Hydropower Production Scheduling using Stochastic Dual Dynamic Programming subject to environmental constraints

Håkon I. Arvidsen

Master Thesis (Msc)
at the University of Bergen, Norway

June 3, 2019

Abstract

In this thesis, we will focus on hydropower production scheduling for the Bergsdalen watercourse, operated by the power producer BKK, in Hordaland. The hydropower production in the Bergsdalen watercourse is subjected to a set of environmental restrictions, imposed on the system by the Norwegian Water Resources and Energy Directorate (NVE). The goal of the thesis is to determine the challenges of implementing the imposed environmental restrictions in an optimization model and to propose approximations of the restrictions. We also investigate the loss of revenues caused by the restrictions.

First, we present hydropower production scheduling in general, for hydropower systems similar to the Bergsdalen watercourse. The scheduling is formulated as a Linear Programming (LP) problem. A Stochastic Dual Dynamic Programming (SDDP) algorithm is presented, in order to solve the scheduling problem. This solution method is commonly used for hydropower production scheduling in Norway. Then, we will discuss the environmental restrictions, and the difficulties of including them as constraints in the optimization model. As the solution method is considered necessary, we propose an approximation of the constraints, in order to solve the scheduling problem for the Bergsdalen watercourse. We have run an SDDP algorithm for two deterministic price series. We present the results of the optimization, discuss the validity of the proposed approximations, and the loss of revenues caused by the approximated environmental constraints.

We found that restrictions depending on inflow and reservoir volume challenge the solution method. The environmental restrictions caused 0.7% and 1.2% loss of revenues for the two price scenarios, respectively.

Acknowledgements

I would like to thank my supervisor Dag Haugland at the Department of Informatics at the University of Bergen. Our frequent meetings have helped significantly in structuring this thesis and ensuring constant progress. Thanks to my co-supervisor at SINTEF Energy, Arild Helseth, for quickly answering questions, and expanding my knowledge on the solution method.

I would also like to thank the staff at BKK Production: Kjetil Trovik Midthun, Frode Haga and Tarjei Lid Riise, for pitching the project and aiding my practical understanding of hydropower production.

Technical terms

Planning period The time period we want to consider when optimizing a production schedule.

Schedule The production schedule regarding volume, production, bypass and overflow for each reservoir, during the planning period.

Hydro system Consisting of one or several watercourses, intended for power production.

Watercourse Consisting of reservoirs and hydro plants, hydrologically connected.

Hydro plant A facility where water can be either let through turbines, to produce electrical power, or bypassed further downstream along the watercourse.

Reservoir A lake or dam where water can be stored.

Module A model representation of a reservoir and a hydro plant.

Inflow Water that is flowing into a reservoir from its surroundings.

Expected future income The expected income for all stages from the subsequent stage until the end of the planning period.

Cut A Benders cut, used for representing the expected future income function

Stage One discrete time step in the planning period.

State The state of a hydro system. Referring to inflow and reservoir volume levels

Inflow realization A possible inflow transition between two stages

Abbreviations

HPS	Hydropower Production Scheduling.
NVE	Norwegian Water Resources and Energy Directorate (Norsk Vassdrag- og Energidirektorat).
BKK	Bergenhalvøens Kommunale Kraftselskap (local energy producer in Hordaland).
LP	Linear Programming.
MILP	Mixed Integer Linear Programming.
SDDP	Stochastic Dual Dynamic Programming.

Symbols

Index sets

$\mathcal{T}$: Index set for stages.

$\mathcal{S}$: Index set for inflow scenarios.

$\hat{\mathcal{S}} \subset \mathcal{S}$: Index set for sampled scenarios.

$\mathcal{I}$: Index set for reservoirs. I is the total number of reservoirs.

$\mathcal{M}_i \subset \mathcal{I}$: Index set for all reservoirs directly upstream of reservoir i.

$\mathcal{K}$: Index set for inflow realizations between stages.

Variables

V_{it}^s : Volume of reservoir i at the end of stage t in scenario s. $10^6 m^3$

Q_{it}^s : Discharge of reservoir i during stage t in scenario s. $10^6 m^3$

B_{it}^s : Bypass of reservoir i during stage t in scenario s. $10^6 m^3$

O_{it}^s : Overflow of reservoir i during stage t in scenario s. $10^6 m^3$

y_t^s : Power sold during stage t in scenario s. Gwh

Parameters

T : Length of the planning period.

S : Total number of inflow scenarios.

$\hat{S}$: Number of sampled inflow scenarios.

K : Number of possible inflow realizations between two arbitrary stages.

$\overline{V}_i$: Max volume of reservoir i. $10^6 m^3$

$\overline{Q}_i$: Max weekly discharge of reservoir i. $10^6 m^3$

$\overline{B}_i$: Max weekly bypass of reservoir i. $10^6 m^3$

$\underline{V}_{iT}^s$: Lower bound on end reservoir volume V_{iT}^s. $10^6 m^3$

$\underline{V}_2$: Lower bound on $V_{2,t}^s$ imposed by environmental restriction (ii). $10^6 m^3$

p_t : Power price during stage t. $10^6 \frac{\text{EUR}}{Gwh}$

η_i : Energy conversion factor for reservoir i. $\frac{Gwh}{10^6 m^3}$

q_{iyw} : Recorded inflow for reservoir i in year y and week w. $10^6 m^3$

$\overline{I}$: Sufficient weekly inflow to activate constraint (i). $10^6 m^3$

Z_{it}^s : Transformed inflow into reservoir i during stage t in scenario s.

$f_{it}(Z_{it}^s)$: Inflow transformation function.

ϕ_{ij} : Inflow correlation matrix. Both i and j are in $\mathcal{I}$.

$\mathcal{E}_t$: Error of inflow transition from stage $t-1$ to t. Stochastic parameter.

$f_{\mathcal{E}_w}$: Probability distribution of $\mathcal{E}_t$.

ϵ_{it}^s : Outcome of $\mathcal{E}_t$ for reservoir i in stage t and scenario s.

$\mathbb{P}(s)$: Probability of scenario s occurring.

$t_A, ..., t_D$: Activation times of the ”precise” environmental constraints, (i) through (iii).

$t_a, ..., t_e$: Activation times of the approximated environmental constraints, (3.21) through (3.26).

U : Upper bound on the objective function of the HPS problem.

$\hat{L}$: Estimated lower bound on the objective function of the HPS problem, from a Monte Carlo simulation.

σ : Standard deviation of the Monte Carlo simulation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to hydropower scheduling

Norway has a large capacity for hydropower. According to *Energy in Norway* (NVE, 2015), hydropower constituted 96% of Norwegian energy production in 2014. In Hordaland, the annual hydropower potential for established hydro plants was roughly 17 TWh, based on yearly average production between 1981 and 2010. With a yearly net consumption of roughly 13 Twh in Hordaland, the annual hydropower potential is enough to meet the demand. We will focus on hydropower in Norway, more specifically, in Hordaland.

Consider a hydropower production company operating a hydro system consisting of multiple watercourses, each with multiple reservoirs and turbines. Among renewable energy sources hydropower has an advantage, due to the ability to store water and produce power at the most profitable times. Therefore, a hydropower producer can profit by determining an optimal production schedule. In general, the producer should save water when the price of power is low, and produce when the price is high. However, with uncertainty of inflow amount over the course of the planning period, the producer should maintain a reasonable water level in their reservoirs. If water levels are too low, the producer reduces ability to produce power during potential high power prices in the future. On the other hand, if water levels are too high, the producer increases possibility of overflow, hence loss of resources and possible damage to equipment and surroundings.

There are several factors to take into account when determining an optimal production schedule. With an increasing contribution from renewable energy in the European energy mix, one can expect increasing fluctuations in power price due to varying availability of cheap renewable energy sources, like solar and wind. It is important to have some idea about the extent of rain during the planning period, and how inflow into each reservoir varies within the planning period. Hydro systems such as this can be large and complex to model. We can either consider one watercourse at a time or the entire system at once. A physically correct model is complex, and contains nonlinearities. Therefore, a balance between model complexity and computational efficiency must be determined. Additionally, downtime due to scheduled maintenance and environmental restrictions are important factors. A hydro system model under the influence of varying and uncertain inflow and price subject to imposed scheduling and environmental restrictions, can result in a quite complex problem. In this thesis we will focus on inflow uncertainty and some environmental restrictions.

Depending on the size of the hydro system and the length of the planning period, there are different factors that play an essential role. For instance, for a hydro system constituting a large part of a power market, the power price will depend on the hydrological situation. When considering longer planning periods, we are subject to greater uncertainty. Hence, the hydro system is often simplified, as we are not interested in the details. Alternatively when considering a short planning period, the price and inflow forecasts have less uncertainty, and a complex physical model can be prioritized. As a consequence, it is common to separate the scheduling to long-, medium-, and short term scheduling. Long term scheduling, typically 3 to 5 years or longer, provides end term values for medium term, usually between 1 and 3 years, which in turn provide end term values for short term scheduling, with a typical planning period of no more than 2 weeks. Here, we will focus on medium term scheduling. For medium term scheduling, it is common practice to set the planning period longer than what is absolutely necessary, in order to minimize the effects of the end term values. End term values can be, for instance, an end reservoir volume bound, or the value of storing water.

The medium term schedule can be used as decision aid for establishing medium term energy contracts. Additionally, power producers are usually interested in the value of their water. As we have a restricted amount of water in each reservoir, we are interested in the optimal dual values of the restrictions on reservoir water volume. These values are interpreted as the marginal cost of water, referred to as shadow prices. They are often more relevant than the medium term schedule, for making daily decisions. Thus, the optimal dual values corresponding to reservoir volume restrictions in the first weeks are interesting. By formulating a hydropower scheduling problem as a Linear Programming (**LP**) problem, we can obtain the shadow prices of the water in each reservoir. Moreover, the shadow prices of water in the first weeks can be used as end conditions for short term optimization.

1.2 Literature

1.2.1 Industry

The hydropower production planning problem considered in this thesis, contains uncertain parameters and a long planning period, where decisions are made sequentially. Therefore, the scheduling task can be considered as a multi-stage stochastic programming problem. Different strategies to solve such problems were discussed by M. Apap and Grossmann (2016). They split uncertainty into two different categories, either endogenous or exogenous. In the endogenous case the outcome of an uncertain parameter depends on the decisions we make, and in the exogenous case it is independent of decisions. Birge (1982) investigated the value of a stochastic solution rather than a deterministic solution using expected values with sensitivity analysis.

As mentioned, in Hordaland it it sufficient to rely almost only on hydropower. However, a lot of the relevant research for hydropower planning has been done on systems also containing thermal power, such as gas or coal plants. These power systems are usually referred to as hydro-thermal systems in the literature. For hydro-thermal systems, we consider a hydro-thermal coordination problem, where the objective can be to cover a certain power need using limited cheap hydropower and supply with more expensive thermal power at the right times. To do so, it is necessary to determine the value of water in relation to the cost of burning fuels, to determine how to cover a load in a cost effective way. A lot of the research mentioned will therefore also consider hydro-thermal production planning.

For hydro-thermal scheduling on a short timescale, the short term Unit Commitment (UC) problem was described by Antonio et al. (2011). Mixed Integer Linear Programming (**MILP**) and Lagrangian Relaxation have been used to solve the UC problem, but with different strengths and weaknesses each. Antonio et al. (2011) tried to unify both solution methods.

For hydro-thermal scheduling on a large scale, a stochastic dual dynamic programming (**SDDP**) algorithm was presented by Pereira (1989). The algorithm uses a multi-stage benders decomposition approach to construct future cost functions, in order to solve a stochastic dynamic programming problem, without a state-space discretization. State-space discretization limits the size of the hydro system to be optimized, allowing for a multi reservoir system to be computationally feasible. Gjelsvik et al. (2010) applied the SDDP solution method to a Nordic hydro power system. The model has been extended to include pumped-storage and wind power by Helseth et al. (2013) and extended further to include maintenance scheduling by (Ge et al., 2018). This solution method requires the problem to be formulated as an LP problem.

For an exact physical representation, the scheduling problem needs to include non-linear relationships. Turbine efficiency- and head height functions are usually non-linear, and may lead to non-convex constraints. Cerisola et al. (2012) presented a model to include such relationships. de Souza Zambelli et al. (2013) used a deterministic non-linear optimization model in a model predictive control approach. This method was compared to an SDDP method. Hjelmeland et al. (2019) created a version of the SDDP algorithm to include integer variables, thus extending the solution method presented by Gjelsvik et al. (2010) to solve a MILP problem. Hjelmeland et al. (2019) found that introducing integers considerably increases computation time.

For an extensive review of work done on multi reservoir hydro system scheduling, we refer to Labadie (2004). The article provides a good overview of the foundation for the field today.

1.2.2 Theory

This section gives a brief overview over the theory literature that will be used in the thesis. A thorough introduction to stochastic programming is given by Kall and Wallace (1994). The solution method we will use is based on Stochastic Dynamic Programming, explained in depth by Ross (1983), where Benders decomposition (Benders, 1962), is used for coupling the stages. The method is based on Bellmans principle of optimality (Bellman and Kalaba, 1966).

Chapter 2

The Hydropower Production Scheduling Problem

2.1 Problem components

In this section the key components of the Hydropower Production Scheduling Problem (**HPS**) are presented. Our goal is to achieve an optimal production schedule over a certain time period, starting in the present until the end of our **planning period**. We optimize the production **schedule** of a **hydro system**, under the influence of **inflow** and price, considering some environmental restrictions. We assume that the system is in a liberalized market, with no obligation to meet any power supply demand. Additionally, we assume the system is relatively small and does not affect the price of power.

2.1.1 Hydro system

Hydro systems vary greatly in size and complexity. They consist of several **hydro plants** and **reservoirs**, which may or may not be connected, constructing one or several **watercourses**.

It is up to the producer whether or not to model all watercourses in one system, or split them up into smaller sub systems. Figure 2.1 is an example of a hydro system consisting of several watercourses, which is a sub system of the entire hydro system operated by the producer **BKK**. If watercourses are completely separated hydrologically, that is no water flows between them, and we can consider price as an exogenous factor, then we can split the scheduling problem into a problem for each separate watercourse.

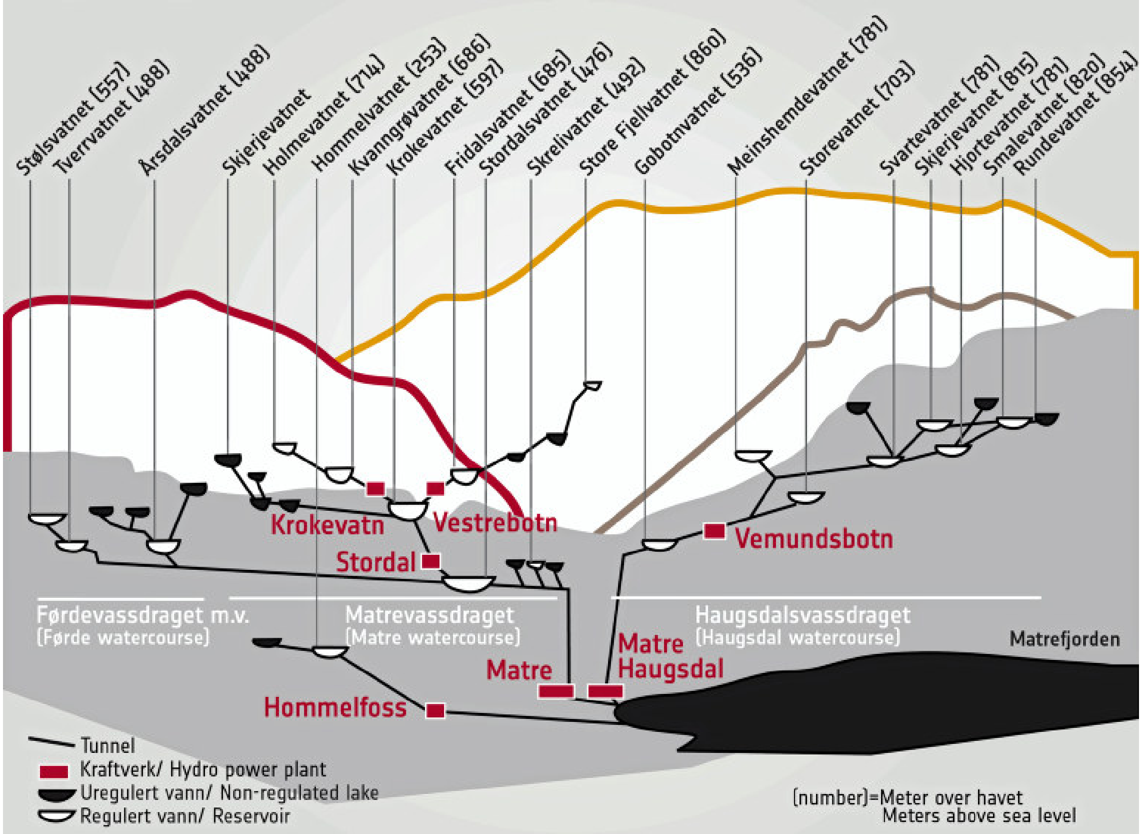


Figure 2.1: *Matre and adjacent watercourses (bkk.no, 2019)*

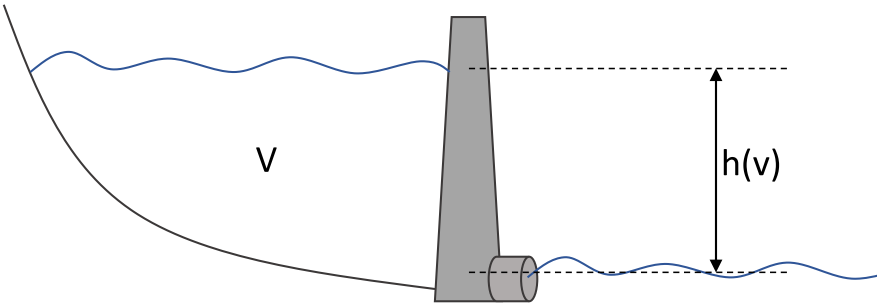


Figure 2.2: *Example of a reservoir with hydro plant*

Hydro plants

The main component of a hydro plant is the turbine. Figure 2.2 illustrates a hydro plant with a connected reservoir. As water is discharged through the turbine, it produces energy J with unit (Wh) given by

$$J(Q, V) = Q \cdot \eta(Q) \cdot \rho \cdot g \cdot h(V) \tag{2.1}$$

where

Q : water discharge through turbine (m^3/s)
V : reservoir volume (m^3)
$\eta(Q)$: turbine efficiency at discharge intensity Q
ρ : density of water (kg/m^3)
g : gravity acceleration (m/s^2)
$h(V)$: difference in reservoir and discharge water level at reservoir volume V, head height (m).

Note that hydro plants can be positioned in the watercourse in different ways than illustrated in Figure 2.2. For instance, some reservoirs discharge water through a pipeline leading to a turbine, resulting in greater head height.

Since turbines have a maximum discharge capacity it is possible to bypass the turbine, allowing water to flow downstream in order avoid reservoir overflow.

Reservoirs

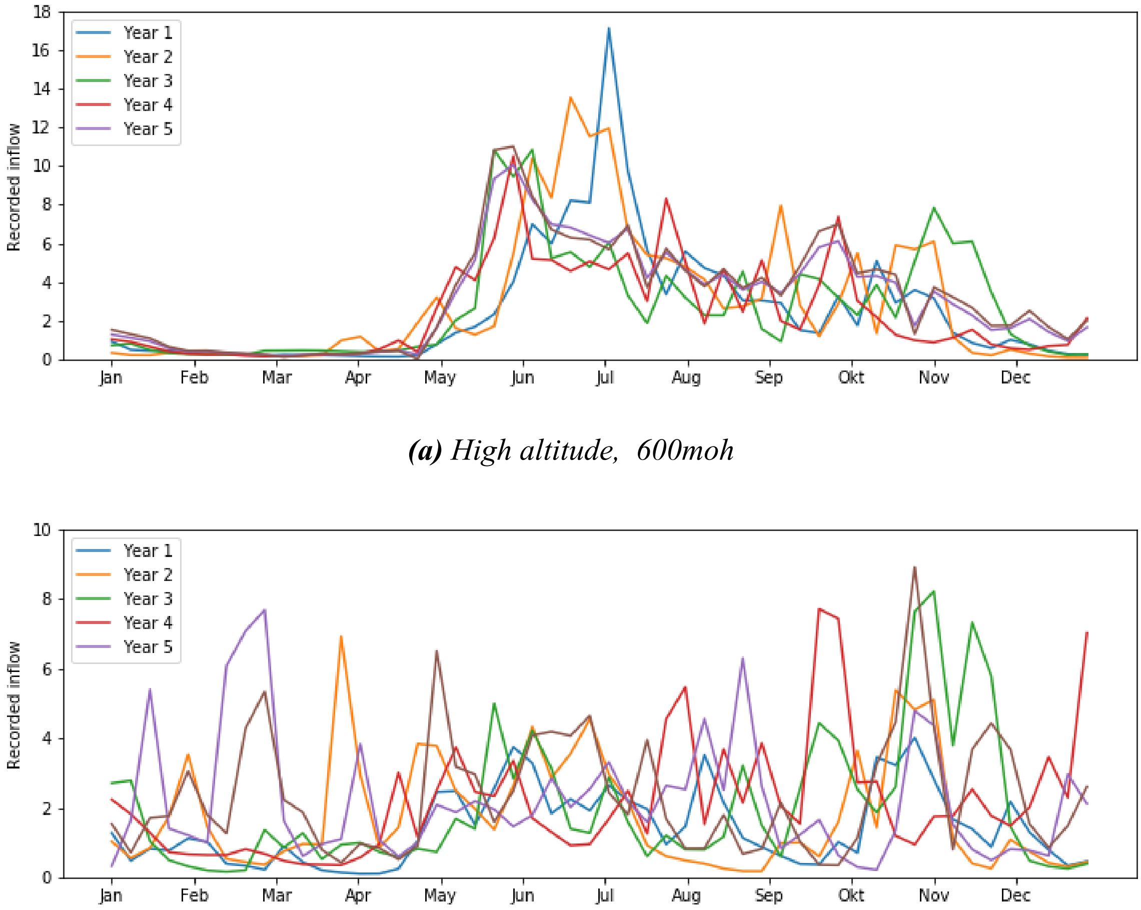
Reservoirs are usually regulated lakes or dams with either a hydro plant or another reservoir immediately downstream. It is reasonable to model a group of reservoirs with no plants in between as one reservoir, since the optimal strategy for these reservoirs is to ensure that the group is always able to release water to a downstream plant. In Hordaland, the watercourses are relatively steep, so delay between reservoirs will not be addressed here.

Due to environmental concerns, it is not common to be allowed to empty reservoirs completely. Hence, the allowed production volume is not equal to actual reservoir volume. If a reservoir overflows, the overflow can get directed back into reservoirs downstream, or get spilled out of the system.

2.1.2 Inflow

Inflow is an important aspect of the HPS problem. If storage levels are too high at the beginning of a period with high inflow, we risk overflowing reservoirs, which in turn may result in environmental damage and loss of resources. When dealing with a long planning period, it is hard to be certain about inflow forecasts. The producer should brace for the possibility of either entering wet or dry seasons. Hence, minimize the possibility of overflowing reservoirs, while maintaining production capability when prices are high.

Even though precipitation around the entire system may have low variation, inflow might vary substantially. Inflow into a reservoir depends on the size, topography and temperature of the inflow catchment area. Reservoirs in high altitudes tend to have longer and drier winters, and a larger spring melt, than low altitude reservoirs, where it tends to be warmer. Figure 2.3 shows recorded inflow into two different reservoirs, at two different elevations, for 5 different years. We see that inflow series may vary in both size and characteristics.



