at the stage or the minimum optimality gap of the stage.

4.1.4 Special ordered sets (SOS)

In a branch and bound algorithm, branching on sets of variables, instead of individual variables, can reduce the computational time. Special Ordered Sets (SOS) allow specifying sets of variables for branching decisions [2]. A set of variables ordered by a reference value, and with at most n consecutive non-zero variables in the set, can be specified as a SOS of type n (SOS-n), where $n \leq 2$. When branching on a SOS-1, a position in the ordered set is chosen, and all variables above and below the chosen position are forced to a zero value [2].

In the master problem (4)–(10), (36)–(38), the variables z_{itk} form a set ordered by k, for each time period t and powerhouse i. Thus, we replace the binary condition on z_{itk} (5) with the following SOS-1 definition

$$\text{SOS-1}_{it} = \{z_{itk} \to k \,:\, k \in \mathcal{K}(i,t)\} \;\forall\; (i,t) \in \{\mathcal{I} \times \mathcal{T} : |\,\mathcal{K}(i,t)\,| > 2\},$$

where the arrow symbol → indicates that k is the ordering value of the set. Since SOS work better when the cardinality of the set is not very small [14], we define a SOS-1_{it} only when the size of the set is greater than 2.

Moreover, when $B_t^- \geq B_t^+$ and $A_t \leq \bar{P}_{ik} \leq W_t$, the order of the variables z_{itk} in the master problem can be enforced by the constraint,

$$z^{SP} \leq \sum_{t \in \mathcal{T}} B_t^+ \Bigg(\sum_{\substack{i \in \mathcal{I}, \\ k \in \mathcal{K}(i,t)}} \bar{P}_{ik} z_{itk} - A_t \Bigg), \tag{45}$$

which defines an upper bound of the subproblem objective value (25). In (45), the order of the variables z_{itk} for each set (i,t) is determined by the generation capacity $\bar{P}_{ik}$, which increases with the number of generators k. Since $B_t^- \geq B_t^+$, buying electricity for selling it (i.e., electricity arbitrage) is suboptimal, so in an optimal solution, only the surplus electricity production can be sold. For a given number k of active generators, the maximum surplus electricity is the capacity $\bar{P}_{ik}$ minus the load A_t (29). When the assumption $A_t \leq \bar{P}_{ik} \leq W_t$ does not hold, (45) must be replaced by the inequality in Appendix A.4.

4.1.5 Combinatorial cuts (CC)

Combinatorial Benders cuts (CBC) [9] have been proposed to remove infeasible solutions in mathematical programs with binary variables. In contrast with the traditional Benders feasibility cuts, which are computed from the subproblem dual extreme rays, CBC exclude the current binary solution $\bar{x}$ by forcing a change of value in at least one variable of $\bar{x}$. Given the variables x_j with index set $\mathcal{J}$, CBC are defined as

$$\sum_{j \,\in\, \mathcal{S}} (1 - x_j) + \sum_{j \notin \mathcal{S}} x_j \geq 1, \tag{46}$$

where $\mathcal{S}$ is the set of variables in $\bar{x}$ with value 1, i.e., $\mathcal{S} = \{j \in \mathcal{J} : \bar{x}_j = 1\}$, and its complement is $\mathcal{S}' = \{j \in \mathcal{J} : \bar{x}_j = 0\}$. We obtain a stronger inequality than (46), by forcing at least one variable in each set, $\mathcal{S}$ and $\mathcal{S}'$, to have a different value, i.e,

$$\sum_{j \,\in\, \mathcal{S}} x_j \leq |\mathcal{S}| - 1, \tag{47}$$

$$\sum_{j \,\notin\, \mathcal{S}} x_j \geq 1 \tag{48}$$

Then, from (47) and (48), we obtain

$$\sum_{j \,\in\, \mathcal{S}} x_j - \sum_{j \,\notin\, \mathcal{S}} x_j \leq |\mathcal{S}| - 2. \tag{49}$$

Proposition 1 *The combinatorial cut (49) dominates the standard CBC (46).*

Proof. Inequality (46) can be rewritten as

$$\sum_{j \,\in\, \mathcal{S}} x_j - \sum_{j \,\notin\, \mathcal{S}} x_j \leq |\mathcal{S}| - 1. \tag{50}$$

As (50) and (49) have equal left hand side, and the right hand side of (50) is greater than the right hand side of (49), then (49) dominates (46). □

Applying (49) to cut a suboptimal solution $\bar{y}$ in (4)–(10), (36)–(38), gives

$$\sum_{(m,t)\,\in\,\mathcal{S}_y} y_{mt} - \sum_{(m,t)\,\notin\,\mathcal{S}_y} y_{mt} \leq |\mathcal{M}| - 2, \tag{51}$$

where $\mathcal{S}_y = \{(m,t) \in \mathcal{M} \times \mathcal{T}(m) : \bar{y}_{mt} = 1\}$. Notice that $|\mathcal{S}_y| = |\mathcal{M}|$, since for each activity m there is a variable $\bar{y}_{mt} = 1$ (6).

Furthermore, when the costs of the tasks are independent of the starting time, i.e., when $C_{mt} = C_m, \forall (m,t) \in \mathcal{M} \times \mathcal{T}(m)$, different solutions $\bar{y}$ can have the same objective value. In this case, a valid cut is

$$\sum_{(i,t,k)\in\mathcal{S}_z} z_{itk} - \sum_{(i,t,k)\notin\mathcal{S}_z} z_{itk} \leq |\mathcal{I}||\mathcal{T}| - 2. \tag{52}$$

In (52), $\mathcal{S}_z = \{(i,t,k) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t) : \bar{z}_{itk} = 1\}$, with cardinality $\mathcal{S}_z = |\mathcal{I}||\mathcal{T}|$, since by (9), for each time period t and powerhouse i, exactly one variable $\bar{z}_{itk}$ is equal to 1. To prevent removing optimal solutions, we only apply the cuts (52) and (51) when the objective value of the solution $(\bar{y}, \bar{z})$ is lower than the cutoff value, that is, when $Z^P(\bar{y}, \bar{z}) < LB^P - \epsilon$, where ϵ is as defined in Section 4.1.2.

