European Journal of Operational Research 259 (2017) 1156–1168

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Stochastic short-term hydropower planning with inflow scenario trees

Sara Séguin[c,1,*], Stein-Erik Fleten[c], Pascal Côté[b,d], Alois Pichler[c,e], Charles Audet[a,b]

[a] Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3A7, Canada
[b] GERAD, 3000 ch. de la Côte-Sainte-Catherine, Montréal, Québec H3T 2A7, Canada
[c] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim NO-7491, Norway
[d] Rio Tinto, 1954 Davis, Saguenay, Québec G7S 4R7, Canada
[e] Chemnitz University of Technology, Faculty of Mathematics, 09107 Chemnitz, Germany

ARTICLE INFO

Article history:
Received 29 September 2015
Accepted 11 November 2016
Available online 18 November 2016

Keywords:
Large scale optimization
Nonlinear programming
OR in energy
Scenarios
Stochastic programming

ABSTRACT

This paper presents an optimization approach to solve the short-term hydropower unit commitment and loading problem with uncertain inflows. A scenario tree is built based on a forecasted fan of inflows, which is developed using the weather forecast and the historical weather realizations. The tree-building approach seeks to minimize the nested distance between the stochastic process of historical inflow data and the multistage stochastic process represented in the scenario tree. A two-phase multistage stochastic model is used to solve the problem. The proposed approach is tested on a 31 day rolling-horizon with daily forecasted inflows for three power plants situated in the province of Quebec, Canada, that belong to the company Rio Tinto.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Hydroelectric producers invest time and resources in developing optimization tools to gain efficiency in the use of water, since even small improvements lead to significant savings. Short-term optimization is used at the power plant level to dispatch available water for production between the turbines. Each turbine has a different efficiency. The amount of water available for production, or reservoir trajectories, is determined from the medium-term optimization and considers demand, uncertainty in the inflows, and travel time of the water between the plants. Short-term optimization is often considered to be deterministic (Taktak & D'Ambrosio, 2016) by making the assumption that the inflows are known (Finardi & da Silva, 2006) or by neglecting water balance constraints (Arce, Ohishi, & Soares, 2002) at such a short time scale, but does not allow planning under different forecasts. Also, Schwanenberg et al. (2015) have shown that considering uncertainty in short-term decision models may lead to improvements.

The focus of this paper is stochastic optimization applied to the short-term hydropower optimization problem. By considering uncertain inflows, turbines will be used in a more efficient manner since the stochastic model results in a compromise between high and low forecasted inflows. For example, in situations where reservoirs are nearly full, considering uncertain inflows when high inflows are expected prevents lowering the reservoir and force turbines into inefficient zones, which results in energy production loss in the future if these high inflows do not occur.

Few papers have looked specifically into short-term hydropower models with uncertain inflows. In Séguin, Côté, and Audet (2016), a short-term hydropower optimization model treats deterministic inflows. Water head variations are considered and nonlinearities and nonconvexities of the hydropower production function are accounted for. In Fleten and Kristoffersen (2008), uncertainty of prices and inflows is considered. The authors use time series analysis to model the water inflows, which is represented by a scenario tree in the stochastic programming model. Start-up costs are considered and a multistage stochastic model is approximated by a two-stage model. A mixed-integer linear program is used. The net water head is assumed to vary with the water discharge only, so hydropower production functions depend only on the water discharge.

In Philpott, Craddock, and Waterer (2000), the only uncertainty considered is demand. The deterministic model is a linear integer model, which is an approximation of a nonlinear mixed integer model. Once again, the hydropower production function depends only on water discharge. For some hydropower systems, neglecting the water head is not a possible avenue since many of the reservoirs have small capacities. Consequently, the water head effect

* Corresponding author.
E-mail addresses: sara.seguin.1@ulaval.ca (S. Séguin), stein-erik.fleten@iot.ntnu.no (S.-E. Fleten), pascal.cote@riotinto.com (P. Côté), alois.pichler@mathematik.tu-chemnitz.de (A. Pichler), charles.aude@gerad.ca (C. Audet).

[1] Permanent address: Department of Civil and Water Resources Engineering, Université Laval, 1065 Av. de la Médecine, Québec, Québec G1V 0A6, Canada.

http://dx.doi.org/10.1016/j.ejor.2016.11.028
0377-2217/© 2016 Elsevier B.V. All rights reserved.

is important in a short-term optimization, even with short time steps.

Many assumptions are made when solving the short-term unit commitment model, since they are complex to solve. They have a large amount of variables, power production functions are nonlinear and efficiency is different for every turbine. The most common assumption is to neglect water head variations leading to linear power production functions.

When uncertainty arises and one wants to solve the optimization models, two main streams of ideas have been applied in the optimization community. Stochastic dynamic programming has been used extensively to solve hydropower optimization models (Siqueira, Zambelli, Cicogna, Andrade, & Soares, 2006; Tejada-Guibert, Johnson, & Stedinger, 1993), as well as variants such as sampling stochastic dynamic programming (Côté, Haguma, Leconte, & Krau, 2011) or stochastic dual dynamic programming (Shapiro, 2011). These models are well suited for long or medium-term horizons but for short-term models, the state space is huge and it is very difficult, if not impossible, to solve them. In order to prevent the optimization process to empty out the reservoirs in the short-term model, values are assigned to the remaining water at the end of the planning horizon, which are obtained with stochastic dynamic programming or stochastic dual dynamic programming for example. In Lohmann, Hering, and Rebennack (2016), a new method to generate inflows, based on periodic autoregressive models, is used as input to a stochastic dual dynamic programming algorithm that allows to schedule a hydro-thermal system located in Brazil.

The other stream is stochastic programming. A two-stage stochastic model (Birge & Louveaux, 2011) consists of two stages of decisions. The first-stage decisions need to be taken without knowing the realization of the uncertainty in the future, while the second stage decisions are taken when the uncertainty is revealed.

Usually, uncertainty is represented by scenarios. Each scenario is a possible realization of the uncertainty. Multiple scenario generation methods have been used in the past to approximate the distributions of the stochastic parameters. An overview of these methods, as well as evaluating the quality of a scenario tree is found in Kaut and Wallace (2003). In De Ladurantaye, Gendreau, and Potvin (2009), a periodic autoregressive process is used to fit historical data of the prices and to generate prices for the stochastic model. The scenario tree is built by sampling the distribution fitted with the model for the different nodes. Another method creates a discrete distribution of the uncertain parameter by matching some specific statistical properties. In Høyland and Wallace (2001), the first four moments, mean, variance, skewness and kurtosis are matched. Multiple pitfalls arise from this method and one must ensure the scenario tree represents possible outcomes of the uncertainty. A survey of techniques for generating scenario trees appears in Dupačová, Consigli, and Wallace (2000) and includes recombining of data paths, contamination method and matching. Also, copulas have been used to generate scenarios for two-stage stochastic problems (Kaut & Wallace, 2003). This method offers the advantage of treating dependencies better than with correlation alone. Other methods are scenario reduction (Feng & Ryan, 2014; Growe-Kuska, Heitsch, & Romisch, 2003). An initial scenario tree is required and forward selection, or backward reduction is applied in order to reduce it and minimize the computational time required to solve the stochastic optimization model. The effect of the reduction on the solution accuracy, applied to a cascaded system of hydropower reservoirs is found in Xu, Zhong, Zambon, Zhao, and Yeh (2015).

Other methods to deal with uncertainty on the inflows include robust optimization techniques (Babonneau, Vial, & Apparigliato, 2010) and probabilistic constrained programming (van Ackooij, Henrion, Möller, & Zorgati, 2013). Robust optimization solves models that have uncertain parameters over uncertainty sets. Therefore, the optimization seeks to find a solution that is feasible regardless of the outcome of the uncertainty. In Apparigliato (2008), a rolling-horizon scheme is used and robust optimization is applied to the decision of day 1 while the rest of the horizon is considered deterministic. This is interesting as the uncertainty is applied to the important decisions. A drawback of robust optimization is the formulation of the uncertainty. In the historical records, some values of inflows may be very low and others very high. Therefore, it is difficult to define what are the best bounds for the uncertainty set, as well as capturing any nonlinear dynamics present. In probabilistic constrained programming, constraints are to be respected given a certain probability. A cascaded hydropower system is optimized with probabilistic constrained programming in van Ackooij et al. (2013). As with robust optimization, parameters on security-level and probability measures are to be given to the model, which is a difficult task in practice.

We contribute to the existing literature by considering inflow uncertainty in the short-term hydropower model. Few papers have looked specifically in stochastic short-term models and we extend the modeling proposed in Séguin et al. (2016) to consider uncertain inflows. For the producer, it is interesting to consider a stochastic model since it gives a production plan for the whole planning horizon. Applying the theory outlined in Pflug and Pichler (2015), we also detail/provide a nonparametric scenario generation approach that relies on the information in the history of inflows. We expand (Séguin et al., 2016) by introducing stochasticity to both the loading and unit commitment problems.

The paper is organized as follows. Section 2 presents data available for inflows. Section 3 describes the method to generate scenario trees. Section 4 gives an overview of the short-term hydropower problem and details the optimization models. Numerical results are presented in Section 5 and final remarks are presented in Section 6.

2. Scenario fan of inflows

This section presents the data available for the inflows. In the province of Quebec in Canada, consumers and producers of hydroelectric energy, except Hydro-Quebec, are not allowed to bid on the spot markets (Boomsma, Juul, & Fleten, 2014). The province-owned integrated utility performs all power market activities. Hence, only uncertainty related to inflows in the reservoirs is considered in this paper.