***(a)** High altitude, 600moh*

***(b)** Low altiude, 50moh*

***Figure 2.3:** Recorded inflow years into two different reservoirs (Haga, 2019)*

2.1.3 Price

As mentioned previously, the goal of the HPS problem is to achieve an optimal production schedule. In this case, that means the schedule that maximizes revenues. To do so, the producer want to sell their available resources at the highest prices. Therefore, it is important to have a good idea about the price of power during the planning period. However, this is challenging, and price forecasts always come with an uncertainty. The optimal production schedule is then the schedule that maximizes expected revenues.

2.1.4 Environmental restrictions

When regulating water levels of large lakes, it is important to respect the interests of local population, and keep environmental impacts at an acceptable level. In parts of the hydro system operated by BKK, there have been established some restrictions by the Norwegian Water Resources and Energy Directorate (**NVE**) on how BKK is allowed to regulate water levels at different times of the year. The following restrictions are imposed on the operation of the Hamlagrøvatn reservoir in the Bergsdalen watercourse.

During the winter, when local activity on the water is low, they are allowed to regulate water levels freely. During this time we have little inflow, we call this the *low inflow period*. When snow melts during the spring flow, we exit the low inflow period, and fill up the reservoir in order to maintain a suitable water level for the summer. The low inflow period ends:

- Never before 15. April.
- Between 15. April and 1. May the first day after experiencing
    - 5 consecutive days of inflow above the yearly mean, and there is less then 90% chance of reaching a water level of 584 meters above sea level (m.a.s.l.) within 1. July without ceasing production at that time. Probability calculations are made by NVE.
- Between 1. May and 15. May if not already ended, the first day after experiencing
    - 5 consecutive days of inflow above the yearly mean.
- 15. May if it has not already ended.

After the end of the low discharge period all discharge must cease until a water level of 584 m.a.s.l. is reached. After that, BKK is allowed to regulate water level freely above this water level until 15. August. From 15. August until 1. September discharge is only allowed as long as it does not lower the water level. This means that BKK can only discharge the inflow and water released from the reservoir above, named Torfinnsvatn.

2.2 Problem formulation

Here, we present the mathematical model of the scheduling problem, in order to formulate the HPS problem. As the uncertainty of inflow is prioritized we require a solution method that is able to solve stochastic problems. In order to be able to use the solution method proposed in Chapter 3, the HPS problem will be formulated as an LP problem. As mentioned in Section 1.1, the LP formulation has the advantage of giving shadow prices. In the data provided for this thesis, time is considered in discrete weekly timesteps. Thus, time will be considered in weekly **stages**, $t \in \{1, 2, ...T\}$, where T is the length of the planning period. Due to the length of our planning period, the weekly time-resolution is considered appropriate for the problem at hand.

2.2.1 System model

There are numerous strategies on how to model the structure of watercourses. Therefore, we propose a commonly accepted simplification of a hydro system. Assume that we aggregate nearby reservoirs, as mentioned in section 2.1.1, and assume that every reservoir has a turbine immediately downstream. All modules can release water to their immediate downstream reservoir (or out of the system at the bottom of the watercourse) by producing power (discharge) or bypassing the plant. We assume all overflow is lost. Then we can build the hydro systems of so called hydropower **modules**. Figure 2.4 is an example of a module representation of the Matre watercourse from Figure 2.1. The blue lines indicate inflow into each module, and black lines indicate waterways.

The efficiency function $\eta(Q)$ in (2.1) is not linear in Q. Usually, it is concave with maximum efficiency just below maximum production capacity. We can approximate $\eta(Q)$ as a piecewise linear function and include it in the HPS problem. Additionally, the head height function $h(V)$ depends on the shape of the reservoir, and might be convex. However, for large reservoirs the head height variations are small compared to the head height, and can be dealt with heuristically, as described by Gjelsvik et al. (2010). This work shows that an LP model of the HPS problem can approximate equation (2.1). However, we simplify the efficiency function significantly, and use an energy conversion factor for each reservoir, as the main focus of this thesis is to investigate effects of the constraints from section 2.1.4.

2.2.2 Inflow model

In order to make predictions for inflow during the planning period, we construct an inflow model. For modelling inflow, we assume a correlation between stages. If we know the inflow volumes for all reservoirs in stage t, we assume that we can somewhat predict all inflow volumes in stage $t + 1$. In order to create a model for predicting inflow, it is common to normalize the inflow series, in an attempt to remove seasonal variations and make the process stationary. Then, we need to define a function $f_{it}(Z_{it})$ to transform inflow values back to un-normalized values. We will go more into detail

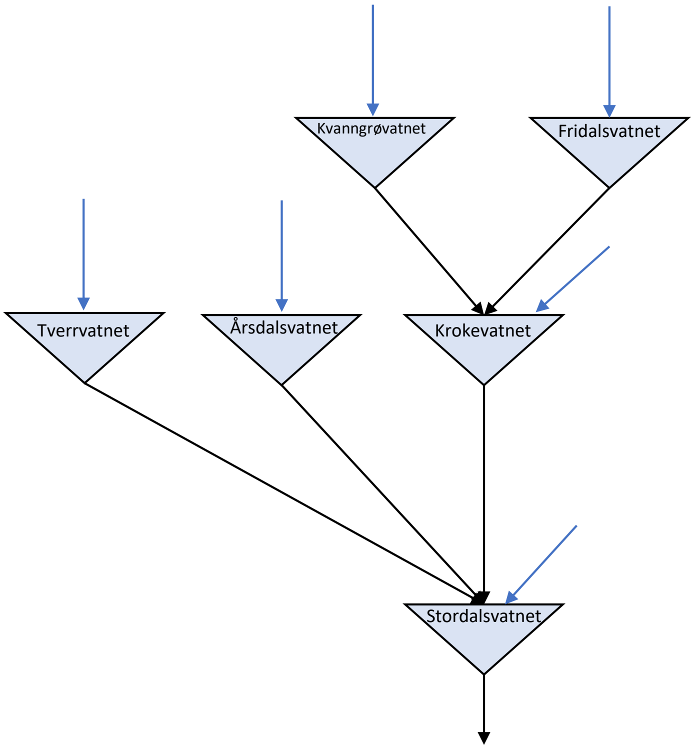


Figure 2.4: *Example of a hydro system modeled with hydro power modules*

about the inflow transformation in Section 4.1. Consider the normalized inflow Z_{it} into reservoir i, during time stage t. We will often refer to normalized inflow without specifying the fact that it is normalized. As the transformation presented in Section 4.1 is linear and continuous, the specification is often superfluous. Let ϕ_{ij} be a transition matrix for predicting inflow during week t (Z_{it}) given the previous week ($Z_{i,t-1}$), and ϵ_{it}^{k} for $k \in \{1, 2, ...K\}$ be K different prediction errors for week t. Then we can construct inflow series using the first order correlation

$$Z_{it} = \phi_{ij} Z_{i,t-1} + \epsilon_{it}^{k}. \tag{2.2}$$

Consider an arbitrary reservoir i. When constructing possible inflow scenarios we start with the known inflow of the present week $Z_{i,0}$, and use (2.2) with K different prediction errors $\{\epsilon_{i,1}^{1}, \epsilon_{i,1}^{2}, ..., \epsilon_{i,1}^{K}\}$, resulting in K possible values of $Z_{i,1}$. Proceeding the same way for all K values of $Z_{i,1}$ we get K^2 realizations of $Z_{i,2}$ and so forth. Figure 2.5 is a representation of a scenario tree with two possible prediction errors at every stage. The prediction errors are referred to as **inflow realizations**. We see that this inflow scenario tree has K^T possible scenarios. An exponentially growing scenario tree limits our methods of solving the HPS problem. However, a detailed inflow representation is prioritized due to the importance of handling inflow uncertainty. This motivates the stochastic solution method presented in Chapter 3.

Moreover, a correlation of more than one stage may reduce modelling error, but this increases the complexity of the solution method. However, a first order correlation is considered sufficient.

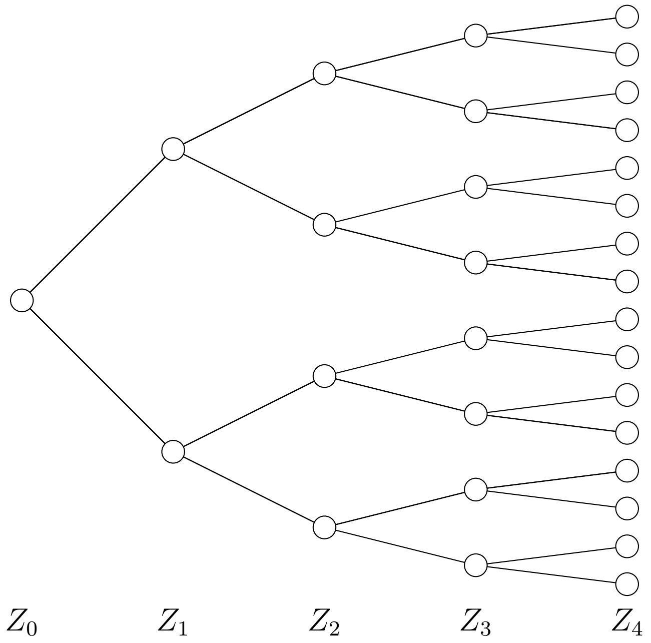


Figure 2.5: *Inflow scenario tree with* $K = 2, T = 4$

2.2.3 Price model

As mentioned, price forecasts always have some uncertainty, and should be modeled as a stochastic process. We assume that the producer is a *price taker*, meaning the producer does not affect the price when acting on the power market. Therefore, we consider the price as an exogenous factor. Due to the solution method we will arrive at in Section 3.4, the price component needs to be split up in discrete levels, and subproblems are to be solved for every price level. This is explained by Gjelsvik et al. (2010) and Pereira and Pinto (1991). Since the goal of this thesis is to investigate how the environmental restrictions from 2.1.4 affect the solution, we choose only one price series as a deterministic component, as the restrictions do not depend directly upon price. We can think that the environmental restrictions affect the subproblems for every price level in the same way. Therefore, the effects of state dependent environmental restrictions can be tested using a deterministic price series.

2.2.4 Environmental constraints

We suggest a slight simplification to the environmental restrictions presented in section 2.1.4. Since our model considers weekly stages, we must determine a weekly inflow amount $\overline{I}$ corresponding to five days of consecutive inflow above the yearly mean. Additionally, we disregard the possibility of not ending the low inflow period after sufficient inflow between 15. April and 1. May. Lastly, the dates cannot be represented exactly due to weekly resolution.

In order to formulate the environmental constraints we introduce the following symbols.

$$
\begin{aligned}
V_{2,t} &: \text{Reservoir volume of Hamlagrøvatn at end of week } t. \\
Q_{2,t} &: \text{Discharge volume of Hamlagrøvatn during week } t. \\
Q_{1,t} &: \text{Discharge volume of Torfinnsvatn during week } t. \\
B_{2,t} &: \text{Bypass volume of Hamlagrøvatn during week } t. \\
B_{1,t} &: \text{Bypass volume of Torfinnsvatn during week } t. \\
Z_{2,t} &: \text{Normalized inflow into Hamlagrøvatn during week } t. \\
\overline{I} &: \text{Sufficient inflow volume to end low inflow period.} \\
\underline{V}_2 &: \text{Reservoir volume corresponding to a water level of 584 m.a.s.l. in Hamlagrøvatn.} \\
t_{\text{15.April}} &: \text{Week containing 15. April.} \\
t_{\text{1.May}} &: \text{Week containing 1. May.} \\
t_{\text{15.May}} &: \text{Week containing 15. May.}
\end{aligned}
$$

The notation will formally be introduced in Section 2.2.5. Now we define the kick-in-times for the environmental constraints as

$$
\begin{aligned}
&t_A = \min\{\, \min\{t \geq t_{\text{15.April}} : f_{2,t}(Z_{2,t}) > \overline{I}\},\ t_{\text{15.May}}\}, && \text{End of low inflow period} \\
&t_B = \min\{t \geq t_0 : V_{2,t} \geq \underline{V}_2\}, && \text{Time when sufficient water level is reached} \\
&t_C : \text{Week containing 15. August} \\
&t_D : \text{Week containing 1. September}
\end{aligned}
$$

where all the kick-in-times are discrete values, since time is considered discretely. Then, the ”precise” constraints are formulated as

$$
\begin{aligned}
Q_{2,t} + B_{2,t} &= 0 \ \ \forall t \in [t_A, t_B)\,, && (2.3a) \\
V_{2,t} &\geq \underline{V}_2 \ \ \forall t \in [t_B, t_C)\,, && (2.3b) \\
Q_{2,t} + B_{2,t} &\leq f_{2,t}(Z_{2,t}) + Q_{1,t} + B_{1,t} \ \ \forall t \in [t_C, t_D]\,. && (2.3c)
\end{aligned}
$$

Constraint (2.3a) states that BKK is not allowed to release any water from the Hamlagrøvatn reservoir from the end of the low inflow period until the reservoir volume is greater than or equal to $\underline{V}_2$. Constraint (2.3b) states that once this $\underline{V}_2$ level is reached in the Hamlagrøvatn reservoir, then $\underline{V}_2$ is the lower bound on reservoir volume, until stage t_C. Lastly constraint (2.3c) states that the total water released from the Hamlagrøvatn reservoir cannot surpass total water coming into the reservoir, during any week between t_C and t_D. Now t_B is a discrete variable, depending on V_t. Additionally, as we explain in Section 4.3.2, inflow will be considered as a variable, meaning t_A also depends on a variable. Thus, we cannot represent the constraints (2.3) exactly in our LP formulation of the HPS problem. Therefore, the constraints do not work with the solution method proposed in Chapter 3. However, we prioritize dealing with the inflow uncertainty, thereby simplifying the constraints in order to try to respect them in our solution. In Section 3.6 we present a simplification to deal with constraint (2.3a) and a heuristic to deal with (2.3b).

2.2.5 Optimization problem

Here we define the HPS problem as an LP problem, without considering the environmental constraints (2.3). We assume unlimited power transfer capacity from outside the system. This means the producer does not have an obligation to meet power demand, and can choose to let consumers buy power from outside the system.

For the presented modeling components we define the following components.

Index sets

$\mathcal{T} = \{1, 2, ...T\}$, index set for stages.

$\mathcal{S} = \{1, 2, ...S\}$, index set for inflow scenarios.

$\mathcal{I} = \{1, 2, ...I\}$, index set for reservoirs.

$\mathcal{K} = \{1, 2, ...K\}$, index set for inflow realizations.

$\mathcal{M}_i$: index set for all reservoirs immediately upstream of reservoir i.

Parameters

T : Length of the planning period

S : Number of inflow scenarios

I : Number of reserviors

K : Number of Inflow realizations

p_t : Power price during stage t.

Z^s_{it} : Normalized inflow into reservoir i during time t in scenario s.

$f_{it}(Z^s_{it})$: Inflow transformation function, for transforming normalized inflow.

η_i : Energy conversion factor for reservoir i.

$\mathbb{P}(s)$: Probability of scenario s occurring.

$\Phi(V_{1,t}, V_{2,t}, ...V_{IT})$: Value of the remaining water at the end of the planning period.
as a function of end reservoir volumes.

$\overline{V}_i$: Max volume of reservoir i.

$\overline{Q}_i$: Max weekly discharge of reservoir i.

$\overline{B}_i$: Max weekly bypass of reservoir i.

Variables

V_{it}^s : Volume of reservoir i at the end of time t in scenario s.
Q_{it}^s : Discharge of reservoir i during time t in scenario s.
B_{it}^s : Bypass of reservoir i during time t in scenario s.
O_{it}^s : Overflow of reservoir i during time t in scenario s.
y_t^s : Power sold during time t in scenario s.

Then the HPS problem is formulated as,

$$\underset{V_{it}^s, Q_{it}^s, B_{it}^s, O_{it}^s, y_t^s}{\text{maximize}} \quad \sum_{s \in \mathcal{S}} \mathbb{P}(s) \left(\sum_{t=1}^{T} p_t y_t^s + \Phi(V_{1,T}, V_{2,T}, ...V_{IT}) \right) \tag{2.4a}$$

subject to

$$\text{Non-anticipativity constraints ,} \tag{2.4b}$$

$$V_{it}^s + Q_{it}^s + B_{it}^s + O_{it}^s - V_{i,t-1}^s - \sum_{j \in \mathcal{M}_i} Q_{jt}^s + B_{jt}^s = f(Z_{it}^s), \quad \forall (i,s,t) \in \mathcal{I} \times \mathcal{S} \times \mathcal{T}, \tag{2.4c}$$

$$\sum_{i \in \mathcal{I}} \eta_i Q_{it}^s - y_t^s = 0 \quad , \quad \forall (s,t) \in \mathcal{S} \times \mathcal{T}, \tag{2.4d}$$

$$\overline{V}_i \geq V_{it}^s \geq 0 \quad , \quad \forall (i,s,t) \in \mathcal{I} \times \mathcal{S} \times \mathcal{T}, \tag{2.4e}$$

$$\overline{Q}_i \geq Q_{it}^s \geq 0 \quad , \quad \forall (i,s,t) \in \mathcal{I} \times \mathcal{S} \times \mathcal{T}, \tag{2.4f}$$

$$\overline{B}_i \geq B_{it}^s \geq 0 \quad , \quad \forall (i,s,t) \in \mathcal{I} \times \mathcal{S} \times \mathcal{T}, \tag{2.4g}$$

$$O_{it}^s \geq 0 \quad , \quad \forall (i,s,t) \in \mathcal{I} \times \mathcal{S} \times \mathcal{T}, \tag{2.4h}$$

$$y_t^s \geq 0 \quad , \quad \forall (s,t) \in \mathcal{S} \times \mathcal{T}. \tag{2.4i}$$

The optimization problem (2.4) formulates the HPS problem as an LP problem with discrete stages and a scenario tree to represent inflow uncertainty. In Chapter 3, we investigate a solution method for this problem. In Section 3.6 we present a linear approximation of the environmental constraints (2.3) for the Bergsdalen watercourse.

The objective function (2.4a) is the expected total revenue of the planning period. However, to avoid emptying reservoirs at the end of the planning period, which of course would make the most money during the planning period, we need to formulate a reward for saving some of the water for future use. This reward is reflected by $\Phi(\cdot)$. The production schedule is given by the distribution of

$$\{V_{it}^s, Q_{it}^s, B_{it}^s, O_{it}^s \quad \forall (i,t,s) \in \mathcal{I} \times \mathcal{S} \times \mathcal{T}\},$$

among scenarios $\mathcal{S}$.

Non-anticipativity constraints

The constraints (2.4b) represent non-anticipativity. They state that scenarios with a common history should have the same set of decisions. That is, a decision made in each $(s,t) \in \mathcal{S} \times \mathcal{T}$, should be mutual for all scenarios sharing the same history as that decision. Such constraints reflect the fact that, we can't anticipate the future. As an example, consider Figure 2.6, representing 4 inflow scenarios. Let x_t^s be decisions made in each stage and scenario. The non-anticipativity constraints for this scenario tree are

$$\begin{aligned} x_0^1 = x_0^2 = x_0^3 = x_0^4, \\ x_1^1 = x_1^2, \\ x_1^3 = x_1^4. \end{aligned}$$

Possible examples of the decisions x_t^s are the decision variables $V_{it}^s, Q_{it}^s, B_{it}^s, O_{it}^s, y_t^s$ in the optimization problem (2.4). As the non-anticipativity constrains are cumbersome to formulate, and we implicitly respect them in the proposed solution method, we do not give them in full detail for this problem.

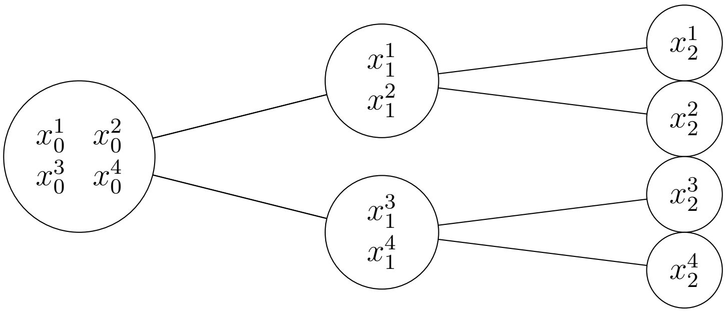


Figure 2.6: *Inflow scenario tree with $K = 2, T = 2$*

Water balance

To model the flow of water in the system in an arbitrary scenario s and stage t, consider the following equation:

$$V_{i,t-1}^s + \big(f_{it}(Z_{it}^s) + \sum_{j \in \mathcal{M}_i} (Q_{jt}^s + B_{jt}^s)\big) - \big(Q_{it}^s + B_{it}^s + O_{it}^s\big) = V_{it}^s.$$

We have a volume $V_{i,t-1}^s$ of water at the start of stage t. Consider the flow of water in and out of the reservoir, as illustrated by Figure 2.7. Inflow and water from the immediate upstream hydro plant flows into the reservoir, discharge and bypass are passed downstream, and overflow is lost. Then at the end of the week, the volume is V_t^s. Constraint (2.4c), referred to as *water balance*, reflects the water flow. Note that overflow is allowed to happen even when the reservoir is not full, which is not a correct physical representation. However, overflow leads to loss of resources, hence loss of revenues. Thus, we assume positive overflow values only when absolutely necessary.

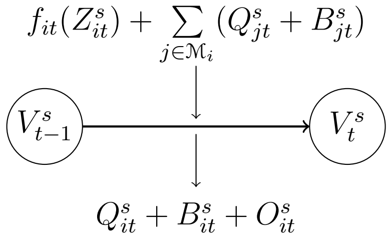

$f_{it}(Z^s_{it}) + \sum_{j\in\mathcal{M}_i}(Q^s_{jt} + B^s_{jt})$
V^s_{t-1}
V^s_t
$Q^s_{it} + B^s_{it} + O^s_{it}$

Figure 2.7: *Representation of the flow of water in a stage*

Power balance

Constraint (2.4d), *power balance*, is strictly not necessary. However, if the piecewise linear approximation of $\eta(Q)$ in equation (2.1), discussed in Section 2.2.1, is to be implemented, we can reformulate the constraint to include the approximation.

Decision variable bounds

The constraints (2.4e), (2.4f) and (2.4g) are bounds on volume, discharge and bypass, respectively. As overflow leads to a loss of revenues, we do not specify an upper bound. Note that volume levels have zero as lower bound, even though we are not allowed to empty any reservoirs completely. Volumes are thus allowed production volumes, not actual volumes of reservoirs .

Chapter 3

Solution method

In this chapter, we will investigate a solution method for problem (2.4), the HPS problem. This solution method is commonly used for multi-stage stochastic problems, and the solution algorithm we arrive at is inspired by the one presented in Gjelsvik et al. (2010). We will illustrate the solution method for a general example, which reflects the features of the an LP.

Consider a general multi-stage stochastic problem, as formulated by problem (3.1), where each stage t depends on the previous, $t-1$. This problem reflects the features of the HPS problem (2.4). Non-anticipativity constraints are as described in Section 2.2.5. The goal is to maximize expected total revenues of a hydro system, with respect to an uncertain parameter $\boldsymbol{D}_t$. In order to illustrate the solution method we define the following components.

$\mathbf{E}_s$: Expectation value operator, with respect to scenarios.

Parameters:

T : Planning horizon.

n : Number of decision variables in each stage.

m : Number of constraints in each stage.

$\boldsymbol{c}_t \in \mathbb{R}^n$: Revenues of decisions in stage t. Known parameter.

$\boldsymbol{x}_0^s \in \mathbb{R}_+^n$: Initial state. Known parameter, equal for all $s \in \mathbb{S}$.