4.1.6 Presolve (PS)

As presolve is a key element for efficiently solving MILP problems [5], several MILP solvers presolve the problem before the branch and cut procedure [5]. A presolve routine reduces the problem through several operations such as tightening bounds, coefficient reduction, removal of redundant columns and rows, and fixing variables based on logical implications or dual information [14; 5]. By reducing the domain of the variables and removing fractional solutions, presolve can improve the upper and the lower bound of MILP problems [5]. However, as in Benders decomposition only part of the original problem information is included into the RMP, the potential of presolving the RMP is reduced. Furthermore, as new rows are included at each iteration of the Benders algorithm, presolve operations such as reduced cost fixing can produce inconsistent solutions if applied to the RMP and fixed for subsequent iterations. In contrast, presolving the complete problem gives problem reductions that are valid for the RMP through all iterations. Therefore, we can accelerate the Benders algorithm with an initialization step that 1) applies to the complete problem (4)–(21) a presolve routine with all presolve operations activated, and 2) in the RMP fixes for all iterations of the Benders algorithm the binary variables that after presolving the complete problem are set to one of their bounds. Notice that the values of the variables fixed during presolve must be explicitly retrieved from the MILP solver because their values can be different from the linear relaxation solution.

4.1.7 Integer rounding cuts (IRC)

Let $c^\intercal$ be the coefficient vector of y in the master problem. Since the lower bound LB^P of the complete problem is also valid for the master problem, combining the bound $LB^P \leq c^\intercal y + Q^{SP}$, with the optimality cut $Q^{SP} \leq a^\intercal y + b$, gives the inequality $LB^P \leq (c+a)^\intercal y + b$, which can be tightened with integer rounding and division by the Greatest Common Divisor (GCD) of $\lceil c+a \rceil$ [27; 8]. Thus,

$$\frac{\lceil LB^P - b \rceil}{\text{GCD}} \leq \left(\frac{\lceil c+a \rceil}{\text{GCD}} \right)^\intercal y, \tag{53}$$

is a valid cut for the master problem. As the bound LB^P increases as the algorithm progresses, an IRC (53) can become weak in subsequent iterations. In the tested instances of the SGMSP, we observed that keeping only the most recent IRC had a better impact on the computational time than keeping all the generated IRC (53) and updating their constant term when the bound LB^P improves.

4.1.8 Parallelization

For the parallelization of the Benders algorithm, we implemented a master-slave approach, where the slave processors solve the subproblem and compute the cut terms, and the master process includes the cuts, solves the master problem and controls the execution of the algorithm (Figure 4). The master process runs on a computer server with a MILP solver, and the slave processes run independently on a computer cluster with an open source linear programming solver.

We used the Message Passing Interface (MPI) standard as a parallel programming protocol. Although MPI requires explicit instructions for communications among processes, some MPI implementations are portable, free and can use both shared and distributed memory. Furthermore, MPI incorporates routines for high performance collective communication that are suitable for our master-slave implementation of the decomposition algorithm.

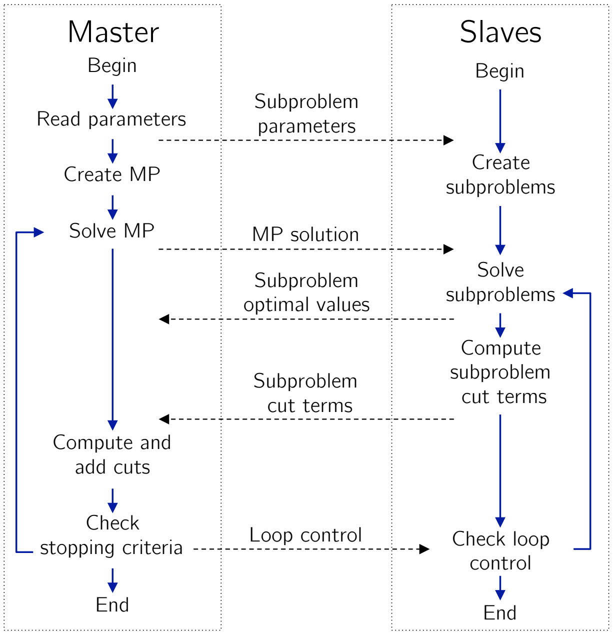


Figure 4: Simplified representation of the parallel Benders decomposition algorithm, implemented with MPI.

4.2 Implementation details

The code was written in C++ with the modeling libraries Xpress BCL. The master problem was solved with the MILP solver Xpress-MP, and the subproblems were solved with the open source linear programming solver Clp. For the parallelization we used MPICH and the Intel MPI Library. In BCL, we specified the Benders optimality cuts as *delayed rows*. This cut definition is appropriate when most of the cuts are unlikely to be active, since only the violated cuts are reintroduced by the solver when a new solution is found. Other cuts that we proposed (valid inequalities, combinatorial cuts and integer rounding cuts) were defined in BCL as *model cuts*, since they can be included by the solver to remove fractional solutions, but are not necessary to obtain feasible solutions. Furthermore, to avoid a large number of combinatorial cuts and integer rounding cuts, we kept only the cuts generated in the previous iteration.