Before presenting the method for generating the scenario trees used in the optimization models, we describe the available data sets. Precipitation forecasts are obtained from Environment Canada (Environment Canada prevision models). A 7 day deterministic precipitation forecast is issued. The 7 day forecast is split in two groups: the first 3 and the last 4 days. We make the assumption that the error for both groups is independent from a meteorological point of view, as the correlation in precipitations between days is negligible. This assumption is motivated by the high variability of the weather conditions on our watershed from 1 day to the next. For example, we could have a few days of snow, followed by no precipitations then a few days of rain. The last 15 years of historical data of precipitation forecasts is searched for a given number (a) of precipitation forecasts that are the closest, in precipitation forecast (millimeters) to the first 3 days, and they are retained. The same is conducted for the second group. Since the error is assumed independent, the scenarios found for the first and second group are mixed and matched to create a^2 precipitation scenarios for the first 7 days. Note that the actual realizations of precipitation on these days are used as scenarios. Then, considering that the forecast has no value after 7 days,

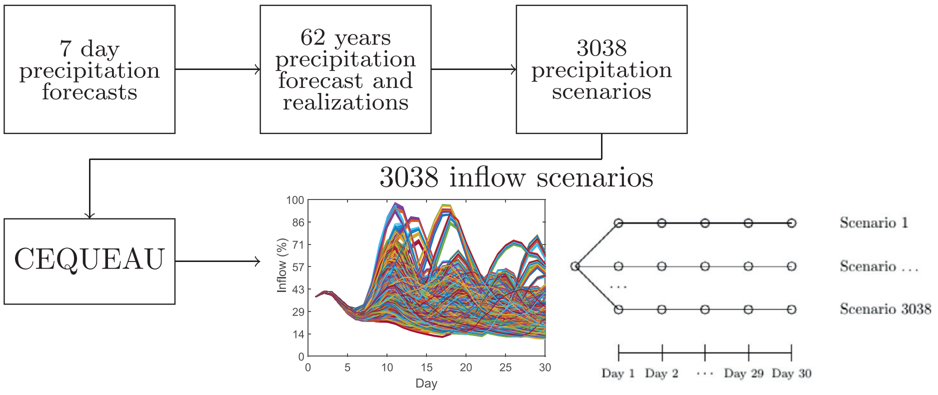


Fig. 1. Building inflow scenarios from a 7 day deterministic precipitation forecast.

the 62 years of available history of realizations is appended to all of the scenarios for the first 7 days with $a = 7$, yielding a total of $a^2 \times 62 = 3038$ scenarios of precipitation for 30 days of prevision. Then, these precipitation scenarios are given as input to the CEQUEAU hydrological model (Brisson, Boucher, & Latraverse, 2015) which outputs inflow previsions for the reservoirs.

Fig. 1 illustrates this process. The goal of the scenario tree generation method, in Section 3, is to create a scenario tree from the scenario fan of inflows.

3. Scenario tree generation

The method chosen to construct a scenario tree suitable for the stochastic optimization is taken from Pflug and Pichler (2016, 2015). The method is applied to real hydropower data. First, the structure of the scenario tree is fixed, then stochastic approximation is used to improve the values of inflow of the nodes, considering all the data available for every approximation. Improvement goes on until a convergence criterion, based on the nested distance and explained in Section 3.4, is reached.

3.1. Fixing the initial scenario tree structure: k-means clustering

The number of stages and the number of nodes per stage of the tree are fixed initially. Aggregation is necessary since the scenario tree structure can be different from the data available. The aggregation is straightforward: values of inflows for each day are summed up.

K-means clustering (Lloyd, 1982) is used to partition the data paths into clusters in order to assign initial values to the scenario tree nodes. Note that initially no probabilities are allocated to the nodes: simply values for the nodes. This clustering method minimizes the distance from every data point to the mean of the cluster to which it belongs. As an example, the k-means algorithm is applied to the 3038 inflow scenarios to form a scenario tree which has a structure as shown in Fig. 3b.

3.2. Improvement of the clusters

The method to improve the scenario tree nodes consists of two steps. First, from the initial data paths, a random data path, that is not in the paths available, is generated using density estimation. Next, the distance between this random path and the closest state of the scenario tree nodes is minimized in a stochastic approximation step in order to improve the tree. This method is repeated for a given number of iterations and is explained in what follows.

3.2.1. Step I: density estimation

In order to generate a new random path, kernel density estimation is used. We generate a random path that is close to the distribution of the data paths and conditional on previous stages. To do so, the conditional probability density function is estimated. For each stage of the desired scenario tree structure, a value of inflow is generated that is close in distribution to all of the data paths and incidental to the past.

A random path $\xi_k^d = (\xi_1^d, \dots, \xi_K^d)^T$ is to be generated using available data paths $X_{ik}^d = (X_{i1}^d, \dots, X_{iK}^d)^T$ where i is the index of available data paths, d is the dimension and K is the number of stages. The conditional density estimator (Pflug & Pichler, 2016) is:

$$\hat{f}(\xi_k|\xi_1, \dots, \xi_{k-1}) = \sum_{i=1}^{n} \prod_{j=1}^{k-1} \frac{\kappa\left(\frac{\xi_j - X_{ij}}{h_j}\right)}{\sum_{m=1}^{n} \kappa\left(\frac{\xi_j - X_{mj}}{h_j}\right)} \times \kappa\left(\frac{\xi_k - X_{ik}}{h_k}\right) \times \frac{1}{h_k}, \tag{1}$$

where the dimension d is dropped for clarity, n is the number of available data paths, κ is the kernel and h_j and h_k are the bandwidths.

The analytical representation of the actual distribution is not computed, as only samples from Eq. (1) are necessary which can be generated quickly. In practice, this is achieved by assigning weights to every data path available. The closer the observation is to the path, the higher is the weight. For every stage from $1, \dots, k-1$, the weights of the data path at each stage are multiplied. With these weights calculated, a value of inflow is to be generated at stage k.

To illustrate refer to Fig. 2. There are three data paths of inflow. The random value of inflow has been generated for stage 1 and is located with a star marker. From there, a value of inflow is to be generated for subsequent stages, always conditional on the past. As per the figure, it is necessary to find a value of inflow at stage 2 that is consistent with the conditional distribution. Therefore, weights are calculated as follows, in this case for stage k:

$$w_i(\xi_1, \dots, \xi_{k-1}) = \prod_{j=1}^{k-1} \frac{\kappa\left(\frac{\xi_j - X_{ij}}{h_j}\right)}{\sum_{m=1}^{n} \kappa\left(\frac{\xi_j - X_{mj}}{h_j}\right)}, \tag{2}$$

where $\sum_{i=1}^{n} w_i = 1$ and $w \geq 0$.

The value of inflow ξ_k at stage k is generated as follows. A data path with index i^* is chosen randomly among the available data paths at stage $k-1$ to satisfy

$$\sum_{i=1}^{i^*-1} w_i(\xi_1, \dots, \xi_{k-1}) \leq \text{rand}_u \leq \sum_{i=1}^{i^*} w_i(\xi_1, \dots, \xi_{k-1}), \tag{3}$$

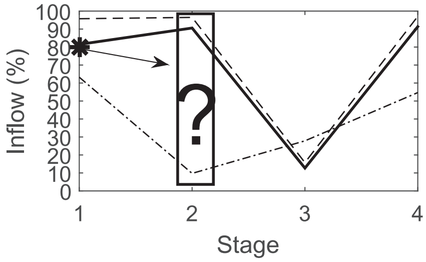


Fig. 2. Generation of a random path based on three available data paths of inflows. The generated value of inflow for stage 1 is shown with a star marker.

where rand_u is chosen from the uniform random distribution on the interval [0, 1]. The cumulative sum of the weights leads to a high probability of picking a data path near an observation.

The value of inflow ξ_k is obtained by setting the value at stage k to

$$\xi_k = X_{i^*k} + \text{rand}_{\kappa_{h_k}}, \tag{4}$$

where $\text{rand}_{\kappa_{h_k}}$ is a random value sampled from the kernel estimator using the composition method (Pflug & Pichler, 2015).

This newly generated inflow value is according to the distribution of density of the current stage and dependent on the history of all the data paths.

Referring again to Fig. 2, weights are calculated for the 3 data paths as per Eq. (2). Then, a data path is chosen randomly at stage 1 and the solid line has a high probability of being picked. Consider it is the case. To generate the value of inflow at stage 2, the value of the solid line at stage 2 is perturbed randomly. This method is then repeated at each stage in order to generate a random data path and is represented on Fig. 3a with a thick dashed line.

It is shown that the choice of the kernel does not have an important effect on the density estimation (Jones, 1990). Hence, in this paper, the logistic kernel is used:

$$\kappa(\xi) = \frac{1}{e^{\xi} + 2 + e^{-\xi}}. \tag{5}$$

The bandwidth is the smoothing factor applied to the estimation of the density. Silverman's rule of thumb (Silverman, 1986) is employed to determine the optimal bandwidth:

$$h_k = \sigma(X_{ik}) n^{-\frac{1}{d+4}} = \sigma(X_{ik}) n^{-\frac{1}{7}}, \tag{6}$$

where n is the number of data paths, d is the dimension and σ is the standard deviation. In this paper, $d = 3$ because there are three values of inflows per scenario tree node, representing three different reservoirs.