$\boldsymbol{D}_t \in \mathbb{R}_+^n$: Upper bound on decisions in stage t. Stochastic variable.

$\boldsymbol{d}_t^s \in \mathbb{R}_+^m$: Realization of $\boldsymbol{D}_t$ in scenario s.

$\boldsymbol{d}_0^s \in \mathbb{R}_+^m$: Initial realization of $\boldsymbol{D}_t$, equal for all $s \in \mathbb{S}$.

$A \in \mathbb{R}^{m,n}$: Coefficient matrix.

$F \in \mathbb{R}^{m,n}$: Coefficient matrix.

p^k : Probability of realization k of $\boldsymbol{D}_t$.

$\mathbb{P}(s)$: Probability of scenario s occurring.

Index sets:

$\mathcal{T} = \{1, 2, ...T\}$, index set for stages.

$\mathcal{S}$: Index set for scenarios.

Variables:

$\boldsymbol{x}_t^s \in \mathbb{R}_+^n$: Vector of decision variables in stage t and scenario s.

We assume that the outcomes of $\boldsymbol{D}_t$ are finite, then, $\mathcal{S}$ is a finite set. The scenarios are all possible sequences of realizations of $\boldsymbol{D}_t$ for all t in the ordered set $\mathcal{T}$. The uncertain parameter $\boldsymbol{D}_t$ represents inflow modelling error. The variables in $\boldsymbol{x}_t^s$ represents decisions (i.e. production, bypass) and state (i.e. reservoir volume, inflow) of our hydro system in $(s,\ t)$. The decisions made in $(s,\ t-1)$ affects the state of our system, and hence the decisions that can be made in $(s,\ t)$. We begin in the present state $\boldsymbol{x}_0^s$ and optimize for all possible sequences of $\boldsymbol{D}_t$ towards the planning horizon T. Then the general multi-stage stochastic problem is

$$\begin{array}{lll} \underset{\boldsymbol{x}_t^s}{\text{maximize}} & \mathbf{E}_s \left[\displaystyle\sum_{t \in \mathcal{T}} \boldsymbol{c}_t' \boldsymbol{x}_t^s \right] & \\ \text{subject to} & & \text{Non-anticipativity constraints,} \\ & A\boldsymbol{x}_t^s + F\boldsymbol{x}_{t-1}^s \leq \boldsymbol{d}_t^s, & \forall (s,t) \in \mathcal{S} \times \mathcal{T}, \\ & \boldsymbol{x}_t^s \geq \mathbf{0}\ , & \forall (s,t) \in \mathcal{S} \times \mathcal{T}. \end{array} \tag{3.1}$$

In stage $t = 1$ the term $F\boldsymbol{x}_{t-1}^s$ is known and should be on the right hand side of the equation. The matrices A and F represent coefficients for the set of constraints in each $(s,t) \in \mathcal{S} \times \mathcal{T}$. $\mathbf{0}$ is the zero vector, containing n zeroes.

Assume that for every stage t there are K possible outcomes of $\boldsymbol{D}_t$, each with a probability of p^k $\forall k \in \{1, 2, ..., K\}$, such that

$$\sum_{k=1}^{K} p^k = 1.$$

That is

$$\boldsymbol{D}_t \in \{\boldsymbol{d}_t^1, \boldsymbol{d}_t^2, ...\boldsymbol{d}_t^K\}, \quad \forall t \in \mathcal{T} \tag{3.2}$$

with the probability mass function

$$f_{\boldsymbol{D}_t}(\boldsymbol{d}) = p^k, \quad \boldsymbol{d} = \boldsymbol{d}_t^k, \quad \forall k \in \{1, 2, ...K\}.$$

The realizations of $\boldsymbol{D}_t$ can be represented by a scenario tree as illustrated in Figure 2.5. The index set for scenarios is

$$\mathcal{S} = \{1, 2, ...K^T\}, \tag{3.3}$$

which we see grows exponentially with planning horizon T.

3.1 The Deterministic Equivalent

If the planning horizon T is aptly short, the optimization problem (3.1) can be solved as the deterministic equivalent of problem (3.1), formulated as

$$\begin{aligned} \underset{\boldsymbol{x}_t^s}{\text{maximize}} \quad & \sum_{s\in\mathcal{S}} \mathbb{P}(s) \sum_{t\in\mathcal{T}} \boldsymbol{c}_t' \boldsymbol{x}_t^s \\ \text{subject to} \quad & \text{Non-anticipativity constraints}, \\ & A\boldsymbol{x}_t^s + F\boldsymbol{x}_{t-1}^s \le \boldsymbol{d}_t^s, \quad \forall (s,t) \in \mathcal{S}\times\mathcal{T}, \\ & \boldsymbol{x}_t^s \ge \boldsymbol{0}\ , \quad \forall (s,t) \in \mathcal{S}\times\mathcal{T}, \end{aligned} \tag{3.4}$$

where the probability of each scenario is the product of all outcomes in each scenario

$$\mathbb{P}(s) = \prod_{t=1}^{T} f_{\boldsymbol{D}_t}(\boldsymbol{d}_t^s).$$

However, in the problem presented in Chapter 2, we want to be able to handle a planning horizon ranging between 1 and 3 years, with weekly resolution, using the inflow model described in 2.2.2. In the best case the planning horizon is $T = 52$ and the inflow realizations per stage are $K = 2$. Then, problem (3.4) has $K^T \sim 4.5 \cdot 10^{15}$ sets of decision variables and constraints. For formulating a computationally feasible solution method, we will instead use the principles of stochastic dynamic programming and Benders decomposition to converge to an optimal solution of problem (3.1). First however, for the purpose of illustration, we will describe the solution method without considering uncertainty.

3.2 A Dynamic Programming Approach

Consider a version of problem (3.1) with no uncertainty of $\boldsymbol{D}_t$. That is, $K = 1$ and $\mathcal{S} = \{1\}$. The optimal solution can of course be found by solving problem (3.4), however we illustrate a dynamic programming approach in order to be able to expand to $K > 1$ and T of significant size. For now we omit the scenario superscript.

For the purpose of illustration, assume that each $\boldsymbol{x}_t$ can take J discrete values, or **states**. For now, accept the fact that we have an integer programming problem, opposed to an LP problem as intended. By applying the Bellman principle of optimality we can solve this problem recursively. First, we split Problem (3.1) into T stages, each with a **future income** function $\alpha_t(\boldsymbol{x}_t)$ as such,

$$\begin{aligned} \alpha_{t-1}(\boldsymbol{x}_{t-1}) = \underset{\boldsymbol{x}_t}{\text{maximize}} \quad & \boldsymbol{c}_t'\boldsymbol{x}_t + \alpha_t(\boldsymbol{x}_t) \\ \text{subject to} \quad & A\boldsymbol{x}_t \le \boldsymbol{d}_t - F\boldsymbol{x}_{t-1}, \\ & \boldsymbol{x}_t \ge \boldsymbol{0}, \end{aligned} \tag{3.5}$$

for all $t \in \mathcal{T}$. Note that, $\alpha_T(\boldsymbol{x}_T) = 0$ since we do not consider any stages after T, and $\alpha_0(\boldsymbol{x}_0)$ is redundant. Then, by solving problem (3.5) in stage t for all J states of $\boldsymbol{x}_{T-1}$

we find $\alpha_{T-1}(\boldsymbol{x}_{T-1})$, which we in turn use for solving (3.5) in stage $t-1$ for all J states of $\boldsymbol{x}_{T-2}$. By repeating for all $t \in \{T, T-1, ...1\}$, we get the optimal solution in the first stage as

$$\boldsymbol{c}_1'\boldsymbol{x}_1^* + \alpha_1(\boldsymbol{x}_1^*),$$

where $\boldsymbol{x}_t^*$ is the optimal solution of problem (3.5) in stage $t = 1$.

To use this solution method on an LP problem, that is when $\boldsymbol{x}_t$ is continuous, we need to determine a way to approximate $\alpha_t(\boldsymbol{x}_t)$ from a finite set of points, $\{\hat{\boldsymbol{x}}_t^1, \hat{\boldsymbol{x}}_t^2, ...\} \subset \mathbb{R}_+^n$. One way of doing so is simply choosing a sufficient number of points to interpolate between, in order to obtain a sufficiently accurate approximation of $\alpha_t(\boldsymbol{x}_t)$. If R states of $\boldsymbol{x}_t$ are chosen, then we can solve problem (3.5) in stage $t+1$ for all $\boldsymbol{x}_t \in \{\hat{\boldsymbol{x}}_t^1, \hat{\boldsymbol{x}}_t^2, ...\hat{\boldsymbol{x}}_t^R\}$ to get points to interpolate between. However, if $\boldsymbol{x}_t$ has a large number of components N and we need J values of each component for a sufficiently accurate approximation, we need to solve problem (3.5) J^N times for every stage. This is bad when considering a multi-reservoir hydro system, where n should be allowed to be a high number. Therefore, we propose an alternative approach using Benders decomposition.

3.3 Future income function from the Dual Problem

Consider a two-stage version of problem (3.1) with no uncertainty. That is, $\mathcal{T} = \{1, 2\}$ and $\mathcal{S} = \{1\}$. We still omit the scenario index. The problem can then be written as

$$\begin{aligned} \underset{\boldsymbol{x}_1, \boldsymbol{x}_2}{\text{maximize}} \quad & \boldsymbol{c}_1'\boldsymbol{x}_1 + \boldsymbol{c}_2'\boldsymbol{x}_2 \\ \text{subject to} \quad & A\boldsymbol{x}_1 \leq \boldsymbol{d}_1 - F\boldsymbol{x}_0, \\ & A\boldsymbol{x}_2 + F\boldsymbol{x}_1 \leq \boldsymbol{d}_2, \\ & \boldsymbol{x}_1, \boldsymbol{x}_2 \geq \boldsymbol{0}. \end{aligned} \tag{3.6}$$

We can decompose problem (3.6) into

$$\begin{aligned} \underset{\boldsymbol{x}_1}{\text{maximize}} \quad & \boldsymbol{c}_1'\boldsymbol{x}_1 + \alpha(\boldsymbol{x}_1) \\ \text{subject to} \quad & A\boldsymbol{x}_1 \leq \boldsymbol{d}_1 - F\boldsymbol{x}_0, \\ & \boldsymbol{x}_1 \geq \boldsymbol{0}. \end{aligned} \tag{3.7a}$$

$$\begin{aligned} \alpha(\boldsymbol{x}_1) = \underset{\boldsymbol{x}_2}{\text{maximize}} \quad & \boldsymbol{c}_2'\boldsymbol{x}_2 \\ \text{subject to} \quad & A\boldsymbol{x}_2 \leq \boldsymbol{d}_2 - F\boldsymbol{x}_1, \\ & \boldsymbol{x}_2 \geq \boldsymbol{0} \end{aligned} \tag{3.7b}$$

Since problem (3.7b) is an LP problem, $\alpha(\boldsymbol{x}_1)$ is a concave piecewise linear function. Assume that we have an initial guess, $\hat{\boldsymbol{x}}_1$, of the optimal decision in the first stage, $\boldsymbol{x}_1^*$. Now consider the dual of problem (3.7b),

$$\begin{aligned} \alpha(\boldsymbol{x}_1) = \underset{\boldsymbol{\pi}}{\text{minimize}} \quad & (\boldsymbol{d}_2 - F\boldsymbol{x}_1)'\boldsymbol{\pi} \\ \text{subject to} \quad & A'\boldsymbol{\pi} \geq \boldsymbol{c}_2, \\ & \boldsymbol{\pi} \geq \boldsymbol{0}. \end{aligned} \tag{3.8}$$

Since the dual problem (3.8) is a linear minimization problem, we know that,

$$\alpha(\boldsymbol{x}_1) = \min_{\boldsymbol{\pi}}\{(\boldsymbol{d}_2 - F\boldsymbol{x}_1)'\boldsymbol{\pi} : A'\boldsymbol{\pi} \geq \boldsymbol{c}_2\} \leq (\boldsymbol{d}_2 - F\boldsymbol{x}_1)'\hat{\boldsymbol{\pi}}, \tag{3.9}$$

for any feasible $\hat{\boldsymbol{\pi}}$. Now choose

$$\hat{\boldsymbol{\pi}} \in \arg\min_{\boldsymbol{\pi}}\{(\boldsymbol{d}_2 - F\hat{\boldsymbol{x}}_1)'\boldsymbol{\pi} : A'\boldsymbol{\pi} \geq \boldsymbol{c}_2\}, \tag{3.10}$$

for the initial guess $\hat{\boldsymbol{x}}_1$. Then we have an upper bound for $\alpha(\boldsymbol{x}_1)$

$$\alpha(\boldsymbol{x}_1) \leq (\boldsymbol{d}_2 - F\boldsymbol{x}_1)'\hat{\boldsymbol{\pi}}. \tag{3.11}$$

Note that, for $\boldsymbol{x}_1 = \hat{\boldsymbol{x}}_1$ inequality (3.11) is fulfilled as an equality.

By including equation 3.11 into the first stage problem 3.7a, we have obtained a linear upper bound for the future income function, known as a Benders **cut**. We can now solve

$$\underset{\boldsymbol{x}_1, \alpha}{\text{maximize}} \quad \boldsymbol{c}_1'\boldsymbol{x}_1 + \alpha \tag{3.12a}$$

$$\text{subject to} \quad A\boldsymbol{x}_1 \leq \boldsymbol{d}_1 - F\boldsymbol{x}_0, \tag{3.12b}$$

$$\alpha + \boldsymbol{x}_1'F'\hat{\boldsymbol{\pi}} \leq \boldsymbol{d}_2'\hat{\boldsymbol{\pi}}, \tag{3.12c}$$

$$\boldsymbol{x}_1 \geq \boldsymbol{0} \tag{3.12d}$$

to obtain a new guess of the optimal state $\boldsymbol{x}_1^*$. Here, $\alpha(\boldsymbol{x}_1)$ is represented by a free variable which, due to the problem being a maximization problem, will take a value on its upper bound.

To get an accurate representation of $\alpha(\boldsymbol{x}_1)$, we have to obtain an upper bound for each vertex in the solution space of the dual problem (3.8) of the second stage problem. If there are too many vertices we can keep solving equation (3.10) for different values of $\hat{\boldsymbol{x}}_1$, and keep adding cuts, until we have a sufficiently accurate representation of $\alpha(\boldsymbol{x}_1)$.

Since problem (3.12) is a maximization problem we have that $\alpha \geq \alpha(\boldsymbol{x}_1)$, which is fulfilled as an equality when the cuts represent $\alpha(\boldsymbol{x}_1)$ precisely. Hence,

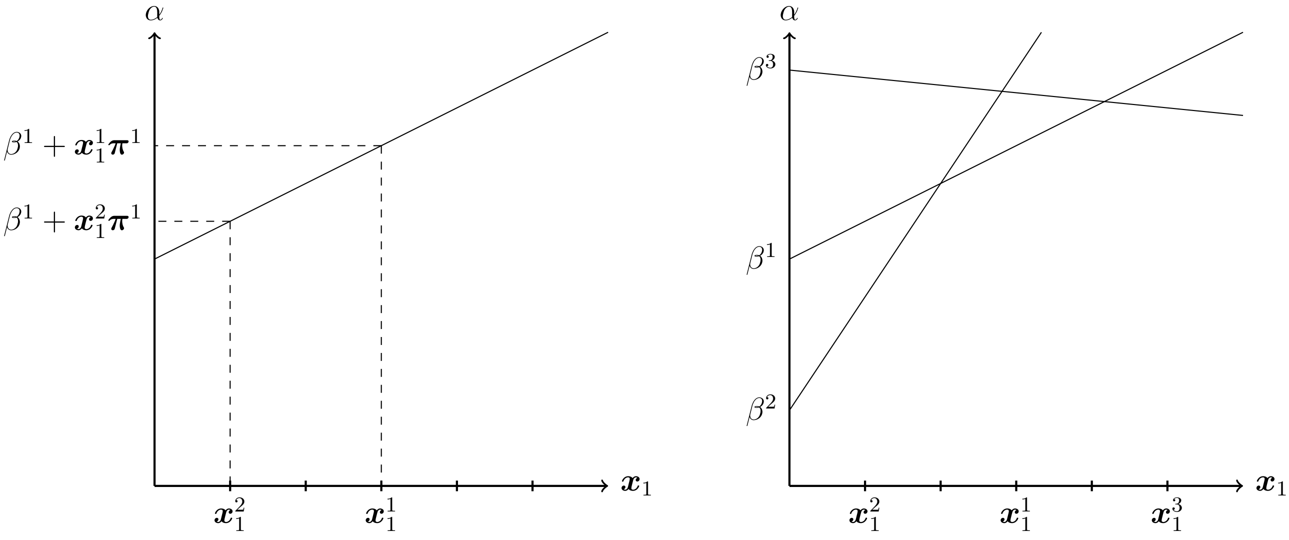
$$\boldsymbol{c}_1'\boldsymbol{x}_1 + \boldsymbol{c}_2'\boldsymbol{x}_2 \leq \boldsymbol{c}_1'\boldsymbol{x}_1 + \alpha. \tag{3.13}$$

By solving the two stages iteratively, we get a new upper bound for $\alpha(\boldsymbol{x}_1)$ from solving problem (3.7b) and a new value of $\hat{\boldsymbol{x}}_1$ from solving (3.7a). At each iteration, since we get a better representation of $\alpha(\boldsymbol{x}_1)$, we get closer to the optimal state $\boldsymbol{x}_1^*$, hence each new cut is more ”relevant”. The optimal solution is then where inequality (3.13) is fulfilled as an equality. Note that by iteratively improving $\hat{\boldsymbol{x}}_1$ towards its optimal value, we might not need to construct cuts for all vertices in the second stage dual problem.

For notational convenience, we propose a reformulation of the Benders cuts. Consider equation (3.11), which we write as

$$\alpha - \boldsymbol{x}_1' F' \hat{\boldsymbol{\pi}} \leq \beta, \tag{3.14}$$

where $\beta = \boldsymbol{d}_2' \hat{\boldsymbol{\pi}}$. Figure 3.1 illustrates how cuts are made iteratively to improve the representation of the future income function $\alpha(\boldsymbol{x}_1)$. The superscripts of $\boldsymbol{x}_1$ indicate different values $\hat{\boldsymbol{x}}_1$.



(a) *Cut calculated for* $\boldsymbol{x}_1^1$

(b) *Piecewise linear future income function*

Figure 3.1: *One dimensional representation of cuts*

3.4 Expected Future Income Function

Consider now the two-stage version of problem (3.1), where all parameters of the first stage are known, and the second stage has two possible outcomes of $\boldsymbol{D}_2$, $\boldsymbol{d}_2^1$ and $\boldsymbol{d}_2^2$. Let p^1 and p^2 be the probabilities of outcome $\boldsymbol{d}_2^1$ and $\boldsymbol{d}_2^2$ respectively, such that $p^1 + p^2 = 1$. Then, $K = 2$ for the second stage, and $\mathcal{S} = \{1, 2\}$. We can decompose this problem into

$$\begin{aligned} \underset{\boldsymbol{x}_1, \overline{\alpha}}{\text{maximize}} \quad & \boldsymbol{c}_1' \boldsymbol{x}_1 + \overline{\alpha} \\ \text{subject to} \quad & A\boldsymbol{x}_1 \leq \boldsymbol{d}_1 - F\boldsymbol{x}_0, \\ & \boldsymbol{x}_1 \geq 0. \end{aligned} \tag{3.15a}$$

$$\begin{aligned} \alpha(\boldsymbol{x}_1|\boldsymbol{d}_2^1) = \underset{\boldsymbol{x}_2^1}{\text{maximize}} \quad & \boldsymbol{c}_2'\boldsymbol{x}_2^1 \\ \text{subject to} \quad & A\boldsymbol{x}_2^1 \leq \boldsymbol{d}_2^1 - F\boldsymbol{x}_1, \\ & \boldsymbol{x}_2^1 \geq 0 \end{aligned} \tag{3.15b}$$

$$\begin{aligned} \alpha(\boldsymbol{x}_1|\boldsymbol{d}_2^2) = \underset{\boldsymbol{x}_2^2}{\text{maximize}} \quad & \boldsymbol{c}_2'\boldsymbol{x}_2^2 \\ \text{subject to} \quad & A\boldsymbol{x}_2^2 \leq \boldsymbol{d}_2^2 - F\boldsymbol{x}_1, \\ & \boldsymbol{x}_2^2 \geq 0 \end{aligned} \tag{3.15c}$$

Note that problem (3.15a) is unbounded before we have found any cuts for $\overline{\alpha}$. In order to handle the uncertainty in $\boldsymbol{D}_2$ we calculate expected cuts, for approximating the **expected future income** function, $\overline{\alpha}(\boldsymbol{x}_1)$. Assume an initial guess of optimal decision in the first stage $\hat{\boldsymbol{x}}_1$. Then, by solving both problems (3.15b) and (3.15c) for $\boldsymbol{x}_1 = \hat{\boldsymbol{x}}_1$, we obtain their dual values $\boldsymbol{\pi}^1, \boldsymbol{\pi}^2$ and their objective function values β^1, β^2 as described in section 3.3. Then, the coefficients of the expected cut are

$$\begin{aligned} \overline{\boldsymbol{\pi}} &= p^1\boldsymbol{\pi}^1 + p^2\boldsymbol{\pi}^2 \\ \overline{\beta} &= p^1\beta^1 + p^2\beta^2, \end{aligned}$$

assuming the 2 outcomes are equiprobable. Lastly, the expected cut is

$$\overline{\alpha} - \boldsymbol{x}_1'F'\overline{\boldsymbol{\pi}} \leq \overline{\beta}, \tag{3.16}$$

which we add as a constraint to problem (3.15a). Now we can solve problem (3.15a) to obtain a better guess $\hat{\boldsymbol{x}}_1$ of the optimal solution to the first stage problem. This also gives a value for expected future income; the optimal value of $\overline{\alpha}$. By repeating the process we obtain a better representation of $\overline{\alpha}(\boldsymbol{x}_1)$. As we did in the previous section with inequality (3.13), we have the optimal solution when

$$\boldsymbol{c}_1'\hat{\boldsymbol{x}}_1 + p^1\boldsymbol{c}_2'\hat{\boldsymbol{x}}_2^1 + p^2\boldsymbol{c}_2'\hat{\boldsymbol{x}}_2^2 = \boldsymbol{c}_1'\hat{\boldsymbol{x}}_1 + \overline{\alpha}.$$

Here, $\hat{\boldsymbol{x}}_2^1$ and $\hat{\boldsymbol{x}}_2^2$ are the optimal solutions of problems (3.15b) and (3.15c) respectively, with $\boldsymbol{x}_1 = \hat{\boldsymbol{x}}_1$.

3.5 A Stochastic Dual Dynamic Programming Algorithm

In this section we will outline the two main components of our Stochastic Dual Programming (SDDP) Algorithm, the backward recursion and forward simulation. The algorithm is used to solve problem (3.1). This is an abstract version of the algorithm we use to solve the HPS problem (2.4). We add the following definitions in order to describe the algorithm.

$\hat{S}$: Number of scenarios to sample at every iteration.

$\hat{\mathcal{S}} \subset \mathcal{S}$: Index set for sampled scenarios. Sampled from $\mathcal{S}$.

R : Number of cuts at each stage.

$\hat{\boldsymbol{x}}_t^s \in \mathbb{R}_+^n$: Initial guess of optimal decision for all $(s,t) \in \hat{\mathcal{S}} \times \mathcal{T}$.

K : Number of possible parameter realizations between each stage.

$\mathcal{K} = \{1, 2, ...K\}$: Index set for possible parameter realizations between each stage.

U : Upper bound on the optimal objective function value.

L : Lower bound on the optimal objective function value.