5 Computational experiments

In this section we select the combination of acceleration techniques with the best performance on a set of test instances and we evaluate the impact of the parallelization on the computational times of the decomposition algorithm. In these experiments, a treatment corresponds to a combination of acceleration techniques or to the specific configuration of one of them.

5.1 Selection of acceleration techniques

For this section, we used a testbed of 24 instances adapted from a real hydropower system in Canada, with the attributes in Table 3. Each instance corresponds to a SGMSP with 30 inflow scenarios, 4 powerhouses, 15 time periods and 6 to 8 maintenance tasks. For each powerhouse and number of generators, the hydropower function was approximated with 30 hyperplanes in (17).

Table 3: Basic attributes of the hydropower system. Powerhouses are ordered from upstream to downstream.

System type	Number of generators	Installed capacity (MW)	Maintenance tasks
Reservoir	5	205	4
Run of the river	5	210	5
Reservoir	12	402	4
Run of the river	17	1587	5
Total	39	2404	18

The decomposition algorithm was executed in parallel on a 200-core computer cluster, with one thread dedicated to each subproblem and with up to 10 threads for solving the master problem on an Intel® Xeon® computer at 2.7 GHz.

Since the computational times can differ significantly between instances, we defined as a performance metric the normalized time $\bar{t}_{jb}$ per instance

$$\bar{t}_{jb} = \frac{t_{jb} - \mu_j}{\sigma_j}, \tag{54}$$

where t_{jb} is the computational time of the instance $j \in \mathcal{J}$ on treatment $b \in \mathcal{B}$, and μ_j, σ_j are respectively, the mean and standard deviation of the computational times of instance $j \in \mathcal{J}$ in all treatments.

5.1.1 Best combination of acceleration methods

Since the first 7 techniques of Section 4.1 can be combined in $2^7 = 128$ different ways, we are interested in identifying which combination has the lowest average computational time. For this purpose, we ran two experiments in sequence. In the first experiment, we applied each of the 7 techniques individually: Valid Inequalities (VI), Warm Start (WS), Multi-phase Relaxation (MR), Special Ordered Sets (SOS), Combinatorial Cuts (CC), Pre-solve (PS) and Integer Rounding Cuts (IRC). In this experiment, MR is the relaxation sequence 4, and VI is the combination of valid inequalities (43) and (44), which reached the smallest computational time in preliminary tests (see Appendix B).

As shown in Figure 5 and Table 4, WS achieved the lowest computational times, followed by PS and SOS. Through one-sided t-tests against the basic method, we confirmed that the effect of these three acceleration techniques was highly significant on the computational time (p-value < 0.001 in Table 4).

Table 4: Summary statistics of the acceleration methods applied independently. The column Diff. shows the difference between the mean time of each technique and the mean time of the basic method (first row).

Treatment	Mean	Std.Dev.	Diff.	p-value
-	0.62	0.27	0.00	-
CC	0.69	0.31	0.07	0.81
IRC	0.59	0.23	−0.03	0.37
MR	0.62	0.43	0.00	0.53
PS	−0.32	0.09	−0.94	6.7e-16
SOS	0.22	0.37	−0.40	5.5e-05
VI	0.56	0.25	−0.06	0.24
WS	−1.68	0.14	−2.30	2.2e-16

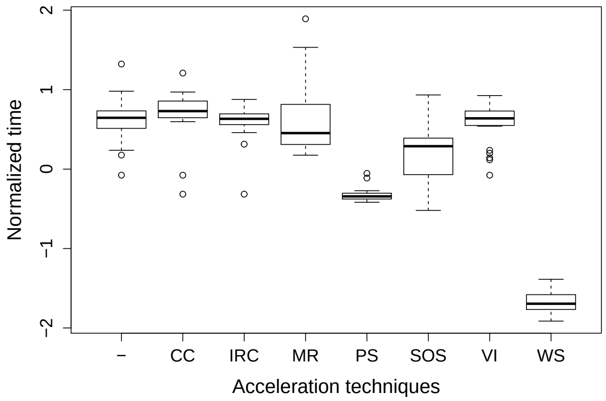


Figure 5: Boxplots of normalized computational times of 7 acceleration techniques and the basic method.

From these results, we fixed, as part of the basic configuration, the techniques with the lowest computational time (PS, SOS and WS). For selecting the final configuration, we ran a full factorial experiment with the remaining 4 techniques: CC, IRC, MR and VI, which corresponds to $2^4 = 16$ treatments. As shown in Table 5, an ANOVA applied to the results of this experiment indicates that CC and IRC had a significant effect (p-value < 0.05) on decreasing the computational time ($\beta < 0$), while MR had the opposite effect and VI was not statistically significant. Therefore, in a second ANOVA, we considered only the factors CC and IRC and their interaction term CC·IRC (Table 6). This ANOVA showed that the effects of CC and IRC were statistically significant (p-value < 0.01) on reducing the computational time ($\beta < 0$). Notice that the main effects of CC and IRC (with estimates -0.996 and -0.339, respectively) dominate interaction term CC·IRC (with estimate 0.242), which was not statistically significant (p-value 0.169).

Table 5: Summary of linear regression model with techniques VI, MP, CC and IRC as main factors, with normalized computational time as response variable.

	β estimate	p-value
(Intercept)	0.288	0.003
VI	0.089	0.295
MP	0.428	7.6e-07
CC	−0.875	< 2e-16
IRC	−0.218	0.011

Table 6: Summary of linear regression model with factors CC and IRC and interaction term. Normalized computational time as response variable.

	β estimate	p-value
(Intercept)	0.607	1.9e-11
CC	−0.996	1.2e-14
IRC	−0.339	0.006
CC·IRC	0.242	0.169

From these results, and the previously selected acceleration techniques (Table 4), we determined that the recommended combination of the acceleration techniques for the considered problem is: PS, SOS, WS, CC and IRC. In additional tests, this configuration achieved speedups of up to 4 times, with respect to the basic Benders decomposition approach.

