

# Energy Systems

## A policy relaxation algorithm for seasonal hydropower planning

--Manuscript Draft--

<b>Manuscript Number:</b>	ENSY-D-22-00443	
<b>Full Title:</b>	A policy relaxation algorithm for seasonal hydropower planning	
<b>Article Type:</b>	Original Paper	
<b>Keywords:</b>	Stochastic programming; Policy relaxation; Hydropower reservoir management	
<b>Corresponding Author:</b>	Håkon Skirstad Grini, MS Norwegian University of Science and Technology: Norges teknisk-naturvitenskapelige universitet Trondheim, NORWAY	
<b>Corresponding Author Secondary Information:</b>		
<b>Corresponding Author's Institution:</b>	Norwegian University of Science and Technology: Norges teknisk-naturvitenskapelige universitet	
<b>Corresponding Author's Secondary Institution:</b>		
<b>First Author:</b>	Håkon Skirstad Grini	
<b>First Author Secondary Information:</b>		
<b>Order of Authors:</b>	Håkon Skirstad Grini	
	Anders Strømmen Danielsen	
	Stein-Erik Fleten	
	Andreas Kleiven	
<b>Order of Authors Secondary Information:</b>		
<b>Funding Information:</b>	Norges Teknisk-Naturvitenskapelige Universitet	Student Researcher Håkon Skirstad Grini
<b>Abstract:</b>	<p>Hydropower producers need to plan several months or years ahead to estimate the opportunity value of water stored in their reservoirs. The resulting large-scale optimization problem is computationally intensive, and model simplifications are often needed to allow for efficient solving. Alternatively, one can look for near-optimal policies using heuristics that can tackle non-convexities in the production function and a wide range of modelling approaches for the price- and inflow dynamics. We undertake an extensive numerical comparison between the state-of-the-art algorithm stochastic dual dynamic programming (SDDP) and rolling forecast-based algorithms, including a novel algorithm that we develop in this paper. We name it Scenario-based Two-stage ReOptimization abbreviated as STRO. The numerical experiments are based on convex stochastic dynamic programs with discretized exogenous state space, which makes the SDDP algorithm applicable for comparisons. We demonstrate that our algorithm can handle inflow risk better than traditional forecast-based algorithms, by reducing the optimality gap from 2.5% to 1.3% compared to the SDDP bound.</p>	

[Click here to view linked References](#)

# A policy relaxation algorithm for seasonal hydropower planning

Håkon S. Grini<sup>1</sup>, Anders S. Danielsen<sup>1</sup>, Stein-Erik Fleten<sup>1</sup>, and Andreas Kleiven<sup>1</sup>

Norwegian University of Science and Technology, Trondheim 7491, Norway

**Abstract.** Hydropower producers need to plan several months or years ahead to estimate the opportunity value of water stored in their reservoirs. The resulting large-scale optimization problem is computationally intensive, and model simplifications are often needed to allow for efficient solving. Alternatively, one can look for near-optimal policies using heuristics that can tackle non-convexities in the production function and a wide range of modelling approaches for the price- and inflow dynamics. We undertake an extensive numerical comparison between the state-of-the-art algorithm stochastic dual dynamic programming (SDDP) and rolling forecast-based algorithms, including a novel algorithm that we develop in this paper. We name it Scenario-based Two-stage ReOptimization abbreviated as STRO. The numerical experiments are based on convex stochastic dynamic programs with discretized exogenous state space, which makes the SDDP algorithm applicable for comparisons. We demonstrate that our algorithm can handle inflow risk better than traditional forecast-based algorithms, by reducing the optimality gap from 2.5% to 1.3% compared to the SDDP bound.

**Keywords:** Stochastic programming · Policy relaxation · Hydropower reservoir management.

## Acknowledgments

We gratefully acknowledge support from the research center HydroCen, RCN No. 257588. We are thankful to Montel for supplying price data, and to Eviny for supplying data regarding their hydropower plants.

## 1 Introduction

Sequential decision-making in the presence of uncertainty is of relevance in many diverse applications, including inventory control problems, production scheduling, and system modelling [3]. Typically, the goal is to minimize cost or maximize profits by dynamically adapting decisions to exogenous information. These problems are often computationally intractable because of exponential growth in the

number of scenarios needed to evaluate future expected costs or profits. Therefore, approximate dynamic programming (ADP) algorithms are often used to obtain near-optimal policies [22].

In some applications, one can utilize the problem structure in developing computationally tractable algorithms, such as decomposition-based algorithms [29, 15]. The stochastic dual dynamic programming algorithm (SDDP), is an example [19]. This algorithm is state of the art for solving the seasonal hydropower production planning problem [11, 25]. The algorithm requires the problem formulation to be convex and uncertainty to be stage-wise independent [20]. Therefore, certain aspects of the real problem need to be approximated, to adhere to the convexity requirement. To handle non-convexities, some recent papers have proposed to extend the algorithm [14, 10], or to approximate the problem using McCormick envelopes [6]. To adhere to the requirement of stage-wise independence, serial correlation is typically accounted for by either imposing a linear dependence structure and increasing the state vector, or discretizing the state space before applying the algorithm [18]. The first approach greatly restricts the dynamics of exogenous factors, while the latter approach induces a discretization error.

An alternative class of algorithms that often provide good policies is forecast-based reoptimization heuristics, e.g. the rolling intrinsic (RI) heuristic [5]. The heuristic repeatedly solves deterministic problems based on the expectation of exogenous factors. When averaging revenues from sufficiently many sample paths, it provides a feasible policy and an estimated lower bound on maximization problems. Reoptimization heuristics are widely used by practitioners in various fields. In gas storage management, numerical experiments indicate that the rolling intrinsic algorithm provides near-optimal policies for price- and storage-dependent injections and withdrawals [16, 30]. Despite the ability to handle complex relationships, few works have assessed the performance of reoptimization-based heuristics in hydropower scheduling. An exception, albeit for short-term operations, is [17].

In this work, our main contributions are to provide an extensive comparison of the rolling intrinsic algorithm, the state-of-the-art algorithm SDDP, and a novel, heuristic-based, algorithm that does not impose any restrictions on the modelling of exogenous factors. We call this algorithm Scenario-based Two-stage ReOptimizaton (STRO). The main innovative aspect is that the requirement that the algorithm must yield a policy is relaxed. Still, the algorithm provides an estimate of the lower bound on the optimal value, from which marginal water values can be derived. This concept is referred to as *policy relaxation*.

STRO samples possible future outcomes and solves two-stage programs repeatedly to make decisions. The idea is that this will better capture the range of possible realizations of exogenous factors, compared to the rolling intrinsic heuristic which uses a deterministic forecast based on expectations. Capturing the range of possible realizations is crucial in applications where all risk factors cannot be hedged, e.g. a hydropower producer which may risk spillage. In the setting of hydropower scheduling using SDDP, our algorithm allows for a more

realistic representation of inflow dynamics, e.g. by using dynamic artificial neural networks [28], as no restrictions are imposed on the modelling of the exogenous factors. Another benefit of the algorithm is the opportunity for scalable parallel processing. Because scenarios are generated independently, their respective policies can be computed individually. The use of parallelization allows for a drastic reduction in the processing time required. As seen in previous works, traditional parallelization schemes do not scale well for SDDP [1, 13]. Finally, the algorithm does not explicitly require the problem formulation to be convex, unlike SDDP. We provide an extensive numerical comparison of optimality gaps and computation time of the proposed algorithm, the state-of-the-art algorithm SDDP, and the rolling intrinsic algorithm.