Start with a set of sampled scenarios for $\boldsymbol{d}_t^s$, where $s \in \hat{\mathcal{S}}$. Then, recursively solve all sub problems for the stages $t \in \{T, T-1, ...2\}$ and construct expected cuts for all $t-1$ stages, as described in Algorithm 1. The upper limit U will be used for checking convergence. In this example we assume that there is no future income in the last stage. We initialize $R = 0$.

Algorithm 1 Backward recursion

```
for $t = [T-1, T-2, ...1]$ do
  Set $u = R + 1$, as new cut index in stage $t$
  for $s \in \hat{\mathcal{S}}$ do
    for $k \in \mathcal{K}$ do
      if $t + 1 = T$ then
        Replace constraints (3.17c) with $\overline{\alpha}_T = 0$
      end if
      Solve for the $k$'th realization of $D_t$
        $\underset{\boldsymbol{x}^s_{t+1}, \overline{\alpha}_{t+1}}{\text{maximize}} \quad \boldsymbol{c}'_{t+1}\boldsymbol{x}^s_{t+1} + \overline{\alpha}_{t+1}$   (3.17a)
        subject to $\quad A\boldsymbol{x}^s_{t+1} \leq \boldsymbol{d}^k_{t+1} - F\hat{\boldsymbol{x}}^s_t,$   (3.17b)
        $\overline{\alpha}_{t+1} - (\boldsymbol{x}^s_{t+1})'F'\boldsymbol{\pi}^r_{t+2} \leq \beta^r_{t+2} \quad \forall r \in \{1, 2, ...R\},$   (3.17c)
        $\boldsymbol{x}_{t+1} \geq \mathbf{0}.$   (3.17d)
      Get cut coefficients for the $k$'th realization
      $\boldsymbol{\pi}^k =$ dual values of constraints (3.17b)
      $\beta^k = \boldsymbol{d}'^k_t \boldsymbol{\pi}^k$
    end for
    Calculate expected cut coefficients to include in stage $t$
    $\beta^u_t = \sum_{k \in \mathcal{K}} p^k \beta^k$
    $\boldsymbol{\pi}^u_t = \sum_{k \in \mathcal{K}} p^k \boldsymbol{\pi}^k$
    $u = u + 1$
  end for
end for
Update set index for cuts
$R = R + \hat{S}$
$U = \max_{s \in \hat{\mathcal{S}}} \boldsymbol{c}'_1 \boldsymbol{x}^s_1 + \alpha^s_1$
```

Note that $\hat{S}$ cuts are constructed for all but the last stage, stage T, in one run of the backwards recursion. Each stage shares cuts across scenarios. After constructing new sets of cuts in every stage, we can sample new scenarios by simulation to obtain new values for $\hat{\boldsymbol{x}}^s_t$, as described in Algorithm 2. Here, we use a Monte Carlo simulation to estimate expected income, as the mean of revenues of the $\hat{S}$ sampled scenarios.

Algorithm 2 Forward simulation

```
Sample $\hat{S}$ random scenarios, such that $\hat{\mathcal{S}} \subset \mathcal{S}$
for $s \in \hat{\mathcal{S}}$ do
  for $t \in \mathcal{T}$ do
    if $t = T$ then
      Replace constraints (3.17c) with $\alpha_T = 0$
    end if
    Solve
      $$\begin{aligned} \underset{\boldsymbol{x}_t^s, \overline{\alpha}_t}{\text{maximize}} \quad & \boldsymbol{c}_t' \boldsymbol{x}_t^s + \overline{\alpha}_t \\ \text{subject to} \quad & A\boldsymbol{x}_t^s \leq \boldsymbol{d}_t^s - F\hat{\boldsymbol{x}}_{t-1}^s, \\ & \overline{\alpha}_t - (\boldsymbol{x}_t^s)' F' \boldsymbol{\pi}_{t+1}^r \leq \beta_{t+1}^r \qquad \forall r \in \{1, 2, ...R\}, \\ & \boldsymbol{x}_t \geq \boldsymbol{0}. \end{aligned}$$
    Update optimal state guess, $\hat{\boldsymbol{x}}_t^s = \boldsymbol{x}_t^s$
  end for
end for
$\hat{L} = \frac{1}{\hat{S}} \sum_{s \in \hat{\mathcal{S}}} \sum_{t \in \mathcal{T}} \boldsymbol{c}_t' \boldsymbol{x}_t^s$
```

We now have an upper bound U and an estimated lower bound $\hat{L}$. Since we used a Monte Carlo simulation we have an uncertainty of the lower bound. Therefore, we determine a confidence interval for L, for example the 95% confidence interval

$$[\hat{L} - 1.96\sigma,\ \hat{L} + 1.96\sigma], \tag{3.18}$$

given that L is normal distributed. Here, σ is the standard deviation of the revenues of the simulated scenarios

$$\sigma = \sqrt{\frac{\sum\limits_{s \in \hat{\mathcal{S}}} \Big(\sum\limits_{t \in \mathcal{T}} \boldsymbol{c}_t' \boldsymbol{x}_t^s - \hat{L} \Big)^2}{S - 1}}.$$

By increasing the number of sample scenarios $\hat{S}$, we obtain greater accuracy, but increase computation time. By alternating between the two algorithms, we can improve our solution by adding more cuts. Hence, $R = 0$ initially, and increases after every backward recursion 1. As a convergence criterion, we can check when the upper bound U is inside the confidence interval (3.18). Note that since we simulate a random subset of all scenarios, the estimated lower bound $\hat{L}$ may exceed the upper bound U at random.

3.6 Modelling the environmental restrictions as linear constraints

Recall the restrictions (2.3) presented in Section 3.6,

$$
\begin{aligned}
&t_A = \min\{\ \min\{t \geq t_{15.\text{April}} : f_{it}(Z_{2,t}) > \overline{I}\},\ t_{15.\text{May}}\}, && \text{End of low inflow period}\\
&t_B = \min\{t \geq t_A : V_{2,t} \geq \underline{V}_2\}, && \text{Time when sufficient water level is reached}\\
&t_C : \text{Week containing 15. August}\\
&t_D : \text{Week containing 1. September}
\end{aligned}
$$

$$
\begin{aligned}
Q_{2,t} + B_{2,t} = 0 \ \ \forall t \in [t_A t_B) && (i),\\
V_{2,t} \geq \underline{V}_2 \ \ \forall t \in [t_B, t_C) && (ii),\\
Q_{2,t} + B_{2,t} \leq f_{it}(Z_t) + Q_{1,t} + B_{1,t} \ \ \forall t \in [t_C, t_D] && (iii).
\end{aligned}
$$

As mentioned, the HPS problem (2.4) is formulated as an LP problem. This is a key property for using the above presented solution method. In the solution method, cutting planes are shared among inflow scenarios in each stage, and scenarios may vary due to random sampling in each iteration. Thus, the cutting planes are represented by linear inequalities in $(V_{it}^s,\ Z_{it}^s)$ for all $i \in \mathcal{I}$. Therefore, constraints that are scenario dependent are not straightforward to implement. The solution method is considered necessary in order to deal with inflow uncertainty, which is prioritized. However, restrictions (i) and (ii) are not linear, thus they cannot be included in our model. Therefore, we need to introduce the constraints (2.3) in a way that maintains linearity.

If we consider t_A and t_B as discrete variables depending on the $Z_{2,t}$ and $V_{2,t}$ respectively, then we introduce integers and non-linearities to the problem. The activation time of (i), as represented by t_A, depends on the scenario, and the activation time of (ii), represented by t_B, depends on a decision variable. Thus, we do not accept this formulation. Note that, restriction (iii) is scenario independent, since t_C and t_D are equal for all scenarios. Therefore, (iii) is straightforward to implement, and we will not discuss it further.

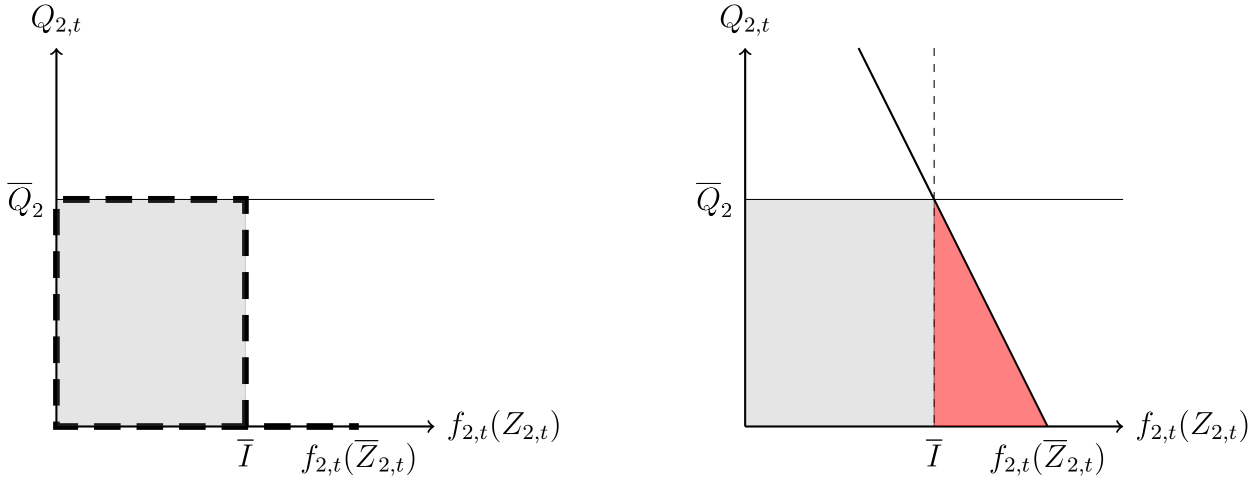
3.6.1 Activation of restriction (i)

The first restriction (i) is activated once we have sufficient inflow into Hamlagrøvatn. That is, if $f_{it}(Z_{2,t}) > \overline{I}$ in a certain time period, between 15. April and 15. May, or at 15. May if it is not already activated. Thus, we will introduce a constraint to prevent production and bypass in Hamlagrøvatn if inflow is greater than $\overline{I}$, for the period where the activation time is uncertain. The solution space is illustrated in Figure 3.2a. The desired solution space is represented by the dashed line in Figure 3.2a. As the desired solution space is not convex, we can't represent it using linear constraints. Since we need to formulate the problem as an LP problem, we will replace the solution space with its convex hull.

For further simplification of (i), we replace the set depicted in 3.2a by its convex hull. In order to do so we add the linear inequality in $Q_{2,t}$ and $Z_{2,t}$

$$f_{2,t}(Z_{2,t}) \leq f_{2,t}(\overline{Z}_{2,t}) - \frac{f_{2,t}(\overline{Z}_{2,t}) - \overline{I}}{\overline{Q}_2} Q_{2,t}, \tag{3.20}$$

where $\overline{Q}_2$ is the upper bound on production in Hamlagrøvatn, and $\overline{Z}_{2,t}$ is the maximum possible simulated inflow in stage t. Thus, we approximate restriction (i) as shown in Figure 3.2b, for all $t \in [t_{15.\text{ April}}, t_{15.\text{ May}})$. The reasoning is equivalent for $B_{2,t}$ and $Z_{2,t}$.



(a) *Desired solution space* ***(b)*** *Convex hull of desired solution space*

Figure 3.2: *Solution space for environmental constraint* (i)

Note that, since we expanded the solution space, we might obtain a solution which is infeasible with respect to (i). Additionally, if inflow decreases after reaching $\overline{I}$ the constraint will not be active, which does not represent the environmental constraint well. However, due to the inflow trend, that is inflow is usually increasing in this period, we consider this a reasonable simplification. Moreover, since the period where the activation time of (i) is uncertain is relatively short, we expect the consequence of the simplification to be small.

When we reach 15. May, production and bypass will be ceased in all scenarios. Therefore, we can include (i) as a constraint for all $t \in [t_{15.\text{May}}, t_B]$. However, this requires determining t_B, which is our next challenge.

3.6.2 Activation of restriction (ii)

Restriction (ii) is a bit more challenging. The period where activation time is uncertain is significantly longer than for (i), and a simplification analogously to the previous one does not make sense. Therefore, we consider only scenario independent values of t_B. That is, we assume that t_B has the same value in all scenarios. Then, we will try to solve the problem by complete enumeration of the possible values of t_B. Note that, a scenario dependent t_B can give better results, and is a more correct representation of the restriction. However, for a scenario dependent values of t_B, the equivalent enumeration would be unreasonably large, with n^S possible combinations, where n is the number of possible values for t_B in one scenario, and S is the number of scenarios. Moreover, as the producer determines reservoir volume, and t_B depends on the reservoir volume, the producer also somewhat determines t_B. Thus, we consider it a reasonable approximation.

In order to find the optimal scenario independent value for t_B, we run the optimization for every possible value $t_B \in \{t_{15.\text{April}}, t_{15.\text{April}} + 1, ..., t_C\}$ and compare the results. This strategy is a way to estimate the optimal deactivation time of (i) and activation time of (ii). Note that, if the planning period spans multiple summers, then the enumeration increases considerably, as we need to include a t_B variable for each summer.

Alternatively, we could consider the historical water level of Hamlagrøvatn, in order to determine a fixed value of t_B. In Figure 3.3 we see recorded water levels for years 2008 through 2018, and predictions for 2019 and 2020. The horisontal axis indicate week of the year and the vertical axis represents altitude in m.a.s.l. The figure indicates that the mean (dashed line) reaches the water level of 584 m.a.s.l. in week 26, where almost all records reach the water level between week 21 and 35. This means, $\underline{V}_2$ is reached on average in week 26. The ending time for constraint (ii) t_C is known, and scenario independent.

Since we solve each stage separately, providing starting conditions for the next stage, we might provide starting conditions where constraint (ii) renders the next stage infeasible. Therefore, we must include (ii) as a soft constraint which we are allowed to break, at some penalty cost. By increasing the penalty cost, we can ”prioritize” the necessity of maintaining the constraint.

3.6.3 Approximated environmental constraints

In order to include the approximated constraints as mentioned above, we propose the following formulation. We define new activation times in order to reflect the approximation of the environmental constraints. Note that, since t_C and t_D are fixed values, and equal for all scenarios, they are easily included. Let,

$$\begin{aligned} t_a = t_{15.\text{April}} &: \text{Starting time for when constraint } (i) \text{ might activate} \\ t_b = t_{15.\text{May}} &: \text{Definitive activation time for constraint } (i) \\ t_c = \{t_0, t_0+1, ...t_3\}, &: \text{Activation time for constraint } (ii) \\ t_d = t_C &: \text{Activation time for constraint } (iii) \\ t_e = t_D &: \text{Deactivation time for constraint } (iii) \end{aligned}$$

The slope a_Q and constant b_Q of the constraint defining the convex hull illustrated by Figure 3.2b is calculated from the two points $(\overline{I}, \overline{Q}_2)$, $(\overline{Z}_{2,t}, 0)$. Assume that we know $\overline{Z}_{2,t}$, which is the maximum possible value of simulated inflow into Hamlagrøvatn in stage t. Equivalently, we can obtain a_B and b_B for the constraint on bypass and inflow.

$$\begin{aligned} Q^s_{2,t} - a_Q f_{2,t}(Z^s_{2,t}) - b_Q \leq 0, && \forall (s,t) \in \mathcal{S} \times [t_a, t_b), && (3.21) \\ B^s_{2,t} - a_B f_{2,t}(Z^s_{2,t}) - b_B \leq 0, && \forall (s,t) \in \mathcal{S} \times [t_a, t_b), && (3.22) \\ Q^s_{2,t} + B^s_{2,t} = 0, && \forall (s,t) \in \mathcal{S} \times [t_a, t_b), && (3.23) \\ \gamma^s_t + V^s_{2,t} \geq \underline{V}_2 && \forall (s,t) \in \mathcal{S} \times [t_a, t_b), && (3.24) \\ \gamma^s_t \geq 0 && \forall (s,t) \in \mathcal{S} \times \mathcal{T}, && (3.25) \\ Q^s_{2,t} + B^s_{2,t} - Q^s_{1,t} - B^s_{1,t} \leq f_{2,t}(Z^s_{2,t}) && \forall (s,t) \in \mathcal{S} \times [t_a, t_b). && (3.26) \end{aligned}$$

Constraints (3.21)-(3.23) are the approximation of (i), constraints (3.24)-(3.25) are the approximation of (ii) and constraint (3.26) is (iii). The variable γ_t is the penalty variable of breaking constraint (3.24). This variable is included in the objective function in problem (4.6) with a corresponding cost c_γ. Note that t_c is allowed to be earlier than t_b, as early as t_a. Activation of (ii) always implies deactivation of (i), meaning if (ii) is activated before (i), then constraint (i) is skipped entirely. Also if $t_c = t_d$, then constraint (ii) will never be activated.

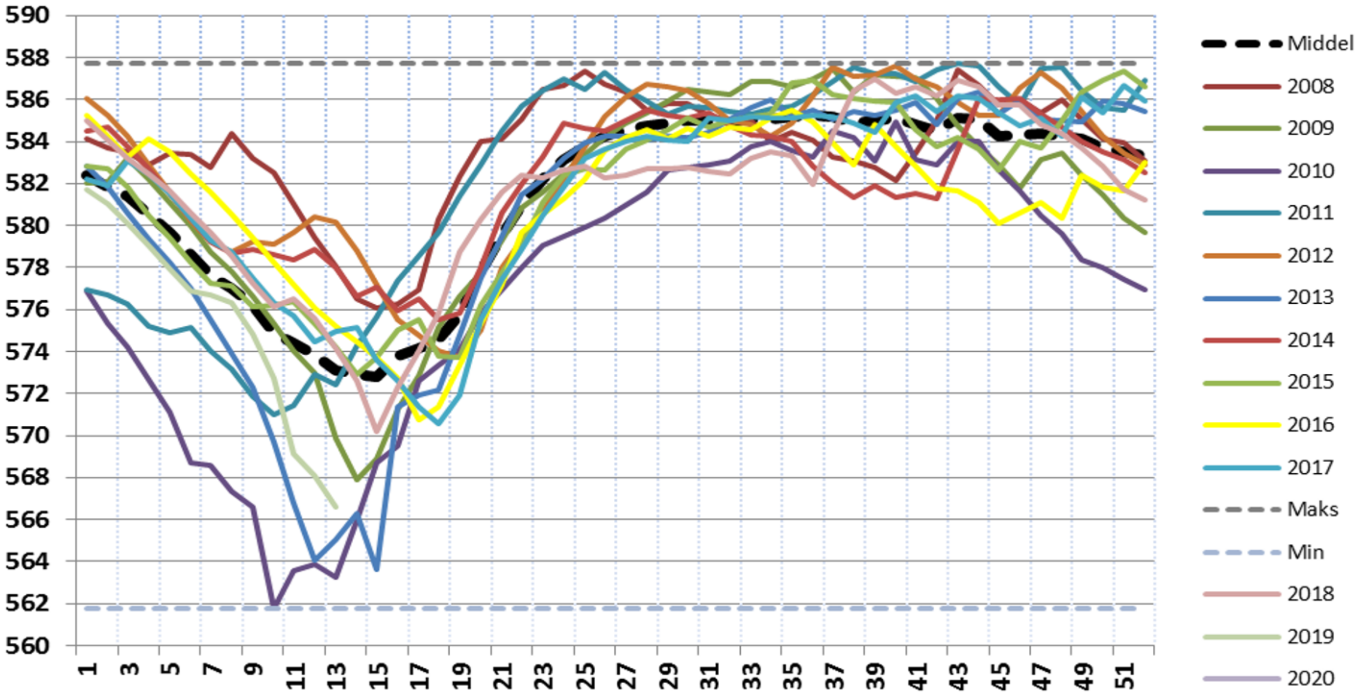


Figure 3.3: *Historical and predicted water level of Hamlagrøvatn (Haga, 2019)*

Chapter 4

Optimizing the Bergsdalen watercourse

In this chapter we present a version of the HPS problem (2.4), as presented in Chapter 2, applied to the Bergsdalen watercourse. The inflow model is also thoroughly described. Figure 4.1 illustrates the watercourse, and Figure 4.2 illustrates its module representation. In addition to the main HPS problem, we formulate the subproblems to be solved in the solution method, described in Chapter 3, in order to solve the main problem.

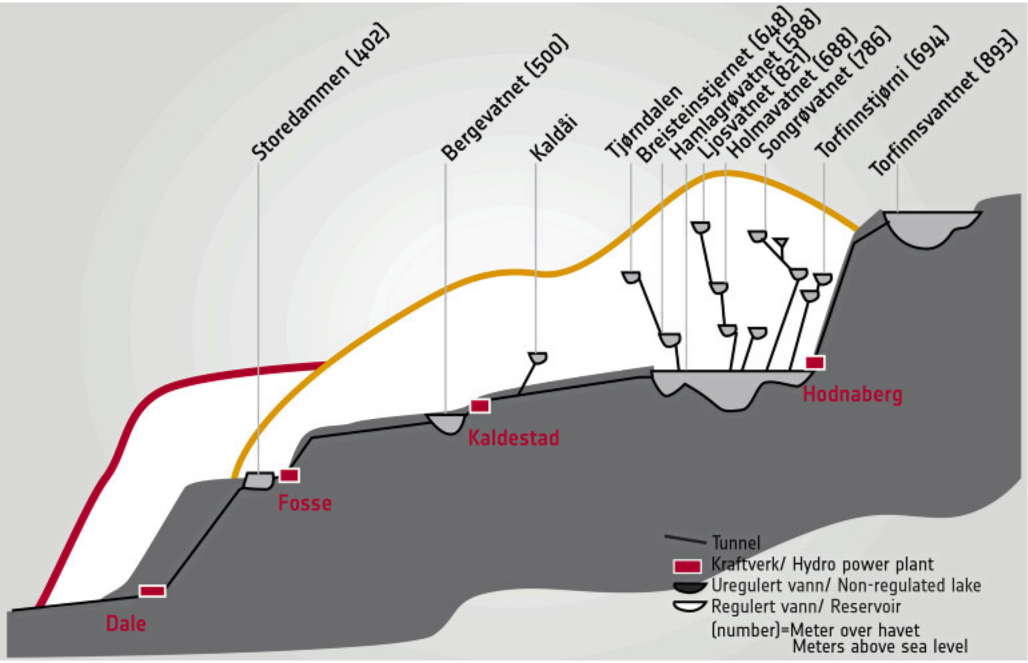


Figure 4.1: The Bergsdalen watercourse (bkk.no, 2019)

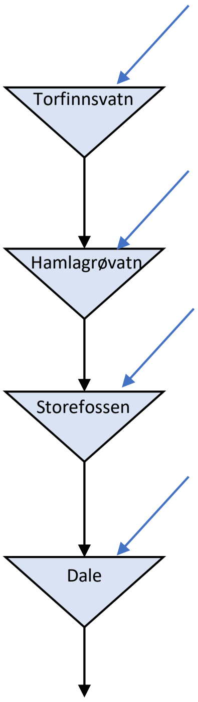


Figure 4.2: *Module representation of the Bergsdalen watercourse*

4.1 Inflow model

As described in Section 2.2.2, we require a model for generating inflow series. We base the model on time series containing recorded historical inflow corresponding each reservoir. The inflow modelling procedure is roughly the same as described by Gjelsvik et al. (2010).