5.2 Effect of parallelization

For the operation of hydropower systems, as many as 3000 scenarios can be generated to represent the uncertainty of the water inflows [29]. In the SGMSP, a large number of scenarios should also be considered in order to achieve high quality solutions. Nevertheless, due to the increase in the problem size, a compromise on the number of scenarios must be accepted in practice, depending on the available computational resources and the time limit for obtaining solutions. As a practical example, we consider a case with data adapted from a real 4-powerhouse system, with 8 maintenance tasks to be completed in a planning horizon of 15 days. As in the previous section, for each powerhouse and number of generators, the power production function was approximated with 30 hyperplanes.

We used the same computer cluster and computing server as in Section 5.1 for solving the subproblems, and the master problem. To avoid overlapping of subproblems on the 200 available threads, we considered a maximum of 200 scenarios, with 1 subproblem for each thread. With a time limit of 1000 seconds, the decomposition method was benchmarked against the straightforward MILP solution approach, i.e., solving model (4)–(21) with the MILP solver Xpress-MP. To observe the effect of the number of scenarios on the computational times, we kept constant all the problem parameters, except the size and composition of the set of inflow scenarios. From an initial set of 3028 scenarios, we randomly sampled 12 sets of 200 scenarios each, and we ran tests with 1, 50, 100, 150 and 200 scenarios of each set.

The results indicate that above some point between 50 to 100 scenarios, the parallel Benders decomposition with acceleration techniques outperformed the computational time of the solution with a MILP solver (Figure 6). Furthermore, in instances with 150 and 200 scenarios, the MILP solver reached the 1000-second time limit, with average optimality gaps of 4.6 % and 6.3 %, respectively, while the Benders decomposition approach reached optimal solutions in less than 800 seconds (Figure 6). The results also confirm that, in contrast with the MILP-based solution, the parallel Benders decomposition method is highly scalable. For example, between 50 to 100 scenarios the computational time of the MILP approach increased by 231.7 %, while the computational time via parallel Benders decomposition increased only by 11.5 % (Table 7).

The need for considering a sufficiently large number of scenarios is apparent in Table 8, where the objective values of the optimization model tend to converge as the number of scenarios increases. For example, in Table 8, the variability of the objective values in instances with 150 scenarios (St. dev. 121.6) is less than a half of the variability corresponding to 50 scenarios (St. dev. 267.8). Naturally, this reduction of the variability leads to a better estimate of the actual objective function value.

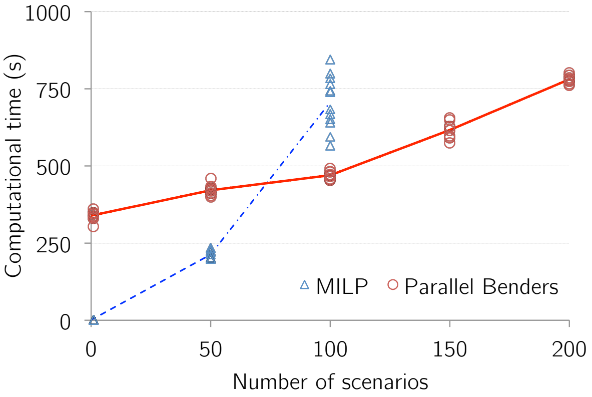


Figure 6: Computational time of solving the SGMSP with a MILP solver and with Benders decomposition.

Table 7: Statistics on the computational times with parallel Benders decomposition and MILP-based solution, with different numbers of inflow scenarios.

Number of scenarios	Benders Time		MILP Time	
	Mean	St. dev.	Mean	St. dev.
1	338.9	14.0	0.9	0.3
50	421.2	15.4	213.0	14.0
100	469.8	11.3	706.5	86.0
150	616.5	24.0	-	-
200	780.7	11.8	-	-

Table 8: Mean, standard deviation and 95 % confidence interval of the objective function values, with 12 replicates for each number of scenarios.

Number of scenarios	Objective function value		
	Mean	St. dev.	95% CI
1	13702.6	1799.8	[12559.1, 14846.2]
50	13418.9	267.8	[13248.7, 13589.0]
100	13511.1	211.5	[13376.8, 13645.5]
150	13496.9	121.6	[13419.6, 13574.1]
200	13516.6	100.9	[13452.5, 13580.7]

6 Conclusions and future work

We developed a two-stage stochastic program for the hydropower generator maintenance scheduling problem, with binary scheduling decisions in the first stage, and hydropower operation decisions in the second stage. This formulation incorporates relevant aspects of hydropower systems, such as the nonlinearity of hydroelectric production and the uncertainty of the water inflows. Furthermore, we derived necessary conditions on the problem parameters for a feasible solution.

In order to solve instances with a large number of inflow scenarios, we implemented a Benders decomposition method, and we proposed 7 techniques for accelerating its execution: valid inequalities (VI), warm start (WS), multi-phase relaxation (MR), special ordered sets (SOS), combinatorial cuts (CC), presolve (PS) and integer rounding cuts (IRC). Using statistical methods such as experimental design and analysis of variance, we found that the decomposition algorithm with the combination of PS, SOS, WS, CC and IRC reached the lowest computational time, among the explored combinations. This combination of acceleration techniques achieved speedups of up to 4 times with respect to the basic Benders decomposition approach. Using the MPI protocol, we parallelized the decomposition algorithm for its execution on a computing server and a 200-core computer cluster. In tests with up to 200 scenarios, we confirmed the high scalability of the parallelization on the number of scenarios.

