Short-term Hydropower Scheduling At Watercourses With Flooding Risks

Tormod Bjelland Litlabø

Master's Thesis
Department of Informatics
and
Geophysical Institute
University of Bergen

Monday 30th May, 2022

Acknowledgements

I would like to thank my supervisor Professor Dag Haugland, for his support, guidance and tireless proofreading throughout the project.

I would like to thank SINTEF for granting me access to SHOP, and Eviny Fornybar AS for letting me access their data and optimization model regarding the Fossdal watercourse.

I would also like to thank my co-supervisors Per Aaslid (SINTEF Energy Research) and Tarjei Lid Riise (Eviny Fornybar AS), for countless meetings with great insight and support.

Abstract

Mathematical programming models have proved to be useful tools for optimizing short-term hydropower schedules. This thesis focuses on a set of applications representing a significant challenge to the performance of such models: At watercourses where a flooding risk is present, it is essential that the model recognizes the connection between overflow and water level of the reservoirs. To that end, the current work proposes a computationally efficient mixed integer programming formulation.

A traditional approach to scheduling at watercourses where floods may occur, is to apply a continuous optimization model. Owing to the non-linear nature of the interdependence between reservoir level and volume, as well as turbines described by non-linear production curves, the model is typically solved by a successive linear programming (SLP) technique. Decision variables can be introduced to ensure that the model respects that flooding occurs if, and only if, the water level exceeds the flood limit. Because discrete variables significantly increase the computational burden of the model, and thereby its solution time, it is common to define these variables on a continuous 0-1 scale. A natural and commonly observed shortcoming, is that the model then predicts overflow even when the reservoir levels are below the flood limit.

This thesis suggests an approach to avoiding such impractical solutions. While the SLP-technique is pursued, binary variables are introduced in order to ensure that flooding occurs only in instances where it makes physical sense. The goal is that the unavoidable increase in running time does not make the model impractical to use.

All binary variables occur in constraints representing relations between flooding and water level in the reservoirs. The upper bound on the volume of the

overflow appears as a "big-M" coefficient of the aforementioned variables. Precise assessment of such coefficients can have a positive effect on the solution time. Rather than setting all overflow bounds to a fixed value, identical for all time periods, they are considered as time and reservoir dependent parameters.

The overflow bounds can be estimated using the start volume and inflow for each time step and reservoir. This is done in the first iteration of the SLP-procedure. In later iterations, the volume of overflow is based on the result from the previous iteration. This is shown to give an even more precise approximation of the volume of overflow. Such estimates of the maximum overflow are further shown to agree well with the theoretical value, assessed by an optimization model with the objective to maximize overflow volumes.

The proposed method has been implemented in the short-term hydro scheduling model SHOP on the Fossdal watercourse in Western Norway, operated by Eviny Fornybar AS. The result is a reduction in solution time of the problem, from just below 1000 seconds with constant overflow bounds, and down to 11 seconds using the time dependent bounds.

It has so far not been examined whether, and to what extent, the positive results obtained at Fossdal carry over to other watercourses. However, the experimental results indicate a potential for substantial reductions in running time by rather simple means. It should therefore be investigated whether other variable and constraint modifications can yield further reductions in the running time of scheduling models for watercourses with flooding risks.

List of Figures

List of Tables

Acronyms

CPLEX IBM ILOG CPLEX Optimization Studio. v, 44, 54

DP Dynamic Programming. 7, 8

HPF Hydropower Production Function. 7–9, 29, 33, 35, 39

HRL Highest Regulated Level. 6, 11, 12, 14, 15, 24, 44

LP Linear Programming. 7, 9–11, 40, 41

LR Lagrangian Relaxation. 9

LRL Lowest Regulated Level. 24, 44

MAP Method of Approximation Programming. 9

MILP Mixed Integer Linear Programming. 8, 10, 11, 38, 40

MINLP Mixed Integer Non-Linear Programming. 8

MIP Mixed Integer Programming. 12

NLP Non-Linear Programming. 7

SHOP Short-term Hydro Optimization Program. vi, 4–6, 10–14, 18, 21, 32, 33, 38, 40, 41, 44, 45, 47, 48

SLP Successive Linear Programming. 9, 10, 33, 38, 40, 45, 50

STHS Short-term Hydro Scheduling. 2, 3, 7–9, 11

UC Unit Commitment. 38, 40

ULD Unit Load Dispatch. 38, 40

List of Symbols

Constants

C^B	Cost of using bypass gate	[NOK/MWh]
C^L	Load penalty cost	[NOK/MWh]
C^R	Reservoir penalty cost	[NOK/MWh]
C^{End}	End volume penalty cost	[NOK/MWh]
C^{GS}	Generator start up cost	[NOK/MWh]
C^{Max}	Max volume reservoir penalty cost	[NOK/MWh]
C^{Over}	Overflow reservoir penalty cost	[NOK/MWh]

Parameters

α_s	Friction loss coefficient in the production tunnel from intake reservoir r_s to power power plant s	[NOK/MWh]
$\Gamma_{t,s}$	Generator discharge cost at power plant s in time period t	[NOK/MWh]
$\sigma_{r,t}(g)$	Value of modelling parameter $M_{r,t}$	$[m^3]$
θ_r	Value of modelling parameter $M_{r,t}$	$[m^3]$
ζ_r	Value of modelling parameter $M_{r,t}$	$[m^3]$
E_t	Electricity price in time period t	[NOK/MWh]
L_s	Height at which the water outlet from power plant s is situated	$[m]$
$M_{r,t}$	Modelling parameter	$[m^3]$

Sets

A Set of edges in the directed acyclic graph $D, (i,j) \in A, \ \forall \ i,j \in N$

B Set of breakpoints of the I/O curve, $b \in B$

N Set of nodes in the directed acyclic graph D, $i \in N$

R Set of reservoirs, index $r \in R$

S Set of plants, $s \in S$

T Set of time periods, index $t \in T, T = [T_0, T_n]$

U_r Set of neighboring upstream reservoirs of reservoir r, $\bar{r} \in U_r$

State-dependent functions

$\eta_s^{Gen}(p_{s,t})$ Generator efficiency at power plant s in time period t $[\%]$

$\eta_s^{Turb}(q_{r_s,t}, h_{s,t}^{Net})$ Turbine efficiency at power plant s in time period t $[\%]$

$h_{s,t}^{Gross}$ Gross head at power plant s in time period t $[m]$

$h_{s,t}^{Net}$ Net head at power plant s in time period t $[m]$

$o(x,r)$ Flood curve for reservoir r $[m^3]$

P_s^{Max} Maximum production limit at power plant s $[MW]$

P_s^{Min} Minimum production limit at power plant s $[MW]$

$Q_{s,t}^{Max}(h_{s,t}^{Net})$ Maximum discharge limit at power plant s in time period t $\left[\frac{m^3}{s}\right]$

$Q_{s,t}^{Min}(h_{s,t}^{Net})$ Minimum discharge limit at power plant s in time period t $\left[\frac{m^3}{s}\right]$

Variables

$\delta_{r,t}$ Binary decision variable, 1 indicates overflow at reservoir r in time period t $\{0,1\}$

$\omega^S_{s,t}$	Start up variable for the generator in power plant s in time period t, 1 indicates starting of the generator	$\{0,1\}$
$\omega_{s,t}$	On/off variable for generator unit i in plant s in time period t, 1 indicates that the generator is on	$\{0,1\}$
$f_{r,t}$	Inflow to reservoir r in time period t	$[m^3]$
lp^D_t	Load penalty variable at end of period t, imbalance caused by too little load on the grid	[MW]
lp^U_t	Load penalty variable at end of period t, imbalance caused by too much load on the grid	[MW]
p^B_t	Power bought from the market in time period t	[MW]
p^S_t	Power sold to the market in time period t	[MW]
$q^B_{r,t}$	Water discharge through bypass gate at reservoir r, in time period t	$\left[\frac{m^3}{s}\right]$
$q_{b,s,t}$	Water discharge for segment b, at power plant s, in time period t	$\left[\frac{m^3}{s}\right]$
$q_{r_s,t}$	Water discharge from intake reservoir r_s, to power plant s in time period t	$\left[\frac{m^3}{s}\right]$
$s_{r,t}$	Spillage from reservoir r in time period t	$\left[\frac{m^3}{s}\right]$
x^{End}_r	Water volume at reservoir r at the end of the scheduling horizon	$[m^3]$
x^{Max}_r	Maximum permissible volume of water at reservoir r	$[m^3]$
$x_{r,t}$	Water volume of reservoir r in time period t	$[m^3]$
$xs_{r,t}$	Volume of overflow at reservoir r in time period t	$[m^3]$

$z^D_{t,r}$	Penalty variable caused by too little water in reservoir r, in time period t	$[m^3]$
$z^U_{t,r}$	Penalty variable caused by too much water in reservoir r, in time period t	$[m^3]$
ze^D_r	End penalty variable caused by too little water in reservoir r, at the end of the scheduling horizon	$[m^3]$
ze^U_r	End penalty variable caused by too much water in reservoir r, at the end of the scheduling horizon	$[m^3]$

Contents

1 Introduction

1.1 Hydropower

Ever since running water was first used to power grain mills in the Roman empire [3], hydropower has been a vital part of the evolution of human society. At the beginning of the 19th century, it played a vital part in the industrial revolution, providing mechanical energy to large machines in fabric factories. Later in the 19th century and at the beginning of the 20th century, it was first used to produce electricity, using one or more turbines to harvest kinetic energy from running water. This became an important turning point in the modernization of society, eventually leading us to where we are today.

Today, hydropower is the world's leading renewable energy source. It has dominated Norway's energy production since the early 20th century. Mainly due to the Norwegian landscape and climate, since high mountains and a great amount of precipitation are vital components for hydropower production. These conditions have made it possible for Norway to become the largest producer of hydropower in Europe, and the seventh-largest producer in the world [4].

The global demand for energy and electricity has been rapidly increasing since the 1950s, as seen in Figure 1. Since fossil fuels such as oil, coal and gas are the main energy sources used in energy production, the emission of greenhouse gases has also increased rapidly. Thus the need for clean, renewable energy has never been higher. Since hydropower is a relatively old technology, a great share of its potential around the world today is already in use. This has forced the hydropower producers to be as efficient as possible regarding production, ultimately leading to the use of optimization to develop a production schedule.