3.2.2. Step II: stochastic approximation

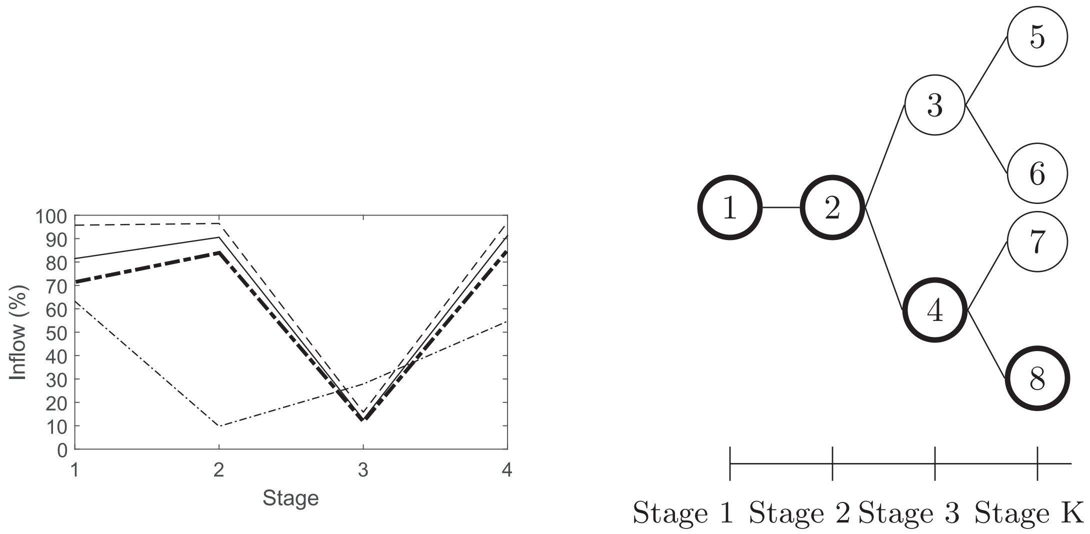
Once the new random path of inflows is generated, a stochastic approximation step is conducted. This step allows to update the value of some scenario tree nodes. During this step, a scenario from the scenario tree, more precisely a path of nodes in the scenario tree is identified. This path of nodes in the scenario tree minimizes the Wasserstein distance W between the randomly generated path during Step I of the algorithm, found in Section 3.2.1, and current scenario tree nodes values.

The Wasserstein distance is minimized as follows:

$$W^2 = \min_{\omega \in \Omega} \sum_{k=1}^{K} ||\Gamma(\omega) - \xi_k||^2, \tag{7}$$

where Ω are the scenario tree paths, $\Gamma(\omega)$ are the states corresponding to the nodes in the path ω in the scenario tree, from the set of all possible scenarios Ω, and ξ_k is the value of inflow generated randomly at stage k. Referring to Fig. 3b, $\Omega = \{(1, 2, 3, 5), (1, 2, 3, 6), (1, 2, 4, 7), (1, 2, 4, 8)\}$. Eq. (7) allows to find this path of nodes and is identified as nodes (1, 2, 4, 8) on Fig. 3b.

To achieve this, a stochastic gradient descent method that minimizes the nested distance is used. Starting from the root of the scenario tree, W is computed for the children node. The



(a) Randomly generated path of inflows, shown with thick dashed line, from three available data paths of inflows.

(b) Scenario tree. The path of nodes in the scenario tree that minimizes the Wasserstein distance is shown in bold.

Fig. 3. Illustration of the two steps of the algorithm. Generation of a random path of inflows from available data paths of inflows and stochastic approximation to improve the value of some scenario tree nodes.

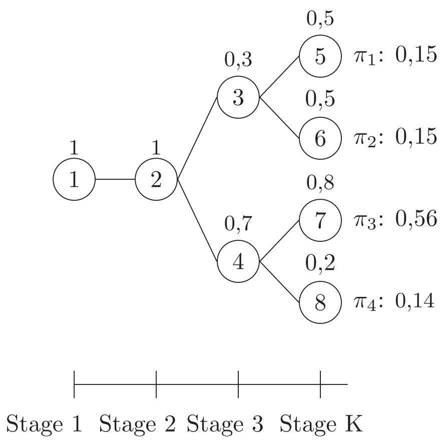


Fig. 4. A scenario tree with node probabilities (over the node) and scenario i probabilities (indicated with π_i).

children node with the smallest value of W becomes the parent node. W is then computed for the children node of the new parent node and so on until a leaf node has been reached.

The identified path of scenario tree nodes values $\Gamma(\omega)$ that minimizes the Wasserstein distance for the current stochastic approximation iteration $p = 1, 2, \ldots$ is updated in the following manner:

$$\Gamma(\omega)_{p+1} = \Gamma(\omega)_p - \alpha_p \nabla W_p, \tag{8}$$

where $\Gamma(w)$ are the values of the scenario tree nodes to improve, α_p is the step-size and ∇W_p the gradient of the distance.

The step-size $\alpha_p = \frac{1}{(p+30)^{3/4}}$, where p is the stochastic approximation iteration, is chosen since it is shown that the method will converge (Pflug & Pichler, 2016) given $\alpha_p > 0$, $\sum_p \alpha_p = \infty$ and $\Sigma_p (\alpha_p)^2 < \infty$.

As an illustration, consider one iteration of the algorithm and refer to Fig. 3. First, a random data path of inflows is generated using kernel density estimation. This can be seen on Fig. 3a: it is the thick dashed line. The Wasserstein distance between this new generated path of inflows and the current values of the scenario tree nodes is minimized and a path of nodes in the scenario tree is retrieved for potential improvement. The path of nodes minimizing this distance is shown on Fig. 3b. Hence, the value of the inflows for the thick nodes, which are 1, 2, 4 and 8 will be improved using Eq. (8).

3.3. Probabilities

During the first stochastic approximation iteration, assigned probabilities of the nodes are 0 since, as explained in Section 3.1, the scenario tree is initialized with values for the nodes only.

Node probabilities are updated at every stochastic approximation iteration. A counter is assigned to each node and initialized at 0. Every time a path of nodes minimizing the Wasserstein distance is retrieved, the corresponding counters of the nodes in this path are incremented by 1.

Once the stochastic approximation iterations are completed, probabilities are computed by dividing the counter value by the number of stochastic approximation iterations, which yields sums of child nodes probabilities equal to 1, as in Fig. 4.

In a multistage stochastic program, each path from the root to a leaf node represents a scenario. The unconditional probabilities of a scenario is obtained by multiplying the unconditional probabilities of all the nodes in the scenario, yielding probabilities π_j, where j is the scenario in Fig. 4.

An interesting feature of the scenario tree generation method is that the extreme (low and high) scenarios are accounted for, according to their occurrence in the historical data set. The law of large numbers insures that the probabilities are asymptotically consistent.

3.4. Termination criterion

The scenario tree generation algorithm terminates when the nested distance has converged to a certain ϵ for the 10 last iterations. Thus, Step I and Step II of the algorithm are repeated until convergence is obtained. Depending on the inflow forecasts, the number of iterations to converge varies. As an example, for a given test case, it took an average of 1038 iterations for the method to converge and generate one scenario tree.

The main advantage of the scenario tree generation method presented in this section is that all of the data paths are used at every iteration to improve the value of the scenario tree nodes. By doing so, the underlying discrete distribution of the available data paths, approximated by a scenario tree, is improved consistently with the data. The scenario trees are prepared before the optimization is conducted.

4. Stochastic short-term hydropower model

The two-phase deterministic optimization models taken from Séguin et al. (2016) are updated to consider stochastic inflows. This section presents the modeling of the short-term problem as well as the mathematical formulations.

4.1. Modeling of the short-term problem

The modeling of the problem considers head-dependency, as well as efficiencies of each turbine. Power P(kilowatt) produced by a single turbine is defined as

$$P(h_n, Q) = \eta(Q) \times G \times Q \times h_n(Q_{tot}, v), \tag{9}$$

where G is the gravitational acceleration (meter per square second), Q is the unit water flow and Q_{tot} is the total water flow (cubic meter per second), $\eta(Q)$ is the efficiency of the turbine and h_n is the net water-head (meter). The net water-head is a function of the forebay elevation h_f (meter), the tailrace elevation h_t (meter) and losses in the penstock φ (meter) that is given by:

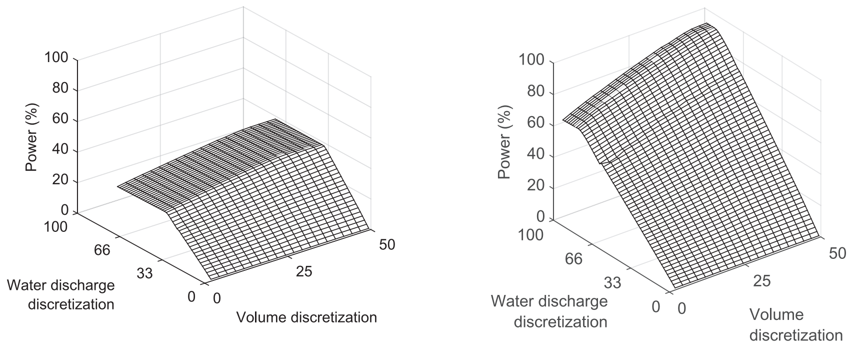
$$h_n(Q_{tot}, v) = h_f(v) - h_t(Q_{tot}) - \varphi(Q_{tot}), \tag{10}$$

where v is the volume of the reservoir (cubic hectometers). For notational purposes and since there is a relation between net water head and volume, we consider that power is a function of the volume and water flow. We propose a modeling with combinations of units instead of single units. To achieve this, a dynamic programming algorithm, where each sub-problem is a turbine, is used to calculate the power produced by a combination of units. As an example, if a power plant has a total of 5 turbines and requires three active turbines, there is a total of 10 combinations of 3 turbines, 5 combinations of 4 turbines and 1 combination of 5 turbines. Water flows are discretized and the dynamic programming algorithm is executed for each possible combinations, 16 in this case, for each power plant and discretizations of reservoir volumes and water flows.