The most important findings are that both the RI and STRO methods achieve revenues that are less than 3% from the estimate of the upper bound on optimal revenue based on several trial runs of the SDDP algorithm. Furthermore, by increasing the number of samples per decision for the STRO method, the performance of the algorithm increases substantially. When using 2 or more samples per decision, STRO outperforms RI by a decent margin and achieves revenues less than 2% from the estimated upper bound. The benefit of increasing the number of samples for the STRO heuristic is largest when moving from 1 to 2 samples, and the benefit from increasing the number of samples further was found to diminish quickly.

The paper is structured as follows. First, a detailed mathematical formulation of the problem is given, along with an in-depth problem description, in section 2. Then the novel heuristic is presented in section 3. We then give an illustrative example of how this heuristic works in section 4. In section 5 we give an overview of the specific case the algorithms were tested on, as well as the software and hardware used. Finally, the results of our experiments are outlined in section 6 before we give our final remarks in section 7.

## 2 Mathematical formulation

In this section, we provide a problem description and describe the stochastic processes and a mathematical model for the seasonal hydropower planning problem.

### 2.1 Problem description

We consider the medium-term reservoir management problem, where the objective is to value-maximize production over one to two years. The production decisions for medium-term reservoir management are typically of a weekly granularity. Modern hydropower plants commonly consist of multiple interconnected reservoirs, that allow for coordinated water release that maximizes profit for the system as a whole. The state of the system at a given time is given by the amount of water that is stored in the different reservoirs. Furthermore, as regulations can put constraints on the water stored on a seasonal basis, these constraints can vary for different times  $t$ .

Several simplifications and assumptions are often made when modeling the reservoir management problem. This includes ignoring operating costs, as these variable costs are usually negligible compared to the revenue. This makes the problem a revenue-maximizing one. Another common simplification is to make the energy coefficient constant, to keep the problem convex. This coefficient is the factor that gives how much energy a production facility can get from a single unit of water. In reality, this factor is dependent on the head (the height difference between the turbine and the variable water level), as well as the intensity of the flow through the turbine. The head can vary substantially with the amount of water in the reservoir. In the sample system, the variations represented between 4.0% and 6.5% of the overall head for the different reservoirs, but these variations can be substantially higher or lower depending on the system. Since potential energy increases linearly with the head, this incentivizes producers to keep their reservoir levels high, thereby generating more electricity per unit of water. This poses an interesting dynamic, as the risk of spillage increases with higher water levels. To capture this dynamic, head variations are implemented in the model with a variable energy coefficient. This leads to bi-linearity which is accounted for using McCormick envelopes, similarly to [6].

### Marginal water values

The main goal of the medium-term reservoir management problem is to estimate marginal water values. These values naturally depend on the amount of water in the system. To illustrate this one can compare the additional value of an extra unit of water in a full reservoir and an empty reservoir. In the case of a full reservoir, one would have to discharge the unit immediately, independently of price, to avoid spillage. If the full reservoir is already producing at full capacity, the additional unit would naturally be spilt, and the marginal value would be 0. For the other extreme case where the reservoir is empty, the marginal value would be much higher as the producer could wait until the prices are high to discharge the unit, without having to worry about spillage. In practice, that means that the marginal value of water decreases, as the water level in the reservoir increases.

The marginal water value represents the current alternative cost of discharging a unit and is therefore used to make short-term production decisions. The way the medium-term reservoir management problem relates to this is that one can calculate the expected discounted revenue over a 1-2 year horizon starting with different water levels. If one plots the expected revenue against the starting water level, the slope of the curve would represent the current marginal water value for different water levels.

Note that this implies that the objective of the medium-term reservoir management problem is to estimate revenue and therefore finding an implementable policy is unnecessary. As a result, the requirement that the optimization method must yield an implementable policy can be relaxed without negatively impacting the utility.

## 2.2 Nomenclature

Before formally defining the optimization model, we provide the notation used for the model in the following section. Some of the notation is only used for the additional constraints seen in the appendix.

### Sets and Indices

$R$	-	Reservoirs, indexed by $i$ and $j$
$R^P$	-	Reservoirs with electricity production
$R_i^{Dis}$	-	Upstream reservoirs releasing water into reservoir $i$
$R_i^{Spill}$	-	Upstream reservoirs spilling water into reservoir $i$
$S_i$	-	Set of points on the piece-wise linear curve representing the relation between head and water level for reservoir with electricity production $i$ , indexed by $k$
$\Pi$	-	Set of feasible policies

### Decision Variables

$x_{i,t}$	-	Discharge from reservoir $i$ in period $t$	$[m^3]$
$r_{i,t}$	-	Slack variable for water spillage from reservoir $i$ in period $t$	$[m^3]$
$l_{i,t}$	-	Water level in reservoir $i$ at the beginning of time period $t$	$[m^3]$
$l_{i,t}^{avg}$	-	Average water level in reservoir $i$ through time period $t$	$[m^3]$
$h_{i,t}$	-	Water head of reservoir $i$ in period $t$	$[m]$
$w_{i,t}$	-	Substitution variable for the bi-linear term $h_{i,t}x_{i,t}$	$[m]$
$\lambda_{i,t,m}$	-	Weight of point $k$ on the piece-wise linear graph describing the relation between water level and head in reservoir $i$ at time $t$	

**Parameters**

$T$	-	Stages (time periods) in planning horizon	
$\beta$	-	Discount factor	
$\eta_i$	-	constant factor describing the efficiency of the turbine of reservoir $i$	$[MWh/m^3]$
$g$	-	gravitational acceleration constant	$[m/s^2]$
$\rho$	-	density of water	$[Kg/m^3]$
$P_t$	-	Power price in stage $t$	$[€/MWh]$
$I_{i,t}$	-	Water inflow intensity for reservoir $i$ in period $t$	$[m^3]$

**2.3 Stochastic processes**

As restructured energy markets have uncertain prices for different times, modellers have to consider the exogenous price denoted  $P_i$ . Furthermore, as weather cannot be forecasted perfectly, the inflow to reservoirs is uncertain. The set of all reservoirs is denoted by  $R = \{1, 2, \dots, |R|\}$ . We denote inflow to reservoir  $i \in R$  at time  $t$  by  $I_{i,t}$ .

For prices, we apply the Schwartz-Smith two-factor model [23]. This model assumes that the logarithm of the price can be modelled as the sum of a short-term factor, a long-term factor, and finally a seasonal factor, as the energy price typically follows a seasonal pattern. The model parameters are estimated using Kalman filtering and maximum likelihood estimation [12]. Data points are obtained from synthetic futures curves [8] using the approach in [2].