Future work should address further refinements to the decomposition approach for this problem, such as cut stabilization methods [15], and branch-and-Benders-cut [17]. Alternative decomposition approaches, in combination with constraint programming, are also potential directions of future research.

Appendix A: Model supplement[1]

A.1 Set reduction

In [26], the set of numbers of generators is defined as

$$\mathcal{K}(i,t) = \big\{\, k \in \mathbb{Z} \,:\, \underline{K}_{it} \leq k \leq \bar{K}_{it} \big\},\ \forall\, (i,t) \in \mathcal{I} \times \mathcal{T} \tag{55}$$

[1] Supplementary material of Rodriguez et al., Stochastic hydropower generator maintenance scheduling via Benders decomposition. Submitted to *European Journal of Operational Research*

where

$$\underline{K}_{it} = \max\{\bar{G}_{it} - O_{it},\, \bar{G}_{it} - \bar{R}_{it}\}, \tag{56}$$

$$\bar{K}_{it} = \bar{G}_{it} - \underline{R}_{it}. \tag{57}$$

In (56)–(57), $\bar{G}_{it}$ denotes the maximum number of available generators at $(i,t) \in \mathcal{I} \times \mathcal{T}$, O_{it} is the maximum number of maintenance outages, and $\bar{R}_{it}$, $\underline{R}_{it}$ denote, respectively, the maximum and minimum number of activities simultaneously in execution at (i,t), according to their time windows, i.e.,

$$\underline{R}_{it} = \big|\{\, (m,t) \in \mathcal{M}(i) \times \mathcal{T}(m) \;:\; L_m \;\leq t \;\leq\; E_m + D_m \,\}\big|, \tag{58}$$

$$\bar{R}_{it} = \big|\{\, (m,t) \in \mathcal{M}(i) \times \mathcal{T}(m) \;:\; E_m \;\leq t \;\leq\; L_m + D_m \,\}\big|, \tag{59}$$

where for each activity $m \in \mathcal{M}$, we denote by D_m, E_m and L_m its duration, earliest starting time and latest starting time, respectively.

A.2 Conditions for feasible subproblems

From the viewpoint of computational efficiency, *complete recourse* and *partially complete recourse* are desirable properties of stochastic programming problems [4]. In problems with these properties, the Benders decomposition method will only generate feasible solutions at each iteration. A stochastic program is said to have *complete recourse* if the second stage problem (i.e., the subproblem) is always feasible. If the stochastic program has *partially complete recourse*, the second stage problem is feasible for any feasible first stage solution and scenario realization. Following these definitions, we notice that the subproblem (25)–(35) has *partially complete recourse* (i.e., is feasible for any inflow scenario and master problem feasible solution), if the following conditions are met:

1. The system (26), (31)–(33) is feasible for any inflow realization $\xi_{it\omega}$, where $(i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega$.
2. In all time periods, the electricity load A_t is not greater than the upper bound of the electricity purchase, i.e., $0 \leq A_t \leq \bar{W}_t^-$, $\forall\, t \in \mathcal{T}$.

Without loss of generality, we assume that the instances of the SGMSP satisfy conditions 1 and 2. Notice that these conditions can be guaranteed with proper values of the variable bounds (32)–(35). If either of these conditions are not met, it would be necessary to include feasibility cuts at some iterations of the Benders algorithm. Alternatively, the partial complete recourse property can be reestablished with the introduction of artificial variables in (26), (29), and with a penalization of these variables in the objective function (25).

A.3 Valid inequalities

1. The first family of valid inequalities comes from the observation in [26] that in a powerhouse i, if at least one maintenance task $m \in \mathcal{M}(i)$ is in execution at time t, then the binary variable corresponding to $\bar{G}_{it}$ active generators must be equal to zero, i.e., $z_{itk} = 0$, for $k = \bar{G}_{it}$. Thus,

$$\sum_{\substack{m \in \mathcal{M}(i) \\ t' \in \mathcal{T}(m) \cap [\, t - D_m + 1,\, t \,]}} y_{mt'} + z_{itk} \leq 1, \qquad \text{for } k = \bar{G}_{it}, \forall\ (i,m,t) \in \mathcal{I} \times \mathcal{M}(i) \times \mathcal{T}, \tag{60}$$

   are valid inequalities. Naturally, such inequalities are unnecessary when $\bar{K}_{it} < \bar{G}_{it}$ (55) or when the set $t' \in \mathcal{T}(m) \cap [\, t - D_m + 1,\, t \,]$ is empty.

2. The second family of valid inequalities comes from the fact that for any (i,t), when the number of maintenance outages is zero, i.e., $r_{it} = 0$, then all $\bar{G}_{it}$ generators are active ($z_{itk} = 1$, for $k = \bar{G}_{it}$) [26]. By (9), it follows that $z_{itk} = 0$ for $k < \bar{G}_{it}$, which is equivalent to

$$\sum_{k \in \mathcal{K}(i,t) \setminus \{\bar{G}_{it}\}} z_{itk} \;\leq r_{it},\ \forall\ (i,t) \in \mathcal{I} \times \mathcal{T}. \tag{61}$$

   Such inequalities are also unnecessary when $\bar{K}_{it} < \bar{G}_{it}$.