4.1.1. Dynamic programming algorithm

The objective of the problem is to maximize the power output and it is found recursively. Given state s^j, the dynamic programming algorithm seeks to choose decision variables q^j that solves:

$$f^{*j}(s^j) = \max_{q^j} P(s^j, v) + f^{*j+1}(s^j - q^j), \tag{11}$$



(a) Five turbines in the combination

(b) Twelve turbines in the combination

Fig. 5. Maximum output surfaces.

where $j = n-1, n-2, \ldots, 1$, n is the number of turbines at the power plant, $s^j \in \{1, 2, \ldots, r\}$ is the remaining water to dispatch given the number of discretizations r and $q^j \in \{1, 2, \ldots, \min\{\overline{q}^j, Q\}\}$ the water flow with $\overline{q}^j$ maximum water flow. The optimal water flow is $q^{*j} = s^j$ that maximizes $f^{*j}(s^j)$. For $j = n$, the optimal power output is given by $f^{*j}(s^j) = P(s^j)$.

4.1.2. Maximum power output surfaces

We then build a set of surfaces of the maximum power output for each power plant. For a plant with 5 turbines requiring at least 3 working, three surfaces are built, more precisely one for 3 turbines working, one for 4 turbines working and one for 5 turbines working. The maximum power output for every possible combination of number of working turbines is retained for every discretization of volume and water flow. Such surfaces can be viewed in Fig. 5. To obtain them, the dynamic programming algorithm is executed for every number of turbines in the combination, every discretization of the reservoir volume, every discretization of the water flow and every power plant. The surfaces of maximum power outputs are then modeled using polynomial equations in the objective-function of the optimization problem. Modeling of the hydropower production functions is done by constraining these functions with two surfaces.

One constrains the water discharge without spill and the other constrains the water discharge with spill. Therefore, in the optimization problem, we have only one variable of the water discharge, which combines processed and spilled. But since we are constraining with the two different surfaces and that we are maximizing, the model will try and avoid spill since it reduces power production. When redistributing the total water discharge to the plant, we can then see if there is some spill or not. There is an upper bound on the water discharge, which is the maximum including spill.

A two-phase optimization strategy is used to penalize the startup of turbines. The first phase, namely the loading problem, optimizes values of water discharges, volumes and number of turbines in the combination for every plant and node. The second phase, namely the unit commitment problem, uses the solution of the first optimization model to determine the exact combination of turbines working at each plant and node in the scenario tree. Startups of turbines are penalized with a fixed cost. Multistage stochastic models are developed for both optimization phases, in order to consider uncertainty in the inflows of the reservoirs.

4.2. Phase I: loading problem

Optimization variables of this nonlinear stochastic multistage mixed integer problem are water flows, volumes and number of working turbines, for each node and plant in the scenario tree. There is only one variable for the water flow, but it includes processed and spilled water. We have shown (Séguin et al., 2016) that relaxing the variables that determine the number of working turbines leads to an integer solution. Therefore, we solve a nonlinear stochastic multistage continuous problem, as the coefficients of the matrix of constraints are also totally unimodular given the stochastic version of the model.

The objective is to maximize total energy production in stage 0, expected energy production in future stages and expected value of the water remaining in the reservoir at the end of the planning horizon:

$$\max_{y,q,v} \sum_{c\in C}\sum_{s=1}^{n_0^c} \chi_{s0}^c y_{s0}^c \zeta_0 + \sum_{c\in C}\sum_{j\in K}\pi_j^c \left(\sum_{i\in N_j}\sum_{s=1}^{n_i^c}\chi_{si}^c y_{si}^c \zeta_i + \sum_{p\in E_j}\Phi_p^c(v_p^c)\right) \tag{12}$$

$$\text{subject to:}\quad \chi_{si}^c \le \Psi_s^{Ac}(q_i^c, v_i^c), \quad \forall c \in C, \quad \forall i \in N, \quad \forall s \in \{1, 2, \ldots, n_i^c\}, \tag{13}$$

$$\chi_{si}^c \le \Psi_s^{Bc}(q_i^c, v_i^c), \quad \forall c \in C, \quad \forall i \in N, \quad \forall s \in \{1, 2, \ldots, n_i^c\}, \tag{14}$$

$$\delta_i^c = v_{i+1}^c - v_i^c + \gamma w_i q_i^c - \sum_{m=1}^{u^c} \gamma w_m q_i^m, \quad \forall i \in N_j, \quad \forall j \in K, \quad \forall c \in C, \tag{15}$$

$$\sum_{s=1}^{n_i^c} y_{si}^c \le 1, \quad \forall i \in N, \quad \forall c \in C, \tag{16}$$

$$y_{s0}^c = \hat{y}_{s0}^c, \quad \forall s \in \{1, 2, \ldots, n_i^c\}, \quad \forall c \in C, \quad \forall i \in N, \tag{17}$$

$$v_{min}^c \le v_i^c \le v_{max}^c, \quad \forall i \in N, \quad \forall c \in C, \tag{18}$$

$$q_{min}^c \le q_i^c \le q_{max}^c, \quad \forall i \in N, \quad \forall c \in C, \tag{19}$$

$$q_i^c \ge 0, \quad \forall i \in N, \quad \forall c \in C, \tag{20}$$

$$v_i^c \ge 0, \quad \forall i \in N, \quad \forall c \in C, \tag{21}$$

$$y_{si}^c \ge 0, \quad \forall s \in \{1, 2, \ldots, n_i^c\}, \quad \forall i \in N, \quad \forall c \in C. \tag{22}$$

Hydropower production surfaces are constrained by (13)–(14). Water balance constraints are represented by (15) and the choice of a single number of active turbines is shown in (16). Constraints

(17) are the initial number of active turbines while constraints (18)–(19) are the bounds on reservoir volumes and water discharges. Finally, constraints (20)–(22) impose nonnegativity. Index $i+1$ of variables v^c_{i+1} takes the value of the node in the set N_j. For example, if $N_j=\{1,3,5,7\}$ and $i=1$, $i+1=3$ since the index takes the value of the node at position $i=2$ in the set N_j.

The above short-term loading problem is described in more details in Séguin et al. (2016). We now show how to integrate a water-value function for the remaining water at the end of the planning horizon.

Water-value function. The water-value function is the expected energy production in the future at the end of the planning horizon. In a deterministic framework, inflows are known with certainty, thus volume in the reservoir at the end of the horizon is easier to determine. In a stochastic framework, it is not possible to give a goal for the volume at the end of the horizon since it may not be feasible for every scenario. On the other hand, neglecting this feature will cause the optimization to empty the reservoir at the end of the horizon, since the objective is to maximize energy. Hence, maximizing the expected value of future energy production, or water-value function, will prevent the optimization of doing this. The water-value functions are computed with a stochastic dynamic algorithm (Côté & Leconte, 2015) at Rio Tinto. A planning horizon of 1 year, with weekly time steps is used.

4.3. Phase II: unit commitment

This linear stochastic multistage integer model is solved using solution found in Phase I. The purpose of this model is to determine the on-off schedule of the turbine combinations (found in Phase I). Given water flows and reservoir volumes found in the loading problem, the dynamic programming algorithm is used to calculate power outputs for every possible combination of turbines, given the number of working turbines found in Phase I, and are stored in parameter β^c_{li}. The model maximizes the energy production and penalizes turbine startups. Initial combination of turbines working at stage 0 is given in $\hat{x}^c_{l0}$.

The objective is to maximize energy production at stage 0 and future energy production and penalize startup of turbines at stage 0 as well as future expected startups:

$$\max_{x,d} \sum_{c\in C} \zeta_0 \left(\sum_{l=1}^{n^c_0} \beta^c_{l0} x^c_{l0} - \sum_{t=1}^{T^c} d^c_{t0}\theta \right) + \sum_{j\in K}\sum_{c\in C} \left(\pi^c_j \left(\sum_{i\in N_j} \zeta_i \left(\sum_{l=1}^{n^c_i} \beta^c_{li} x^c_{li} - \sum_{t=1}^{T^c} d^c_{ti}\theta \right)\right)\right) \tag{23}$$

$$\text{subject to:} \quad \sum_{l=1}^{n^c_i} x^c_{li} = 1, \quad \forall i\in N, \quad \forall c\in C, \tag{24}$$

$$x^c_{li} f^c_{lit} - x^c_{li-1} f^c_{li-1t} \le d^c_{ti}, \quad \forall l\in\{1,2,\dots,n^c_i\}, \quad \forall i\in N_j, \quad \forall j\in K, \quad \forall c\in C, \quad \forall t\in\{1,2,\dots,T^c\}, \tag{25}$$

$$x^c_{l0} = \hat{x}^c_{l0}, \quad \forall l\in\{1,2,\dots,n^c_i\}, \quad \forall i\in N, \quad \forall c\in C, \tag{26}$$

$$x^c_{li}, d^c_{it} \in \mathbb{B}, \quad \forall l\in\{1,2,\dots,n^c_i\}, \quad \forall i\in N, \quad \forall t\in\{1,2,\dots,T^c\}, \quad \forall c\in C. \tag{27}$$

The choice of a single turbine combination is given by (24). Constraints that allow to penalize a startup by flagging them is shown in constraints (25). The initial combinations are given in (26) and imposition of binary variables are constraints (27). Index $i-1$ of parameters f^c_{li-1t} takes the value of the node in the set N_j. For example, if $N_j=\{1,3,5,7\}$ and $i=4$, $i-1=5$ since the index takes the value of the node at position $i=3$ in the set N_j.