The inflow model is primarily based on the one seen in [11], where the inflow is modelled according to AR-1 processes. To reduce dimensionality and capture the covariance of inflow to different reservoirs, a PCA transformation of the inflow data is performed to project it to a lower dimensionality, as seen in [7]. The original inflow data is normalized to follow a standard normal distribution, before being transformed using PCA. After fitting AR-1 models to the lower dimension data, one can produce scenarios using the model for every dimension  $k$  in the projected space. After generating the low-dimensional scenarios, we can transform the generated scenario back to the original dimensionality, using the PCA matrix. We then denormalize this data, to ensure that the generated scenarios follow the distribution of the historical observations. To circumvent the issue of negative inflows generated by the AR model, negative values are converted to 0 according to equation (1).

$$I_{i,t} = \max(I_{i,t}, 0) \quad (1)$$

## 2.4 Actions and transition function

We present here stage- $t$  actions and transition functions. As only a subset of these reservoirs have connected turbines allowing for energy production, we denote this by subset  $R^P = \{1, 2, \dots, |R^P|\}$ . These sets are both indexed by  $i$  and  $j$ , for the modelling of interconnections between reservoirs. Every reservoir  $i$ , have two sets of reservoirs  $R_i^{Dis}$  and  $R_i^{Spill}$ , that represent the reservoirs whose discharged water flows directly into reservoir  $i$  and reservoirs whose spilt water flows into  $i$ , respectively.

The vector of decisions in stage  $t$  includes discharge  $x_{i,t}$  and spillage  $r_{i,t}$  for each station and reservoir, where  $\Pi_t(l_t, I_t)$  is the feasible stage- $t$  action set and  $(x_{i,t}, r_{i,t}, i \in R) \in \Pi_t(l_t, I_t)$ . The discharge variable has both upper and lower bounds depending on physical limitations and regulations that can be seasonal. The spillage and the discharge can naturally not be negative.

The transition function consists of three components  $(l_t, P_t, I_t)$ . The exogenous components, includes price  $P_t$  and the vector of inflows,  $I_t = (I_{i,t}, i \in R)$ . These factors get updated according to the stochastic processes discussed in the previous section, independently of the stage- $t$  action. The endogenous component is the vector of reservoir volumes in the watercourse,  $l_t = (l_{i,t}, i \in R)$ . Executing an action  $(x_t, r_t)$  at stage  $t$  and state  $(l_t, P_t, I_t)$  leads to the following update of the endogenous state,

$$l_{i,t+1} = l_{i,t} + I_{i,t} - r_{i,t} - x_{i,t} + \sum_{j \in R_i^{Spill}} r_{j,t} + \sum_{j \in R_i^{Dis}} x_{j,t}, \quad i \in R. \quad (2)$$

This function ensures energy balance in time and the topology of the watercourse. It gives the relationship between the inflow, production decisions, spillage, and the water level of a given reservoir  $i$ . It ensures that spillage happens when the net inflow is higher than the reservoir capacity. Due to the longer time steps taken with medium-term reservoir management, water delay is not taken into account, and it is assumed that water can flow through the entire system within a single time step.

## 2.5 Immediate reward and policy

At each stage, the immediate reward is given by electricity production and electricity price (3)

$$P_t \sum_{i \in R^P} g(l_{i,t}, x_{i,t}), \quad (3)$$

where  $g(l_{i,t}, x_{i,t})$  is the energy output from discharge  $x_{i,t}$  at reservoir level  $l_{i,t}$ . We denote the set of feasible policies by  $\Pi$ . A policy  $\pi$  is a collection of stage-dependent actions, mapping states at time  $t$  to feasible actions. We let  $l_{i,t}^\pi$  and  $x_{i,t}^\pi$  respectively denote the endogenous state reached at, and action made in, stage  $t$  for reservoir  $i$ , when following policy  $\pi$ . We aim to find a policy that



maximizes the expected accumulated discounted reward,

$$V_0(l_0, m_0) = \max_{\pi \in \Pi} \mathbb{E} \left[ \sum_{t \in N} \beta^t P_t \sum_{i \in R^P} g(l_{i,t}^\pi, x_{i,t}^\pi) + \beta^T \sum_{i \in R} \Phi_i(l_{i,T}^\pi) \middle| m_0 \right], \quad (4)$$

where  $\beta$  is the discount factor and  $\Phi(l_{i,T})$  is the end of horizon value for each reservoir, given by how much water is remaining.

## 2.6 Relaxation and approximations

The reward function contains the term  $g(l_{i,t}, x_{i,t})$ . This function is typically concave in  $l_{i,t}$  and  $x_{i,t}$ . The higher the reservoir, the higher the generation output. Similarly, as a function of discharge, the efficiency of a turbine is first increasing and then possibly decreasing. Piece-wise linear approximations of the discharge function can handle the latter feature. In our model, we assume it to be a constant factor for each reservoir  $i$  denoted  $\eta_i \in [0, 1]$ , i.e.  $g(l_{i,t}, x_{i,t}) = g(l_{i,t})x_{i,t}\eta_i$ . The dependency of reservoir volume, i.e. head variations, is complicated since it leads to a bi-linear objective and thereby non-convexity. This makes it challenging to optimize with conventional methods such as SDDP, which we use as a baseline. To deal with this issue, we relax the problem using McCormick envelopes. We start by modifying the reward in (3) to

$$P_t \sum_{i \in R^P} h_i(l_{i,t})x_{i,t}g\rho\eta_i \quad (5)$$

The functions  $h_i(\cdot)$  represent the head in reservoir  $i$ , while  $g$  is the gravitational acceleration constant and  $\rho$  is the density of water. The  $h_i(\cdot)$  functions depend on  $l_{i,t}$  and is calculated using piece-wise linear approximations for the individual reservoirs. These piece-wise linear graphs are represented as a sets of mappings between the water level to the head for reservoir  $i$ , denoted  $S_i$ . To find the head  $h_{i,t}$  given the water level  $l_{i,t}$  we introduce the variables  $\lambda_{i,t,k}$  representing the weight of every point in the set  $S_i$ . By ensuring that all these weights are non-negative and sum to 1, and that at most two points adjacent to each other can be non-zero, we can find the exact point on the y-axis on the piece-wise linear graph, given the value on the x-axis.

The restriction requiring that at most two points can be non-zero, and that these must be adjacent to each other, is called a special ordered set of type 2 (SOS2) restriction. This restriction can be ignored in cases where the piece-wise linear function is concave, as the head  $h_{i,t}$  would be the same with or without the SOS2 constraint. The approximated graphs for the energy-producing reservoirs in the Matre Haugsdal system can be seen in Figure 1. It can be seen that these are all concave, ensuring that the SOS2 constraints are unnecessary.