For each hydropower module we have a time series of recorded inflow from 1958 until 2015, q_{jyw}. Thus, we have 58 recorded inflow years for each reservoir. The recorded inflow series are not from the Bergsdalen watercourse, but from lakes with similar characteristics, and need to be scaled. We denote the set of lakes with recorded inflow as $\mathcal{J}$, such that each lake corresponds to a module in the Bergsdalen system, that is $\mathcal{J} \mapsto \mathcal{I}$. Each module has an average yearly inflow volume $q_{i,mean}$, for a reference period of 30 years, between 1981 and 2010, which we use for scaling the recorded inflow series to the correct size. In order to scale the inflow series, we consider the scaled mean

$$q_{i,mean} = \frac{1}{30} \sum_{y=1981}^{2010} \sum_{w=1}^{52} k_i \, q_{jyw} \quad \forall (i,j) \in \{\mathcal{I} \times \mathcal{J} \; : \; i = j\} \tag{4.1}$$

of yearly inflow during the reference period. Then

$$k_i = \frac{30 \, q_{i,mean}}{\sum\limits_{y=1981}^{2010} \sum\limits_{w=1}^{52} q_{jyw}}$$

is the scaling coefficient for series i, found by solving equation (4.1) for k_i. Thus, the inflow values for the modules in the Bergsdalen system are

$$q_{iyw} = k_i \; q_{jyw} \quad \forall (i, js) \in \{\mathcal{I} \times \mathcal{J} \; : \; i = j\}$$

for each year y and week w for all 58 recorded inflow years.s

Some part of the inflow series may be explained purely by seasonal variations. Therefore, before fitting the auto regressive model described in Section 2.2.2, we propose a transformation to remove seasonal variations from the inflow modelling. First, we find the weekly mean over all inflow years $\overline{q}_{iw}$, and the corresponding standard deviation σ_{iw}

$$\overline{q}_{iw} = \frac{1}{58} \sum_{y=1958}^{2015} q_{iyw} \qquad\qquad \sigma_{iw} = \sqrt{\frac{\sum\limits_{y=1958}^{2015} (q_{iyw} - \overline{q}_{iw})^2}{58 - 1}}.$$

Then, the transformed recorded inflow series Z_{iyw} is

$$Z_{iyw} = \frac{q_{iyw} - \overline{q}_{iw}}{\sigma_{iw}}.$$

By doing the same for all 4 time series we get the back-transformation

$$f_{it}(Z_{it}^s) = \sigma_{iw} Z_{it}^s - \overline{q}_{iw} \qquad\qquad , w = t \;\; mod \;\; 52, \tag{4.2}$$

to be used on simulated inflow values Z_{it}^s. Since the planning period may be longer than a year, we should index σ and $\overline{q}$ with week of the year, not stage number in the planning period. Therefore, the w subscript is the modulus of t and amount of weeks in the year, 52. We let mod denote the modulus operation, such that $a \;\; mod \;\; b$ is the remainder of the integer division a/b. The simulation is described later in this section.

Now, instead of considering the transformed recorded inflow series as years and weeks, Z_{iyw}, consider them as 4 series with all recorded values consecutively from start to finish, $Z_{it} \;\; , \; i \in \mathcal{I}, \; t \in \{1, 2, ...58 \cdot 52\}$. For fitting the model we use a first order vector auto regression. In essence, we find the cross correlation matrix ϕ_{ij} by minimizing the sum of squared prediction errors

$$\min_{\phi_{ij}} \quad \sum_{t=2}^{58 \cdot 52} \sum_{i \in \mathcal{I}} (Z_{it} - \sum_{j \in \mathcal{I}} \phi_{ij} Z_{j,t-1})^2.$$

Now we can simulate inflow series with

$$Z_{it} = \sum_{j \in \mathcal{I}} \phi_{ij} Z_{j,t-1} + \epsilon_i, \tag{4.3}$$

where ϵ_i is a modeling error.

Let $\boldsymbol{\epsilon} = [\epsilon_1, ..., \epsilon_4]$ be a vector of errors made for all reservoirs for one stage. We assume that in each week of the year, the model makes a similar modeling error. Then, for each week of the year, we get 58 error vectors. For each week we perform a principal component analysis (PCA), in order to find the 4 eigenvectors of errors in each week. Note $\boldsymbol{\epsilon} \in \mathbb{R}^4$ in this case. For each week we choose the first principal compontent, i.e. the eigenvector explaining the most variance of errors. Then, we multiply the eigenvector with the amount of variance it explains, to construct $\hat{\boldsymbol{\epsilon}}_w$. Thus, we have values for the most common mistake made in week w. The values of $\hat{\boldsymbol{\epsilon}}_w$ will be used for ϵ_{it} in equation (4.3).

When simulating transformed inflow, we start with a known present inflow value Z_{i0}. Then, we use equation (4.3) to obtain Z_{i1} etc. As a random error term we use the discrete random vector $\mathcal{E}_w$, which can take values from $\{\hat{\boldsymbol{\epsilon}}_w, \mathbf{0}, -\hat{\boldsymbol{\epsilon}}_w\}$. Its probability mass function is

$$f_{\mathcal{E}_w}(\boldsymbol{\epsilon}) = \begin{cases} 0.2, & \boldsymbol{\epsilon} = \hat{\boldsymbol{\epsilon}}_w \\ 0.6, & \boldsymbol{\epsilon} = \mathbf{0} \\ 0.2, & \boldsymbol{\epsilon} = -\hat{\boldsymbol{\epsilon}}_w. \end{cases}$$

We assume that we are most likely not to make an error, while there is a possibility to make an error along either direction of the first principal component of errors.

Here, $\hat{\boldsymbol{\epsilon}}_w$ is the first principal component of errors in week w multiplied with its explained variance, and $\mathbf{0}$ is the zero vector. The outcomes of $\mathcal{E}_w$ represent the three possible *inflow realizations* between every stage. If $\hat{\boldsymbol{\epsilon}}_w$ is too large, we risk simulating negative inflow. Therefore a coefficient g_t is chosen for all $t \in \mathcal{T}$ such that inflow will not be negative. Some weeks have high inflow due to seasonality, making it possible to simulate larger variation, while avoiding negative inflow. Thus, g_t is set larger when possible, to increase variation in inflow simulation. Variance in inflow simulation and avoiding negative inflow requires tuning the inflow simulation error coefficients g_t, to achieve an acceptable balance. Now, every week we get the inflow value

$$Z_{it} = \sum_{j \in \mathcal{I}} \phi_{ij} Z_{i,t-1} + \ \epsilon_{it} \tag{4.4}$$

$$\epsilon_{it} = g_t \cdot \text{random sample of } \mathcal{E}_w \qquad , w = t \ mod \ 52 \tag{4.5}$$

The PCA and scaling of the error term reduces the number of possible inflow scenarios, resulting in a loss of the actual inflow uncertainty representation. To avoid this, we can use more principal components and modify the transformation f_{it} to avoid negative inflow. However, we accept the loss of inflow scenarios in order to maintain a relatively simple model.

Figure 4.3 shows an example of 50 simulated inflow series for all 4 reservoirs of the Bergsdalen watercourse over one year, in blue. The red lines are simulations where the error term is either always positive, always negative or always zero.

4.2 End water values

The end water volume may affect the entire production schedule. If the end water value is not specified, we will obtain an optimal schedule that leads to an unreasonably low end volume, and vica versa if the end water value is too high. This leads to a schedule that spends more water than what is realistic, which may in affect the beginning of the production schedule. If the beginning of the schedule is sensitive to the end volume levels, we risk making wrong decisions based on the optimized but unrealistic ”bold” schedule. As the beginning of the schedule tends to be less sensitive to end water volume when the planning period is long, it is common practice to set the planning period longer than what is absolutely necessary. However, since we do not have an adequate measure of the value of storing water at the end of the planning period, we instead introduce a bound on end reservoir volumes.

Let $\underline{V}_{iT}$ be a lower bound for all i end reservoir volumes. Spending water increases the objective function when there is no value specified for saving water. Therefore, the optimal solution will seek a low end reservoir volume, thus a lower bound is sufficient. In order to determine values for $\underline{V}_{iT}$ we can run the optimization with a longer planning period, to determine a reasonable lower bound in stage $t = 52$. We do not go into detail about the determination of end volume bounds. However, since the enumeration of activation times for environmental restriction (ii) t_c increases significantly when the planning period spans several years, the proposed method to deal with restriction (ii) does not work well with a long planning period. Due to the same problem with suddenly implementing a volume restriction as mentioned with constraint (ii) in Section 3.6, we implement the end reservoir volume restrictions as soft constraints.

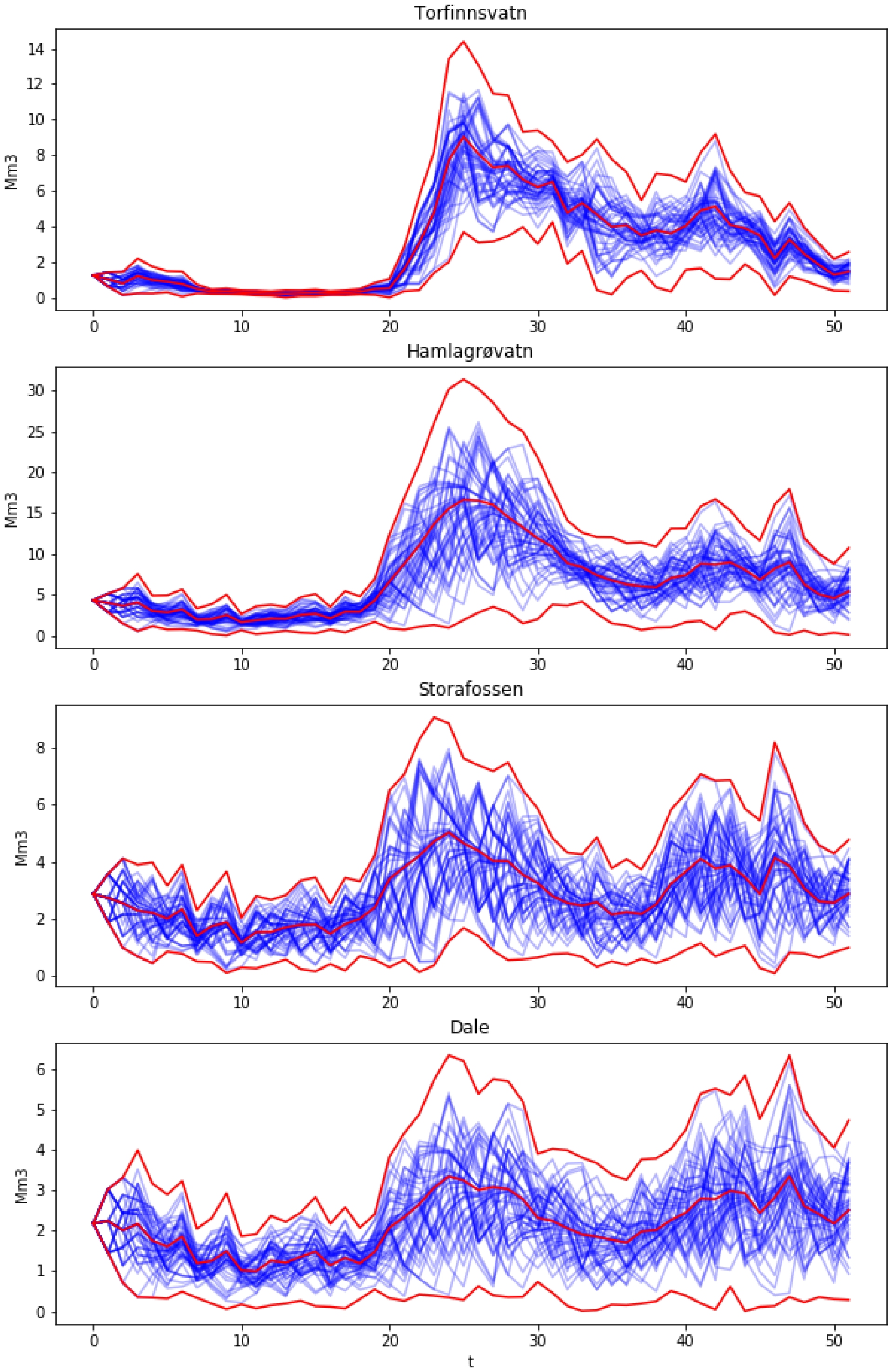


Figure 4.3: *Simulated inflow for all reservoirs*

4.3 The Bergsdalen HPS problem

Recall the definition of problem (2.4) from Chapter 2. We add the following definitions as presented in this chapter and in Section 3.6.

Parameters

$\underline{V}_{iT}$: End reservoir volume bound for reservoir i.
ϵ^s_{it} : Inflow modelling error for reservoir i in stage t and scenario s.
ϕ_{ij} : Inflow transition matrix.
c_γ : Cost of breaking soft constraints.
π^r_{itV} : Expected optimal dual value of the waterbalance constraint (4.7d).
π^r_{itZ} : Expected optimal dual value of the inflow transition constraint (4.7c).
π^r_{itV} : Expected optimal dual value of waterbalance constraint (**??**).
β^r_{t+1} : Expected right hand side of cut.

Variables

γ^s_t : Penalty variable for breaking soft constraints.

We now formulate an extention of this problem, to be solved in order to determine an optimal production schedule for the Bergsdalen watercourse.

$$\text{maximize} \quad \sum_{s\in\mathcal{S}} \mathbb{P}(s) \sum_{t=1}^{T} \left(p_t y^s_t - c_\gamma \gamma^s_t\right) \tag{4.6a}$$

subject to

$$\text{Non-anticipativity constraints ,} \tag{4.6b}$$

$$\text{Environmental constraints (3.21) - (3.26) ,} \tag{4.6c}$$

$$V^s_{it} + Q^s_{it} + B^s_{it} + O^s_{it} - V^s_{i,t-1} - \sum_{j\in\mathcal{M}_i} \left(Q^s_{jt} + B^s_{jt}\right) = f_{it}(Z^s_{it}), \quad \forall (i,s,t) \in \mathcal{I}\times\mathcal{S}\times\mathcal{T}, \tag{4.6d}$$

$$\sum_{i\in\mathcal{I}} \eta_i Q^s_{it} - y^s_t = 0 \quad , \quad \forall (s,t) \in \mathcal{S}\times\mathcal{T}, \tag{4.6e}$$

$$\gamma^s_T + V^s_{iT} \geq \underline{V}_{iT} \quad , \quad \forall (i,s) \in \mathcal{I}\times\mathcal{S}, \tag{4.6f}$$

$$\overline{V}_i \geq V^s_{it} \geq 0 \quad , \quad \forall (i,s,t) \in \mathcal{I}\times\mathcal{S}\times\mathcal{T}, \tag{4.6g}$$

$$\overline{Q}_i \geq Q^s_{it} \geq 0 \quad , \quad \forall (i,s,t) \in \mathcal{I}\times\mathcal{S}\times\mathcal{T}, \tag{4.6h}$$

$$\overline{B}_i \geq B^s_{it} \geq 0 \quad , \quad \forall (i,s,t) \in \mathcal{I}\times\mathcal{S}\times\mathcal{T}, \tag{4.6i}$$

$$O^s_{it} \geq 0 \quad , \quad \forall (i,s,t) \in \mathcal{I}\times\mathcal{S}\times\mathcal{T}, \tag{4.6j}$$

$$y^s_t \geq 0 \quad , \quad \forall (s,t) \in \mathcal{S}\times\mathcal{T}, \tag{4.6k}$$

$$\gamma^s_t \geq 0 \quad , \quad \forall (s,t) \in \mathcal{S}\times\mathcal{T}. \tag{4.6l}$$

Problem (4.6) is an extension of the HPS problem (2.4), where we have included environmental restrictions and end volume constraints. The non-anticipativity constraints (4.6b) are as described in Section 2.2.5. The constraints (4.6c) are the approximated environmental constraints given in Section 3.6.3. We have introduced the penalty variables $\gamma_t^s \ \forall (s,t) \in \mathcal{S} \times \mathcal{T}$, in order to handle the soft constraints in the environmental constraints (4.6c) and the end reservoir volume constraints (4.6f), as presented in Section 4.2.

4.3.1 Subproblems

The HPS problem (4.6) will be solved by splitting it into subproblems (4.7), as described in Section 3.4. The subproblems will be solved iteratively to construct cuts, and improve the solution, as described in Section 3.5. The backward recursion algorithm (1) constructs cuts, given a guessed solution, and the forward simulation algorithm (2) improves the solution when new cuts are made. The subproblems to be solved by the algorithm are formulated as,

$$
\begin{aligned}
\text{maximize} \quad & p_t y_t^s + \alpha_t^s - c_\gamma \gamma_t && && \text{(4.7a)}\\
\text{subject to} \quad & && && \\
& \text{Environmental constraints (3.21) - (3.26)} && , && \text{(4.7b)}\\
& Z_{it}^s = \sum_{j \in \mathcal{J}} \phi_{ij} Z_{j,t-1}^s + \epsilon_{it}^s \forall i \in \mathcal{J}, && && \text{(4.7c)}\\
& V_{it}^s + Q_{it}^s + B_{it}^s + O_{it}^s - \sum_{j \in \mathcal{M}_i} \left(Q_{jt}^s + B_{jt}^s\right) - f_{it}(Z_{it}^s) = V_{i,t-1}^s && \forall i \in \mathcal{J}, && \text{(4.7d)}\\
& \sum_{i \in \mathcal{J}} \eta_i Q_{it}^s - y_t^s = 0, && && \text{(4.7e)}\\
& \alpha_t^s - \sum_{i \in \mathcal{J}} \left(\pi_{i,t+1,V}^r V_{it} + \pi_{i,t+1,Z}^r \ \phi_{ij} Z_{it}\right) \leq \beta_{t+1}^r && \forall r \in \{1, 2, ...R\}, && \text{(4.7f)}\\
& \gamma_t^s + V_{it}^s \geq \underline{V}_{it} && \forall i \in \mathcal{J} \ \text{ if } t = T, && \text{(4.7g)}\\
& \overline{V}_i \geq V_{it}^s \geq 0 && \forall i \in \mathcal{J}, && \text{(4.7h)}\\
& \overline{Q}_i \geq Q_{it}^s \geq 0 && \forall i \in \mathcal{J}, && \text{(4.7i)}\\
& \overline{B}_i \geq B_{it}^s \geq 0 && \forall i \in \mathcal{J}, && \text{(4.7j)}\\
& O_{it}^s \geq 0 && \forall i \in \mathcal{J}, && \text{(4.7k)}\\
& y_t^s \geq 0, && && \text{(4.7l)}\\
& \gamma_t^s \geq 0, && && \text{(4.7m)}
\end{aligned}
$$

for each $(s,t) \in \mathcal{S} \times \mathcal{T}$. The constraint (4.7c) represents the inflow model, from equation (4.4). Constraints (4.7f) represents the expected Benders cuts, as presented in equation (3.16). The values π^r_{itV} and π^r_{itZ} are the expected optimal dual values of constraint (4.7d) and (4.7c), calculated iteratively as described in Section 3.5.

Now every subproblem provides starting conditions for inflow and reservoir volumes, to the subsequent subproblem, similar to $\boldsymbol{x}_t$ in Chapter 3. The inflow modelling error ϵ_{it} is the outcome of the random parameter $\boldsymbol{\mathcal{E}}_w$, corresponding to $\boldsymbol{D}_t$ in Chapter 3.

4.3.2 State variables

Recall the decision variables x^s_t from Chapter 3. Every decision made in stage $t-1$ decided the state in stage t. Similarly for this problem, due to the water balance constraint (4.7d) and the inflow transition in equation (4.4), we have the same coupling of stages through states. However, the inflow is not a real decision, but we can interpret it as a state variable in order to construct cuts that are used for all states in each stage. Thus in an arbitrary scenario s, both the decision on reservoir volumes $V^s_{i,t-1}$ and inflow realizations $Z^s_{i,t-1}$ for all $i \in \mathcal{I}$ in the subproblem previous stage $t-1$ determines the state of the subproblem in the next stage t.

As we have seen, the inflow states $Z_{i,t-1}$ $\forall i \in \mathcal{I}$ helps determine the inflow state Z_{it}. Now, in every subproblem to be solved in the algorithm, we can create Benders cuts that are shared among inflow scenarios, in addition to reservoir volume states. Hence, we interpret both inflow scenario and reservoir volume as state. Since all cuts are shared within each stage, the subproblems optimize with respect to all future scenarios. Additionally, scenarios with the same history will have the same state values from the previous stage, and make the same decisions. Hence, the non-anticipativity constraints are respected implicitly.

4.4 Main algorithm

Recall the algorithms (1) and (2) presented in Section 3.5. Here we compute the upper limit U and the estimated lower limit $\hat{L}$ of the objective function in Problem (4.6), with the standard deviation of simulated lower limits σ. To initialize the algorithm, that is obtain a start guess of the optimal solution, we propose to perform a forward simulation where expected future income α_t^s is always fixed as zero. Then, we can write the SDDP algorithm to solve the Bergsdalen HPS problem (4.6) as,

Algorithm 3 SDDP algorithm

```
Initialize by solving all subproblems (4.7) with $\alpha_t^s = 0$
Initialize convergence criteria: Error $= 1$, Tolerance $= 0$
while Error $>$ Tolerance do
    Perform the backward recursion algorithm (1) on subproblems (4.7)
    Perform the forward simulation algorithm (2) on subproblems (4.7)
    Update $\sigma$

    Error $= |U - \hat{L}|$
    Tolerance $= 1.96\sigma$
end while
```

In each call of the forward simulation (2), we sample a new inflow scenario and update our guess of an optimal production schedule. Then in the following backward recursion (1) we construct cuts using the recently sampled scenario and production schedule.

Chapter 5

Experiments

5.1 Goal

The goal of the experiments is to find an optimal production schedule for the Bergsdalen watercourse, and determine how well we are able to represent the environmental restrictions presented in Section 3.6. We will run the optimization for all possible values of t_c, in order to determine the optimal scenario independent time for activation constraint (3.24). Then, we will discuss the estimated optimal schedule.

5.2 Setup

The optimization problem described in Section 4.3 is solved by implementing Algorithm 3. The implementation is done in Python. The Pyomo library (Hart et al. (2011) and Hart et al. (2017)) is used for modelling the subproblems (4.7) and interfacing with a chosen LP solver. We chose the Gurobi direct solver (Gurobi Optimization, 2018), utilizing warm starts when possible. As the subproblems do not change drastically between iterations, warm starts help performance significantly. As multiple scenarios are solved separately, the algorithm would benefit from implementing parallel processing, as shown by Helseth and Braaten (2015). However, this is not utilized in this implementation.

To construct the inflow model described in Section 4.1 we used the vector auto regression method from the Python library StatsModels (Seabold and Perktold, 2010). The principal component analysis is done with the PCA method from the machine learning library Scikit-Learn (Pedregosa et al., 2011).

We have scaled the input data, in order to keep numerical values at a reasonable size relative to each other. Bad scaling may crash the solver. Water volumes are considered in $10^6 m^3$, and power price in $\frac{\text{EUR}}{Gwh}$. Thus the energy conversion factor η_i is $\frac{Gwh}{10^6 m^3}$. As most values are considered in 10^6, we let M denote 10^6. Keep in mind that we let M$m^3 = 10^6 m^3$, as we do not refer to the SI prefix *mega*. Additionally, as Python is zero-indexed, the stages are indexed from 0 to 51 in the plots presented, and not from 1 to 52 we have presented the stage indexes previously.

Preliminary tests

Some preliminary tests were completed, where we observed many of the cuts in each stage were similar. As a result, the data passed to the solver each time a subproblem was solved grew unnecessarily large, and computation time increased. Therefore, we propose a heuristic to choose fewer but ”more descriptive” cuts in Section 5.2.1.

We noticed the convergence was sensitive to the choice of penalty cost c_γ. If we set the penalty too low, some scenarios might be ”sacrificed” in order to have a ”bolder” schedule in other scenarios, and increase the expected revenues. Also, if the penalty is set too high we can obtain a large standard deviation from the forward simulation, when some scenarios are penalized heavily and others are not. Recall the convergence criterion in Algorithm 3, stating that the algorithm should stop when $|U - \hat{L}| < 1.96\sigma$, where U is upper bound, $\hat{L}$ is estimated lower bound and σ is the standard deviation from the forward simulation. If σ is large, we will accept a large error $|U - \hat{L}|$, thus the algorithm may converge prematurely.