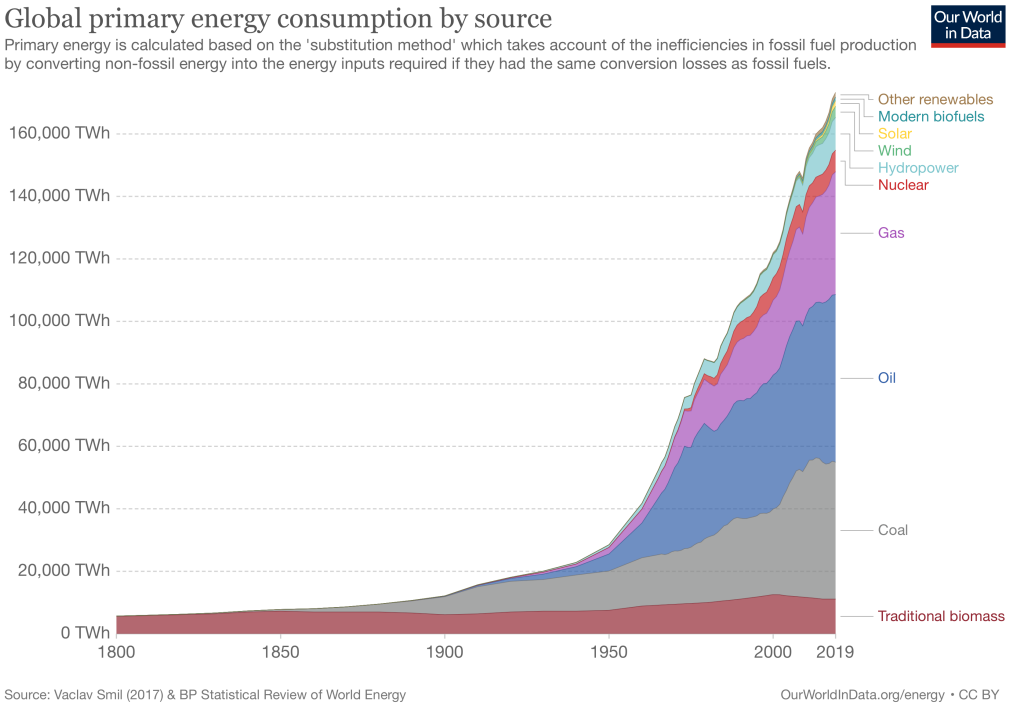

Figure 1: Global primal energy consumption, showing a clear increase in energy demand in the last 70 years [1].

1.2 Hydropower scheduling

Hydropower scheduling is divided into three main categories regarding the time horizon being investigated; short-, mid-, and long-term, explained more closely in Figure 2. They all have the same target, which is to optimize the power generation schedule of the accessible hydropower units, which generate maximum energy by utilizing the available potential during a specific period [5].

Using optimization allows power producers to schedule their production according to their specific target. Common objectives for the *Short-term Hydro*

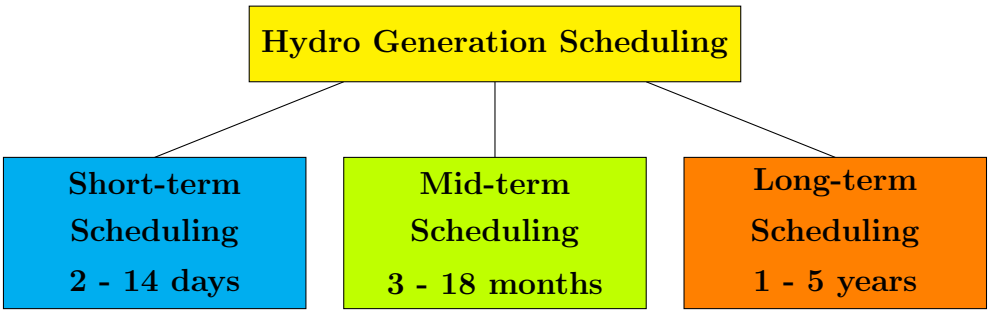

Figure 2: The different categories regarding time horizon, used in hydropower scheduling.

Scheduling problem (STHS) are to maximize the total revenue, minimize total operational costs, or minimize the value of water used or spilt. To maximize the power generated for a given flow, minimize the total discharge from the units, and maximize the efficiency of the power plant's total energy conversion, are other objectives that often are addressed [6]. The objectives are dependent on the structure of the power plant, e.g the number and size of the generator units, and the layout and size of the reservoirs, tunnels and gates. It is also dependent on the structure of the market in which the power is sold. Along with the optimization problem comes a group of various constraints regarding several physical limits connected to the water level of the reservoirs, discharge, maximum and minimum production limit etc. It also connects short-term to mid-, and long-term scheduling.

One important condition the power producers need to take into consideration when calculating a production schedule is to avoid unregulated spillage, i.e. overflow, from the reservoirs in the hydro system. This is often formulated as a constraint regarding the water level of the reservoirs and the forecasted inflow. Such constraints are described in detail in [6, 7].

1.3 Short-Term Hydro Optimization Program

Short-term Hydro Optimization Program (SHOP) is a programming tool developed by SINTEF [8]. It allows power producers around the world to schedule their production for optimal results, e.g. minimizing costs, maximizing profits, and at the same time taking care of vital constraints regarding limitations of the hydro system. Although SHOP is very well developed, it has some limitations when the risk of flooding, i.e. overflow, from the hydro system is high. These limitations will be investigated and outlined more closely, and ultimately a solution to it is developed.

1.4 Data basis

For this thesis, Eviny has provided data from the Fossdal watercourse, consisting of the power plant Fossmark and reservoirs Gråsidevatn and Fossdalsvatn. Fossmark is located in Vaksdal municipality, not far from Bergen where Eviny's headquarters are located. The power plant was built in 1917 and renovated in 1989. Rivers run from the upper reservoir, Gråsidevatn, down to the intake reservoir Fossdalsvatn situated at 440 meters above sea level. Fossmark has one Pelton turbine and a maximum production capacity of 9 MW. Approximately 43 GWh is produced at Fossmark each year.[9]

The layout of the Fossdal watercourse is shown in Figure 3. From the upper reservoir Gråsidevatn, two gates lead to the lower reservoir Fossdalsvatn. The dashed line from Gråsidevatn is a flood gate leading overflow from Gråsidevatn to Fossdalsvatn, whilst the full line represents a regulated gate which leads scheduled water release from Gråsidevatn to Fossdalsvatn. From Fossdalsvatn the main water flow goes to Fossmark through the production tunnel, whilst there is one flood gate and one bypass gate leading water directly into the sea. The data is from a specific time period in which the reservoirs are full, and the forecasted inflow for the next two weeks in-

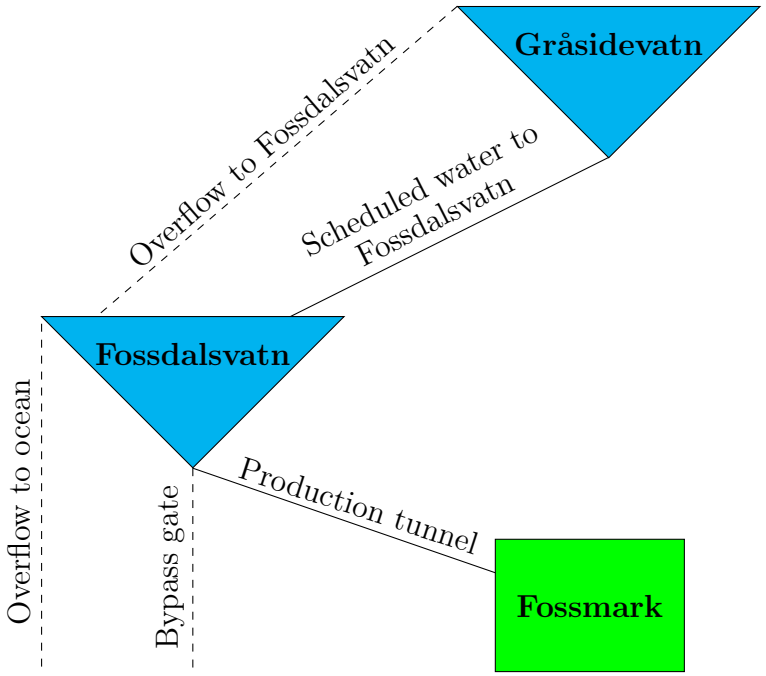

Figure 3: Layout of the Fossdal watercourse.

creases the risk of overflow. It is specifically chosen because it challenges some of SHOP's limitations regarding flooding risks to appear. The Fossdal watercourse is also quite simple, lowering the risk of unwanted complications caused by a too complex hydro system.

1.5 Unphysical overflow

The mentioned limitation of SHOP regarding flooding risks, is the occurrence of unphysical overflow, which is scheduled overflow when the water level is below the upper limit of the reservoir, making it physically impossible for overflow to occur. If the forecast of inflow indicates that flooding from the hydro system is highly likely, or even inevitable, SHOP will schedule flooding at a convenient time. This is dependent on the chosen objective which, as mentioned previously, often is to maximize the total revenue, or to minimize

the total operational costs [6]. The convenient time could correspond to the time when producing power is most profitable, i.e. when the electricity price is at its highest. It could also be the time just before the downstream reservoir runs empty, and a lot of water from an upstream reservoir is needed, which can be seen in Figure 4. Since this does not always happen at a time where the reservoirs are overfilled, with water physically flowing through the flood gate, this phenomenon is referred to as unphysical overflow.

Different methods are available to use to avoid unphysical overflow. SHOP provides an option to use binary decision variables corresponding to overflow from each reservoir r and in each period t. With these decision variables active, SHOP will only tolerate flooding from the system if the water level of reservoir r, in period t, is above the *Highest Regulated Level* (HRL). In other words, SHOP now only tolerates overflow from the hydro system if the water is physically flooding from the reservoir. Usage of these binary variables and how they impact the results are discussed further in Section 3.2.

2 Background

2.1 Optimization methods used in hydropower

Hydropower optimization is challenging. It is mainly due to decision variables being coupled in time, meaning that the optimization problem includes state variables. Reservoir level and climate-dependent variables, such as inflow, are examples of such state variables. Thus the full-dimensional optimization problem is decomposed into sub-problems, as mentioned in Section 1.2, each solved by different solution techniques [10]. Since this thesis will concentrate on the short-term sub-problem, solution methods for the long- and medium-term will not be a subject of discussion.

Short-term Hydro Scheduling is not at all a simple task. The earlier mathematical formulations of the problem were widely based on *Linear Programming* (LP) [6, 11, 12]. However the handling of non-linear and non-convex elements, together with state-dependencies, is a real challenge.

Such non-linear and non-convex elements are present at the very core of the STHS problem, namely the *Hydropower Production Function* (HPF). It is a function that models the relationship between water discharge as input and produced power as output. It consists of several complex elements, such as the net head of the reservoir, turbine efficiency, and generator efficiency. The HPF will be addressed more in detail in Section 4.1.6.