The piece-wise linear approximation of the head function  $h_i(l_{i,t})$  is explained in further detail in Appendix A. This approximation leads to bi-linear terms in (5). To approximate these terms, we introduce the substitution variable  $w_{i,t} =$

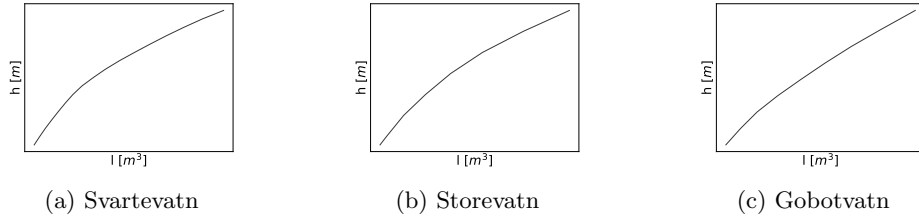


Fig. 1: Volume-head relations for reservoirs with production

$h_i(l_{i,t})x_{i,t}$  and use McCormick envelope constraints. This gives us the modified linear reward

$$P_t \sum_{i \in R^P} w_{i,t} g \rho \eta_i, \quad (6)$$

which leads to a convex and linear multistage optimization problem, satisfying the requirements of SDDP. For additional constraints associated with the McCormick relaxation, see Appendix B.

### 3 Scenario-based Two-stage Re-Optimization heuristic

One of the main ideas behind rolling horizon methods, such as RI, is that a finite number of sampled scenarios can be used to approximate the sample space of a continuous-state problem. The RI heuristic has shown promising performance in various applications and is simple and intuitive. We utilize a similar concept as the RI heuristic in developing a novel Scenario-based Two-stage Re-Optimization heuristic (STRO). Instead of using expectations of exogenous variables as forecasts from any given state, we propose to generate possible scenarios from the stochastic model and repeatedly solve two-stage problems based on these scenarios.

By generating  $N$  possible scenarios from time  $t$  and state  $(l_t, m_t)$  to the end of the horizon and building and solving a two-stage stochastic program from these, one can find a first-stage decision before moving to the next stage. To distinguish between the outer scenarios that are the realizations of the environment that are unknown to the heuristic, and the inner scenarios that are generated to build the two-stage stochastic programs that the heuristic solves to find the first-stage decision, we refer to the outer scenarios as level 1 scenarios and the inner ones as level 2 scenarios.

A visualization of how a two-stage stochastic program is built from generated level 2 scenarios to explore the underlying sample space can be seen in Figure 2. Starting in a state in stage  $\tau$ , one can generate  $N$  possible realizations of the future, starting in the current state. Then, these scenarios can be combined into a two-stage stochastic program, which can be solved to find the first-stage decision.

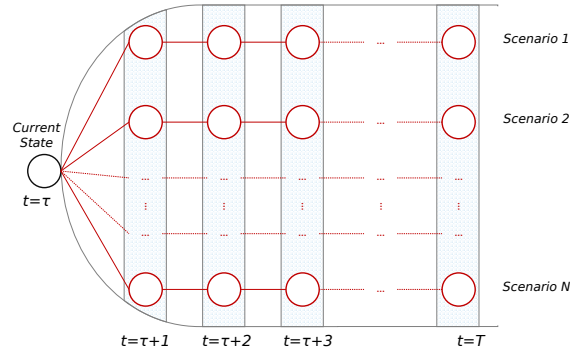


Fig. 2: Two-stage stochastic program built from  $N$  level 2 scenarios. These are solved to find an implementable decision for stage  $\tau$  in a level 1 scenario.

Given that stochastic programs are built for every state in every level 1 scenario, one will approximate the continuous sample space of the exogenous variable from each state. This approximation is illustrated in Figure 3. The main difference compared to RI is that STRO makes multiple explorations (level 2 scenarios) from every state in every underlying level 1 scenario and uses these to approximate the sample space from that point. The red nodes in the figure represent the level 2 scenarios. New information about the underlying level 1 scenario is revealed with each step of the horizon. Thus, new two-stage stochastic programs must be built for every decision by generating new level 2 scenarios.

None of the decisions made by STRO's heuristic are based on any knowledge of the future in the underlying price-inflow scenario. Following this, the value of the feasible policies found for every individual level 1 scenario yields lower bounds for the optimal revenue for that given scenario. Same as for RI, this means that with enough sampled scenarios, one will get an estimated lower bound of the expected revenue, subject to sampling error.

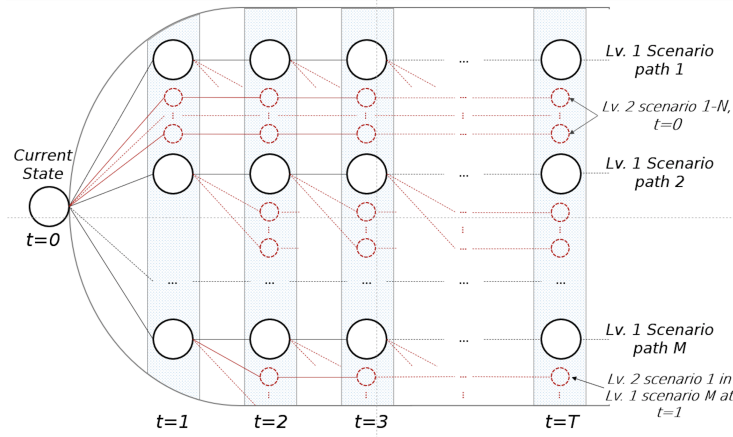


Fig. 3: Sample space approximation for STRO with  $N$  level 2 scenarios per decision

As every generated level 2 scenario represents possible realizations of the sample space, one would gradually get a better approximation of all possible realizations of the future in all possible states by generating more scenarios per decision.

As mentioned, the STRO heuristic finds the first-stage decision by considering these realizations as a two-stage program. Using two stages is an approximation, and a more accurate method would be to solve a multistage problem instead. However, building and solving multistage stochastic programs is more computationally costly, and approximation through two stages has been shown to be a good solution for hydropower problems in the literature [24].

Due to the growth in complexity when increasing the number of level 2 scenarios per decision, one can in practice only build programs based on a small number of scenarios. Therefore, the performance of STRO hinges on the assumption that most of the value from the approach can be achieved using a small number  $N$  of samples per decision, e.g.  $N = 2$  or  $N = 3$ , which from early experiments seems to be the case.

## Main innovations and benefits

As mentioned in 2.1, medium-term hydropower planning is mainly concerned with calculating the expected discounted revenue, given different initial water levels, to calculate the marginal value of the water in the reservoir. Consequently, there is no requirement to generate an implementable policy that deterministically yields the same production decision when faced with the same state, such as RI and SDDP. Instead, we relax this requirement and allow the STRO heuristic to yield stochastic production decisions based on the level 2 scenarios drawn

from the sample space. We refer to this as policy relaxation. The benefit is that by not having to aggregate the possible outcomes, for instance, by taking the expectation such as RI, we can explore more of the sample space and consider more varied outcomes. As shown in the illustrative example in section 4, this can lead to more accurate revenue estimations.

Another significant benefit of STRO and other rolling horizon methods is that they pose few constraints on the problem class. Unlike SDDP, which either requires inflow and price to be modelled linearly or with a Markov decision process over discretized states, this allows price and inflow to be modelled using non-linear methods such as deep learning-based methods. There is also no strict constraint that requires the problem to be modelled convexly, meaning that one for instance, could model head variations more accurately without having to relax the problem with McCormick envelopes. However, a non-convex model would require a longer processing time.