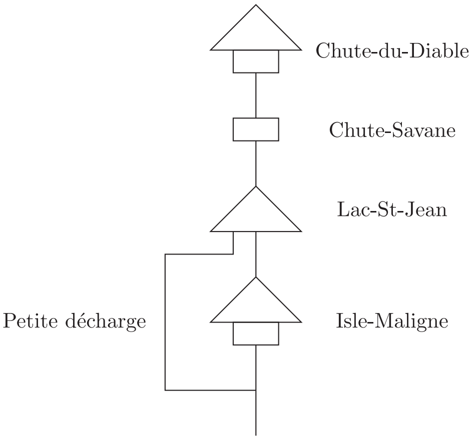


Fig. 6. Hydroelectric system studied.

This two-phase optimization process allows to find a solution efficiently. Also, even though an approximation of the energy produced is conducted in the first phase, the actual energy production is retrieved in the second phase, seeing that the actual hydropower production functions are used to compute the actual energy production given a water discharge and volume, which are solutions of the first phase.

5. Results

This section details the system on which the stochastic hydropower models are tested and results are presented.

5.1. Hydroelectric system studied

The hydroelectric system studied is situated in the Saguenay Lac-St-Jean region in the province of Quebec, Canada and is owned by Rio Tinto. For the purpose of this paper, three hydroelectric plants, which are Chute-du-Diable, Chute-Savane and Isle-Maligne are considered. The two first plants have 5 turbines each and the latter has 12. Fig. 6 represents the system studied. Triangles represent reservoirs and squares power plants.

Chute-du-Diable, Chute-Savane and Isle-Maligne plant reservoirs are quite small, respectively 344.6 cubic hectometers, 75 cubic hectometers and 120 cubic hectometers. In the optimization model, there is no water value function associated with these plants since they have small reservoirs. Instead, a full reservoir constraint at the last period is imposed as a goal in the model. The only water-value function used is for the Lac-St-Jean reservoir, therefore volume of this reservoir at the last period is an optimization variable. The capacity of this reservoir is of 4500 cubic hectometers. Water flow in *Petite décharge* is limited by a function dependent on the volume of Lac-St-Jean.

5.2. Rolling-horizon procedure

A rolling-horizon methodology is retained to validate the optimization models developed in this paper. The planning horizon of the rolling-horizon is of 31 days. For every day of the rolling-horizon, the forecast is for 30 days. For day 1 of the rolling-horizon, previsions are from days 2 to 31, for day 2 of the rolling-horizon, previsions are from days 3 to 32, and so on. The stochastic optimization models presented in Section 4 are solved every day, but only the solution for the first-stage is retained. Forecasts are updated daily. Once the forecast is updated,

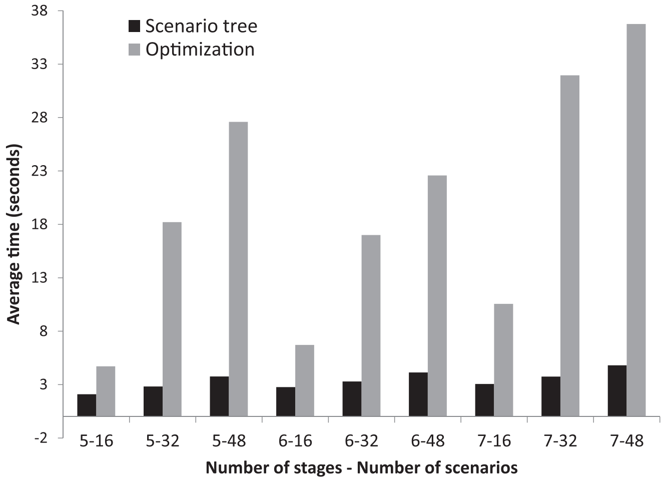


Fig. 7. Average computational time of scenario tree generation and optimization for 1 day in the rolling-horizon.

the scenario tree is generated for the corresponding day. The two-phase optimization process is launched and the first-stage solution is retained, that is: volume, water discharge and turbine combination. Then, considering the actual realization of the inflow, the water balance constraints are used to determine the actual volume of the reservoirs at the end of the period. More precisely, the water discharge from the optimization is combined with the actual realization of the inflow in order to calculate the reservoir volumes. The same process is repeated for the 31 days. In the end, a production plan for 31 days is available, which consists of the reservoir volumes, total water discharges at the plants and turbine combinations in use. See Aasgård, Andersen, Fleten, and Haugstvedt (2014) for a different approach to rolling-horizon evaluation of short-term hydropower operation.

The solution obtained from the scenario tree generation is compared to the solution obtained from the median scenario of the inflows. Therefore, we compare our method to a rolling median. Every day, the median scenario is found throughout all available scenarios and a scenario tree of 1 node per stage is solved in a deterministic fashion.

5.3. Numerical results

The scenario tree generation method is coded in Matlab (MATLAB, 2015). The optimization models are coded using AMPL (Fourer, Gay, & Kernighan, 2003). The optimization software for the loading problem, which is the relaxation of a nonlinear mixed-integer problem, is IPOPT (Wächter & Biegler, 2006) and the unit commitment model, a linear integer problem, is solved with XPRESS (Xpress).

Six test cases, which consist of monthly periods are available. The biggest problems to solve have 7 stages with 48 scenarios, 1123 nonlinear variables, 33 linear variables and 1237 constraints for the loading problem and 3475 binary variables and 825 constraints for the unit commitment problem.

Different stages, more precisely 5, 6 or 7, as well as different number of scenarios, namely 16, 32 or 48 are tested.

5.3.1. Computational time

The average time to construct the scenario tree and to optimize is shown on Fig. 7. The average time is in seconds, for a single day in the rolling-horizon procedure, more precisely for one problem including construction of the scenario tree and optimization of the two-phase process. It takes less than 5 seconds to build the scenario trees for all test cases, while the optimization requires more time given higher numbers of scenarios. Less than 42 seconds, for a single day in the rolling-horizon are necessary to construct the scenario tree and optimize the two-phase process, which is acceptable in the real operating environment. The current implementation of the scenario tree generation method and optimization is tested on three cascaded hydropower plants. For this specific producer, the whole hydropower systems consists of five hydropower plants, therefore calculation time would be acceptable for the whole system. Considering another system of, for example, 50 hydropower plants, the actual method would take approximately 350 minutes. The proposed method in this paper is applicable to a larger system, probably by decomposing the system in smaller sub-systems. To do so, the system is to be studied and depending on its configuration, distances between plants and others, modeled in an acceptable manner. Depending on the scope of the application, the calculation time may or may not be satisfactory. If a producer does not mind solving a 7 hour model every day, then the computational time is satisfactory. In order to diminish computing time, an avenue is to solve the model for a given number of days then weeks. In this way, the number of variables is greatly reduced and so is the computing time. This model is applicable to a larger hydropower system, but it would be necessary to decompose the system in sub-systems and review the modeling to diminish the number of optimization variables, given a producer requiring fast computational time.

5.3.2. Results

Table 1 illustrates the difference in energy, in terawatt hours, produced throughout the 31 days rolling-horizon combined with the value of water remaining in the reservoir at the end of the planning horizon. This implies that the difference in energy can be compared to annual production but absolute numbers are unfortunately not thus interpretable. A positive value indicates the scenario tree method produces more than the median scenario and a negative value indicates the contrary. For 4 of the test cases, the stochastic solution produces more energy. For 1 test case, the median scenario solution produces more energy. Finally, for the August case, the stochastic solution produces more energy with a 5 stage or 6 stage scenario tree, and the median scenario with a 7 stage. For the 4 test cases for which the scenario tree produces more energy than the median scenario, average improvements are 0.0679812% for June, 0.0273551% for July, 0.1620522% for September 2011 and 0.0251653% for September 2010. Despite the low percentages, this represents significant savings for the producer. As an example, the current value of a 1 gigawatt hour improvement, in the province of Quebec, is around 20,000$. Therefore, for June, the 0.0679812% higher production translates into 10,932,489$.

5.3.3. In-sample stability test

An in-sample stability test allows to verify if the scenario tree generation method is consistent. It is taken from King and Wallace (2012). Since the scenario tree is generated from random samples, one wants to verify if the solution given by the optimization, with a different scenario tree each time, give more or less the same solution. If so, then the scenario tree method is consistent.

As an example, July 2011 and June 2011 data sets were chosen for this verification. For both data sets, 6 scenario trees were generated with the same number of stages and scenarios. Then, the optimization was conducted on all of these scenario trees to verify the effect on the objective function value. Table 2 gives, for these two data sets and 6 instances each, the values of the objective function, for the scenario tree and median scenario methods.

Results show that the scenario tree generation method is consistent, as the difference between the objective functions of the stochastic and median scenario methods present slight variations. For the July test case, the median is 0.2077 terawatt hours, the

Table 1
Results for 6 test cases (5 are data sets from the year 2011 and 1 from 2010). Energy produced by the stochastic solution and the median scenario rolling-horizon is given. Also, the difference in energy between both solutions is shown.