LP does not fully represent these physical characteristics of the hydro generation, leading to the use of other optimization methods [13]. *Non-Linear Programming* (NLP) can handle non-linearity and non-convexity well, and is often combined with *Dynamic Programming* (DP) algorithm to solve specific types of hydropower scheduling problems. For the case of a single hydropower plant with few units and low capacity, such as Fossmark, DP has been one of the most popular optimization techniques used to solve the STHS problem.

However, as the size and installed capacity of a plant increases, it becomes harder to apply DP to solve the problem [6, 14].

The recent development of computational and mathematical techniques allows for more details of the problem to be assessed. The discrete nature of the problem, e.g. the on/off condition of the units, is too important to be ignored. Such features are modelled by introducing binary integer variables to the mathematical formulation. If piecewise linear functions are used to approximate the non-linearity of the HPF, e.g. turbine efficiency, and used together with the binary integer variables, the STHS problem is modeled as a *Mixed Integer Linear Programming* (MILP) problem [6]. J. Kong et al. refer to several relevant articles regarding MILP-modelling of the STHS problem.

Adding constraints to the optimization problem and effectively finding a solution are vital features for optimization. These are features that MILP has good performance with, and this is one of the reasons for it being widely used to solve large scale STHS problems [6, 15].

The downside of using MILP to solve the STHS problem is the high demand for computational capacity, which also increases with the scale of the STHS problem. It is illustrated in Section 3.2. Linearizing non-linear functions can affect the values calculated by the MILP model, resulting in deviations between these and the real values, which can ultimately lead to an infeasible solution [6].

Mixed Integer Non-Linear Programming (MINLP) is another optimization technique that can be used to solve the STHS problem. It is preferred when the STHS problem is modelled with more details. The MINLP problem relies on the hydropower plants to not be too large for it to be solvable. [6]

Decomposing STHS problems of larger scale into a set of smaller subproblems is an important part of optimization. These subproblems are mathematical

solvable and demand much less computational power to be solved. One of the most popular and successful decomposition methods is *Lagrangian Relaxation* (LR). The LR decomposition technique is divided into three steps: (1) dualization, (2) solving the dual subproblem, and (3) finding a feasible optimal or near-optimal primal solution [6].

Again the non-linear and non-convex HPF is the complicating factor. Although the LR method can relax some complex linking constraints, the HPF makes it very difficult to obtain the real dual function. This and the fact that the process of finding a primal feasible solution often is based on heuristics are the main problems with the LR method [6].

As mentioned LP has been a popular approach to the STHS problem, due to its simplicity and efficiency. The issue of non-linear and non-convex elements needs to be dealt with to use LP successfully. To tackle this, *Successive Linear Programming* (SLP) can be applied, as in [10].
SLP was first introduced in [16], where it is referred to as *Method of Approximation Programming* (MAP). Here it is shown how a non-linear problem can be linearized by expansion as a Taylor's series, and ignoring the terms of a higher order than linear. This new linear problem can now be solved by an LP algorithm, which is repeated until an optimal or near-optimal solution is found. The algorithm uses the previous solution as input and builds on this for the new solution.

In [10] a linear model for STHS, which is easily solved by LP, is provided. Then a solution to the issue of accounting for important non-linear elements, e.g. the efficiencies regarding head of water and energy per m^3 of utilized water, is given. It is done by formulating the non-linear term as a first-order approximation and updating it within the iterative process of solving the linear model. Thus a new SLP model of the problem, where the efficiencies are added as a linear term in the objective function, is given.

2.2 SHOP

As mentioned in Section 1.3, SHOP is a programming tool for developing short-term hydro schedules. It was developed by SINTEF's Energy Research group in 1989, with financial support from the state-owned power and grid company Statkraftverkene. The first operational version was finished in 1996. SHOP was used operationally for the first time by the Norwegian company Statkraft in 2003. Since then, SHOP has been redesigned in 2008 and again in 2016, to make sure the solution methods are state of the art [17].

The optimization problem in SHOP was previously modelled at plant level to reduce the size of the problem, thus reducing the computational time. The unit-based model, i.e. using binary integer variables to model which generator unit within a power plant is active, was only implemented in certain cases [17, 18]. With the development of computational capability and the need for even more precise calculations, the implementation of unit-based modelling occurred more and more frequently. It ultimately caused a switch of the default optimization model in SHOP, from plant-based to unit-based in 2016 [17].

In the same way that SHOP has been developed over the years regarding plant- and unit-based modelling, so have the optimization methods which are in use. Since the earlier years, SLP has been the main optimization method used in SHOP [10, 19], which means that the problem is formulated and solved as an LP-problem, within each iteration of an SLP-loop. This was mainly because of computational limitations at the time. However, as more and more of the computational limitations vanished, the use of MILP within the SLP-iterations has increased [17]. Using LP within the SLP-loop is still the default optimization method in SHOP, but the option to formulate the problem as an MILP-problem is available. The reason why LP is preferred is that for most cases, it solves the problem easily, and it is typically faster

than MILP. However, for some cases of e.g. discrete nature, LP does not perform that well, and MILP becomes the preferred optimization method. An example of such a case is the handling of unphysical overflow, which LP is sensitive to, whilst MILP handles this nicely at the expense of a possible increase in the computational time.

2.3 Modelling overflow

Modelling the overflow from a hydro system is an important part of STHS problems. Overflow from a reservoir in a hydro system can have many different consequences. Wildlife and nearby society can be affected, but also the hydro system itself and its owners. Some of these consequences are more serious than others, nonetheless, it is important to consider them when scheduling the production for a hydro system.

Overflow is not necessarily bad in every case. If a hydro system contains several reservoirs, and discharge from an upper reservoir is a direct source of inflow to the downstream power plant, then overflow from the upper reservoir can be useful for the production at the power plant. A typical example of this situation is when an upper reservoir (or power plant) acts like a bottleneck, e.g. due to maintenance or a low maximal flow capability. If the electricity price is high, one would want to use the water stored in the upper reservoir, now only accessible by using the flood gate.

However, special cases like this can cause new problems to occur. The previous case is also a good example of why unphysical overflow can occur when scheduling the hydropower production. This sometimes happens in SHOP. Within each SLP-iteration in SHOP, modelling of overflow can be done with either LP, or MILP, or both. LP is sensitive to unphysical overflow since the optimization model contains continuous decision variables regarding if the water level of a reservoir is higher than the HRL. Modelling with MILP does solve the problem of unphysical overflow completely, but with the cost of a

high and perhaps unsatisfactory solution time.

Unphysical overflow is divided into three categories in SHOP [7]:

- **Unphysical overflow**: flood gate is used when the reservoir is not full.

- **Partially unphysical overflow**: If the situation and the settings of the system indicate overflow during the time period, it might be beneficial to advance the overflow. Then one would have both unphysical and physical overflow.

- **Model forced unphysical overflow**: If one, in the iterative process, goes from a full model without a plant description based on MIP, to an incremental model with a MIP-based description of the plant, mathematical inconsistencies can occur. These inconsistencies can then be "solved" in SHOP by unphysical overflow.

A great example of how the first category of unphysical overflow often occurs is:

If the electricity price is high enough, the penalty for letting water flood to a lower plant won't be enough to stop the scheduling problem from getting a profit. The penalty also only concerns the actual overflow from the reservoir, not if it is possible for the water to flood, i.e. if the water level is higher than the HRL. It makes it possible for SHOP to schedule overflow at the most beneficial time, often resulting in unphysical overflow.

3 Experimental study of the current possibilities in SHOP

This chapter gives an overview of the current possibilities in SHOP, regarding how unphysical overflow from the hydro system is handled.

All the scenarios are solved with the same settings and data, e.g. start volume and inflow for each reservoir, electricity prices etc. The tunnel from Gråsidevatn to Fossdalsvatn mentioned in Section 1.4, is closed during the entire planning horizon. This essentially means that one does not wish to spend water that is stored in Gråsidevatn, during the planning horizon. One would rather like to save the water for a later time. The different penalty costs that apply are listed in Table 1. The optimization model used in SHOP,

Penalty cost	**Value [NOK/MWh]**
Overflow	50000
Load	5000
Reservoir endpoint	1000
Reservoir ramping	100
Discharge	100

Table 1: Penalty costs with numerical values, used in SHOP.

which is presented in Chapter 4, is the same for both scenarios. Only a small adjustment is done to the model solved in Scenario 1. It is the relaxation of the binary decision variables, which is 0 if there is no overflow at reservoir r in period t, and it is 1 if there is overflow. For Scenario 1 it is relaxed to be continuous between 0 and 1.

3.1 Scenario 1 - using continuous decision variables

SHOP has a relatively large penalty cost regarding overflow from a reservoir. This is naturally linked with a variable for amount of overflow from a reservoir. In this scenario, the decision variable regarding overflow at reservoir r in period t, is as mentioned continuous between 0 and 1. This usually works very well, and is the preferred model in many practical applications of SHOP. However, when the hydro system is nearly full, and the forecasted inflow suggests flooding, a problem occurs. This type of situation is well illustrated by the data from Eviny's watercourse Fossdal, for the period of two weeks between the 10^{th} of August 2020 and the 24^{th} of August 2020.

When solving the problem with this data set in SHOP, we clearly see that the problem of unphysical overflow occurs. In Figure 4 we see that around 22^{nd} of August, a great amount of water is overflowing from Gråsidevatn to Fossdalsvatn. This would suggest that the head of Gråsidevatn is increasing rapidly just before the time of the overflow. However, when looking at Figure 5, it is clear that the head of Gråsidevatn does not increase rapidly. It is actually below the HRL at the time of this spike of overflow from the hydro system. The overflow in these periods is thus categorized as unphysical overflow.

This kind of result is only made possible by the aforementioned continuous decision variable regarding overflow. It makes it possible to schedule overflow, in periods when the water level of the reservoir is below the HRL. This typically happens at reservoirs with a low discharge capacity. In our experiment, Figure 5 suggests that Fossdalsvatn is about to use up all its stored water, before the end of the planning horizon, possibly missing out on producing power at a time with good electricity prices. To avoid this, and to keep being able to produce at near maximum capacity, SHOP schedules a great amount of overflow from Gråsidevatn, even though the water level

of the reservoir is eventually below HRL. This has fixed the issue regarding Fossdalsvatn possibly running out of water, but it has introduced a new problem, namely unphysical overflow.