Other significant benefits of rolling horizon methods such as STRO and RI is that they are significantly more straightforward to implement than SDDP and are more suited for parallelization since the level 1 scenarios are all solved independently. Although SDDP is also parallelizable, it is harder to implement and does not lead to as linear of a speedup.

We leave to further work to explore the practical merits of these benefits and concentrate our experiments on a case where SDDP can be used as a benchmark.

## 4 Illustrative example

This section illustrates the proposed heuristic using a simple three-stage example. Consider a plant operator with a current reservoir volume of 8, maximum reservoir capacity of 10, and maximum generation capacity of 10. In this setting, the chronology is as follows: First, inflow realizes, then the reservoir volume gets updated. Spillage happens if the current reservoir level plus inflowing water exceeds the reservoir capacity. Then the generation decision is taken before the new inflow arrives.

The price is deterministic and given by  $P_0 = 10$ ,  $P_1 = 11$ ,  $P_2 = 12$ , incentivizing later production. Inflow is stochastic and the different scenarios can be seen in Figure 4. From every state, there is an equal chance that the subsequent inflow is high or low compared to the previous inflow. This results in a binary tree and 4 scenarios since inflow is deterministic in the first stage.

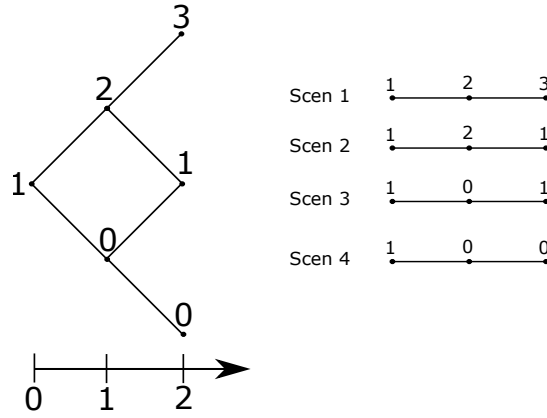


Fig. 4: Illustrative example inflow tree.

As an example: Since the current reservoir volume is 8 and 1 unit of inflow arrives deterministically, no spillage happens, as the reservoir can hold 10 units of water, allowing all units to be used for later generation. Suppose the producer decides to generate nothing at  $t = 0$ , and the high inflow is realized at  $t = 1$  (scenario 1 or 2). In that case, 2 units arrive in addition to the 9 already in the reservoir, resulting in 1 unit of spillage.

To see how RI and STRO with different numbers of level 2 scenarios perform for this example, we compare their different decisions and revenues in Table 1. We also compare them to the optimal strategy. Here we use  $x_{i,j}$  to denote the production decision at time  $i$  and state  $j$ , numbered according to the scenarios in Figure 4. The scenario index for stage 0 is omitted, as all these decisions need to be identical. Similarly, in stage 1, we only use scenario indices  $\{1, 2\}$  since there only are 2 possible inflow realizations at  $t = 1$ .

We observe that it is optimal to produce 1 unit in stage 0 to avoid spillage in the future. Rolling intrinsic, which bases decisions on expectations of the stochastic variables, ignores that there is a 50% chance of 2 units of inflow getting realized in the next stage and only considers the expected inflow, which is  $\frac{0+2}{2} = 1$ , thereby deciding to produce 0 at  $t = 0$ .

The notation STRO( $N$ ) is used in the table to indicate that  $N$  level 2 scenarios are used per decision for the STRO heuristic. Since there are only 4 possible realizations of the future, we use a maximum  $N$  of 4. For STRO(1), STRO(2) and STRO(3) the production decision naturally depends on which of the 4 scenarios are sampled. Since we sample the scenarios without replacement, the number of combinations is  $\binom{4}{N}$ , which results in 4, 6, 4 and 1 combinations for  $N = 1, 2, 3$  and 4 respectively. The production decisions in the first stage are given in the table for all different combinations in curly brackets. To deal with uncertain production decisions, we list the production decisions in the first stage for each combination of samples for every STRO( $N$ ) in curly brackets. Since the production decision at  $t = 0$  is uncertain for  $N = 1$  and  $N = 2$ , the next state is also

uncertain (STRO(3) yields the same production decision at  $t = 0$  independent of the combination of sampled scenarios). For this reason, the production decisions at  $t = 1$  and  $t = 2$  are not listed for these, as these decisions naturally will depend on the previous decision, which is not deterministic. This relates to the discussed idea of policy relaxation, meaning that STRO does not necessarily generate an implementable policy.

	Optimal	Rolling	intrinsic	STRO(1)	STRO(2)	STRO(3)	STRO(4)
$x_0$	1	0	$\{0, 0, 1, 1\}$	$\{0, 1, 1, 1, 1, 1\}$	$\{1, 1, 1, 1\}$	1	
$x_{1,1}$	3	2	-	-	3	3	
$x_{1,2}$	0	0	-	-	3	0	
$x_{2,1}$	10	10	-	-	10	10	
$x_{2,2}$	8	9	-	-	8	8	
$x_{2,3}$	9	10	-	-	9	9	
$x_{2,4}$	8	9	-	-	8	8	
Mean spill	0	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{12}$	0	0	
Mean profit	131.5	125.0	127.0	130.83	131.5	131.5	

Table 1: Solutions and objective values. Note that STRO does not yield an implementable policy since there are several decisions at the first stage.

The table shows that STRO(4) and STRO(3) both find the optimal strategy. This is unsurprising for STRO(4) as it considers all 4 possible scenarios when making decisions. STRO(3) does not have this privilege, but since it only omits a single possible scenario at  $t = 0$ , it will always sample at least one of the 2 scenarios with low inflow at  $t = 1$ . This allows it to see that there is a chance of spillage occurring and thereby plan its production accordingly.

It can further be seen that STRO(1) and STRO(2) both outperform RI, despite not necessarily seeing that there is a chance of high inflow at  $t = 1$ . As STRO(2) will sample 2 scenarios per decision it will not consider the chance of high inflow only in the case that the 2 sampled scenarios are the 2 low inflow scenarios. There is only a  $\frac{1}{6}$  chance that this happens, making it unlikely that the wrong production decision is made. When making a decision at  $t = 1$  there are only 2 possible future inflow realizations, depending on the previously observed inflow. As a consequence of this, STRO(2) will make the optimal decisions at  $t = 1$  since it can consider both possible future realizations from that point.

STRO(1) only samples a single future realization when making decisions. At  $t = 0$  it is only a 50% chance that it samples a high inflow scenario and adjusts production to avoid risking spillage. However, it still outperforms RI, because RI will only consider the expected inflow of 1, meaning that it will always make the wrong production decision at  $t = 0$ . The same effect would be seen if a high inflow scenario was realized at  $t = 1$ . Then RI would not take the fact that a high inflow of 3 could occur at  $t = 2$  into account and only consider the expected inflow of 2. This results in a higher chance of spillage.

## 5 Hydropower Case and Implementation

### 5.1 Implementation environments

**Software** All code is implemented and run in Python 3.8, except the parameter estimation of the price model, which is done in MATLAB2020b [12]. In Python, the gurobipy package is used for the optimization models. A previous Python implementation of SDDP was also used [9].