	June 2011			July 2011			August 2011		
Nb. Sc.	Stoch. (terawatt hours)	Median (terawatt hours)	Diff. (terawatt hours)	Stoch. (terawatt hours)	Median (terawatt hours)	Diff. (terawatt hours)	Stoch. (terawatt hours)	Median (terawatt hours)	Diff. (terawatt hours)
	5 stages								
16	804.5143	804.0265	0.4878	740.2678	740.0631	0.2047	710.1115	710.0795	0.0320
32	804.7050	804.0251	0.6799	740.2783	740.0631	0.2152	710.1108	710.0794	0.0314
48	804.6894	804.0249	0.6645	740.2496	740.0631	0.1865	710.0988	710.0794	0.0194
	6 stages								
16	804.5059	804.1495	0.3564	740.2698	740.0665	0.2033	710.0783	710.0733	0.0050
32	804.6796	804.1479	0.5317	740.2652	740.0665	0.1987	710.1139	710.0733	0.0406
48	804.6715	804.1481	0.5234	740.2608	740.0665	0.1943	709.9826	710.0732	**−0.0906**
	7 stages								
16	804.5171	804.0881	0.4290	740.2676	740.0578	0.2098	710.0693	710.0867	**−0.0174**
32	804.7166	804.0881	0.6285	740.2566	740.0578	0.1988	710.0732	710.0867	**−0.0135**
48	804.7063	804.0879	0.6184	740.2686	740.0578	0.2108	710.0806	710.0867	**−0.0061**
	September 2010			**September 2011**			**October 2011**		
Nb. Sc.	Stoch. (terawatt hours)	Median (terawatt hours)	Diff. (terawatt hours)	Stoch. (terawatt hours)	Median (terawatt hours)	Diff. (terawatt hours)	Stoch. (terawatt hours)	Median (terawatt hours)	Diff. (terawatt hours)
	5 stages								
16	729.5792	729.3811	0.1981	733.0375	731.6799	1.3576	704.7842	704.8494	**−0.0652**
32	729.5841	729.3821	0.2020	733.0530	731.6799	1.3731	704.7847	704.8494	**−0.0647**
48	729.5810	729.3804	0.2006	733.0818	731.6799	1.4019	704.7877	704.8496	**−0.0619**
	6 stages								
16	729.5856	729.3917	0.1939	732.9971	731.7773	1.2198	704.7690	704.8636	**−0.0946**
32	729.5779	729.3929	0.1850	733.0188	731.7773	1.2415	704.7928	704.8636	**−0.0708**
48	729.5800	729.3924	0.1876	733.0937	731.7774	1.3163	704.7326	704.8634	**−0.1308**
	7 stages								
16	729.5854	729.4151	0.1703	732.9428	731.9647	0.9781	704.7608	704.8566	**−0.0958**
32	729.5775	729.4156	0.1619	732.9599	731.9647	0.9952	704.7879	704.8566	**−0.0687**
48	729.5834	729.4139	0.1695	732.9702	731.9648	1.0054	704.7873	704.8567	**−0.0694**

Table 2
Objective function values for 6 random scenario trees with the same number of stages and scenarios, on two data sets.

Data	Inst.	Stoch. (terawatt hours)	Median (terawatt hours)	Diff. (terawatt hours)
July	1	740.2652	740.0665	0.1987
	2	740.2759	740.0665	0.2094
	3	740.2725	740.0665	0.2060
	4	740.2581	740.0665	0.1916
	5	740.2799	740.0665	0.2134
	6	740.2878	740.0665	0.2213
June	1	804.6715	804.1481	0.5234
	2	804.6707	804.1484	0.5223
	3	804.6709	804.1474	0.5235
	4	804.6824	804.1489	0.5335
	5	804.6769	804.1486	0.5283
	6	804.6571	804.1472	0.5099

mean 0.2067 terawatt hours and the variance 0.9308 terawatt hours and for the June test case, the median and the mean are 0.5235 terawatt hours and the variance 0.0516 terawatt hours.

5.3.4. Interpretation of the results

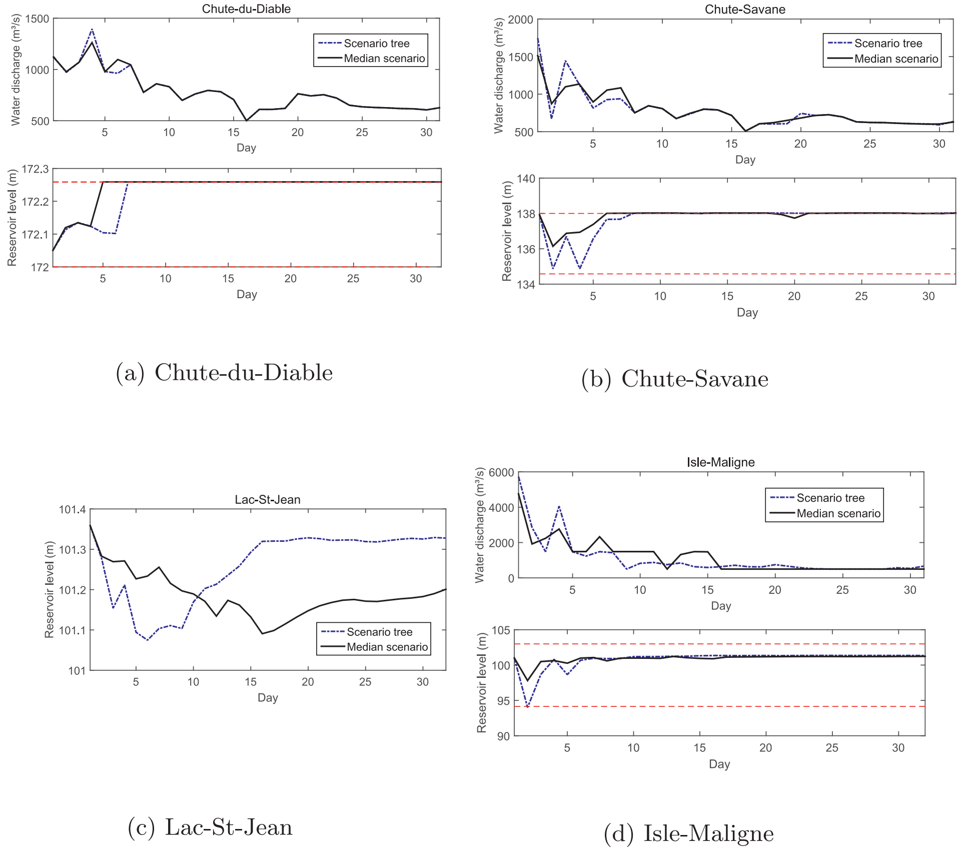
The following figures illustrate the 31 day rolling-horizon backtesting solution more precisely: water discharge and reservoir levels for the power plants and reservoirs studied in this paper.

Fig. 8 pictures June 2011 data set with 5 stages and 16 scenarios. Solutions obtained from the scenario tree method and the median scenario are quite similar. Also note that when a method turbines more water, it is penalized accordingly so it is not advantaged. The absolute difference between the volume, at the end of the 31 day horizon, between the solution obtained with the stochastic model and the solution obtained with the deterministic model is calculated. This volume is then transformed into energy, then added to the method which has a higher end volume, since it is disadvantaged, given the other method processed more water throughout the 31 day planning horizon.

Fig. 9 also illustrates the June 2011 data set with 7 stages and 16 scenarios. Again, results are very similar.

Without any surprise, the numerical experiments reveal that the solutions to the cases with more stages are closer to the operational ones because the hydropower system operation is more realistic. For example, Figs. 8 and 9 show that the solutions with 5 and 7 stages lead to a similar improvement, but the implementation with 7 stages is preferable. Fig. 9a, b and d presents reservoir volumes that are more stable than Fig. 8a, b and d.

The October data set is the only one for which the median scenario produces more energy for all number of stages. The interest of a stochastic method is to account for uncertainty in the future. As we compare our method with the median scenario, if the actual realization of the inflows is close to the median scenario, the stochastic solution will not produce more energy, as the median scenario depicts correctly the future. In practice, this may happen during the fall period, for example when low variability exists in the weather and storms have less chances of developing. This can be seen on Fig. 10. Each subfigure corresponds to a reservoir. Fig. 10a is Chute-du-Diable. The top figure is the day 1 October forecast and the bottom figure is the day 1 September forecast. For the first 15 days, the October forecast median scenario is very close to the inflow realization and therefore, as we keep the day 1 decision only, the median scenario produces more energy. The other subfigures are represented in the same fashion. Again, Fig. 10b and c shows that for Chute-Savane and Lac-St-Jean, the actual inflows in October are very close to the median scenario, therefore there is no gain in using a stochastic optimization model, as the deterministic median scenario allows to obtain a good solution. For this unusual October case, solving the short-term unit commitment and loading problem with a median scenario is acceptable. This affirmation is to be used with caution as situations like these have