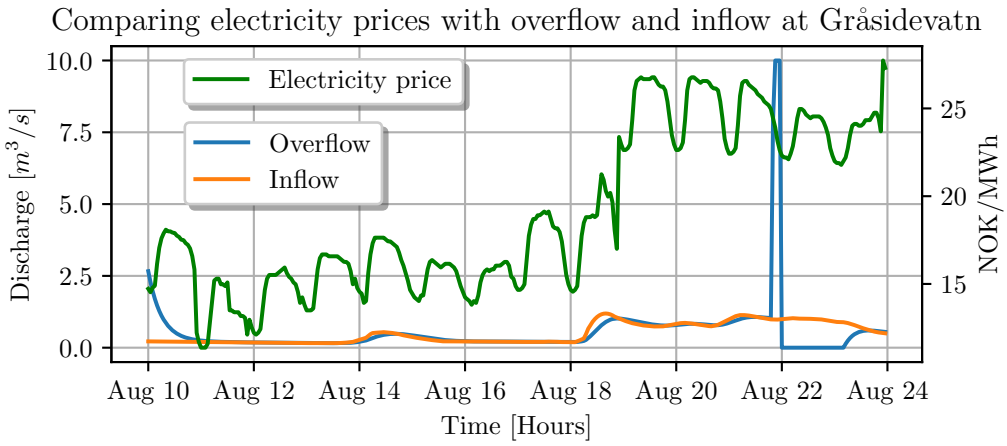

Figure 4: Overflow and inflow at Gråsidevatn, compared with the electricity price.

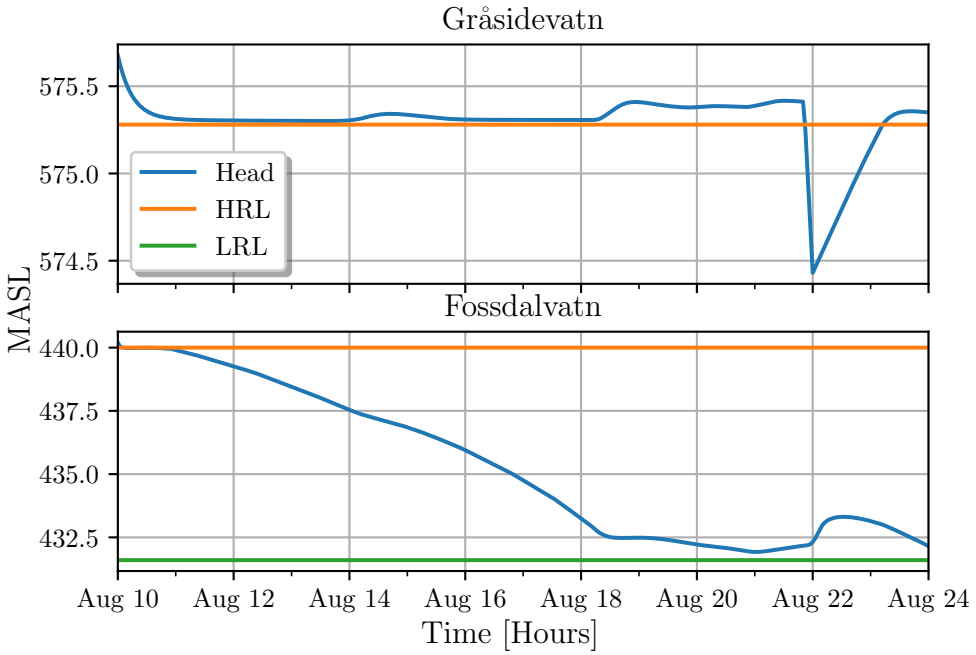

Figure 5: Head of the reservoirs for the whole planning horizon.

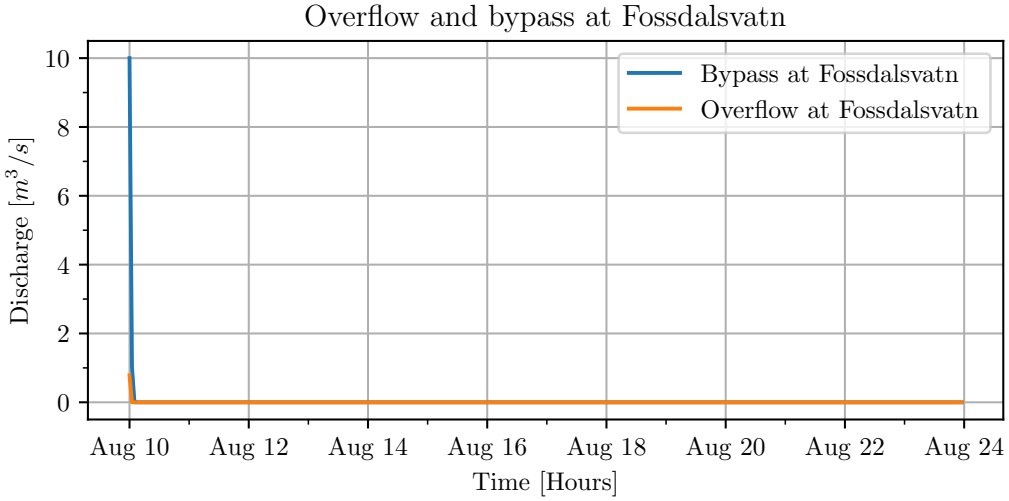

Figure 6: Overflow and bypass at Fossdalsvatn. This runs directly into the ocean.

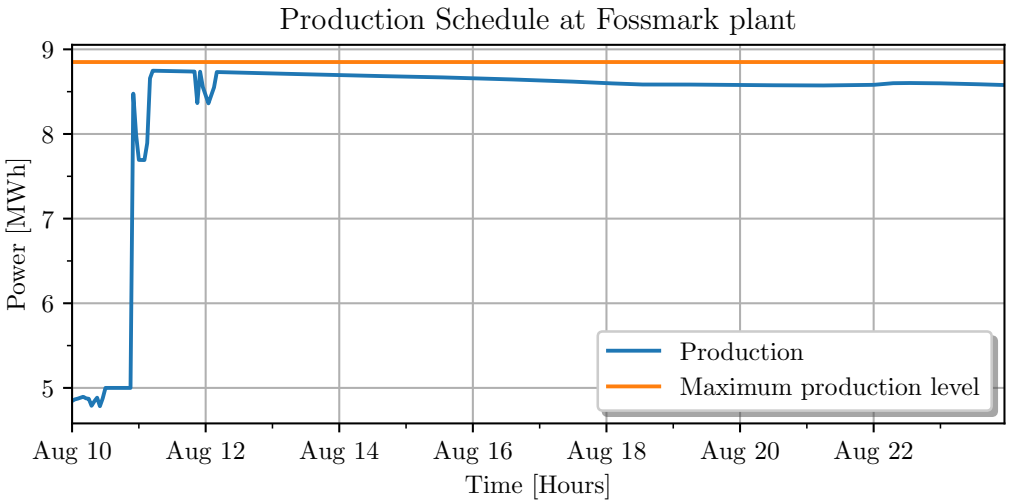

Figure 7: Production at Fossmark power plant.

3.2 Scenario 2 - using binary decision variables

In Scenario 2, the aforementioned continuous decision variables, regarding overflow are replaced by binary decision variables. Thus, the model in SHOP now only allows physical overflow from the hydro system. In terms of finding a solution that is realistic and without any non-physicalities, this works great. However, the computational time of the solver becomes exponential in the number of binary decision variables, in the worst case. The model in SHOP has one binary decision variable, for each reservoir and in each time period. The total number of binary decision variables for our experiment with two reservoirs, namely Gråsidevatn and Fossdalsvatn, a planning horizon of 14 days and a period length of one hour, is 672. Naturally, this increases the solution time dramatically, which is the main problem of this scenario. Figures 9–11 display the results obtained by solving the scheduling problem in SHOP, using the model of Scenario 2.

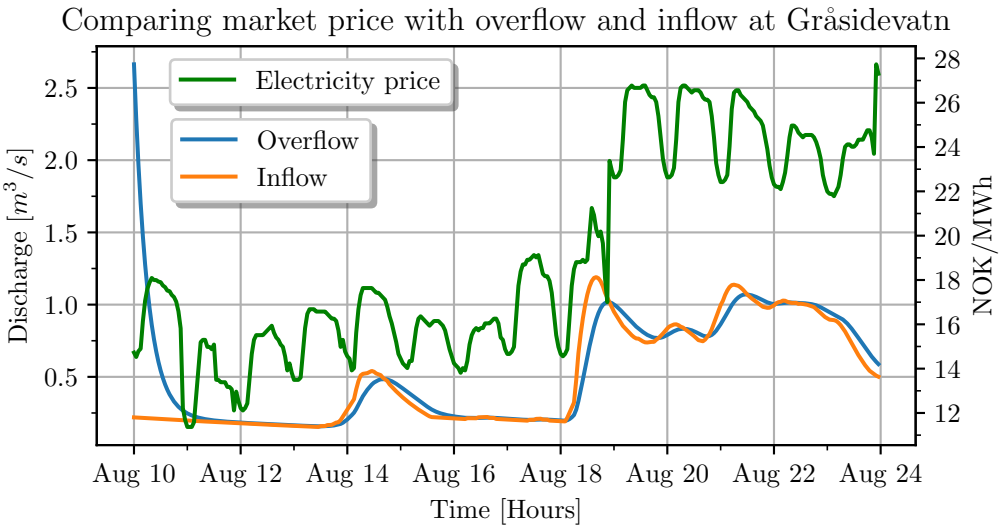

Figure 8: Overflow from Gråsidevatn to Fossdalsvatn, compared to inflow at Gråsidevatn.

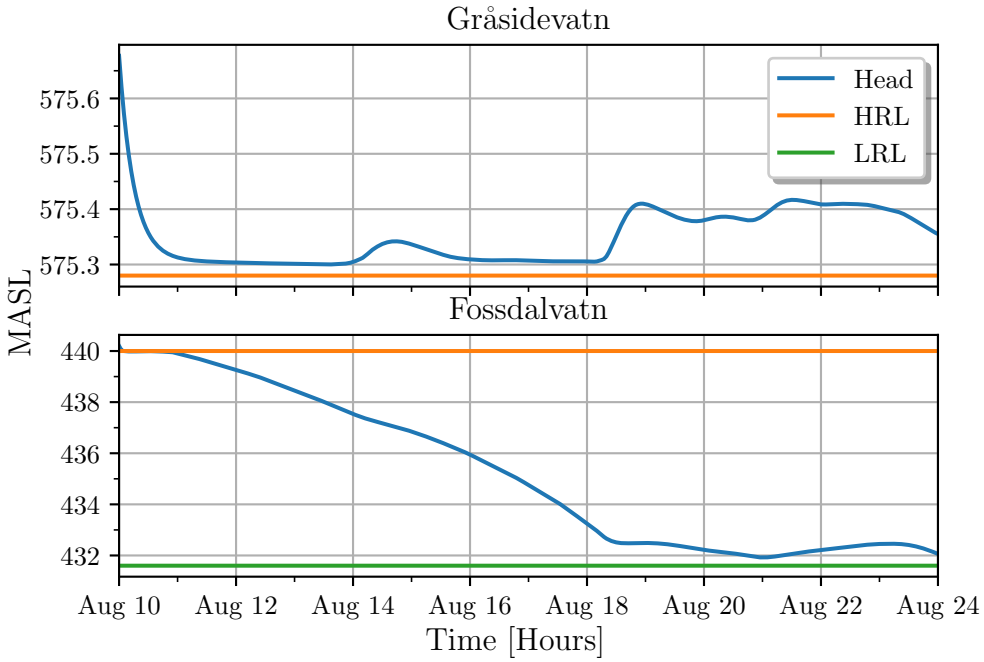

Figure 9: Head of the reservoirs.