**Hardware** The computational tests are conducted on a system that allows for running up to 40 processes in parallel per node. In Table 2, the relevant technical aspects of the hardware are given.

Table 2: Hardware specifications

Specification	Details
Node System	Dell PowerEdge R640
Node CPU:	2x 2.4GHz Intel Xeon Gold 5115 CPU – 10 core
Node RAM:	96Gb

### 5.2 Matre Haugsdal and Vemundsbøtn

This paper is written in cooperation with a major Norwegian hydropower producer. They have provided data for one of their hydropower systems. Matre Haugsdal and Vemundsbøtn are power plants in western Norway. The plants are parts of the more extensive system of interconnected reservoirs built around the Matre watercourse. Together they produce enough electricity to cover the needs of more than 50k households. An overview of the system’s layout as modelled can be seen in Figure 5, and the publicly available technical specifications for the two production plants are given in Table 3.



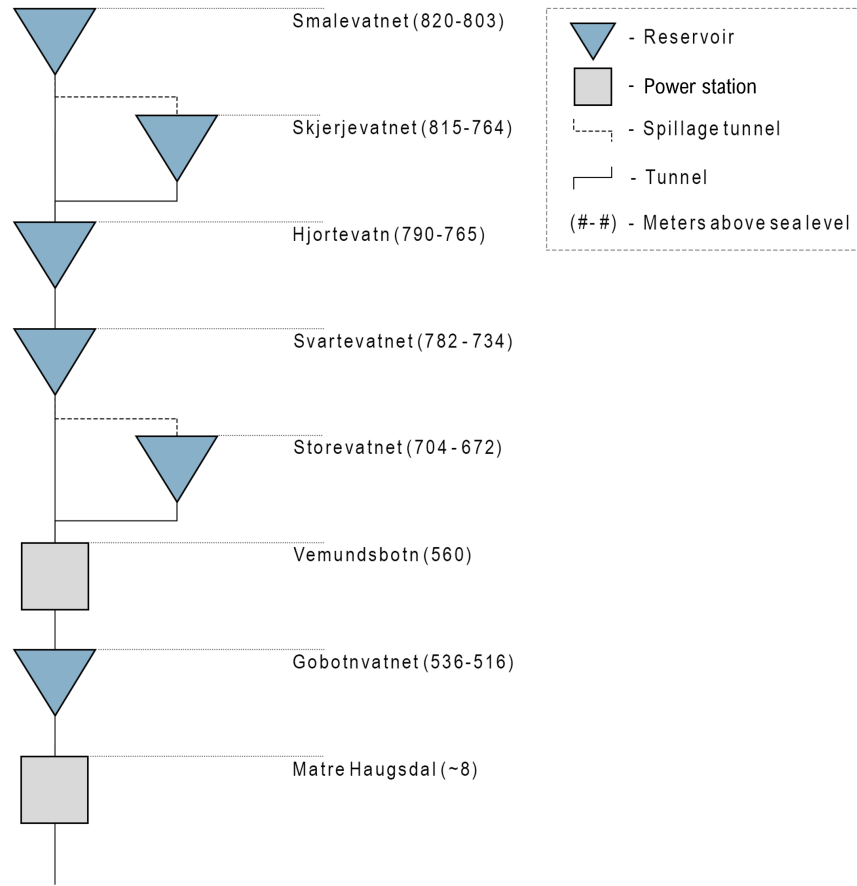


Fig. 5: Matre Haugsdal and Vemundsbøtn, with connected reservoirs.

Table 3: Specifications for plants in the considered system

	Matre Haugsdal Vemundsbøtn	
Build year	2016	1976
Turbines	2	1
Production capacity [MW]	180	45
Elevation [m]	537	(164 / 242)*
Head difference [m]	20.5	(32 / 48)*
*Two connected reservoirs (Storevatnet/Svartevatnet)		

Table 4 shows the average annual inflows for each reservoir, along with their respective volumes and discharge capacities. A glance at the data shows that the reservoirs are of a “fast” nature, meaning that the inflow to the reservoirs is relatively high compared to the storage capacity. We were not given any specified minimum discharge for the reservoirs in question, and thus, no lower bound is set for the model. Furthermore, the water level regulations for the reservoirs do not vary with season, but are constant throughout the year.

Table 4: Reservoir specifications

Reservoir	Volume [ $Mm^3$ ]	Inflow [ $Mm^3/yr$ ]	Production capacity [ $m^3/s$ ]
Gobotvatn	22.0	134.6	37.6
Storevatn	23.3	57.6	18.0
Svartevatn	60.8	66.8	23.6
Skjerjevatn	70.5	61.3	22.0
Hjortevatn	10.7	60.9	10.0
Smalevatn	13.1	96.5	19.8

5.3 Price and Inflow Data

Our price model uses futures data, and to generate the smoothed future curves, we use futures price data for the trading period 2006–2018, retrieved from Montel. The price simulations start 1 January 2019. The data is given per trading day, with closing prices for monthly, quarterly, and annual delivery for up to three years ahead. To fit the inflow model, we use time series inflow data for the reservoirs. The inflow data series provided are of hourly resolution in the period 2009–2019.

6 Computational Study

To see how the STRO method with different number of level 2 scenarios per decision performs, the results are benchmarked to the SDDP method, as well as the most well-known rolling horizon method RI. To avoid disclosing any non-public information about the Matre Haugsdal hydropower system, the results of the models are represented as percentages relative to the baseline given by SDDP.

For these studies, revenue has been used as the primary metric for performance comparisons between different algorithms. Seeded sample scenarios were used for the rolling horizon methods, to make the revenues as comparable as possible. This means that all rolling horizon methods receive the same realizations of inflow and price. This ensures that the objective values of the rolling horizon methods are comparable, as they solve the exact same scenarios. However, this was not achievable for SDDP due to its discretization. We address this

potential source of bias by using a relatively large number of samples for both methods over 10 runs. The results presented are average values from all these runs, along with relative standard deviations of the objective values achieved over different trials using the different methods. This approach should give a reasonable estimate of the difference in performance.

Note that since SDDP yields upper bounds while RI and STRO yield estimated lower bounds, there will always be some discrepancy between their results.

The discretizations are done on a sample base of 200k samples of each exogenous variable, which were discretized to a Markov chain with 125 discrete states per stage.

The results from the trial runs are presented in Table 5. It presents the expected revenue and the average price of energy produced for the different methods. The standard deviation over the different runs and the average price for the energy sold are also given. Every run of the Rolling Horizon methods was made using 1000 level 1 scenarios, and reported revenue is the average over these runs.

Table 5: Average objective values of the compared methods.