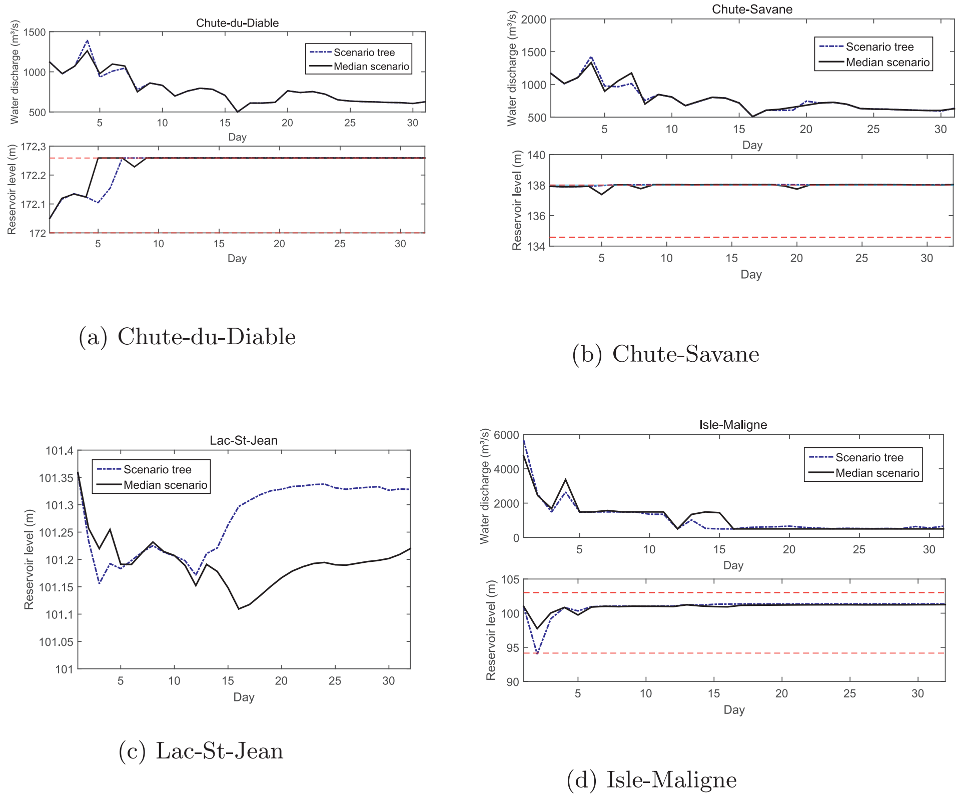
(a) Chute-du-Diable

(b) Chute-Savane

(c) Lac-St-Jean

(d) Isle-Maligne

Fig. 8. Water discharges and reservoir levels for the case June 2011, 5 stages, 16 scenarios.



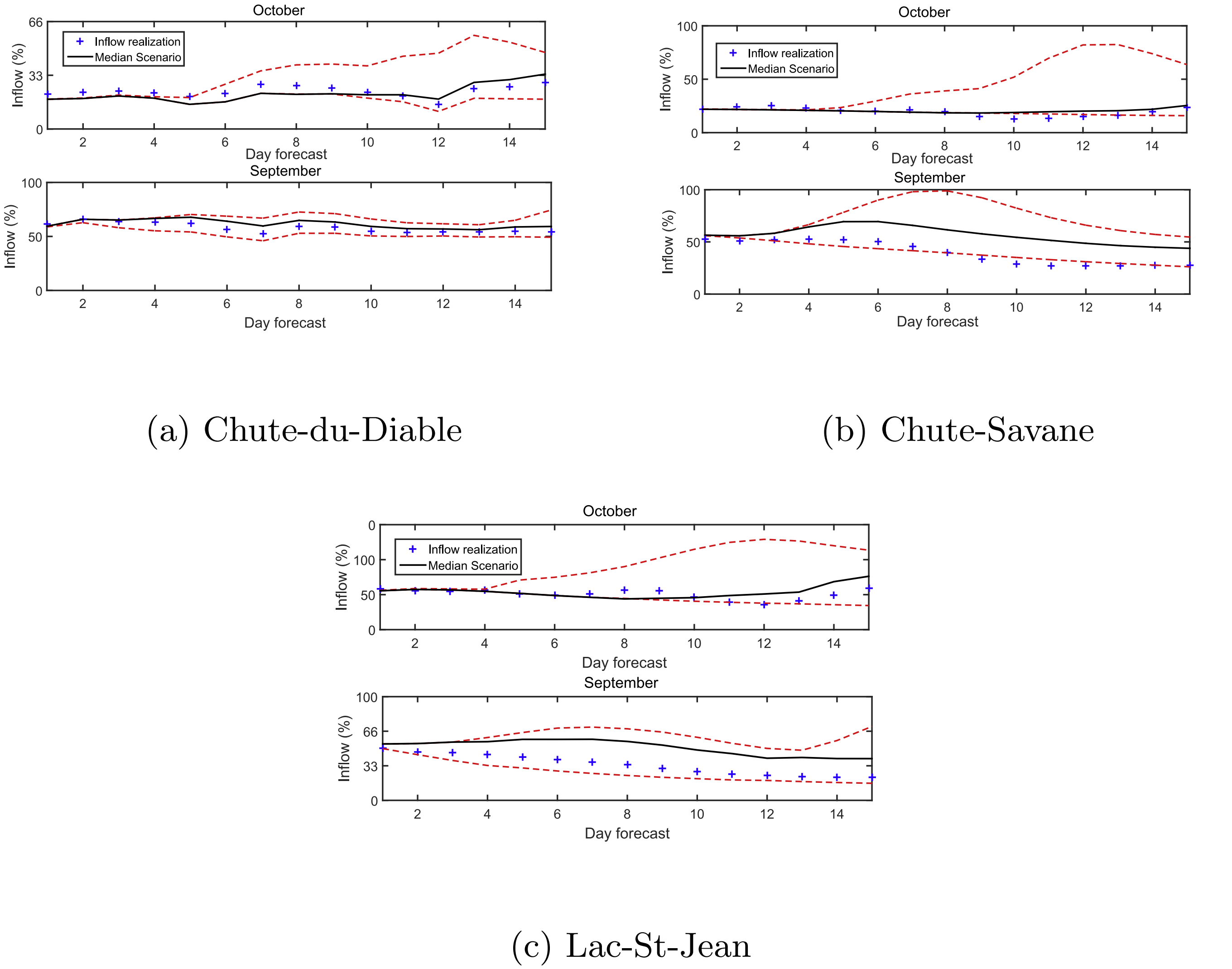
(a) Chute-du-Diable

(b) Chute-Savane

(c) Lac-St-Jean

(d) Isle-Maligne

Fig. 9. Water discharges and reservoir levels for the case June 2011, 7 stages, 16 scenarios.



(a) Chute-du-Diable

(b) Chute-Savane

(c) Lac-St-Jean

Fig. 10. Comparison of September (lower figures in each subfigure) and October (upper figures in each subfigure) day 1 data sets. The dashed lines are the minimum and maximum scenarios. The median scenario is the solid line. The actual realization of the inflows is the plus sign line.

a low probability of occurring. These results show that there is certainly a gain in using a stochastic model for the short-term hydropower optimization model, as relying on the median scenario offers a less robust solution than multiple scenarios.

6. Conclusion

This paper presents a stochastic short-term hydropower optimization method which emphasizes inflow scenario trees. Few papers looked specifically into stochastic short-term models and we extend the modeling presented in Séguin et al. (2016) to consider uncertain inflows. The optimization method considers inflow uncertainty, head variations and nonlinear and nonconvex relationship between discharge and power output. The scenario tree generation method first uses kernel density estimation to generate random values of inflows. Then, the path of nodes, from root to leaf, that minimizes the Wasserstein distance is found in the scenario tree and the corresponding nodes are updated using stochastic approximation. The process is repeated until the termination criterion, which is the convergence of the tree in Wasserstein distance, has been reached. A stability test has shown that the scenario tree generation method is consistent. A highlight of this method is that it uses all data available at each iteration to improve the values of the scenario tree nodes. The scenario trees are inputs to a two-phase optimization process. The first phase, loading problem, allows to find water discharge, volume and number of turbines working in each plant. The second phase, unit commitment, chooses the exact combination of turbines to use, to maximize energy production and penalize unit startups. A major feature of this modeling of the problem is that the water head is not neglected. For this paper, the models are tested on three hydropower plants. A rolling-horizon procedure is retained on a 31 day planning horizon. The stochastic solution is compared to the median scenario. Furthermore, fast computation time allows this method to be scaled in order to be applied in full to the Saguenay-Lac-St-Jean hydroelectric system. Future work based on this paper consists on investigating the complexity required in the scenario tree structure. Since a rolling-horizon framework is retained and that only the solution of the first-stage is kept, tests with scenario fans instead of scenario trees will be conducted.

Acknowledgments

The authors would like to thank Marco Latraverse, and Rio Tinto, for providing data necessary to this study. This work was supported by NSERC, FRQNT and Rio Tinto. Also, a grant for international mobility, awarded by FRQNT through GERAD allowed the research to be conducted at the Norwegian University of Science and Technology. Stein-Erik Fleten acknowledges financial support from the Research Council of Norway through project 243964/E20. Alois Pichler gratefully acknowledges support of the Research Council of Norway through grant 207690/E20. The authors are thankful to two anonymous referees, for providing constructive comments that greatly improved the paper.