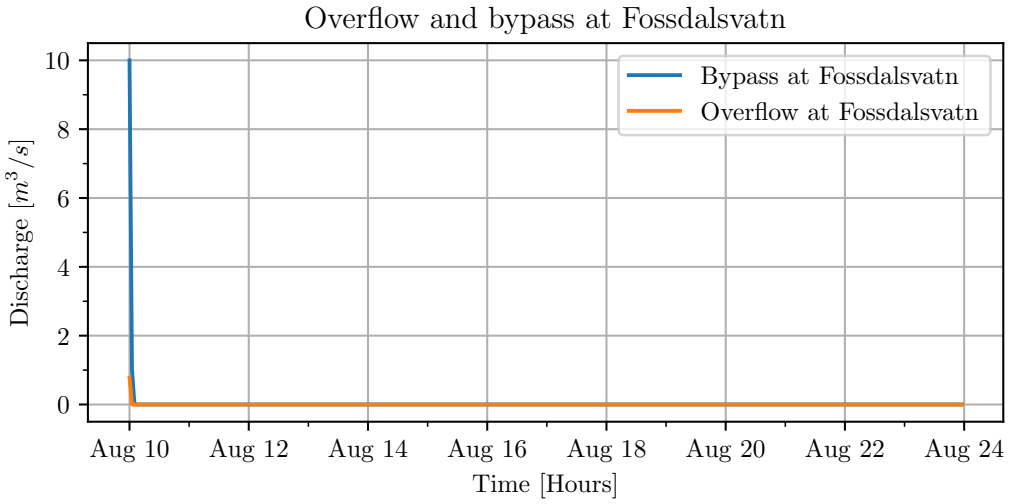

Figure 10: Flood gate and bypass gate from Fossdalsvatn.

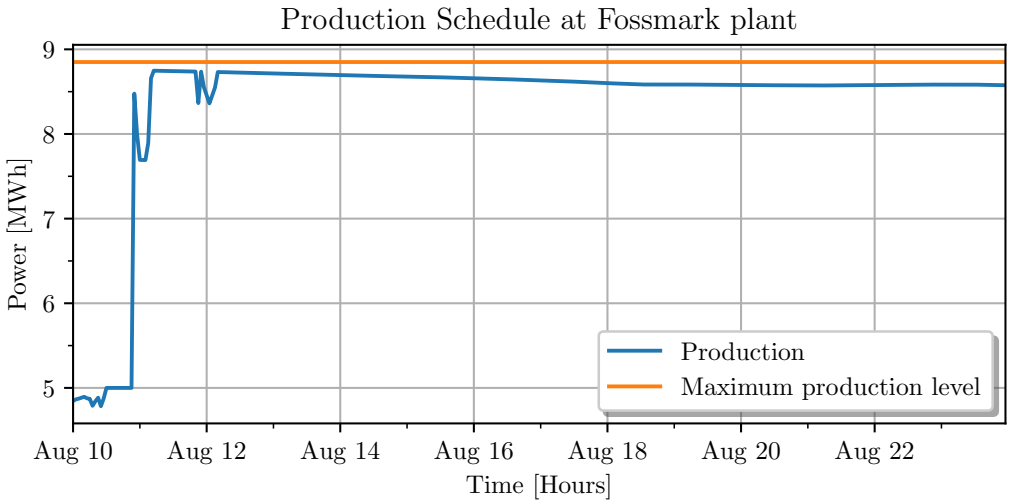

Figure 11: Production at Fossmark.

4 Optimization methods

SHOP provides several customization options of the optimization model, to best fit each specific optimization problem. The objective function, constraints, variables and parameters can all be customized. This chapter presents the standard model in SHOP, and provides suggestions that might have a positive effect on the problem regarding unphysical overflow.

4.1 Mathematical model

The type of optimization problem that is addressed here is often related to low electricity prices and full or nearly full reservoirs. This typically implies that one would like to minimize the "damage" of producing hydropower, which might not be profitable. The objective would then be to either minimize the total operational costs, or to minimize the value of water that is used or spilled. The hydro system can be defined as a directed acyclic graph, $D = (N, A)$. Where the set N consists of two types of nodes, namely reservoir $r \in R$ and power plant $s \in S$. The set N can be defined as $N = R \cup S$. The set of edges A is defined as $A \subseteq N \times N$.

Figure 12 shows an example of a hydro system, as a directed acyclic graph.

Here, $r_1, r_2, r_3, r_4, r_5, s_1, s_2 \in N$, where $s_1, s_2 \in S$ and $r_1, ..., r_5 \in R$. The hydro system is connected by edges, and it is only possible to have discharge from a reservoir node r to node j, if the edge $(r, j) \in A$.

For each reservoir node r, we define the set U_r as the set of all neighboring, upstream reservoir nodes of r. An example from Figure 12 is that for reservoir node r_4, the set $U_{r_4} = \{r_2, r_3\}$, whilst the set for reservoir node r_1 is $U_{r_1} = \emptyset$.

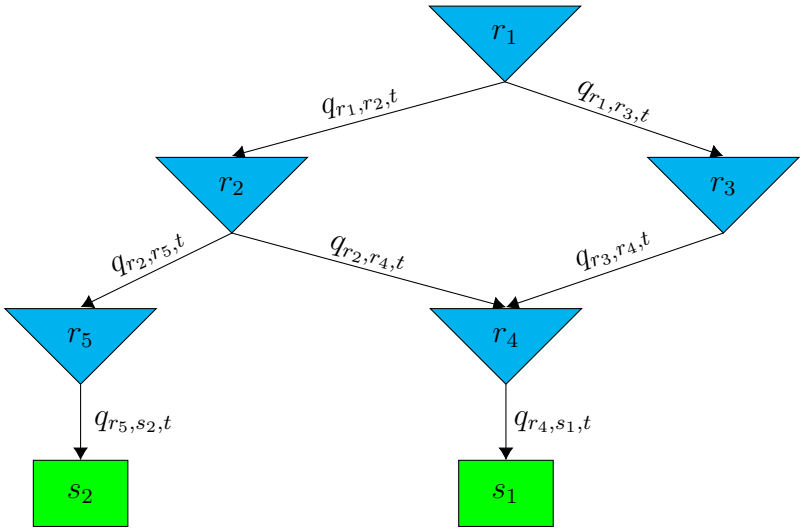

Figure 12: Example hydro system as a directed acyclic graph

We assume that each power plant node s is connected to exactly one upstream reservoir r, which is referred to as intake reservoir r_s. An example of this can be seen in Figure 12, where the reservoirs r_4 and r_5 are the intake reservoirs for power plants s_1 and s_2, respectively.

Scheduled discharge from reservoir r to node j, at time period t, is given by $q_{r,j,t}$. However as shown in Figure 12, discharge can be scheduled from one reservoir node, to multiple other nodes. The total discharge $q_{r,t}$ out of reservoir r, in period t, is thus given by

$$q_{r,t} = \sum_{j \in N:(r,j) \in A} q_{r,j,t}, \qquad \forall \ r \in R, t \in T. \tag{1}$$

Equation (1) holds for all nodes r that is a reservoir, i.e. $r \in R \in N$. The total discharge going from intake reservoir r_s, into power plant s, at time period t is given by $q_{r_s,t}$.

Figure 3, presented in Section 1.4, shows an example of D, for the Fossdal wa-

tercourse. It consists of three nodes, namely the two reservoirs Gråsidevatn and Fossdalsvatn, and the power plant Fossmark. Here, Fossdalsvatn is the intake reservoir that is directly connected to the power plant Fossmark.

4.1.1 Objective function

The objective of our model is to minimize the total cost of the hydropower production, while fulfilling several constraints. The objective function minimizes a weighted sum of several penalty costs, as well as the amount of power bought from the grid, the amount of power sold to the grid, the value of water stored at the end of the planning horizon. The objective function reads

$$\begin{aligned}\textbf{Minimize } \sum_{t\in T}\Bigg(\sum_{r\in R}\Big(&C^{Over}\cdot s_{r,t}+C^{B}\cdot q^{B}_{r,t}+E_t\left(p^{B}_{t}-p^{S}_{t}\right)\\ &+C^{R}\left(z^{U}_{r,t}+z^{D}_{r,t}\right)+C^{L}\left(lp^{U}_{t}+lp^{D}_{t}\right)\Big)\\ &+\sum_{s\in S}\Big(C^{GS}\cdot\omega^{S}_{s,t}+\Gamma_{s,t}\cdot q_{r_s,t}\Big)\Bigg)\\ -\sum_{r\in R}\Big(&C^{End}x^{End}_{r}+C^{R}\left(ze^{U}_{r}+ze^{D}_{r}\right)\Big).\end{aligned} \tag{2}$$

A number of penalty costs are present in the objective function. If not otherwise mentioned, the different penalty costs are equal for all reservoirs.

Penalty cost C^{Over}, penalizes overflow from reservoir r, i.e. if the flood gate at reservoir r is used. This is linked with $s_{r,t}$ which is the amount of spillage from reservoir r, i.e. the amount of water going through the flood gate at reservoir r.

The cost of using the bypass gate C^{B}, at reservoir r, is naturally linked with the amount of discharge through the bypass gate, namely $q^{B}_{r,t}$.

The amount of power bought or sold in time period t is represented, respectively, by p_t^B and p_t^S. Each of them is linked to the electricity price, E_t, for the same period t.