Solution Method	Revenue	
	Relative Value	Standard Deviation
SDDP	100.000%	0.487%
RI	97.546%	0.489%
STRO(1)	97.249%	0.484%
STRO(2)	98.115%	0.487%
STRO(3)	98.411%	0.489%
STRO(4)	98.548%	0.486%
STRO(5)	98.611%	0.485%
STRO(6)	98.654%	0.482%
STRO(7)	98.674%	0.483%

From these results there are multiple takeaways. First, it is apparent that increasing the number of samples per decision in the heuristic leads to a substantial improvement in the results achieved. When comparing the results of STRO for  $N = 1$  to  $N = 7$  it can be seen that the gap to the SDDP algorithm is approximately halved. Furthermore, it can be seen that the improvement in the results is diminishing, and the most significant improvement can be seen when increasing  $N$  from 1 to 2. However, when comparing the results for  $N = 6$  to  $N = 7$ , the gain has tapered out.

Another takeaway is that the STRO algorithm clearly outperforms the traditional RI algorithm for  $N \geq 2$ . The results of the RI algorithm were found to be within 2.5% of the estimated optimum from SDDP, in line with previous findings from [4]. However, the results from the STRO(7) algorithm, is within

1.4% of the results of SDDP. Given that the estimated revenues from the rolling horizon methods, STRO and RI, are lower bounds, whereas SDDP gives an upper bound, this gap is quite close, and it is apparent that the STRO method is well suited for the hydropower planning problem.

As seen in the table, the standard deviations of the results achieved using different methods are quite high compared to the differences in the results achieved with the different algorithms. This implies that the true differences in revenues between SDDP and the other methods can be slightly higher or lower than these experiments indicate. Since seeded scenarios were used for RI and STRO we can be more confident that the comparisons between these methods are more accurate. In future experiments, we will make efforts to reduce the standard deviations by using more level 1 scenarios to ensure more stable revenue estimates over the different trials.

### 6.1 Runtime Analysis

Rolling Horizon methods are well suited for parallelization and multiprocessing, as each run is independent. To minimize the processing time, our implementation runs 40 processes in parallel, which was the maximum allowed by our hardware. With improved hardware, a further speedup is naturally achievable. There exist parallel implementations of SDDP [1, 21, 27]. However, as [1] points out, parallel implementation of SDDP is far from straightforward. Therefore, we opt for a non-parallel version for benchmarking. With such parallel methods implemented, we would naturally have a substantially lower processing time with SDDP.

In table 6, the run times of SDDP and the STRO methods are presented. RI solves a problem of the same size as STRO(1), the computational times for these methods are, therefore, approximately the same. The total run time is given in hours (h), and the average time used to solve each level 1 scenario is given in seconds (s). Note that the run time of SDDP is given without the additional time required for sampling and discretization.

Table 6: Run time for 1000 level 1 scenarios of STRO( $N$ ), and 2000 iterations of SDDP.

Solution Method	Time	
	Total (h)	Per sampled scenario (s)
SDDP	11.38	
STRO(1) / RI	0.26	0.94
STRO(2)	0.73	2.63
STRO(3)	3.47	12.48
STRO(4)	5.94	21.38
STRO(5)	8.19	29.49
STRO(6)	10.58	38.10
STRO(7)	11.88	42.77

The table shows that the most significant increase in run-time comes with the increase from 2 to 3 level 2 scenarios per decision in STRO, followed by a nearly linear increase in computational time. Once the scenarios are sampled and given to the two-stage stochastic program, the program becomes deterministic [26]. By doubling the number of scenarios for the two-stage program, the program size approximately doubles.

## 7 Concluding Remarks

We propose a new approach to solving the medium-term reservoir management problem. The aim is to approximate the value function as it depends on current price levels and current reservoir levels. It is not necessary to have an implementable policy from the medium-term problem since only the value function is passed to the short-term planning problem. Thus, we relax policy implementability, taking care not to relax information constraints; our method relies on scenarios for the future development of prices and inflow rather than on their actual realization. The new approach, called STRO, is compared to SDDP and RI.

Although STRO can be applied to broader classes of problems compared to those solved by SDDP, we designed the experiments to suit SDDP well. We leave for future work to explore the performance on non-convex problems.

Sources of bias potentially impacting the comparisons of the implemented models include both discretization and sampling error subjected to the different methods. Efforts were made to mitigate this by performing multiple trial runs, averaging the results, and performing experiments that, for instance, aimed to bypass the discretization error of SDDP. We have also used seeding to ensure that the exact sampling error is imposed on the rolling horizon methods.

Another factor to consider is that the methods discussed and compared were only tested on a single hydropower system, using only one type of inflow and price model. Specific properties of the case system can make the method more suitable, which can mean that similar results will not be achievable for other systems. These concerns were not explored in the interest of scope, but they should be further examined in the future.

With these points made, we emphasize that the RI algorithm has been shown to produce promising results when applied to the reservoir management problem by other authors with different stochastic models for the exogenous processes. This strengthens our confidence in the results.

The idea of using a sampling-based heuristic for rolling horizon methods can also be explored further. The STRO method only considers building two-stage stochastic convex programs, but in principle, multi-stage non-convex stochastic programs could be built and solved instead.

## References

1. Ávila, D., Papavasiliou, A., Löhdorf, N.: Parallel and distributed computing for stochastic dual dynamic programming. *Computational Management Science* **19**, 199—226 (2022)

2. Benth, F.E., Koekkebakker, S., Ollmar, F.: Extracting and applying smooth forward curves from average-based commodity contracts with seasonal variation. *The Journal of Derivatives* **15**(1), 52–66 (2007)
3. Bertsekas, D.: *Dynamic Programming and Optimal Control: Volume I*. Athena Scientific, Belmont, MA, USA (2005)
4. Braaten, S.V., Gjønnes, O., Hjertvik, K.J., Fleten, S.E.: Linear decision rules for seasonal hydropower planning: Modelling considerations. *Energy Procedia* **87**, 28–35 (2016)
5. Breslin, J., Clewlow, L., Elbert, T., Kwok, C., Strickland, C., van der Zee, D.: Gas storage: Rolling intrinsic valuation. *Energy Risk* **January**, 61–65 (2009)
6. Cerisola, S., Latorre, J.M., Ramos, A.: Stochastic dual dynamic programming applied to nonconvex hydrothermal models. *European Journal of Operational Research* **218**(3), 687–697 (2012)
7. da Costa, J.P., de Oliveira, G.C., Legey, L.V.L.: Reduced scenario tree generation for mid-term hydrothermal operation planning. 2006 International Conference on Probabilistic Methods Applied to Power Systems (2006)
8. Dietze, M., Chávarry, I., Freire, A.C., Valladão, D., Street, A., Fleten, S.E.: A novel semiparametric structural model for electricity forward curves. *IEEE Transactions on Power Systems* pp. 1–11 (2022). <https://doi.org/10.1109/TPWRS.2022.3197982>
9. Ding, L., Ahmed, S., Shapiro, A.: A Python package for multi-stage stochastic programming. *Optimization Online* (2019)
10. Downward, A., Dowson, O., Baucke, R.: Stochastic dual dynamic programming with stagewise-dependent objective uncertainty. *Operations Research Letters* **48**(1), 33–39 (2020)
11. Gjelsvik, A., Mo, B., Haugstad, A.: Long-and medium-term operations planning and stochastic modelling in hydro-dominated power systems based on stochastic dual dynamic programming. In: *Handbook of Power Systems I*, pp. 33–55. Springer (2010)
12. Goodwin, D.: Schwartz-Smith 2-factor model: Parameter estimation. <https://www.mathworks.com/matlabcentral/fileexchange/43352-schwartz-smith-2-factor-model-parameter-estimation> (2020), accessed 2021–11–09
13. Helseth, A., Braaten, H.: Efficient parallelization of the stochastic dual dynamic programming algorithm applied to hydropower scheduling. *Energies* **8**, 14287–14297 (12 2015). <https://doi.org/10.3390/en8121431>
14. Hjelmeland, M.N., Zou, J., Helseth, A., Ahmed, S.: Nonconvex medium-term hydropower scheduling by stochastic dual dynamic integer programming. *IEEE Transactions on Sustainable Energy* **10**(1), 481–490 (2018)
15. Infanger, G., Morton, D.P.: Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming* **75**(2), 241–256 (1996)
16. Lai, G., Margot, F., Secomandi, N.: An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations Research* **58**(3), 564–582 (2010)
17. Löhndorf, N., Wozabal, D.: The value of coordination in multimarket bidding of grid energy storage. *Operations Research Articles in Advance* (2022). <https://doi.org/10.1287/opre.2021.2247>
18. Löhndorf, N., Shapiro, A.: Modeling time-dependent randomness in stochastic dual dynamic programming. *European Journal of Operational Research* **273**, 650–661 (2019)
19. Pereira, M., Pinto, L.: Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming* **52**(1–3), 359–375 (1991)