3. From (56) we notice that
$$\bar{G}_{it} \leq \underline{K}_{it} + \bar{R}_{it}, \quad (i,t) \in \mathcal{I} \times \mathcal{T}. \tag{62}$$
Then, applying (62) on the left hand side of (8) gives
$$r_{it} + \sum_{k \in \mathcal{K}(i,t)} k z_{itk} \leq \underline{K}_{it} + \bar{R}_{it}, \ \forall \ (i,t) \in \mathcal{I} \times \mathcal{T},$$
which by (9) and (55) leads to
$$r_{it} + \sum_{k \in \mathcal{K}(i,t) \setminus \{\underline{K}_{it}\}} (k - \underline{K}_{it}) z_{itk} \leq \bar{R}_{it}, \ \forall \ (i,t) \in \mathcal{I}, \times \mathcal{T}. \tag{63}$$

A.4 Upper bound of subproblem objective value

If the assumption $A_t \leq \bar{P}_{ik} \leq W_t$ does not hold, (45) can be replaced with

$$\begin{aligned} z^{SP} \leq \sum_{t \in \mathcal{T}} B_t^+ \Bigg(& \sum_{\substack{i \in \mathcal{I}, \\ k \in \{\mathcal{K}(i,t): \\ \bar{P}_{ik} > A_t \, ; W_t^+ > \bar{P}_{ik} - A_t\}}} (\bar{P}_{ik} - A_t) z_{itk} \\ & + \sum_{\substack{i \in \mathcal{I}, \\ k \in \{\mathcal{K}(i,t): \\ \bar{P}_{ik} > A_t \, ; W_t^+ < \bar{P}_{ik} - A_t\}}} W_t^+ z_{itk} \Bigg) \\ & - \sum_{t \in \mathcal{T}} B_t^- \Bigg(\sum_{\substack{i \in \mathcal{I}, \\ k \in \{\mathcal{K}(i,t): \\ \bar{P}_{ik} < A_t \}}} (A_t - \bar{P}_{ik}) z_{itk} \Bigg), \end{aligned} \tag{64}$$

where the first term is the maximum sold electricity when the electricity surplus is less than the bound of the electricity sale. The second term is the maximum sold electricity when the bound on the electricity sale is less than the electricity surplus, and the third term is the cost of the electricity purchase when the load exceeds the generation capacity.

Appendix B: Selecting multiple-phase relaxation sequence and valid inequalities[2]

B.1 Valid inequalities

On the set of 24 instances, we ran a factorial experiment with the $2^3 = 8$ combinations of the three families of valid inequalities of Section 4.1.1. To select the best combination of these inequalities, we sequentially applied analysis of variance (ANOVA) with normalized computational time as the response variable. From the results of the first ANOVA, with each family of valid inequalities defined as a categorical factor (Table 9), we dropped the valid inequality family 1 (factor VI1) for increasing the computational times ($\beta = 0.188$) at a significance level of 0.1 (p-value $= 0.055$). With the same experimental data, an ANOVA with the factors VI2 and VI3 and the interaction term VI2·VI3 (see Table 10) shows that the combination of the valid inequalities 2 and 3 (i.e., the interaction term VI2·V3) has the lowest average computational time ($\beta = -0.363$), at a significance level of 0.1 (p-value $= 0.064$).

[2] Supplementary material of Rodriguez et al., Stochastic hydropower generator maintenance scheduling via Benders decomposition. Submitted to *European Journal of Operational Research*

Table 9: Summary of ANOVA with valid inequalities 1, 2 and 3 as main factors, and normalized computational time as response variable.

	β estimate	p-value
(Intercept)	0.078	0.427
VI1	0.188	0.055
VI2	−0.095	0.333
VI3	−0.249	0.011

Table 10: Summary of ANOVA with valid inequalities 1 and 2 and interaction term, and normalized computational time as response variable.

	β estimate	p-value
(Intercept)	0.081	0.408
VI2	0.087	0.531
VI3	−0.067	0.627
VI2·VI3	−0.363	0.064

B.2 Multiple-phase relaxation

We defined the relaxation sequences of Table 2 as treatments. In these sequences, each phase is completed when either a specified maximum number of cuts or a maximum optimality gap is reached (Table 11). According to the results, the sequence without relaxation (i.e., relaxation sequence 0), exhibited the largest variability and the highest computational time (Figure 7). An analysis of variance on the 24 instances indicated that the multi-phase relaxation had a significant effect on the computational times (p-value $=$ 0.00924). Although the computational times of the relaxation sequences 3, 4 and 5 were similar, the relaxation sequence 4 showed the most significant effect (p-value $= 0.007$) in a one-tailed t-test against the method without relaxation (see Table 12). Therefore, the best configuration applies the relaxation sequence $(y, z) \rightarrow (z) \rightarrow (y)$, before solving the master problem without relaxation.

Table 11: Parameters of stages in multi-phase relaxation.

Relax. level	Binary var.	Max. cuts	Max. gap
0	y, z	1000	1.0e-5
1	y	4	0.005
2	z	4	0.005
3	-	20	0.010

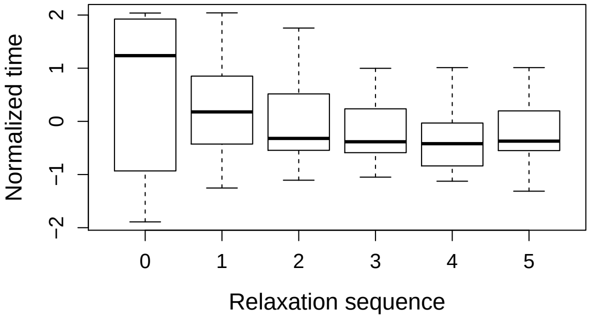


Figure 7: Boxplot of the computational times of the multi-phase relaxation sequences on 24 instances.

Table 12: Summary statistics of normalized computational times of multi-phase relaxations. The column Diff. shows the difference between the mean time of each sequence and the mean of sequence 0.