As the soft constraints are not intended to be allowed to break, we choose a high penalty cost $c_\gamma = 100$ M EUR. In order to prevent premature convergence, an additional convergence criterion was defined, that the half-size of the convergence interval is less than 1 M EUR. The additional convergence criterion was added to Algorithm (3).

5.2.1 Cut selection heuristic

In an effort to keep computation times low, while efficiently constructing expected future income functions, we propose the following heuristic for only calculating the ”most descriptive” cuts. In general, we want to create cutting planes that cut away as much of the solution space as possible, and are the least similar to each other. Cuts with roughly the same coefficients are considered similar to each other. By including the cuts that are the least similar, we can more efficiently represent the future income function. If the future income function representation is not efficiently created, the algorithm converges slowly. We assume that subproblems with similar optimal objective function value creates similar cuts.

Consider an arbitrary iteration of the algorithm. Recall the backward recursion algorithm, Algorithm 1, from Section 3.5. For each stage $t \in \{T-1, T-2, ...1\}$ we construct $\hat{S}$ cuts from stage $t+1$. Let g_t^s be the optimal objective function value in stage t and scenario s. Assume without loss of generality that

$$g_t^1 \leq g_t^2 \leq ...g_t^{\hat{S}}.$$

That is, the scenario indices $s \in \{1, 2, ...\hat{S}\}$ are sorted with non-decreasing value of g_t^s. Then, choose the scenarios $\{1, \frac{\hat{S}}{3}, \frac{2\hat{S}}{3}, \hat{S}\}$, for calculating cuts for the previous stage. Fractions are rounded to the closest integer. In addition, we choose 4 random subproblems for calculating cuts in each stage. Now every iteration adds 8 cutting planes to the expected future income.

5.2.2 Dataset

The input data for this optimization model is provided by BKK. We have received data for price, inflow and system parameters. The optimization starts in week 48 and planning period is set to one year . Thus $t = 0$ indicates the beginning of week 48 etc.

The price data consists of different price scenarios, which we assume are all representable for our planning period, with probabilities of jumping between scenarios. As we do not model price uncertainty in our model, we choose one scenario as a deterministic price series for the planning period ($p_t\ \forall t \in \mathcal{T}$ in problem (2.4)). The inflow data consists of observed inflow for similar reservoirs to those in the Bergsdalen watercourse, from 1958 to 2015. In Section 4.1 we described how we generate the inflow simulation model used in the optimization model. The system parameters determine maximum values for reservoir volume, production and bypass ($\overline{V}_i, \overline{Q}_i, \overline{B}_i\ \forall i \in \mathcal{I}$ in problem (2.4)) and a turbine efficiency factor for each reservoir ($\eta_i\ \forall i \in \mathcal{I}$ in problem (2.4)).

Choice of price series

Figure 5.1 shows the quartile distribution of price values in each stage for all the price periods in our dataset. We used this to sample a price scenario that somewhat reflects the typical price profile during the period. Figure 5.4 shows the price scenario we chose for the optimization.

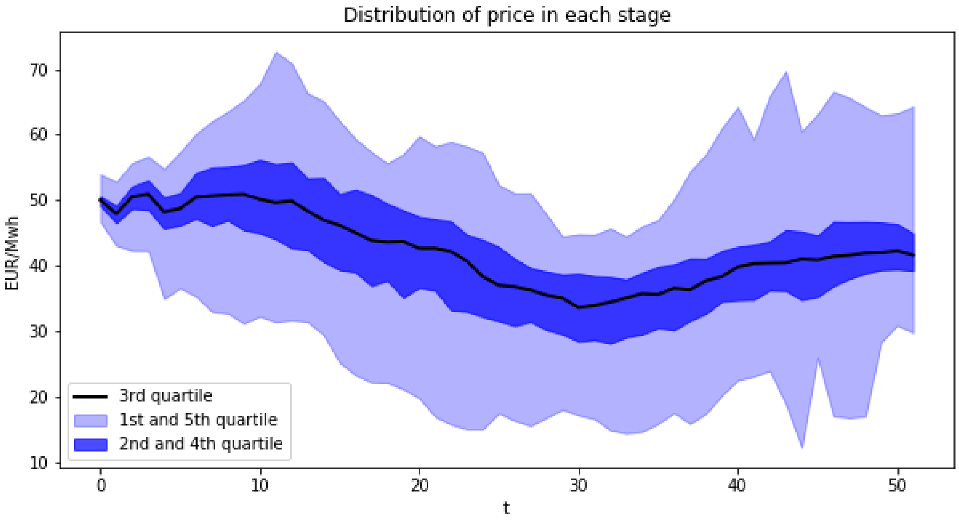


Figure 5.1: *The distribution of price possibilities in each stage*

5.3 Results of experiments

In this section, we present the results of running Algorithm 3 for all possible values of t_c, in order to obtain an estimated optimal solution to Problem (4.6). Recall the definition of the approximated environmental constraints (3.21) - (3.26) from Section 3.6. t_c represent the activation time for the environmental constraint (3.24), which we consider to be scenario independent. Lastly, we evaluate the estimated optimal schedule, and discuss the effect of the approximated environmental constraints.

5.3.1 t_c analysis

We ran Algorithm 3 for all 18 possible values of t_c. In each forward simulation we simulated 60 inflow scenarios. The calculations finished in approximately 10 hours. Note that, comparing the upper and lower bound to expected revenues is only valid when penalties are zero, because the penalty variable does not represent an actual loss in revenues. We will refer to the mean of total revenues over scenarios, from the forward simulation Algorithm 2, as estimated lower bound.

Figure 5.2 shows the results of the analysis. In the upper plot we have plotted the upper and estimated lower bound of expected revenues, for all values of t_c. We know with 95% certainty that the true lower bound is not below the grey dashed line. The grey dashed line is the estimated lower bound minus its confidence interval half-size. The lower plot shows the sum of penalties (γ_t^s) and overflow, summed over time and scenarios, for each value of t_c. We see that the penalties are insignificantly small (note the axis scale), however not zero for all t_c. The cases with the highest values for overflow are $t_c \in \{33, 35, 36, 37\}$. The cases with the highest upper bound are $t_c \in \{29, 30, 33\}$. Note that in $t = 27$ the estimated lower bound is larger than the upper bound. As mentioned in Section 3.5 this may happen, due to sampling variation. Moreover, we observe that the 95% confidence of the lower bound is always below the upper bound.

Determining the estimated optimal t_c

Due to the uncertainty in the estimated lower bound, we can't be 95% certain of improvements of less than 2 M EUR, as the confidence interval half-size of the lower bound is roughly 1 M EUR for all t_c. Additionally, we observe that the bounds are not monotonously increasing. This might be due to the convergence criterion, where the tolerance 1.96σ for the error $|U - \hat{L}|$ depends on the standard deviation σ of the forward simulations. The standard deviation is most likely different for each forward simulation, as we have a large sample space ($K^T = 3^{52}$ as shown in Section 3.1) for inflow scenarios. Thus, determining an optimal scenario independent value of t_c is difficult when comparing schedules that are not significantly different. However, we will choose the estimated optimal schedule as the one that *looks the most promising*.

Recall constraint (3.22), ceasing production and bypass in Hamlagrøvatn until t_c. We can see that if we set t_c too late, we risk overflowing. However, if this overflow is from Hamlagrøvatn, then in reality we would not be restricted by environmental restriction (i), because that means $V_{2,t}^s > \underline{V}_2$ and (ii) should be activated instead. It seems unlikely that we would keep the reservoir volume of Hamlagrøvatn below $\underline{V}_2$ for $t_c > 32$. Although something interesting happens in $t_c = 34$, we will not investigate that production schedule further.

Since the upper bound is a definite upper bound of expected revenues, we will rely mostly on the upper bound in order to choose the estimated optimal scenario independent t_c. A high upper bound allows the objective function of the Bergsdalen HPS problem (4.6) to be high, thus we call schedules with high upper bound *promising*. The candidates are $t_c \in \{29, 30, 33\}$. We exclude 33 due to the reason mentioned for all $t_c > 32$. The upper bound in $t_c = 29$ is slightly larger than $t_c = 30$. The estimated lower bound is larger in $t_c = 30$, however due to the uncertainty of the lower bound, we let the upper bound determine the estimated optimal $t_c = 29$.

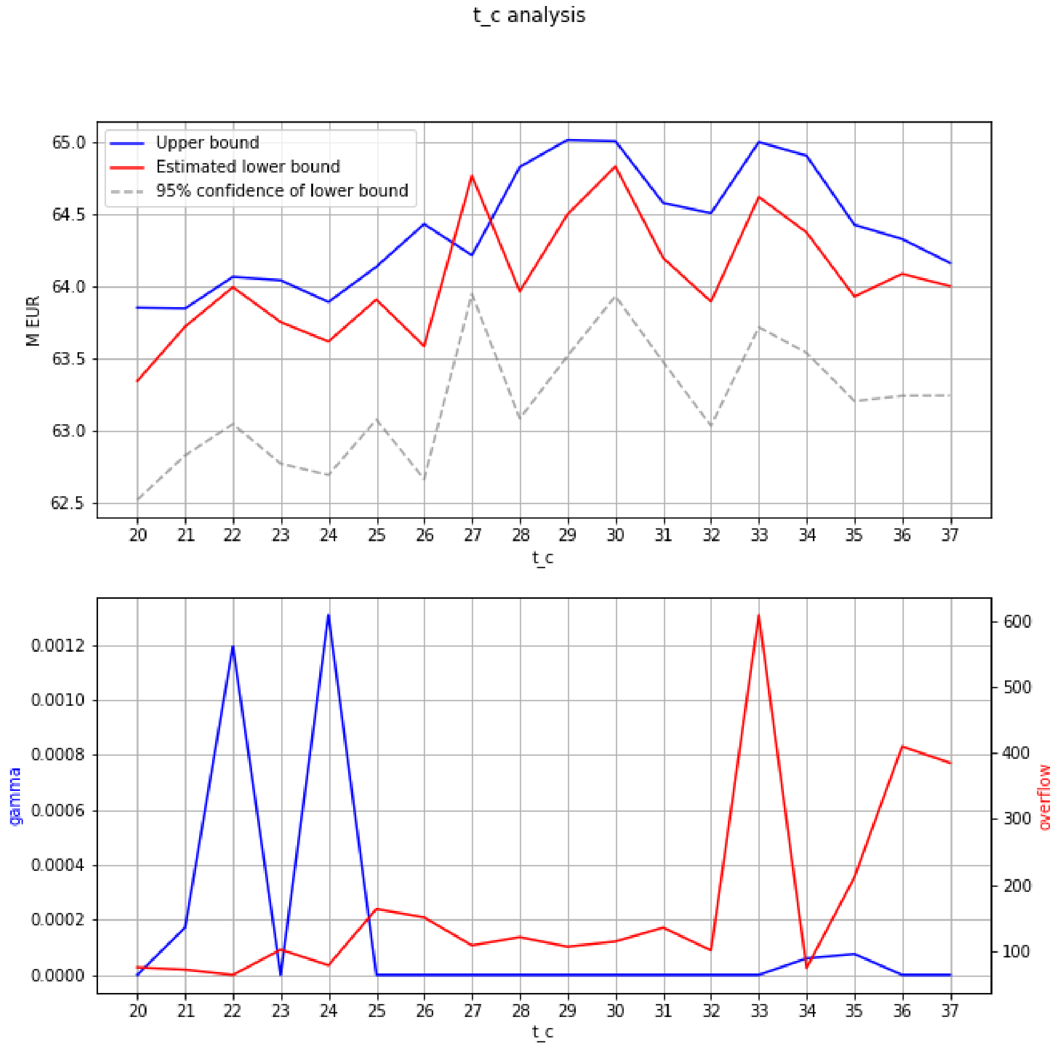


Figure 5.2: *Analysis for estimated optimal scenario independent* t_c

5.3.2 Production schedule for t_c= 29

Results

We now present the estimated optimal production schedule with $t_c = 29$. We have simulated 100 inflow scenarios in each forward simulation. The algorithm converged in roughly 1 hour, with zero penalties.

Table 5.1 shows the upper bound U and estimated lower bound $\hat{L}$ on expected revenues, half-size of the confidence interval 1.96σ for the estimated lower bound and the sum of penalties $\sum\limits_{s\in\hat{\mathcal{S}}}\sum\limits_{t\in\mathcal{T}}\gamma_t^s$, for all iterations. All values except penalties are in M EUR. The estimated optimal production schedule has expected revenues between the upper bound U and lower bound $L = \hat{L} \pm 1.96\sigma$

$$U = 64.5602 \qquad \text{M EUR,}$$
$$\hat{L} \pm 1.96\sigma = 64.1836 \ \pm \ 0.6326 \qquad \text{M EUR,}$$

with 95% confidence of the lower bound. We see that the upper bound is steadily decreasing, whereas the lower bound is only generally increasing after the second iteration. Between iterations 2 and 10 both the half-size of the confidence interval and the sum of penalties are relatively high.

Table 5.1: *Upper and lower bound, with confidence of lower bound and total penalties for all iterations.*

Iteration	U	$\hat{L}$	1.96σ	$\sum\limits_{s\in\hat{\mathcal{S}}}\sum\limits_{t\in\mathcal{T}}\gamma_t^s$
1	74408.4583	41.2196	0.5733	0.0000
2	96.3071	-25075.5509	1338.4892	25113.1818
3	96.3071	-4300.9043	452.7470	4351.7620
4	96.3071	-1388.5201	290.9697	1440.2007
5	93.8197	-673.2475	189.7185	731.2631
6	91.6224	-314.6541	115.5890	374.4670
7	91.6224	-177.0162	97.1947	237.0171
8	91.6224	-115.4890	106.6751	174.9686
9	91.6142	-6.0144	47.5754	66.1176
10	91.5569	18.2698	42.5451	42.3502
11	91.4818	59.7033	4.7890	2.4830
12	79.8821	58.5816	0.7269	0.0000
13	74.0421	57.5564	0.5149	0.0072
14	73.9586	62.9885	0.8027	0.0079
15	64.9756	63.1283	0.6607	0.0000
16	64.9683	63.7342	0.6913	0.0003
17	64.5602	64.1836	0.6326	0.0000

Figure 5.3 shows the upper bound and the estimated lower bound, in the upper plot. The purple points are the revenues of each forward simulation, and the grey dashed line is the confidence interval of the estimated lower bound. The lower plot shows the distribution of overflow and bypass for all simulated scenarios, of the final iteration. The solid line is the 0.5 quantile (median), the dark shaded area represent the 0.25 and 0.75 quantile (lower and upper quartile) and the light shaded area represent the minimum and maximum values (0 and 1 quantile) in every stage. The quantiles show the distribution among scenarios of bypass and overflow, in each stage. This quantile representation will be used frequently.

The minimum value for both bypass and overflow are always zero. For overflow, all values except the maximum values are zero. The maximum overflow is non-zero for $t \in \{28, 29, 43, 44, 46, 49, 50, 51\}$, and towards the end of the period, bypass is more frequent.

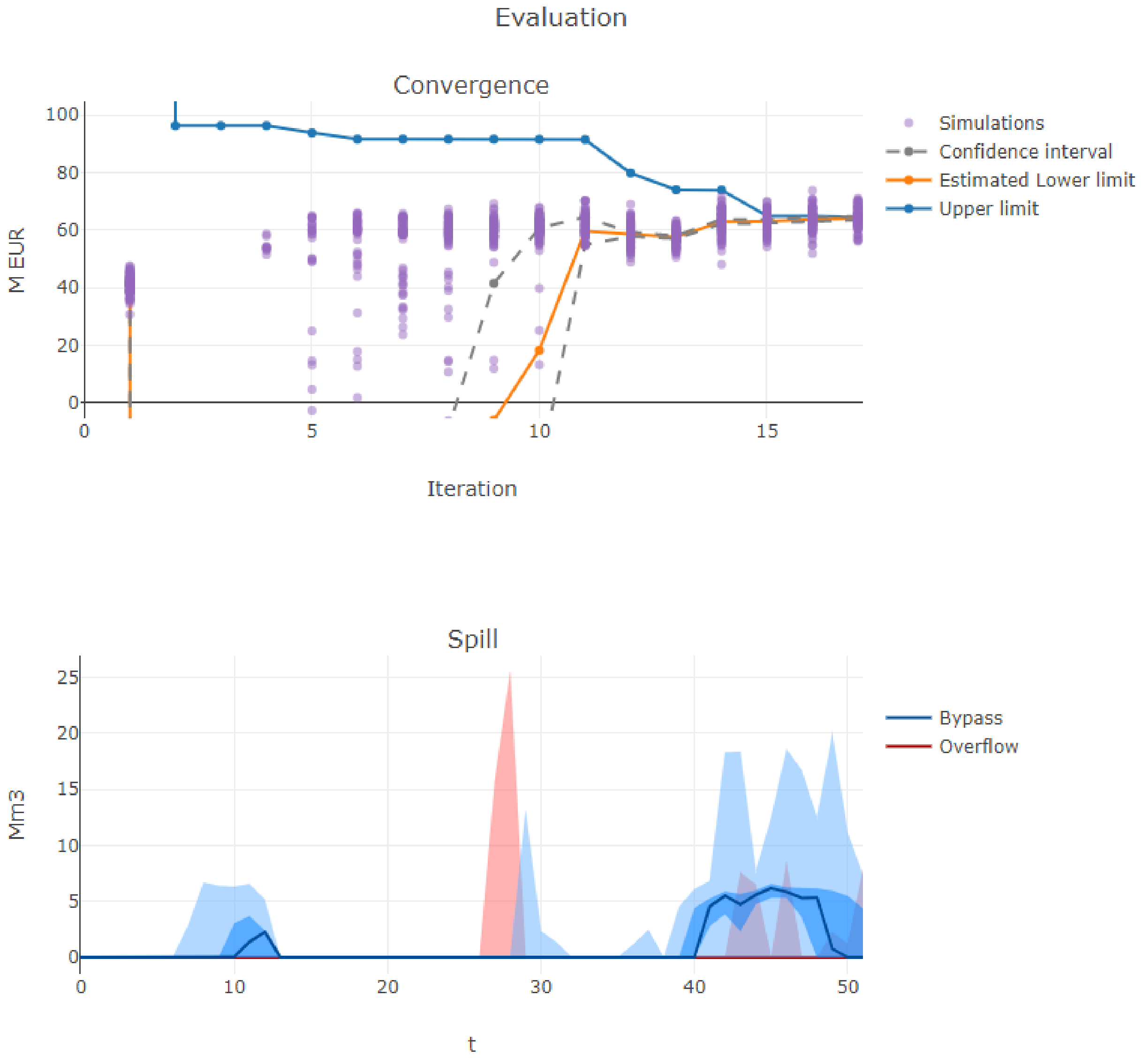


Figure 5.3: Convergence of algorithm and total spill in every stage

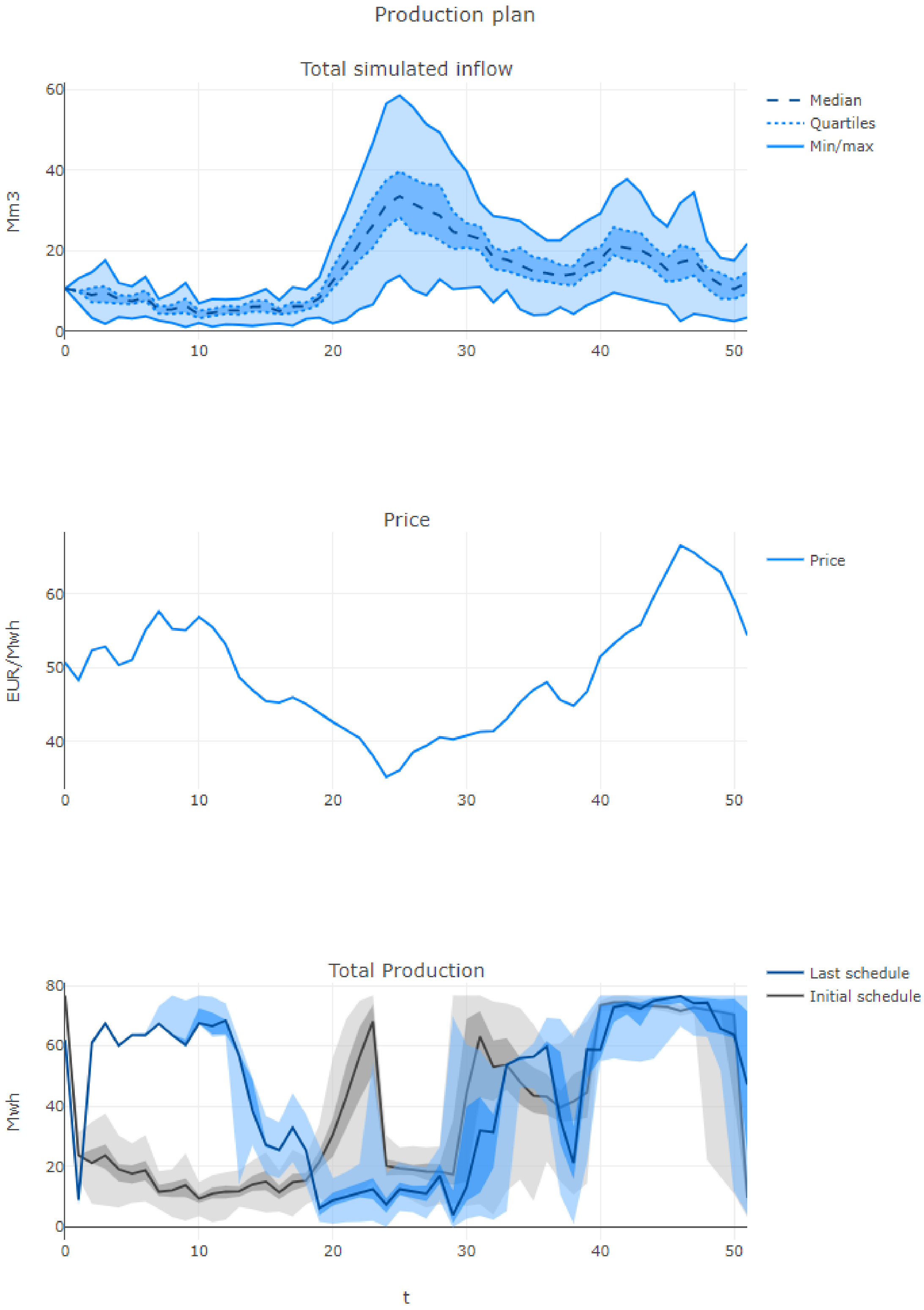


Figure 5.4: *Total simulated inflow from last iteration, input price and estimated optimal schedule.*

Figure 5.4 illustrates the simulated inflow, price input and the estimated optimal production schedule over the planning period. The inflow plot is the total simulated inflow of the last iteration. We have included the production schedule from the initiation step of the algorithm, where expected future income was set to zero, in order to illustrate the improvement caused by constructing an expected future income function in every stage. The estimated optimal schedule has more production when the prices are high, whereas the initial schedule spends the initial volume quickly and is influenced more by the inflow.

Figure 5.5 illustrates the estimated optimal production schedule for Hamlagrøvatn. The bypass is kept insignificantly small throughout the period, and only some scenarios had significant overflow, as presented in Table 5.2. We regard everything above 1 M$m3$ as significant. The red lines represent the intervals where the different approximated environmental constraints are in effect, as indicated by equation references under the horizontal axis.