20. Philpott, A., Guan, Z.: On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters* **36**(4), 450–455 (2008)
21. Pinto, R.J., Borges, C.T., Maceira, M.E.P.: An efficient parallel algorithm for large scale hydrothermal system operation planning. *IEEE Transactions on Power Systems* **28**(4), 4888–4896 (2013). <https://doi.org/10.1109/TPWRS.2012.2236654>
22. Powell, W.: *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons (2007)
23. Schwartz, E., Smith, J.E.: Short-term variations and long-term dynamics in commodity prices. *Management Science* **46**(7), 893–911 (2000)
24. Séguin, S., Audet, C., Côté, P.: Scenario-tree modeling for stochastic short-term hydropower operations planning. *Journal of Water Resources Planning and Management* **143**(12) (2017)
25. Shapiro, A., Tekaya, W., da Costa, J.P., Soares, M.P.: Risk neutral and risk averse stochastic dual dynamic programming method. *European Journal of Operational Research* **13**, 375–391 (2013)
26. Shapiro, A., Nemirovski, A.: On complexity of stochastic programming problems. In: Jeyakumar, V., Rubinov, A. (eds.) *Continuous Optimization. Applied Optimization*, vol. 99, pp. 111–146. Springer, Boston, MA (2005)
27. da Silva, E., Finardi, E.: Parallel processing applied to the planning of hydrothermal systems. *IEEE Transactions on Parallel and Distributed Systems* **14**(8), 721–729 (2003). <https://doi.org/10.1109/TPDS.2003.1225052>
28. Valipour, M., Banihabib, M.E., da Costa, J.P., Behbahani, S.M.R.: Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. *Journal of Hydrology* **13**, 433–441 (2012)
29. Van Slyke, R.M., Wets, R.: L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* **17**(4), 638–663 (1969)
30. Wu, O., Wang, D., Qin, Z.: Seasonal energy storage operations with limited flexibility: The price-adjusted rolling intrinsic policy. *Manufacturing & Service Operations Management* **14**(3), 455–471 (2012)

## A Piece-wise Linear Approximations

As the head  $h_{i,t}$  is a function of the amount of water in a reservoir  $l_{i,t}$  this function needs to be approximated in a piece-wise manner. To do this, the relation between head and water level for a specific reservoir  $i$  can be represented by a set  $S_i$  which contains sample points  $((l_{i,1}, h_{i,1}), \dots, (l_{i,K}, h_{i,K}))$ , representing the mapping between water level and head. Increasing the number of points naturally leads to a better approximation, but comes at the consequence of increased complexity.

To find the head from the water level using the piece-wise linear graph, a set of variables representing the weight of point  $k$  for reservoir  $i$  at time  $t$  denoted  $\lambda_{i,t,k}$ . By ensuring that no weights can be negative (7), the weights sum to 1 (8), and that only two adjacent points are non-zero using a special ordered set of type 2 constraint (SOS2) (9), one can find the head  $h_{i,t}$  based on the water level  $l_{i,t}$ . This is done using equations (10)–(11). Equation (10) forces the weights to correspond with the water level, and equation (11) finds the head given the weights. If the function describing the mapping from water level to head is concave, the SOS2 constraint (9) is redundant.

$$\lambda_{i,t,k} \geq 0, \quad k \in S_i \quad i \in R^P \quad t = 0, \dots, T-1 \quad (7)$$

$$\sum_{k \in S_i} \lambda_{i,t,k} = 1, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (8)$$

$$\lambda_{i,t,k} \in \text{SOS2}, \quad k \in S_i \quad i \in R^P \quad t = 0, \dots, T-1 \quad (9)$$

$$\sum_{k \in S_i} \lambda_{i,t,k} l_{i,k} = l_{i,t}^{avg}, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (10)$$

$$\sum_{k \in S_i} \lambda_{i,t,k} h_{i,k} = h_{i,t}, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (11)$$

As the incoming and outgoing water levels can be quite different, the average water level  $l_{i,t}^{avg}$  is used as the basis for the head for the period. This variable is given according to constraint (12).

$$l_{i,t}^{avg} = \frac{l_{i,t} + l_{i,t+1}}{2}, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (12)$$

## B McCormick Envelope Constraints

In the following the constraints comprising the McCormick envelopes are shown. These are used to ensure that the substitution variable  $w_{i,t}$  closely approximates the bi-linear term  $h_{i,t}x_{i,t}$ . These constraints depend on the variables  $h_{i,t}$  and  $x_{i,t}$ , as well as the maximum and minimum values that these can take on. These maximum and minimum values are denoted  $H_{i,t}^{max}$ ,  $H_{i,t}^{min}$ ,  $X_{i,t}^{max}$  and  $X_{i,t}^{min}$ .



$$w_{i,t} \geq X_{i,t}^{min} h_{i,t} + x_{i,t} H_{i,t}^{min} - X_{i,t}^{min} H_{i,t}^{min}, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (13)$$

$$w_{i,t} \geq X_{i,t}^{max} h_{i,t} + x_{i,t} H_{i,t}^{max} - X_{i,t}^{max} H_{i,t}^{max}, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (14)$$

$$w_{i,t} \leq X_{i,t}^{max} h_{i,t} + x_{i,t} H_{i,t}^{min} - X_{i,t}^{max} H_{i,t}^{min}, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (15)$$

$$w_{i,t} \leq X_{i,t}^{min} h_{i,t} + x_{i,t} H_{i,t}^{max} - X_{i,t}^{min} H_{i,t}^{max}, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (16)$$

$$0 \leq w_{i,t} \leq X_{i,t}^{max} H_{i,t}^{max}, \quad i \in R^P \quad t = 0, \dots, T-1 \quad (17)$$