Relax. Seq.	Mean	St. Dev.	Diff.	p-value
0	0.54	1.51	0.00	-
1	0.21	0.91	−0.33	0.181
2	−0.04	0.71	−0.58	0.048
3	−0.19	0.57	−0.73	0.017
4	−0.34	0.63	−0.88	0.007
5	−0.18	0.60	−0.72	0.019

Appendix C: Nomenclature

Primary sets	
$\mathcal{I}$	Powerhouses
$\mathcal{M}$	Maintenance tasks
$\mathcal{T}$	Planning time periods, $t \in \mathcal{T} = \{1 \dots T\}$
Ω	Scenarios
Parameters	
$\xi_{it\omega}$	Lateral inflows to powerhouse i in period t and scenario ω, [m^3/s].
A_t	Electricity load at time period t.
B_t^+	Electricity sale price in time period t, [\$/MWh].
B_t^-	Electricity purchase price in time period t, [\$/MWh].
C_{mt}	Total cost of maintenance task m started at time period t, [\$].
D_m	Duration of maintenance task m [day].
E_m	Earliest start time period of maintenance task m.
F	Factor for conversion from flow per second in m^3 to flow per day in hm^3 [0.0864·s·hm^3 ·/(day'·m^3)].
$\bar{G}_{it}$	Maximum number of available turbines in powerhouse i at time period t, [turbines].
$\underline{G}_i$	Minimum number of available turbines in powerhouse i [turbines].
L_m	Latest start time period of maintenance task m.
O_{it}	Maximum number of turbine outages in powerhouse i at time period t, [turbines].
$\bar{P}_i$	Generation capacity in powerhouse i, [MWh/day].
$\bar{P}_{ik}$	Generation capacity in powerhouse i when k turbines are active, [MWh/day].
$Q(\bar{y})$	Expected operating cost of solution $\bar{y}$ [\$].
$Q_\omega(\bar{y})$	Expected operating cost of solution $\bar{y}$ in scenario ω [\$].
$\bar{R}_{it}$	Number of maintenance activities that *can* be in execution at powerhouse i in time period t.
$\underline{R}_{it}$	Number of maintenance activities that *must* be in execution at powerhouse i in time period t.
S_{0i}	Initial volume in reservoir of powerhouse i, [hm^3].
$\underline{S}_i, \bar{S}_i$	Limits on stored water in reservoir of powerhouse i at period t [hm^3].
$\bar{U}_{it}$	Maximum discharge rate in powerhouse i, [m^3/s].
$\bar{V}_{it}$	Maximum water spill in powerhouse i, [m^3/s].
$\bar{W}_t^+$	Maximum electricity sale at time t [MWh/day].
$\bar{W}_t^-$	Maximum electricity purchase at time t [MWh/day].
Derived sets	
$\mathcal{T}(m)$	Time periods when maintenance task m can be initiated in order to be completed within $\mathcal{T}$.
$\mathcal{M}(i)$	Maintenance tasks m that should be executed in powerhouse i.
$\mathcal{M}(i,t)$	Maintenance tasks m that can be in execution in powerhouse i at time period t.
$\mathcal{U}(i)$	Powerhouses upstream of powerhouse i ($\mathcal{U}(i) \subset \mathcal{I}$).
$\mathcal{K}(i,t)$	Numbers of generators that can be active at time period t and powerhouse i.
$\mathcal{H}(i,k)$	Hyperplanes for approximating the maximum power of powerhouse i when k turbines are active.
$\mathcal{A}$	set of indices (m,t) of variables y_{mt} with value 1 in solution $\bar{y}$, i.e, $\mathcal{A} = \{(m,t) \in \mathcal{M} \times \mathcal{T} \mid \bar{y}_{mt} = 1\}$.
Parameters with indexes in derived sets	
β_h^u	Coefficient of u_{it} in hyperplane $h \in \mathcal{H}(i,k)$ for bounding the power output of powerhouse i when k generators are active [MWh· s/(m^3·day)].
β_h^s	Coefficient of s_{it} in hyperplane $h \in \mathcal{H}(i,k)$ for bounding the power output of powerhouse i when k generators are active [MWh/(hm^3·day)].
β_h^0	Independent term of hyperplane $h \in \mathcal{H}(i,k)$ for bounding the power output of powerhouse i when k generators are active [MWh/day].

Decision variables	
$p_{it\omega}$	Generation of powerhouse i during time period t in scenario ω [MWh/day].
$p_{itk\omega}$	Generation of powerhouse i during time period t in scenario ω when k generators are active [MWh/day].
$q^{+}_{t\omega}$	Sale of electricity at period t in scenario ω [MWh].
$q^{-}_{t\omega}$	Purchase of electricity at period t in scenario ω [MWh].
r_{it}	Number of maintenance activities in execution at powerhouse i and time period t.
$s_{it\omega}$	Content of reservoir in powerhouse i at the end of period t in scenario ω [hm^3].
$u_{it\omega}$	Water discharge of turbines in powerhouse i at time period t in scenario ω [m^3/s].
$v_{it\omega}$	Water spill of reservoir in powerhouse i at time period t in scenario ω [m^3/s].
y_{mt}	Binary variable with value 1 if maintenance task m initiates at time period t, 0 otherwise.
z_{itk}	Binary variable with value 1 if k hydro-turbines are active in powerhouse i at time t, 0 otherwise.
z^{SP}	Approximated expected profit of the hydroelectric production [$].
z^{SP}_{ω}	Profit of the hydroelectric production in scenario ω [$].
Dual variables	
$\pi^{p}_{it\omega}$	of mass balance constraint (26) in solution p.
$\gamma^{p}_{itkh\omega}$	of power function (27) in solution p.
$\lambda^{p}_{itk\omega}$	of power bound constraint (28) in solution p.
$\psi^{p}_{t\omega}$	of power balance constraint (29) in solution p.
$\theta^{p}_{it\omega}$	of sum of power constraint (30) in solution p.
$\alpha^{pu}_{it\omega}$	of water discharge bound (32) in solution p
$\alpha^{ps}_{it\omega}$	of stored water bounds (33) in solution p
$\alpha^{p+}_{t\omega}$	of electricity sale bounds (34) in solution p
$\alpha^{p-}_{t\omega}$	of electricity purchase bounds (35) in solution p

References

[1] A. Arce. Optimal dispatch of generating units of the Itaipu hydroelectric plant. IEEE Power Engineering Review, 21(11):56–56, 2001.