We see that production is zero for most cases while constraint (3.21) is active. Recall the convex hull (3.21) of the environmental restriction (i), from Section 3.6. We want to examine how often we are breaking restriction (i) by being in the red area in Figure 3.2b. Table 5.3 shows how many scenarios had inflow greater than $\overline{I}$, meaning production and bypass should be zero, according to the environmental restriction. We have compared this to how many scenarios actually has zero production.

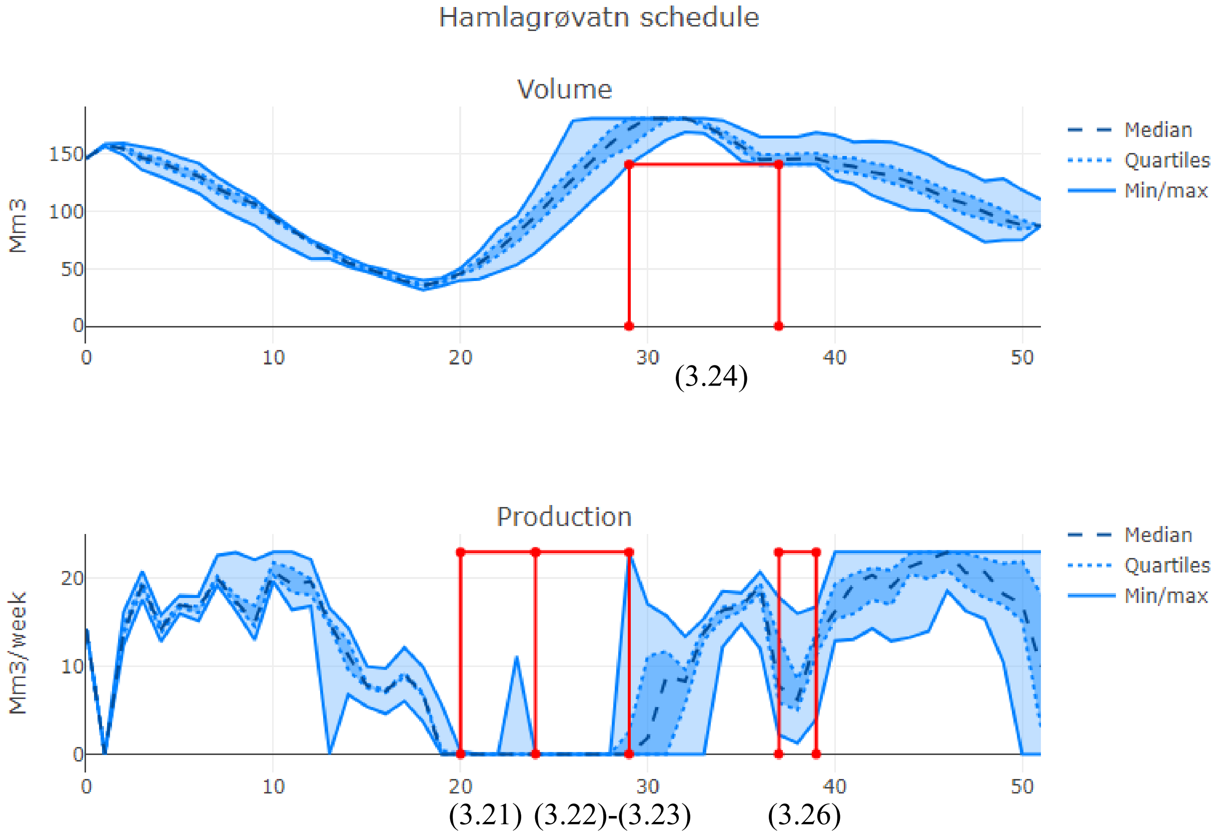


Figure 5.5: *Estimated optimal volume and production schedule for Hamlagrøvatn*

Table 5.2: *Number of scenarios with significant overflow in Hamlagrøvatn, for the estimated optimal schedule*

Stage	Number of scenarios with significant overflow	Maximum overflow
$t = 27$	5	15.76
$t = 28$	13	25.47

Table 5.3: *Number of scenarios where production should be restricted, compared to number of scenarios with zero production.*

Stage	Number of scenarios with $f_{2w}(Z^s_{2,t}) > \overline{I}$	Number of scenarios with zero production
$t = 20$	41	100
$t = 21$	77	100
$t = 22$	87	100
$t = 23$	96	91
$t = 24$	96	100

5.3.3 The value of the environmental constraints

In order to get an estimate of the loss of revenues caused by the constraints, we have run the optimization without including the environmental constraints. For this run, the algorithm converged in roughly 1 hour, after 21 iterations with the upper bound U and lower bound $L = \hat{L} \pm 1.96\sigma$ on the objective function

$$\begin{aligned} U &= 65.0013 && \text{M EUR}, \\ \hat{L} \pm 1.96\sigma &= 64.3980 \ \pm \ 0.7367 && \text{M EUR}, \end{aligned}$$

with 95% confidence of the lower bound, and zero penalty. Figure 5.6 shows the Hamlagrø reservoir volume and production for this solution. Now, the red lines only show where the environmental constraints should have been activated.

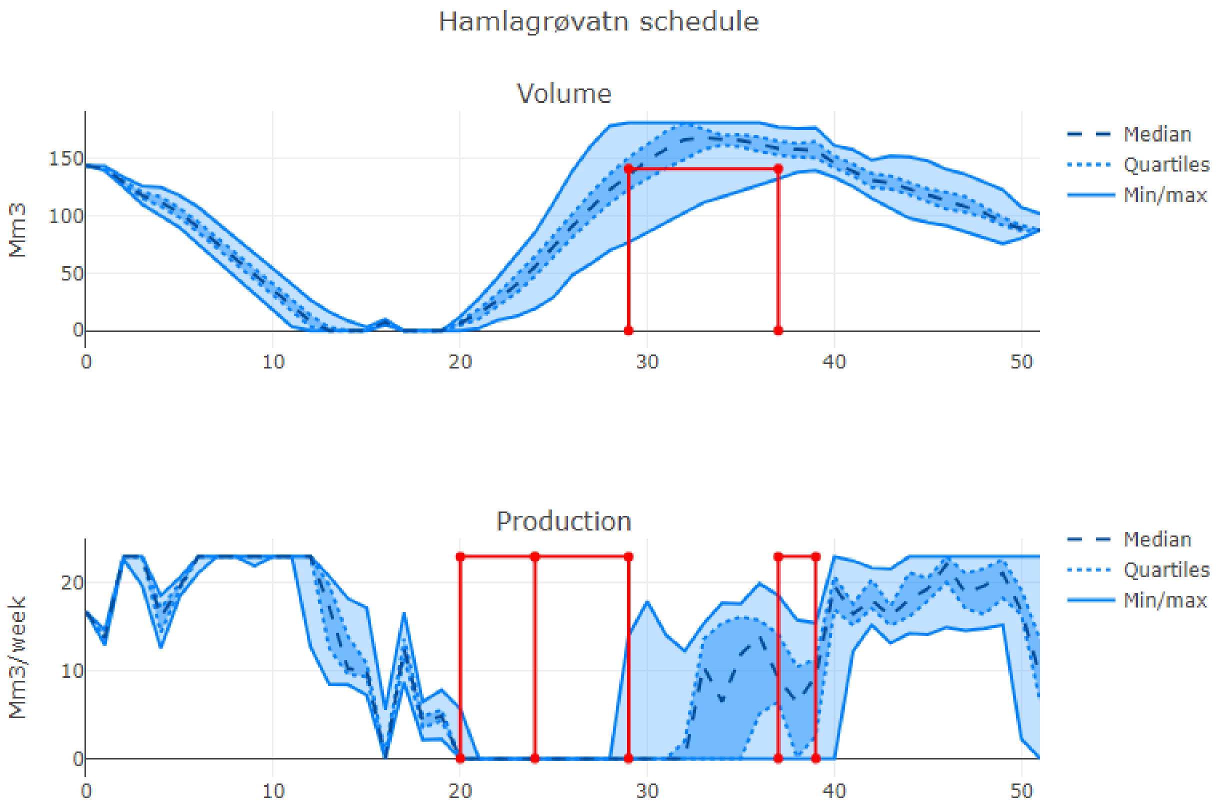


Figure 5.6: *Estimated optimal volume and production schedule for Hamlagrøvatn, without including environmental restrictions.*

5.3.4 Discussion of results

As mentioned in Section 5.2, we observe that the convergence of the algorithm is affected by the penalties. We have set the penalty cost very high, in order to maintain the constraint in all scenarios. From Figure 5.3 we observe that the upper bound does not improve much between iteration 2 and 10. This is arguably the expected future income function is not represented adequately, and is unable to prevent low reservoir volumes, resulting in schedules that break the soft lower volume bounds, and thus gets heavily penalized. Perhaps, a better way of selecting what cuts to include can prevent this effect. We also note that the large confidence interval of the lower bound makes it difficult to certainly choose an estimated optimal t_c.

We have obtained a production schedule which is expected to yield revenues of no more than $U = 64.5602$ M EUR, and no less than $\hat{L} - 1.96\sigma = 63.551$ M EUR with 95% certainty, for the modeled system. However, we have simplified the actual hydro system. By multiplying the annual production for all hydro plants in the Bergsdalen watercourse 1033.6 Gwh (NVE, 2019) with the mean annual price from last year (2018) 43.05 EUR$/Mwh$ (Nordpool, 2019), we get an estimated annual revenues of roughly 44.5 M EUR. Considering we used a deterministic price series, we expect the optimal production schedule is ”bolder”, than if we modeled price as a stochastic parameter. A ”bolder” schedule does not consider uncertainties in price, and will obtain larger revenues if the realized price is the same as we guessed in the beginning of the planning period. We have set the end volume bound slightly lower than the initial volume, resulting in a net loss of water over the planning period, although a large production volume. This may not be representative for the actual decisions made by the producer, BKK. In addition, the simplification of turbine efficiency functions to a scalar, as mentioned in Section 2.2, might yield a higher power output than the actual hydro system. We keep in mind the differences between the actual hydro system and the model, however we assume that the estimated optimal schedule is still relatable to the actual hydro system, in terms of comparing one schedule to another.

Figure 5.4 gives an indication that the estimated optimal schedule is trustworthy. The goal was to store water when prices are low, and produce power when prices are high. We can see that the large inflow volumes between $t = 20$ and $t = 30$ do not affect the estimated optimal production considerably, because the price is low in this period. That means, the schedule is able to make use of the storage capacity of the reservoirs in an efficient manner. We notice that bypass increases considerably towards the end of the planning period. This may be due to the lower bound on end reservoir volume, which may not be ideal end term conditions. However, we do not trust the end of the production schedule due to its sensitivity to the end term conditions as mentioned in Section 1.1. We also notice dips in the productions at $t = 1$ and $t = 38$, where the price is relatively high compared to the rest of the period, however not to the closest price points.

In order to determine how well we are able to represent the environmental restrictions, we inspect the schedule for Hamlagrøvatn, Figure 5.5. Recall that bypass was considered insignificant, and only some overflow was considered significant, as presented in Table 5.2.

We know that restriction (i) is accurately represented from t_b to t_c, for a scenario independent t_c. From t_a to t_b it is not accurately represented. However, production is ceased in most scenarios. When considering the price plot in Figure 5.4 this makes sense, because the price in this interval is relatively low. Hence, the effect of restriction (i) depends on the price. Note that in $t = 23$, we are breaking restriction (i), because we have non-zero production in 9 scenarios, whereas only 4 scenarios have inflow volume less than $\overline{I}$.

The soft lower bound on reservoir volume from t_c to t_d is maintained. Restriction (ii) has been adequately represented for a scenario independent activation time. This means we have set the penalty cost sufficiently high. However, we noticed that the high penalty cost affected the convergence of the algorithm. For this case, it is possible to maintain the restriction in all scenarios. If that is not the case we will severely affect the convergence. Moreover, we have seen that the algorithm captures the productions dependency on power price. This in turn may affect the optimal value of t_c, since production can wait while prices are low and vice versa. In addition, if t_c is set too late, we get a long period with no production and bypass, and we risk overflowing. Hence, it is likely that the optimal schedule can be improved if we are able to estimate a t_c value for each scenario. This could be done by implementing a heuristic utilizing the two observations above, to push t_c back if prices are low, and move t_c forward if we have a high inflow scenario where we risk overflow. However, due to the sharing of cuts among scenarios in each case, it seems like a complicated task.

We notice that volume is stable from t_d to t_e, and restriction (iii) is accurately represented.

Comparing the production schedules in Figures 5.5 and 5.6, we see that the solution is not excessively affected by the environmental constraints. However for the schedule without environmental constraints, we see that the Hamlagrøvatn reservoir is completely emptied around $t = 18$ for all scenarios. Thus, this schedule is able to produce more in the beginning of the period. We have not discussed the production efficiency coefficients. However, the production efficiency is lowest in the Hamlagrøvatn reservoir. Although, we still earn more by producing more in Hamlagrøvatn, it is reasonable to assume that the schedule is able to ”work around” the restriction and keep production high in modules with higher production efficiency. The production efficiency coefficients are available at NVE (2019), as *energiekvivalent*. The upper bound is 0.7% larger, and the estimated lower bound also indicates that the true lower bound is larger without the environmental constraints. This indicates that we might earn a little more if we do not respect the environmental constraints, but we will not increase revenues by more than 0.7%.

5.4 Price scenario sensitivity

In order to check the algorithms sensitivity to the choice of price series, we perform the optimization again with a different choice of price series. Figure 5.7 shows the new price series used for this analysis. This price series is also obtained from the collection of price series depicted in Figure 5.1. We use the same procedure as previously, in order to determine an estimated optimal scenario independent t_c and evaluate the estimated optimal schedule.

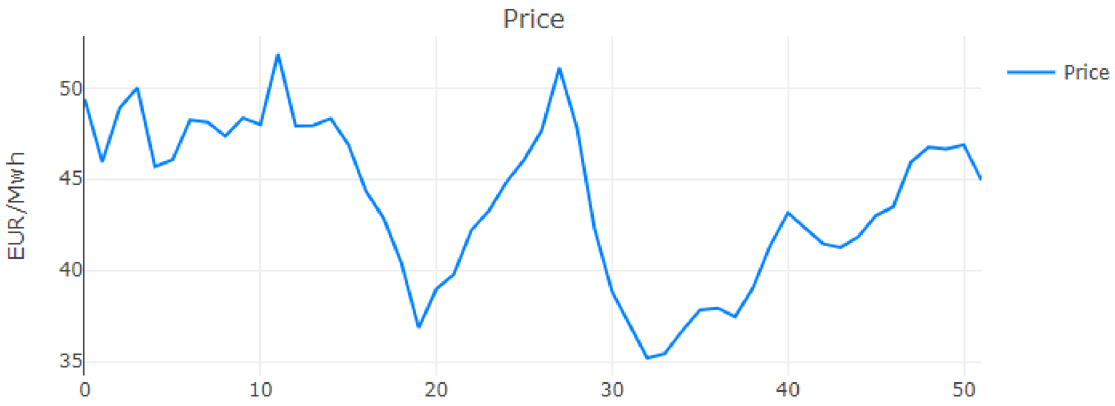


Figure 5.7: *The price series chosen for the price scenario sensitivity analysis*

5.4.1 Second t_c analysis

As we did in the previous optimization procedure, we have run Algorithm 3 for all 18 possible values of t_c. 60 inflow scenarios were simulated in each forward iteration, and the calculations finished in approximately 10 hours. Figure 5.8 shows the result of the t_c analysis. We determined $t_c = 23$ as the estimated optimal scenario independent value for t_c.

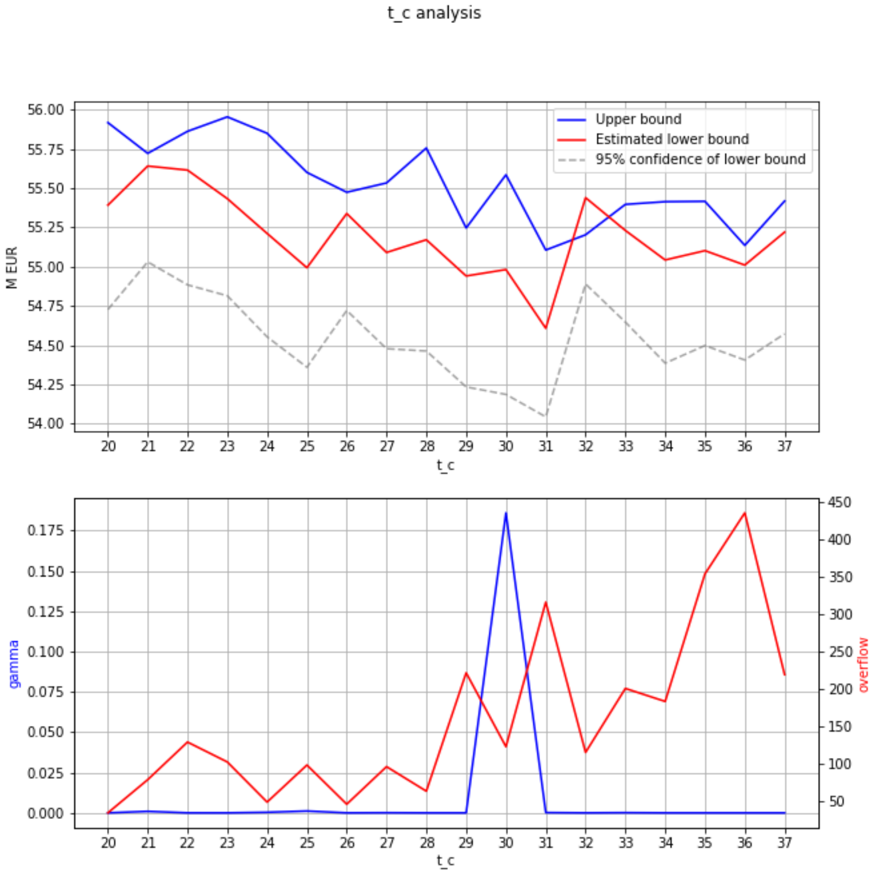


Figure 5.8: *Second analysis for estimated optimal scenario independent t_c*

5.4.2 Production schedule for t_c= 23 with second price series

We ran algorithm (3) with $t_c = 23$, simulating 100 scenarios in the forward simulation. The algorithm converged in roughly 1.3 hours, after 19 iterations. The upper bound U and the lower bound $L = \hat{L} \pm 1.96\sigma$ were

$$U = 55.7924 \qquad \text{M EUR},$$
$$\hat{L} \pm 1.96\sigma = 55.3622 \ \pm \ 0.6876 \qquad \text{M EUR},$$

with 95% confidence, with zero penalty. Figure 5.9 shows the volume and production of the Hamlagrøvatn reservoir. We observe that from the t_c analysis, we have chosen

$t_c < t_b$. As explained in Section 3.6.3, this implies that constraint (3.23) is never activated.

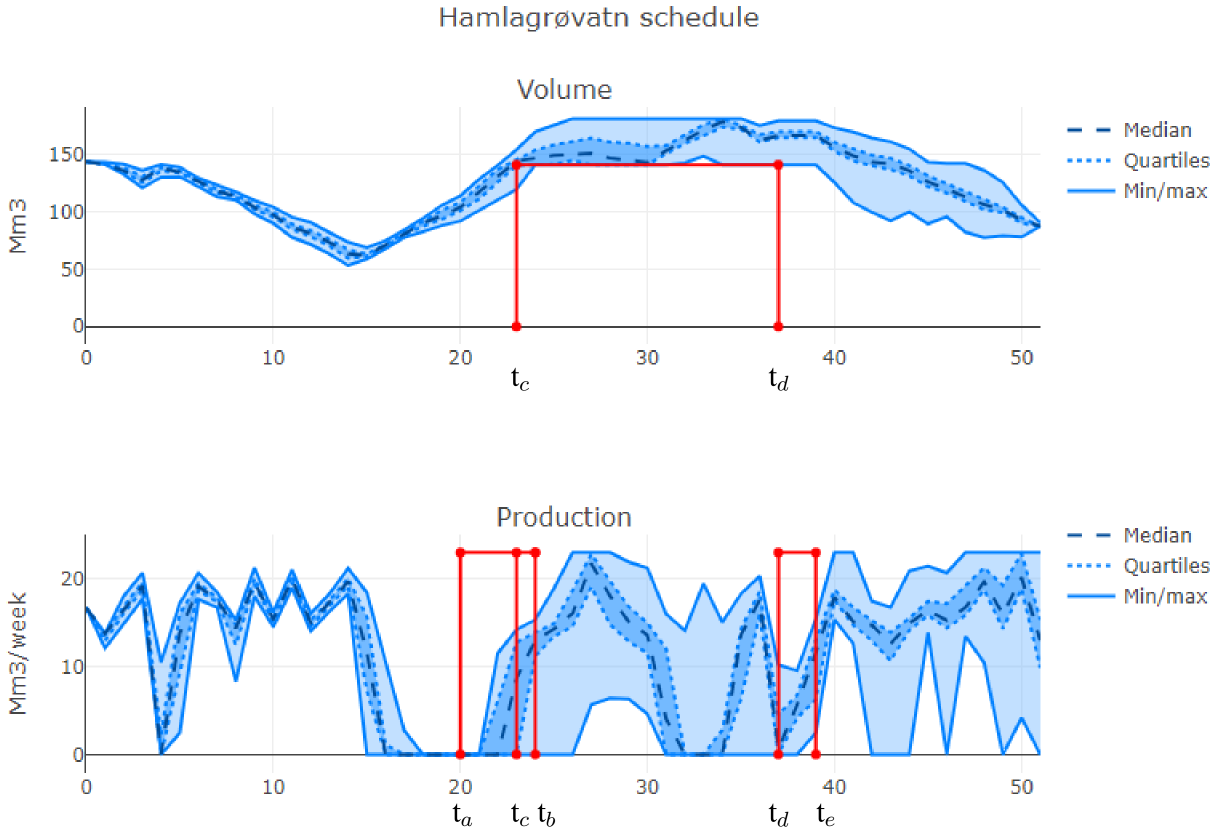


Figure 5.9: Estimated optimal volume and production schedule for Hamlagrøvatn using the second price series

5.4.3 The value of the environmental constraints for the second price series

We ran the same case as above, without including the environmental constraints. For this run, the algorithm converged in roughly 50 minutes, after 17 iterations with the upper bound U and lower bound $L = \hat{L} \pm 1.96\sigma$ on the objective function

$$\begin{aligned} U &= 56.4821 && \text{M EUR}, \\ L \pm 1.96\sigma &= 56.3636 \ \pm\ 0.7082 && \text{M EUR}, \end{aligned}$$

with zero penalty. Figure 5.10 shows the Hamlagrøvatn reservoir volume and production for this solution.