One of the largest valued penalty cost is C^R, and it is connected to $z_{r,t}^D$ and $z_{r,t}^U$ which are penalty variables regarding violation of LRL and HRL, respectively. These variables are used to ensure that the problem is solvable, even though it might be infeasible in practice.
An example of this can be that if power production is scheduled at a time t when there is not enough water in reservoir r. Then $z_{r,t}^D$ will be the volume of water that is needed to make the production schedule feasible. The reservoir penalty cost C^R, is also connected to ze_r^U and ze_r^D. These are reservoir penalty variables regarding a violation at the end of the planning horizon.

Avoiding imbalance in the power grid is really important for power producers, and C^L is a penalty cost regarding this. It is connected to lp_t^U and lp_t^D, which are penalty variables regarding overload and underload of the power grid, respectively.

The cost of starting up the generator is given by C^{GS}, it is connected to the binary decision variable $\omega_{s,t}^S$, which is 0 if the generator in power plant s has not been started in time period t, and 1 if it has been started.

The cost of discharge into power plant s, in time period t, is given by $\Gamma_{s,t}$. It is linked with the total discharge $q_{r_s,t}$, from intake reservoir r_s, into the generator at power plant s, in period t.

The value of water at the end of the planning horizon is given by C^{End}. It is linked with x_r^{End} which is the amount of water in reservoir r, at the end of the planning horizon.

4.1.2 Water balance of the reservoirs

While minimizing the objective function, the optimization model also has several constraints to fulfill. The constraints

$$x_{r,t} = x_{r,t-1} - s_{r,t} + \sum_{\bar{r} \in U_r} \sum_{u=1}^{t-1} \beta_{\bar{r},r,u} \cdot s_{\bar{r},u} - q_{r,t} \\ - q^B_{r,t} + f_{r,t} + z^U_{r,t} - z^D_{r,t} - \left(z^U_{r,t-1} - z^D_{r,t-1}\right), \qquad \forall\, r \in R, t \in T \qquad (3)$$

and

$$x_{r,k} = x^{End}_r + ze^U_r - ze^D_r - z^U_{r,k} + z^D_{r,k}, \qquad \forall\, r \in R, k = T_n \qquad (4)$$

bring water balance of the reservoirs into the model.

Constraint (3) describes how the volume $x_{r,t}$, of reservoir r at the end of time step t, is calculated. In short it equals the volume $x_{r,t-1}$, of the reservoir at the previous time step, plus the total inflow $f_{r,t} + \sum_{\bar{r} \in U_r} \sum_{u=1}^{t-1} \beta_{\bar{r},r,u} \cdot s_{\bar{r},u}$, subtracted by the total volume $s_{r,t} + q_{r,t} + q^B_{r,t}$, of water going out. In addition to this the penalty variables $z^U_{r,t} - z^D_{r,t} - \left(z^U_{r,t-1} - z^D_{r,t-1}\right)$, are added. This is done to secure an artificial water balance, thus keeping the problem feasible in theory, for it to remain easily solvable. If the water in reservoir r is not in balance at period t, then $z^U_{r,t} - z^D_{r,t} \neq 0$, where $z^U_{r,t} > 0$ indicates a shortage or water and $z^D_{r,t} > 0$ indicates too much water. If there was an imbalance in reservoir r in the previous time period $t-1$, then $-(z^U_{r,t-1} - z^D_{r,t-1}) \neq 0$. This is added to the constraint to make sure that a potential imbalance at the previous time period is made up for.

In other words, $x_{r,t}$ relies heavily on $x_{r,t-1}$, as well as the total water discharge $q_{r,t}$, and spillage $s_{r,t}$, from reservoir r in time period t. It also relies on natural inflow $f_{r,t}$, at reservoir r in time period t, as well as spillage $s_{\bar{r},u}$ from the upstream reservoir $\bar{r}$, at time period u. The parameter $\beta_{\bar{r},r,u}$ equals the proportion of the spillage from reservoir $\bar{r}$ that u time periods later flows

into reservoir r.

Example values of $\beta_{\bar{r},r,u}$ is shown in Figure 13. Here it becomes clear that first after 2 to 3 hours the largest amount of spillage arrives. After 5 hours, about 55% of the total spillage has arrived.

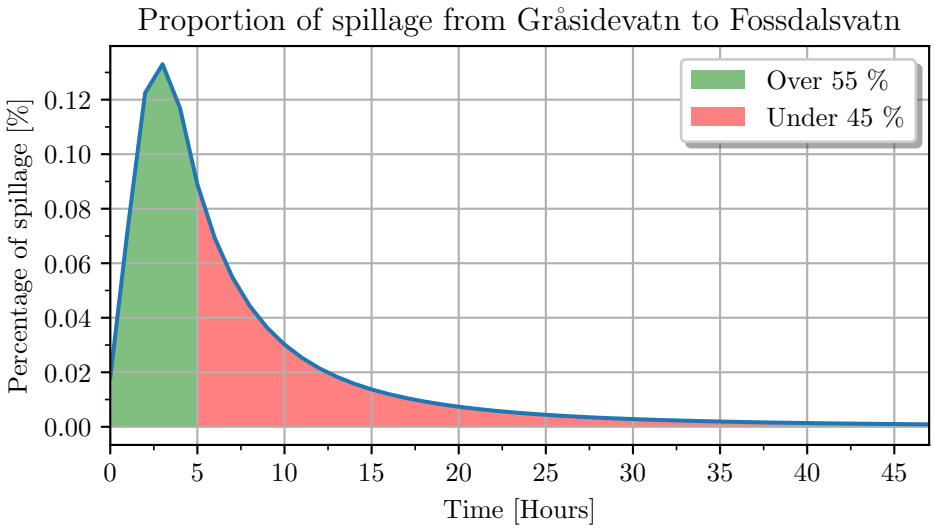

Figure 13: Proportion of spillage from Gråsidevatn to Fossdalsvatn.

Constraint (4) is a final state condition that says that water volume $x_{r,k}$, of reservoir r at the last time step k of the scheduling horizon, needs to be equal to the end volume of the reservoir x_r^{End}.

4.1.3 Modelling overflow from the reservoirs

Description of overflow, and how this is related to e.g. volume of reservoir is described by the following constraints

$$xs_{r,t} \geq x_{r,t} - x_r^{Max}, \qquad \forall\, r \in R, t \in T, \tag{5}$$

$$xs_{r,t} \leq x_{r,t} - x_r^{Max} \cdot \delta_{r,t}, \qquad \forall\, r \in R, t \in T, \tag{6}$$

$$xs_{r,t} \leq M_{r,t} \cdot \delta_{r,t}, \qquad \forall\, r \in R, t \in T, \tag{7}$$

$$s_{r,t} = c_r \cdot xs_{r,t}, \qquad \forall\, r \in R, t \in T. \tag{8}$$

Constraint (5) defines the lower bound of the volume of overflow $xs_{r,t}$, to be greater than or equal to the difference between the volume $x_{r,t}$, in reservoir r at time period t, and the maximum volume x_r^{Max}, in reservoir r. Constraints (6) and (7) defines the upper bounds of $xs_{r,t}$. The binary decision variable $\delta_{r,t}$, is 1 if there is overflow at reservoir r, at time period t, and 0 if there is no overflow from the reservoir. This variable is used in both constraints (6) and (7), and in constraint (6) it makes sure that if there in fact is overflow at the reservoir, then the volume of overflow $xs_{r,t}$, is defined as less than or equal to difference between the volume $x_{r,t}$, in reservoir r at time period t, and the maximum water volume x_r^{Max}, in reservoir r.

It is now clear that when $\delta_{r,t} = 1$, i.e. there is overflow at reservoir r, in time period t, constraints (5) and (6) defines the volume of overflow $xs_{r,t}$, as less than or equal to, and greater than or equal to the difference between the volume $x_{r,t}$ and the maximum volume x_r^{Max}, in reservoir r and at time period t, respectively. The volume of overflow $xs_{r,t}$, can thus be defined as this volume difference.

Constraint (7) ensures that the volume of overflow $xs_{r,t}$, is zero when there is no overflow at reservoir r in period t, i.e. when $\delta_{r,t} = 0$. Further discussion of the definition of the modelling parameter $M_{r,t}$, and its importance on the

solution time of the problem, will occur later in Section 4.4. Constraint (8) defines that the flow of spilled water $s_{r,t}$, going through the flood gate at reservoir r at time period t, is the volume of overflow $xs_{r,t}$ multiplied with a parameter c_r. This parameter converts water volume into water flow. It is reservoir dependent since different structures of the reservoir dam, results in different flow capacities.

4.1.4 Head variation and friction loss

Head is a typical hydropower-term, and it is not very intuitively defined. Nonetheless it is important because of its impact on the amount of power that a power plant can produce. Head is divided into two categories; gross head and net head. The two following constraints define these two categories, respectively

$$h_{s,t}^{Gross} = l_{r_s,t-1}(x_{r_s,t-1}) - L_s, \qquad \forall\ s \in S, t \in T, \tag{9}$$

$$h_{s,t}^{Net} = h_{s,t}^{Gross} - \alpha_s \cdot q_{r_s,t}^2, \qquad \forall\ s \in S, t \in T. \tag{10}$$

Constraint (9) defines gross head $h_{s,t}^{Gross}$, as the height difference between the water level $l_{r_s,t-1}(x_{r_s,t-1})$, at the intake reservoir r_s, and the height L_s, at which the water outlet from power plant s is situated. This difference is the height which the water, used in production, needs to fall in order to reach the turbine.

The net head $h_{s,t}^{Net}$, is defined by constraint (10) as the mentioned height difference, i.e. $h_{s,t}^{Gross}$, subtracted any friction loss $\alpha_s \cdot q_{r_s,t}^2$, that occurs within the production tunnel leading water from intake reservoir r_s, to power plant s [2].

4.1.5 Water discharge and plant limitation

The limitations of the total water discharge $q_{r_s,t}$, from intake reservoir r_s, into power plant s are described by

$$q_{r_s,t} \geq Q_{s,t}^{Min}\left(h_{s,t}^{Net}\right) \cdot \omega_{s,t}, \qquad \forall\ s \in S, t \in T, \tag{11}$$

$$q_{r_s,t} \leq Q_{s,t}^{Max}\left(h_{s,t}^{Net}\right) \cdot \omega_{s,t}, \qquad \forall\ s \in S, t \in T. \tag{12}$$

Constraint (11) states that if the generator at power plant s is active in time period t, then $q_{r_s,t}$ needs to be greater or equal to the lower bound, $Q_{s,t}^{Min}\left(h_{s,t}^{Net}\right)$. Simultaneously, constraint (12) states that $q_{r_s,t}$ needs to be less than or equal to $Q_{s,t}^{Max}\left(h_{s,t}^{Net}\right)$. These bounds are dependent of the net head, $h_{s,t}^{Net}$. The minimum total discharge $Q_{s,t}^{Min}$, is the minimum amount of discharge needed at power plant s, in time period t, for the generator to be activated, i.e. $\omega_{s,t} = 1$. The maximum total discharge $Q_{s,t}^{Max}$, is the maximum permissible amount of discharge at power plant s, in time period t.