Appendix A. Notation

The following notation is used throughout the paper:

N	set of nodes
E	set of leaf nodes
K	set of scenarios
C	set of hydroelectric plants
u^c	number of hydroelectric plants upstream of plant $c \in C$
$s \in \{1, 2, \ldots, n_i^c\}$	index of surfaces corresponding to number of active turbines with hydroelectric plant c and node i
$l \in \{1, 2, \ldots, n_i^c\}$	index of combinations with hydroelectric plant c and node i
$t \in \{1, 2, \ldots, T^c\}$	index of turbines of hydroelectric plant c
π_j^c	probability of scenario j for plant c
ν_i^c	volume of plant reservoir c at node i (cubic hectometers)
q_i^c	water discharge at plant c and node i (cubic meter per second)
θ	start-up penalty for any turbine (megawatt)
β_{li}^c	power generated by combination $l \in n_i^c$ at plant c and node i
$y_{si}^c =$	$\begin{cases} 1 \text{ if surface } s \text{ is chosen at node } i \\ \text{for plant } c \\ 0 \text{ otherwise} \end{cases}$
$f_{lit}^c =$	$\begin{cases} 1 \text{ if turbine } t \text{ of combination } l \\ \text{for plant } c \text{ is active at node } i \\ 0 \text{ otherwise} \end{cases}$
$x_{li}^c =$	$\begin{cases} 1 \text{ if combination } l \text{ of plant } c \\ \text{is chosen at node } i \\ 0 \text{ otherwise} \end{cases}$
$d_{ti}^c =$	$\begin{cases} 1 \text{ if turbine } t \text{ of plant } c \text{ is started} \\ \text{at node } i \\ 0 \text{ otherwise} \end{cases}$
χ_{si}^c	power for surface s at node i and plant c (megawatt)
$\Psi_s^{Ac}(q_i^c, \nu_i^c)$	power production function without spillage for surface s and plant c
$\Psi_s^{Bc}(q_i^c, \nu_i^c)$	power production function with spillage for surface s and plant c
δ_i^c	inflow of plant c at node i (cubic meter per second)
w_i	duration of node i (h)
γ	conversion factor from water discharge (cubic meter per second) to (cubic hectometers per hour)
$\Phi_j^c(\nu_j^c)$	water-value function for plant c and scenario j
ζ_i	conversion factor to energy units (gigawatt hours)
ν_{min}^c	minimal volume of plant c reservoir (cubic hectometers)
ν_{max}^c	maximum volume of plant c reservoir (cubic hectometers)
q_{min}^c	minimum water discharge at plant c (cubic meter per second)
q_{max}^c	maximum water discharge at plant c (cubic meter per second).

References

Aasgård, E. K., Andersen, G. S., Fleten, S.-E., & Haugstvedt, D. (2014). Evaluating a stochastic-programming-based bidding model for a multireservoir system. *IEEE Transactions on Power Systems, 29*(4), 1748–1757. doi:10.1109/TPWRS.2014.2298311.

Apparigliato, R. (2008). *Règles de décision pour la gestion du risque: Application à la gestion hebdomadaire de la production électrique*. Ecole Polytechnique X (Ph.D. thesis).

Arce, A., Ohishi, T., & Soares, S. (2002). Optimal dispatch of generating units of the Itaipu hydroelectric plant. *IEEE Transactions on Power Systems, 17*(1), 154–158. doi:10.1109/59.982207.

Babonneau, F., Vial, J.-P., & Apparigliato, R. (2010). *Uncertainty and environmental decision making: A handbook of research and best practice* (pp. 79–126). Boston, MA: Springer US. doi:10.1007/978-1-4419-1129-2_3.

Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer. doi:10.1007/978-1-4614-0237-4.

Boomsma, T. K., Juul, N., & Fleten, S.-E. (2014). Bidding in sequential electricity markets: The Nordic case. *European Journal of Operational Research, 238*(3), 797–809.

Brisson, C., Boucher, M.-A., & Latraverse, M. (2015). Illustration of the added value of using a multi-site calibration and correction approach to reconstruct natural inflows and inter-catchment transfer flow: a case study. *Canadian Journal of Civil Engineering, 42*(5), 342–352. doi:10.1139/cjce-2014-0270.

Côté, P., Haguma, D., Leconte, R., & Krau, S. (2011). Stochastic optimisation of Hydro-Quebec hydropower installations: a statistical comparison between SDP and SSDP methods. *Canadian Journal of Civil Engineering, 38*(12), 1427–1434. doi:10.1139/l11-101.

Côté, P., & Leconte, R. (2015). Comparison of stochastic optimization algorithms for hydropower reservoir operation with ensemble streamflow prediction. *Journal of Water Resources Planning and Management, 142*(2). doi:10.1061/(ASCE)WR.1943-5452.0000575.

De Ladurantaye, D., Gendreau, M., & Potvin, J.-Y. (2009). Optimizing profits from hydroelectricity production. *Computers and Operations Research, 36*(2), 499–529. doi:10.1016/j.cor.2007.10.012.

Dupačová, J., Consigli, G., & Wallace, S. W. (2000). Scenarios for multistage stochastic programs. *Annals of Operations Research, 100*(1–4), 25–53. doi:10.1023/A:1019206915174.

Environment Canada prevision models. http://collaboration.cmc.ec.gc.ca/cmc/CMOI/product_guide/docs/changes_f.html (Accessed: 21 August 2015).

Feng, Y., & Ryan, S. M. (2014). Scenario reduction for stochastic unit commitment with wind penetration. In *PES general meeting | conference exposition* (pp. 1–5). IEEE. doi:10.1109/PESGM.2014.6939138.

Finardi, E. C., & da Silva, E. L. (2006). Solving the hydro unit commitment problem via dual decomposition and sequential quadratic programming. *IEEE Transactions on Power Systems, 21*(2), 835–844. doi:10.1109/TPWRS.2006.873121.

Fleten, S.-E., & Kristoffersen, T. K. (2008). Short-term hydropower production planning by stochastic programming. *Computers and Operations Research, 35*(8), 2656–2671. doi:10.1016/j.cor.2006.12.022.

Fourer, R., Gay, D. M., & Kernighan, B. W. (2003). *AMPL: A modeling language for mathematical programming* (2nd). Pacific Grove, California: Thomson/Brooks/Cole.

Growe-Kuska, N., Heitsch, H., & Romisch, W. (2003). Scenario reduction and scenario tree construction for power management problems. In *Power tech conference proceedings, 2003 IEEE Bologna: vol. 3* (p. 7). doi:10.1109/PTC.2003.1304379.

Høyland, K., & Wallace, S. W. (2001). Generating scenario trees for multistage decision problems. *Management Science, 47*(2), 295–307. doi:10.1287/mnsc.47.2.295.9834.

Jones, M. C. (1990). The performance of kernel density functions in kernel distribution function estimation. *Statistics & Probability Letters, 9*(2), 129–132. http://dx.doi.org/10.1016/0167-7152(92)90006-Q.

Kaut, M., & Wallace, S. W. (2003). Evaluation of scenario-generation methods for stochastic programming. In *Stochastic Programming E-Print Series*. Institut für Mathematik. Number 14.

King, A. J., & Wallace, S. W. (2012). Scenario-tree generation: With Michal Kaut. In *Modeling with stochastic programming*. In *Springer Series in Operations Research and Financial Engineering* (pp. 77–102). New York: Springer. doi:10.1007/978-0-387-87817-1_4.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory, 28*, 129–137.

Lohmann, T., Hering, A. S., & Rebennack, S. (2016). Spatio-temporal hydro forecasting of multireservoir inflows for hydro-thermal scheduling. *European Journal of Operational Research, 255*(1), 243–258. http://dx.doi.org/10.1016/j.ejor.2016.05.011.

MATLAB (2015). *Version 8.5.0.197613 (R2015a)*. Natick, Massachusetts: The MathWorks Inc.

Pflug, G. Ch., & Pichler, A. (2015). Dynamic generation of scenario trees. *Computational Optimization and Applications, 62*(3), 641–668.

Pflug, G. Ch., & Pichler, A. (2016). From empirical observations to tree models for stochastic optimization: convergence properties. *SIAM Journal on Optimization, 26*(3), 1715–1740. doi:10.1137/15M1043376.

Philpott, A. B., Craddock, M., & Waterer, H. (2000). Hydro-electric unit commitment subject to uncertain demand. *European Journal of Operational Research, 125*(2), 410–424. doi:10.1016/S0377-2217(99)00172-1.

Schwanenberg, D., Fan, F. M., Naumann, S., Kuwajima, J. I., Montero, R. A., & Assis dos Reis, A. (2015). Short-term reservoir optimization for flood mitigation under meteorological and hydrological forecast uncertainty. *Water Resources Management, 29*(5), 1635–1651. doi:10.1007/s11269-014-0899-1.

Séguin, S., Côté, P., & Audet, C. (2016). Self-scheduling short-term unit commitment and loading problem. *IEEE Transactions on Power Systems, 31*(1), 133–142. doi:10.1109/TPWRS.2014.2383911.

Shapiro, A. (2011). Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research, 209*(1), 63–72. http://dx.doi.org/10.1016/j.ejor.2010.08.007.

Silverman, B. W. (1986). Density estimation for statistics and data analysis. *Chapman & Hall/CRC Monographs on Statistics & Applied Probability*. Taylor & Francis.

Siqueira, T. G., Zambelli, M., Cicogna, M., Andrade, M., & Soares, S. (2006). Stochastic dynamic programming for long term hydrothermal scheduling considering different streamflow models. In *International conference on probabilistic methods applied to power systems, 2006. PMAPS 2006* (pp. 1–6). doi:10.1109/PMAPS.2006.360203.

Taktak, R., & D'Ambrosio, C. (2016). An overview on mathematical programming approaches for the deterministic unit commitment problem in hydro valleys. *Energy Systems*, 1–23. doi:10.1007/s12667-015-0189-x.

Tejada-Guibert, J. A., Johnson, S. A., & Stedinger, J. R. (1993). Comparison of two approaches for implementing multireservoir operating policies derived using stochastic dynamic programming. *Water Resources Research, 29*(12), 3969–3980. doi:10.1029/93WR02277.

van Ackooij, W., Henrion, R., Möller, A., & Zorgati, R. (2013). Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering, 15*(2), 509–531. doi:10.1007/s11081-013-9236-4.

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming, 106*(1), 25–57. doi:10.1007/s10107-004-0559-y.

XPRESS, Optimization Suite, Fair Isaac Corporation (FICO). www.fico.com/en/products/fico-xpress-optimization-suite/.

Xu, B., Zhong, P.-A., Zambon, R. C., Zhao, Y., & Yeh, W. W.-G. (2015). Scenario tree reduction in stochastic programming with recourse for hydropower operations. *Water Resources Research, 51*(8), 6359–6380.