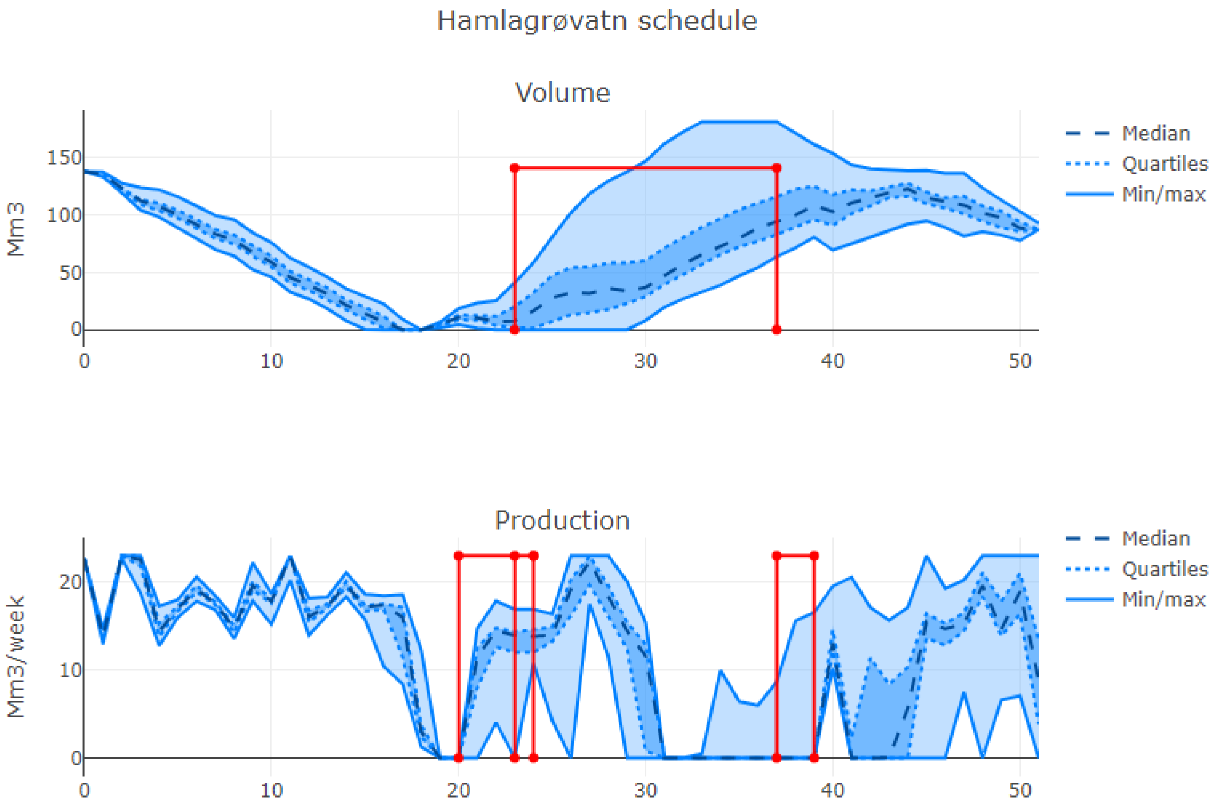


Figure 5.10: *Estimated optimal volume and production schedule for Hamlagrøvatn, without including environmental restrictions.*

5.4.4 Discussion of results

The price scenario chosen for the analysis, was chosen on purpose because it features a high power price in a period where production might be completely restricted, depending on the activation of restrictions (ii), t_c. When comparing the optimization for the two price series, we see that t_c may depend significantly on the price. Thus, a scenario independent representation of t_c is not ideal if we are to extend this implementation to include price uncertainty.

For this case, we see that the production schedules in Figures 5.9 and 5.10 are significantly different. As the price has a peak at $t = 27$, we see that the Hamlagrøvatn reservoir volume is usually kept low at this time, arguably due to high production, and filling start later than in the previous case, Figure 5.6. However, both schedules has a high chance of producing close to maximum capacity at the price peak in $t = 27$. The upper bound is 1.2% larger, and the estimated lower bound also indicates that the true lower bound is larger without the environmental constraints. For this case, we can increase revenues a little more by not respecting the environmental constraints, than in the previous case. However, not by more than 1.2%.

Chapter 6

Conclusions

Hydropower production planning is an important task. Depending on the size and timescale of the problem, different aspects of the task should be prioritized in the modeling. As prognoses for inflow and price are subject to a great deal of uncertainty on a longer timescale, it is important to optimize with respect to these uncertainties, when determining an optimal production schedule. The solution method method presented in this thesis is able to handle the uncertainty of future inflow, at the cost of sacrificing modelling accuracy. In order to use the proposed solution method, we require the problem to be formulated as an LP problem, in order to construct Benders cuts to construct a representation of the expected future cost function in each stage. Additionally, the LP representation has the advantage of giving the shadow prices of the reservoir volume restrictions, called water values. The water values at the beginning of the planning period may be valuable for making daily decisions. We found that the environmental restrictions imposed by NVE on the Bergsdalen watercourse can be included if they are approximated using scenario independent activation times.

6.1 Scheduling under environmental restrictions

When determining an optimal production schedule for the Bergsdalen watercourse, we need to respect the environmental restrictions imposed on the system, by NVE. The restrictions were divided in to three parts, where all three affect the Hamlagrøvatn reservoir in the Bergsdalen watercourse. The restrictions are imposed at the earliest on 15. April, until 1. September. First, once the inflow volume reaches a certain level, during the spring melt, we are required to cease production and bypass until we reach a certain reservoir volume level. Then, we are required to maintain that reservoir volume level until 15. August. Lastly, we are not allowed to lower the reservoir volume level until 1. September. We saw that the first two parts challenges the solution method, and require simplification if we are to model the problem as an LP problem. The third part was implemented with success.

The first part of the environmental restrictions is activated depending on the inflow scenario, hence we classified it as scenario dependent. Thus, the restriction might be activated at different times in different scenarios. We have seen that the solution method requires that the Benders cuts are shared among all scenarios in every stage. Therefore, the restriction is challenging, because all scenarios in each stage need to share the same circumstances. That is, all subproblems should have the same solution space in each stage. In addition, the solution spaces need to be convex, due to the LP requirement.

In order to reflect the first part of the environmental restrictions, we determined a solution space restricting production and bypass when the inflow volume is sufficiently high to activate the scenario. This allows the possibility of ”deactivating” the restriction if inflow decreases. However, during the relevant period, which is the spring melt, inflow usually increases. As a complete stop, that is a discrete activation, of production and bypass after sufficient inflow resulted in a non-convex solution space, we replaced the solution space with its convex hull. As a result the optimal schedule may break the restriction, however to a smaller extent than without any restriction. Since the relevant period was relatively short, we considered the effect of this simplification to be small. Additionally, after analyzing the estimated optimal solution we observed that due to low prices during this period, most scenarios had zero production even when production was not restricted by the constraint. Hence, the price affects whether or not the constraint will be binding.

The second part of the environmental restrictions is activated depending on the reservoir volume, which is a decision variable in the optimization model. If we let the activation time be a variable depending on the reservoir volume, we introduce a non-linearity. Additionally, the discrete activation of the function, similar to the first part, is not consistent with the LP requirement.

As means of approximating the second part of the environmental restrictions, we simplified the activation time to a scenario independent value, and solved the HPS problem (2.4) for all possible activation times. This required a long computation time, and we recommend it only for determining a fixed activation time, for later use. Additionally, it was hard to conclude the optimal activation time for this case, due to the uncertainty in the lower bound. However, the upper bound on the optimal objection function value can determine what activation time *looks the most promising*. If the uncertainty can be lowered, we believe the analysis can be profitable for a hydropower producer. In reality the optimal activation time is most likely scenario dependent. Thus, we can consider a fixed activation time, which we have found from running the analysis of all possible scenario independent activation times. Then, we could investigate a heuristic to move the activation times forward or backwards, depending on the scenario. However, as cuts are shared among scenarios, we are quite restricted as to what strategies could be implemented.

We ran the optimization for a second price series, in order to investigate the effects of the environmental constraints under another price scenario. The second price series was selected in order to try to challenge the environmental restrictions. We found that the scenario independent activation time of the second part of the environmental constraints may change significantly, depending on the price scenario. Additionally, we ran the optimization for both price scenarios without including the environmental constraints. For the first price scenario this increased the upper bound on expected revenues by 0.7%, and by 1.2% for the second price scenario.

6.2 Further work

The HPS problem solved in this thesis, problem 2.4, is a simplified version of the scheduling problem solved by Gjelsvik et al. (2010). There are already established techniques which expand on this solution method such as piecewise linear production efficiency curves (mentioned in Section 2.2.1), and parallel processing (mentioned in Section 5.2). There is great potential in further improving the representation of production efficiency functions, for instance by modelling non-convex efficiency functions (Helseth, 2019). The computation time can also be decreased more by improving parallel processing implementation, as mentioned by Helseth and Braaten (2015).

We have seen that modelling complexity is quite limited when utilizing the solution method presented in this thesis. If greater detail is required, to implement more complex constraints, we recommend investigating other techniques. In Section 1.2, we mentioned some other techniques that have been tried. Modelling the problem as a Mixed Integer Linear Program (MILP) significantly increases the modeling accuracy of the environmental constraints presented in this thesis. However, then we need to limit the amount of inflow scenarios we take into account. Note that, as mentioned in the beginning of this chapter, power producers are interested in the water value of their reservoirs. Thus, a potential solution method should have well a defined dual problem. If we introduce integers to the optimization model, we lose the water values. A scenario reduced MILP was solved by Hjelmeland et al. (2019), with a version av the SDDP algorithm. The introduction of integers resulted in higher computation time, and the trade-off between improved results and increased computation time was not justified. However for the Bergsdalen problem, the number of integers to be introduced is limited, as the environmental constraints can only be activated during the summer. An interesting experiment is to include a scenario dependent activation time for the second part of the environmental constraint in the solution method presented by Hjelmeland et al. (2019), and investigate potential improvements of results.

Appendix A

Central components of the algorithm

Here we present the implementation of the main components of the SDDP algorithm presented in this thesis. We chose to present the first part of the initialization, in order to give an overview of the attributes belonging to an instance of the Problem class. An instance of the Problem class represent an instance of all subproblems defined by (4.7) for all $(s, t) \in \mathcal{S} \times \mathcal{T}$. We have defined object methods for performing backward recursion, Algorithm (1), and forward simulation, Algorithm (2), on an instance of the Problem class. Lastly, we run the SDDP algorithm on the Problem class in the main algorithm, which is outlined by Algorithm (3).

Additionally we have created functionality to read inflow, price and system parameter data, generate the inflow model described in Section 4.1, build the pyomo models of each subproblem (4.7) and plotting results. To access the entire source code, please visit https://bitbucket.org/harvidsen/hydropowersddp/src/master/. Due to restrictions on confidential data, the public will not be able to test the code without obtaining another dataset. Functionality for generating data, may or may not be added to the repository.

A.I Imports

The following are the imported modules for the main script. We have put frequently used operations in "helpers.py".

```
In [ ]: from submodel import buildModel
        import numpy as np
        import matplotlib.pyplot as plt
        import pyomo.environ as pyo
        from pyomo.opt import SolverFactory, SolverStatus, TerminationCondition
        import pandas as pd
        import sys
        from helpers import *
        import math
```

A.II Initialization

The following method initializes an instance of the Problem class. We show only the first part of the initialization, as the simulation part of the initialization is almost equivalent to the forward simulation.

```
In [ ]: class Problem:
            def __init__(self, S,  T, pricedata, inflowModel, params, opt, times, q_ymean):
                '''Initializing the Problem class. Creating all containers for values.
                Building all subproblems and guessing an optimal production plan,
                without considering future income.'''
                #Creating structure for a little overview of dimensions
                self.opt = opt                                  # The chosen solver
                self.params = params                            # Parameters
                self.T = T                                      # Planning horizon
                self.S = S                                      # Scenarios to simulate
                self.I = len(params)                            # Number of reservoirs
                self.IM = inflowModel                           # The inflow model
                D2container = np.full(shape=(S,T), fill_value = -1.0)
                D3container = np.full(shape=(S,T,self.I), fill_value = -1.0)
                self.V = np.copy(D3container)                   # Reservoir volumes
                self.Q = np.copy(D3container)                   # Production
                self.B = np.copy(D3container)                   # Bypass
                self.O = np.copy(D3container)                   # Overflow
                self.Z = np.copy(D3container)                   # Transformed inflow
                self.Y = np.copy(D2container)                   # Sold power
                self.gamma = np.copy(D3container)               # Penalties
                self.EFI = np.copy(D2container)                 # Expected future income
                self.subs = np.copy(D2container).tolist()       # List of subproblems
                self.revenues = np.copy(D2container)            # Revenues in each stage
                self.objs = np.copy(D2container)                # Objective function
                                                                #   values in each stage
                self.Vcoeffs = []                               # Water values for t = 1
                self.betas = []                                 # Cut RHS for t = 1
self.std = []                                   # Standard deviations
                                                #   of each simulation
self.dist = []                                  # Total revenues
                                                #   of each simulation
self.Upper = []                                 # Upper bounds
self.Lower = []                                 # Lower bounds

t0, t1, t2, t3, t4 = times
self.times = times          # Activation times for environmental constraints
self.q_ymean = q_ymean      # Inflow value to activate (i)

#Assigning values
self.p = pricedata                  # Price of power in each stage
# Number of possible inflow error realizations
self.K = self.IM.e_t_k[0].shape[0] #All times must have the same K
self.V0 = np.empty(self.I)          # Initial reservoir volumes
for i in range(self.I):
    self.V0[i] = params[i]['V0']

# In case we want to fix some random scenarios
# self.picks = np.full(shape=(S,T), fill_value = 0)
# for s in range(S):
#     for t in range(T):
#         self.picks[s,t] = np.random.choice(3)

IM = self.IM

#################ACTUAL SCRIPT DOES NOT END HERE##########################
#Perform a forward simulation that generates pyomo models in self.subs[s][t]
#   for all (s,t) in self.S and self.T, while solving with EFI fixed to zero.
```

A.III Backward recursion

The following method is defined in the Problem class. This is the backward recursion method used in the algorithm.

```
In [ ]: def backPass(self):
            '''Creating cuts for stage T-1 to stage 0, by solving subproblems for
            stage T to stage 1. Only creating 8 cuts in each iteration, where
            subproblems to create cuts are chosen heuristically to create only the
            "most relevant" cuts.'''
            #Fetching some values for ease of access
            T = self.T
            S = self.S
            I = self.I
            opt = self.opt
            subs = self.subs
            K = self.K

            for t in [T-i-2 for i in range(T-1)]:
                #Using heuristic to choose subproblems to generate ''most relevant'' cuts
                sampleSampleS, rest = findScatteredVals(self.revenues[:,t+1], S)
                sampleSampleS.extend(np.random.choice(rest, size = 4, replace = False))

                for s in sampleSampleS:
                    sub = subs[s][t+1] #Creating cut from t+1
                    beta = 0
                    pi = np.zeros(shape = (I))
                    pi2 = np.zeros(shape = (I))

                    #Startvalues for t+1
                    V_prev = self.V[s,t]
                    Z_prev = self.Z[s,t]

                    #Solving for all K realizations of inflow modelling error,
                    # and creating cut coefficients.
                    p_ind = 0
                    p_lst = self.IM.p_dist
                    for e in self.IM.e_t_k[(t+1)%52]:
                        p = p_lst[p_ind]
                        p_ind += 1
                        for i in range(I):
                            sub.e[i] = e[i]
                        opt.solve(sub, warmstart = True)
                        beta += p*pyo.value(sub.obj)

                        for i in range(I):
                            dual = sub.dual[sub.waterBalance[i]]
                            dual2 = sub.dual[sub.inflowTransition[i]]
                pi[i] += p*dual
                pi2[i] += p*dual2

        beta -= np.dot(V_prev,pi)
        beta -= np.dot(self.IM.f2((self.IM.cmat @ Z_prev), t+1), pi2)

        if t == 1:
            self.Vcoeffs.append(pi)
            self.betas.append(beta)

        #Adding expected cut all scenarios in stage t
        for u in range(S):
            sub_prev = subs[u][t]
            V_expr = pyo.summation(pi, sub_prev.V)
            # The following three lines should be one line
            Z_expr = sum(self.IM.f(sum(self.IM.cmat[i,j]
            * sub_prev.Z[j] for j in sub_prev.I), t+1, i)
            * pi2[i] for i in sub_prev.I)
            cut = sub_prev.EFI - V_expr - Z_expr <= beta
            sub_prev.cuts.add(cut)


#Calculating new upper limit
lst = []
for s in range(S):
    opt.solve(subs[s][0], warmstart = True)
    lst.append(pyo.value(subs[s][0].obj))

self.Upper.append(max(lst)) #Should be equal anyway
```

A.IV Forward simulation

The following method is defined in the Problem class. This is the forward simulation method used in the algorithm.

```
In [ ]: def forwardPass(self):
            '''Simulating new inflow scenarios, and solving all subproblems. As the
            backward recursion might alter the state of some subproblems, we assign
            them the state of the last simulation.'''
            #Fetching some values for ease of access
            S = self.S
            T = self.T
            I = self.I
            opt = self.opt
            subs = self.subs
            IM = self.IM
            t0, t1, t2, t3, t4 = self.times

            obj_lst = []
            for s in range(S):
                objective = 0
                for t in range(T):
                    #Determining state of subproblem (s,t)
                    if t == 0:
                        V_prev = self.V0
                        Z_prev = IM.Z0
                        realized_noise = [0,0,0,0] #Assuming no error in first week
                        Z = IM.cmat @ Z_prev + realized_noise
                    else:
                        V_prev = self.V[s,t-1]
                        Z_prev = self.Z[s,t-1]
                        ind = np.random.choice(self.K, p = IM.p_dist)
                        # # In case we want to fix the scenarios
                        # realized_noise = self.noise[self.picks[s,t]]
                        realized_noise = IM.e_t_k[t%52,ind]
                        Z = IM.cmat @ Z_prev + realized_noise


                    #Assigning the correct values to the subproblem
                    sub = subs[s][t]
                    for i in range(I):
                        sub.Z0[i] = Z_prev[i]
                        sub.e[i] = realized_noise[i]
                        sub.V0[i] = V_prev[i]


                    #Solving and storing
                    result = opt.solve(sub, warmstart = True)
```

```
        checkFeasibility(sub, result, sub.name)
        self.EFI[s,t] = pyo.value(sub.EFI)
        self.revenues[s,t] = pyo.value(sub.p*sub.Y)
        self.objs[s,t] = pyo.value(sub.obj)
        subObj = pyo.value(sub.obj) - pyo.value(sub.EFI)
        objective += subObj
        if t == T-1: objective += pyo.value(sub.EFI)
        self.Z[s,t] = vecVal(sub.Z)
        self.V[s,t] = vecVal(sub.V)
        self.Q[s,t] = vecVal(sub.Q)
        self.B[s,t] = vecVal(sub.B)
        self.O[s,t] = vecVal(sub.O)
        self.gamma[s,t] = vecVal(sub.gamma)

    obj_lst.append(objective)

lower = 1/S * sum(obj_lst)
this = 0
for obj in obj_lst:
    this += (lower - obj)**2

std = math.sqrt(1/(S**2) * this)
#Updating lower bound
self.std.append(std)
self.Lower.append(lower)
print('Standard deviation of Monte Carlo run: ', std)
#And saving trajectories to plot
self.history.append(np.copy(1/self.S*self.Q.sum(axis = (0,2))))
self.check.append(np.copy(self.Z))
self.dist.append(obj_lst)
```

A.V Main algorithm

This is the main algorithm implemented in this thesis, in order to solve the Bergsdalen HPP. Here we have excluded functionality for logging and plotting, which is present in the actual implementation.

```
In [ ]: #Initialize
        #
        data = Problem(
            S,                          # Number of scenarios to simulate in the forward simulation
            T,                          # Number of weekly time-stages
            pricedata,                  # A deterministic price series
            inflowModel,                # An instance of the inflow model
            params,                     # Hydro system parameters
            opt,                        # The chosen solver instance, from pyomo.opt.SolverFactory
            times,                      # Activation times for the approximated environmental
                                        #    constraints

            q_ymean                     # Inflow amount for activation environmental restriction (i).
        )                               #    Only for plotting.

        maxIter = 30
        e = 1 #Just higher than tol
        tol = 0
        i = 1
        while e > tol or tol > 1:
            # Perform backwards recursion
            data.backPass()

            #Perform forward simulation
            data.forwardPass()

            #Update convergence criterion values
            tol = 1.96*data.std[-1]
            e = abs(data.Upper[-1] - data.Lower[-1])

            #Check iteration number
            if i == maxIter:
                log.write('\nDid not converge after '+str(i)+ ' iterations')
                break
            i += 1
```

Bibliography

Antonio, F., C. Gentile, and F. Lacalandra (2011, 03). Sequential lagrangian-milp approaches for unit commitment problems. *International Journal of Electrical Power & Energy Systems 33*, 585–593.

Bellman, R. and R. Kalaba (1966). *Dynamic Programming and Modern Control Theory*. Elsevier Science.

Benders, J. F. (1962, December). Partitioning procedures for solving mixed-variables programming problems. *Numer. Math. 4*(1), 238–252.

Birge, J. (1982, 12). The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical Programming 24*.

bkk.no (2019). Bkk river systems. Available at https://www.bkk.no/en/hydropower.

Cerisola, S., J. Latorre, and A. Ramos (2012, 05). Stochastic dual dynamic programming applied to nonconvex hydrothermal models. *European Journal of Operational Research 218*, 687–697.

de Souza Zambelli, M., L. S. A. Martins, and S. S. Filho (2013, July). Advantages of deterministic optimization in long-term hydrothermal scheduling of large-scale power systems. In *2013 IEEE Power Energy Society General Meeting*, pp. 1–5.

Ge, X., S. Xia, and X. Su (2018, 5). Mid-term integrated generation and maintenance scheduling for wind-hydro-thermal systems. *International Transactions on Electrical Energy Systems 28*(5).

Gjelsvik, A., B. Mo, and A. Haugstad (2010). *Long- and Medium-term Operations Planning and Stochastic Modelling in Hydro-dominated Power Systems Based on Stochastic Dual Dynamic Programming*, pp. 33–55. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gurobi Optimization, L. (2018). Gurobi optimizer reference manual.

Haga, F. (2019). Conversations and email correspondence.

Hart, W. E., C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola (2017). *Pyomo–optimization modeling in python* (Second ed.), Volume 67. Springer Science & Business Media.

Hart, W. E., J.-P. Watson, and D. L. Woodruff (2011). Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation 3*(3), 219–260.

Helseth, A. (2019). Conversations and email correspondence.

Helseth, A. and H. Braaten (2015). Efficient parallelization of the stochastic dual dynamic programming algorithm applied to hydropower scheduling. *Energies 8*(12), 14287–14297.

Helseth, A., A. Gjelsvik, B. Mo, and U. Linnet (2013, December). A model for optimal scheduling of hydro thermal systems including pumped-storage and wind power. *IET Generation, Transmission Distribution 7*(12), 1426–1434.

Hjelmeland, M. N., J. Zou, A. Helseth, and S. Ahmed (2019, Jan). Nonconvex medium-term hydropower scheduling by stochastic dual dynamic integer programming. *IEEE Transactions on Sustainable Energy 10*(1), 481–490.

Kall, P. and S. W. Wallace (1994). *Stochastic Programming*. John Wiley & Sons.

Labadie, J. (2004, 03). Optimal operation of multireservoir systems: State-of-the-art review. *Journal of Water Resources Planning and Management 130*, 93–111.

M. Apap, R. and I. Grossmann (2016, 12). Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. *Computers & Chemical Engineering 103*.

Nordpool (2019). Market data, elspot day-ahead. Available at https://www.nordpoolgroup.com/Market-data1/Dayahead/Area-Prices/ALL1/Yearly/?view=table.

NVE, Norwegian Water Resources and Energy Directorate. (2015). Energi i Norge. *An overview of electrical energy production and consumption for 2014*.

NVE, Norwegian Water Resources and Energy Directorate. (2019). Hydropower database. Available at https://www.nve.no/energiforsyning/vannkraft/vannkraftdatabase/.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Pereira, M. (1989). Optimal stochastic operations scheduling of large hydroelectric systems. *International Journal of Electrical Power & Energy Systems 11*(3), 161 – 169.

Pereira, M. V. F. and L. M. V. G. Pinto (1991, May). Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming 52*(1), 359–375.

Ross, S. M. (1983). *Introduction to Stochastic Dynamic Programming: Probability and Mathematical*. Orlando, FL, USA: Academic Press, Inc.

Seabold, S. and J. Perktold (2010). Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.