4.1.6 Power production and generator status

The hydropower production function (HPF), which was first mentioned in Section 2.1, is defined as

$$p_{s,t} = G \cdot \eta_s^{Gen}\left(p_{s,t}\right) \cdot \eta_s^{Turb}\left(q_{r_s,t}, h_{s,t}^{Net}\right) \cdot h_{s,t}^{Net} \cdot q_{r_s,t}, \quad \forall\ s \in S, t \in T. \tag{13}$$

It is a complex state-dependent, non-linear and non-convex function [6]. The generator efficiency $\eta_s^{Gen}\left(p_{s,t}\right)$, is a function of the production $p_{s,t}$. The turbine efficiency $\eta_s^{Turb}\left(q_{r_s,t}, h_{s,t}^{Net}\right)$, is a function of both the discharge $q_{r_s,t}$, and the net head $h_{s,t}^{Net}$. The conversion constant G includes both gravity acceleration and water density, making the unit conversion from $[m]$ and $[m^3/s]$ to $[MW]$. The value of G is $9.81 \cdot 10^{-3} kgm^2/s^2$.

The boundaries on power production, power balance, and the status of the

generator are described, respectively, by the following constraints

$$P_s^{Min} \cdot \omega_{s,t} \leq p_{s,t} \leq P_s^{Max} \cdot \omega_{s,t}, \qquad \forall\ s \in S, t \in T, \tag{14}$$

$$p_{s,t} = p_t^S - p_t^B + lp_t^U - lp_t^D, \qquad \forall\ s \in S, t \in T, \tag{15}$$

$$\omega_{s,t} \leq \omega_{s,t-1} + \omega_{s,t}^S, \qquad \forall\ s \in S, t \in T. \tag{16}$$

The lower and upper bounds on power generation variable $p_{s,t}$, i.e. P_s^{Min} and P_s^{Max}, respectively, for the generator at power plant s, is given by constraint (14). They are naturally linked with the binary decision variable $\omega_{s,t}$, which is 1 if the generator at power plant s is on at time period t, and 0 if the generator is off. It secures that power can only be produced at power plant s and period t, if the generator is active.

Maintaining balance of the electricity grid is of crucial importance, and constraint (15) describes this by stating that the total power produced $p_{s,t}$, needs to equal the power sold, p_t^S, subtracted the power that is purchased p_t^B. If this equation is not in balance, one of the load penalty variables, lp_t^U or lp_t^D takes a positive value. A proportional penalty cost C^L, is then incurred in the objective function.

Constraint (16) describes the operating status of the generator at power plant s, in time period t. This is done by stating that the operating status of the generator at power plant s and in time period t, is either the same as the operating status at the previous period $t-1$, or it has been started up in period t. The binary decision variable $\omega_{s,t}^S$, has value 1 if the generator at power plant s has been started up in period t.

4.1.7 Summary of the model

The complete model is

$$\begin{aligned}\textbf{Minimize } \sum_{t\in T}\Bigg(\sum_{r\in R}\Big(&C^{Over}\cdot s_{r,t}+C^{B}\cdot q^{B}_{r,t}+E_t\left(p^{B}_{t}-p^{S}_{t}\right)\\&+C^{R}\left(z^{U}_{r,t}+z^{D}_{r,t}\right)+C^{L}\left(lp^{U}_{t}+lp^{D}_{t}\right)\Big)\\&+\sum_{s\in S}\Big(C^{GS}\cdot\omega^{S}_{s,t}+\Gamma_{s,t}\cdot q_{r_s,t}\Big)\Bigg)\\-\sum_{r\in R}\Big(&C^{End}x^{End}_{r}+C^{R}\left(ze^{U}_{r}+ze^{D}_{r}\right)\Big).\end{aligned}$$

Subject to

$$
\begin{aligned}
q_{r,t} &= \sum_{j \in N:(r,j) \in A} q_{r,j,t}, && \forall \ \ r \in R, t \in T, \\
x_{r,t} &= x_{r,t-1} - s_{r,t} + \sum_{u \in T^*} \beta_u \cdot \bar{s}_{t-u} - q_{r,t} \\
&\quad - q^B_{r,t} + f_{r,t} + z^U_{r,t} - z^D_{r,t} - \left(z^U_{r,t-1} - z^D_{r,t-1}\right), && \forall\ r \in R, t \in T, \\
x_{r,k} &= x^{End}_r + ze^U_r - ze^D_r - z^U_{r,k} + z^D_{r,k}, && \forall\ r \in R, k = T_n, \\
xs_{r,t} &\geq x_{r,t} - x^{Max}_r, && \forall\ r \in R, t \in T, \\
xs_{r,t} &\leq x_{r,t} - x^{Max}_r \cdot \delta_{r,t}, && \forall\ r \in R, t \in T, \\
xs_{r,t} &\leq M_{r,t} \cdot \delta_{r,t}, && \forall\ r \in R, t \in T, \\
s_{r,t} &= c_r \cdot xs_{r,t}, && \forall\ r \in R, t \in T, \\
h^{Gross}_{s,t} &= l_{r,t-1}(x_{r,t-1}) - L_s, && \forall\ s \in S, t \in T, \\
h^{Net}_{s,t} &= h^{Gross}_{s,t} - \alpha_s \cdot q^2_{r_s,t}, && \forall\ s \in S, t \in T, \\
q_{r_s,t} &\geq Q^{Min}_{s,t}\left(h^{Net}_{s,t}\right) \cdot \omega_{s,t}, && \forall\ s \in S, t \in T, \\
q_{r_s,t} &\leq Q^{Max}_{s,t}\left(h^{Net}_{s,t}\right) \cdot \omega_{s,t}, && \forall\ s \in S, t \in T, \\
p_{s,t} &= G \cdot \eta^{Gen}_s\left(p_{s,t}\right) \cdot \eta^{Turb}_s\left(q_{r_s,t}, h^{Net}_{s,t}\right) \cdot h^{Net}_{s,t} \cdot q_{r_s,t}, && \forall\ s \in S, t \in T, \\
P^{Min}_s \cdot \omega_{s,t} &\leq p_{s,t} \leq P^{Max}_s \cdot \omega_{s,t}, && \forall\ s \in S, t \in T, \\
p_{s,t} &= p^S_t - p^B_t + lp^U_t - lp^D_t, && \forall\ s \in S, t \in T, \\
\omega_{s,t} &\leq \omega_{s,t-1} + \omega^{Start}_{s,t}, && \forall\ s \in S, t \in T, \\
x_{r,t},\ s_{r,t} &\geq 0, && \forall\ r \in R, t \in T, \\
h^{Gross}_{r,t},\ h^{Net}_{r,t} &\geq 0, && \forall\ r \in R, t \in T, \\
\delta_{r,t} &\in \{0,1\}, && \forall\ r \in R, t \in T, \\
\omega_{s,t},\ \omega^{Start}_{s,t} &\in \{0,1\}, && \forall\ s \in S, t \in T.
\end{aligned}
$$

4.2 Handling non-linearities

Handling non-linear functions in an effective way is important. How this is done in SHOP will be explained in the next sections.

4.2.1 Linearization of non-linear constraints

SHOP needs to be able to calculate the nonlinear HPF, and at the same time be computationally efficient. To do this, the HPF is converted into a linear input/output (I/O) curve, which is described in detail by [2].

A given number of SLP-iterations are performed to stabilize the head variation, i.e. the gross head $h_{s,t}^{Gross}$, between the intake reservoir r_s and the height L_s of which the water outlet at power plant s is situated. For each SLP-iteration $h_{s,t}^{Gross}$ is linearized. The result, i.e. the volume $x_{r_s,t-1}^*$, and water level $l_{r_s,t-1}(x_{r_s,t-1}^*)$ of intake reservoir r_s, from the previous SLP-iteration, is used as the input point for the current SLP-iteration. Thus, equation (9) can be rewritten as

$$h_{s,t}^{Gross} = l_{r_s,t-1}(x_{r_s,t-1}^*) - L_s, \qquad \forall\ s \in S, t \in T. \tag{17}$$

This result is then used later on, when calculating the net head. The same is done with the gate spillage variable $s_{r,t}$ defined in constraint (8). It is rewritten as

$$\Delta s_{r,t} = c_r \cdot xs_{r,t} - s_{r,t}^*, \qquad \forall\ r \in R, t \in T. \tag{18}$$

Here, $s_{r,t}^*$ is the spillage from the previous SLP-iteration, and $\Delta s_{r,t}$ is the difference in spillage between the previous and the current SLP-iteration. Equation (8) can hence be replaced by equation (18), and equation (3) is replaced by

$$\begin{aligned} x_{r,t} = x_{r,t-1} - \Delta s_{r,t} + \sum_{u \in T^*} \beta_u \cdot \bar{s}_{t-u} - q_{r,t} \\ - q_{r,t}^B + f_{r,t} + z_{r,t}^U - z_{r,t}^D - \left(z_{r,t-1}^U - z_{r,t-1}^D\right), \end{aligned} \qquad \forall\ r \in R, t \in T. \tag{19}$$

Water discharge is also needed to calculate the net head, thus the minimum water discharge $Q_{s,t}^{Min}$, the best efficiency water discharge $Q_{s,t}^{Best}$, and the maximum water discharge $Q_{s,t}^{Max}$, are determined next. They are used as the

initial points in the I/O curve of the generator in power plant s. The rest of the breakpoints of the I/O curve are obtained by uniformly partitioning the interval between $Q_{s,t}^{Min}$ and $Q_{s,t}^{Best}$ into b^{Down} segments, and the interval between $Q_{s,t}^{Best}$ and $Q_{s,t}^{Max}$ into b^{Up} segments. The set B is the set of all breakpoints b, $B = \left[0, b-1, b, b^{Down}, ..., b^{Down} + b^{Up}\right]$. The discharges at the $b^{Down} + 1$ breakpoints are then

$$\bar{Q}_{b,r_s,t} = Q_{s,t}^{Min} + \frac{Q_{s,t}^{Best} - Q_{s,t}^{Min}}{b^{Down}} \cdot b, \quad \forall \;\; b = 0, ..., b^{Down}, s \in S, t \in T. \tag{20}$$

The corresponding discharges at the b^{Up} breakpoints are

$$\begin{aligned} \bar{Q}_{b,r_s,t} = Q_{s,t}^{Best} + \frac{Q_{s,t}^{Max} - Q_{s,t}^{Best}}{b^{Up}} \cdot (b - b^{Down}), \\ \forall \;\; b = b^{Down} + 1, ..., b^{Down} + b^{Up}, s \in S, t \in T. \end{aligned} \tag{21}$$

Additionally, if the optimal water discharge $q_{r_s,t}^{*}$, from the previous SLP-iteration, differs from the breakpoints determined in the next SLP-iteration, it is added to the I/O curve as an extra breakpoint. This is done to improve convergence.