[2] Evelyn Martin Lansdowne Beale and John A Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In James Rowland Lawrence, editor, Proc. of the 5th Int. Conf. on Operations Research, pages 447–454. Tavistock Publications, 1970.

[3] Jacques F. Benders. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik, 4(1):238–252, 1962.

[4] John R. Birge and Francois Louveaux. Introduction to stochastic programming. Springer Science & Business Media, 2011.

[5] E. Robert Bixby, Mary Fenelon, Zonghao Gu, Ed Rothberg, and Roland Wunderling. MIP: Theory and practice—closing the gap. In IFIP Conference on System Modeling and Optimization, pages 19–49. Springer, 1999.

[6] Alberto Borghetti, Claudia D'Ambrosio, Andrea Lodi, and Silvano Martello. An MILP approach for short-term hydro scheduling and unit commitment with head-dependent reservoir. IEEE Transactions on Power Systems, 23(3):1115–1124, 2008.

[7] Salvador Perez Canto. Application of Benders' decomposition to power plant preventive maintenance scheduling. European Journal of Operational Research, 184(2):759–777, 2008.

[8] Der-San Chen, Robert G Batson, and Yu Dang. Applied integer programming: modeling and solution. John Wiley & Sons, 2011.

[9] Gianni Codato and Matteo Fischetti. Combinatorial Benders' cuts for mixed-integer linear programming. Operations Research, 54(4):756–766, 2006.

[10] Antonio J. Conejo, José Manuel Arroyo, Javier Contreras, and Francisco Apolinar Villamor. Self-scheduling of a hydro producer in a pool-based electricity market. IEEE Transactions on Power Systems, 17(4):1265–1272, 2002.

[11] Jean-François Cordeau, Goran Stojković, François Soumis, and Jacques Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. Transportation Science, 35(4):375–388, 2001.

[12] Teodor Gabriel Crainic, Mike Hewitt, and Walter Rei. Partial Benders decomposition strategies for two-stage stochastic integer programs. Technical Report CIRRELT-2016-37, 2016.

[13] Yongqing Feng, Peng Li, and Hao Wang. Hydro-thermal generator maintenance scheduling accommodating both randomness and fuzziness. In Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), 2011 4th International Conference on, pages 734–741. IEEE, 2011.

[14] FICO. Xpress-Optimizer reference manual, 2017.

[15] Matteo Fischetti, Ivana Ljubić, and Markus Sinnl. Redesigning Benders decomposition for large-scale facility location. Management Science, 63(7):2146–2162, 2016.

[16] Wai Kuan Foong, Angus R. Simpson, Holger R. Maier, and Stephen Stolp. Ant colony optimization for power plant maintenance scheduling optimization—– a five-station hydropower system. Annals of Operations Research, 159(1):433–450, 2008.

[17] Bernard Fortz and Michael Poss. An improved Benders decomposition applied to a multi-layer network design problem. Operations Research Letters, 37(5):359–364, 2009.

[18] Aurélien Froger, Michel Gendreau, Jorge E. Mendoza, Éric Pinson, and Louis-Martin Rousseau. Maintenance scheduling in the electricity industry: A literature review. European Journal of Operational Research, 251(3):695–706, 2016.

[19] Bernard Gendron, Maria Grazia Scutellà, Rosario G. Garroppo, Gianfranco Nencioni, and Luca Tavanti. A branch-and-Benders-cut method for nonlinear power design in green wireless local area networks. European Journal of Operational Research, 255(1):151–162, 2016.

[20] Igor Kuzle, Hrvoje Pandžić, and Miljenko Brezovec. Hydro generating units maintenance scheduling using Benders decomposition. Tehnički vjesnik, 17(2):145–152, 2010.

[21] Jeff Linderoth and Stephen Wright. Decomposition algorithms for stochastic programming on a computational grid. Computational Optimization and Applications, 24(2-3):207–250, 2003.

[22] Thomas L. Magnanti and Richard T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. Operations Research, 29(3):464–484, 1981.

[23] Soren S. Nielsen and Stavros A. Zenios. Scalable parallel Benders decomposition for stochastic linear programming. Parallel Computing, 23(8):1069–1088, 1997.

[24] Nikolaos Papadakos. Practical enhancements to the Magnanti–Wong method. Operations Research Letters, 36(4):444–449, 2008.

[25] Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The Benders decomposition algorithm: A literature review. European Journal of Operational Research, 259(3):801–817, 2017.

[26] Jesús A. Rodríguez, Miguel F. Anjos, Pascal Côté, and Guy Desaulniers. MILP formulations for generator maintenance scheduling in hydropower systems. Technical Report G-2017-63, Les Cahiers du GERAD, August 2017.

[27] Tjendera Santoso, Shabbir Ahmed, Marc Goetschalckx, and Alexander Shapiro. A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operational Research, 167(1):96–115, 2005.

[28] Sara Séguin, Pascal Côté, and Charles Audet. Self-scheduling short-term unit commitment and loading problem. IEEE Transactions on Power Systems, 31(1):133–142, 2016.

[29] Sara Séguin, Stein-Erik Fleten, Pascal Côté, Alois Pichler, and Charles Audet. Stochastic short-term hydropower planning with inflow scenario trees. European Journal of Operational Research, 259(3):1156–1168, 2017.