Then, given a water discharge point $\bar{Q}_{b,r_s,t}$ in breakpoint b, for power plant s, in time period t, the net head $H_{b,s,t}^{Net}$, in breakpoint b, is determined, and the turbine efficiency $\eta_s^{Turb}\left(\bar{Q}_{b,r_s,t}, H_{b,s,t}^{Net}\right)$, is calculated. The power output $\bar{P}_{b,s,t}$ at each breakpoint b is then calculated using equation (13).

When both water discharge and power output is determined at every breakpoint in the I/O curve, the slope of the line segment, between breakpoints b and $b-1$, equals

$$\gamma_{b,s,t} = \frac{\bar{P}_{b,s,t} - \bar{P}_{b-1,s,t}}{\bar{Q}_{b,r_s,t} - \bar{Q}_{b-1,r_s,t}}, \qquad \forall \; b \in B, s \in S, t \in T. \tag{22}$$

It is important that the I/O curve is concave, to keep the computational burden as low as possible. Thus, breakpoints that make the I/O curve non-concave, e.g. the white dots in Figure 14, are removed.

The I/O curve is limited by the permissible discharge limits $Q_{s,t}^{Min}$ and $Q_{s,t}^{Max}$, of the turbine at power plant s, in time period t. They define the first and last breakpoints of the I/O curve, respectively. Lastly, the production bounds of power plant s, are included as limits on the curve. They are defined, by the most restrictive rule, as

$$\hat{P}_{s,t}^{Min} = \text{MAX}\left[\bar{P}_{0,s,t}, P_s^{Min}\right], \qquad \forall\ s \in S, t \in T, \tag{23}$$

$$\hat{P}_{s,t}^{Max} = \text{MIN}\left[\bar{P}_{b^{Down}+b^{Up},s,t}, P_s^{Max}\right], \qquad \forall\ s \in S, t \in T. \tag{24}$$

Here P_s^{Min} and P_s^{Max}, are the production limits of the generator at power plant s. The final I/O curve for the generator at power plant s is illustrated, in red, in Figure 14. The minimum operating limit $\hat{P}_{s,t}^{Min}$ at power plant s, in time period t, is constrained by the minimum production limit P_s^{Min} of power plant s. The maximum operating limit $\hat{P}_{s,t}^{Max}$, at power plant s, in time period t, is constrained by the maximum permissible water discharge $Q_{s,t}^{Max}$ at the plant. Thus, the final operating limits, i.e. the first and last points of the I/O curve, are $(\hat{P}_{s,t}^{Min}, \hat{Q}_{s,t}^{Min})$ and $(\hat{P}_{s,t}^{Max}, \hat{Q}_{s,t}^{Max})$, respectively. Here, $\hat{P}_{s,t}^{Min} = P_s^{Min}$, $\hat{Q}_{s,t}^{Max} = Q_{s,t}^{Max}$, while $\hat{Q}_{s,t}^{Min}$ and $\hat{P}_{s,t}^{Max}$ are calculated by linear interpolation.

The HPF is now represented by a piecewise linear I/O curve. Hence the power output, i.e. the total power production, from power plant s in time period t, is given by

$$p_{s,t} = \sum_{b \in B} \gamma_{b,s,t} \cdot q_{b,r_s,t} + \hat{P}_{s,t}^{Min} \cdot \omega_{s,t}, \qquad \forall\ s \in S, t \in T. \tag{25}$$

Equation (25) states that the power production $p_{s,t}$, at power plant s in period t, is given by the sum of the slope of the I/O curve $\gamma_{b,s,t}$, multiplied with the water discharge $q_{b,r_s,t}$, over all breakpoints b. The lower bound of the production $p_{s,t}$, is integrated in equation (25), stating that if the generator at power plant s is active in period t, i.e. $\omega_{s,t} = 1$, then $p_{s,t}$ needs to be at

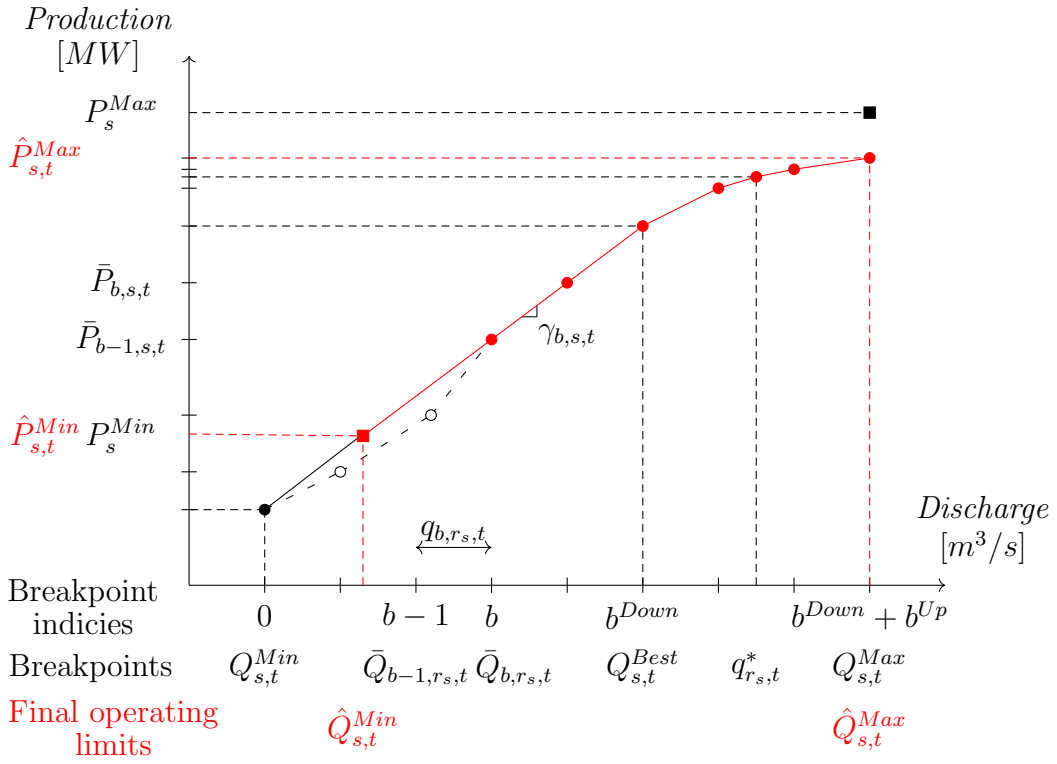

Figure 14: Illustration for determining the concave piecewise linear I/O curve for the generator at power plant s

[2].

least as large as $\hat{P}_{s,t}^{Min}$. The upper bound of the production $p_{s,t}$, is defined by the following inequality

$$p_{s,t} \leq \hat{P}_{s,t}^{Max} \cdot \omega_{s,t}, \qquad \forall\ s \in S, t \in T. \tag{26}$$

Constraint (26) makes sure that, if the generator at power plant s is active in period t, i.e. if $\omega_{s,t} = 1$, the production variable $p_{s,t}$ is less than or equal to its upper bound, $\hat{P}_{s,t}^{Max}$. Constraints (13) and (14) are thus replaced by constraints (25) and (26), respectively.

The total water discharge at power plant s, in period t is now given by

$$q_{r_s,t} = \hat{Q}_{s,t}^{Min} \cdot \omega_{s,t} + \sum_{b \in B} q_{b,r_s,t}, \qquad \forall\ s \in S, t \in T, \tag{27}$$

$$0 \leq q_{b,r_s,t} \leq \bar{Q}_{b,r_s,t} - \bar{Q}_{b-1,r_s,t}, \qquad \forall\ b \in B, s \in S, t \in T. \tag{28}$$

Constraints (11) and (12) are thus replaced by constraints (27) and (28).

4.2.2 Summary of the linearized model

The complex and non-linear model, which was presented in Section 4.1.7, can now be updated by approximating the non-linear constraints by the new, linear constraints found in Section 4.2.1. The new model is

Minimize

$$\begin{aligned}
&\sum_{t \in T} \Bigg(\sum_{r \in R} \Big(C^{Over} \cdot \Delta s_{r,t} + C^{B} \cdot q_{r,t}^{B} + E_t \left(p_t^B - p_t^S \right) \\
&+ C^{R} \left(z_{r,t}^{U} + z_{r,t}^{D} \right) + C^{L} \left(lp_t^U + lp_t^D \right) \Big) \\
&+ \sum_{s \in S} \Big(C^{GS} \cdot \omega_{s,t}^{S} + \Gamma_{s,t} \cdot q_{r_s,t} \Big) \Bigg) \\
&- \sum_{r \in R} \Big(C^{End} x_r^{End} + C^{R} \left(ze_r^U + ze_r^D \right) \Big)
\end{aligned}$$

Subject to

$$
\begin{aligned}
&(1),(4)-(7),(10),(15)-(16), \\
x_{r,t} &= x_{r,t-1} - \Delta s_{r,t} + \sum_{u \in T^*} \beta_u \cdot \bar{s}_{t-u} - q_{r,t} && \forall\ r \in R, t \in T, \\
&\quad - q^B_{r,t} + f_{r,t} + z^U_{r,t} - z^D_{r,t} - \left(z^U_{r,t-1} - z^D_{r,t-1}\right), \\
\Delta s_{r,t} &= c_r \cdot x s_{r,t} - s^*_{r,t}, && \forall\ r \in R, t \in T, \\
h^{Gross}_{s,t} &= l_{r,t-1}(x^*_{r,t-1}) - L_s, && \forall\ s \in S, t \in T, \\
p_{s,t} &= \sum_{b \in B} \gamma_{b,s,t} \cdot q_{b,r_s,t} - \hat{P}^{Min}_{s,t} \cdot \omega_{s,t}, && \forall\ s \in S, t \in T, \\
p_{s,t} &\leq \hat{P}^{Max}_{s,t} \cdot \omega_{s,t}, && \forall\ s \in S, t \in T, \\
q_{r_s,t} &= \hat{Q}^{Min}_{s,t} \cdot \omega_{s,t} + \sum_{b \in B} q_{b,r_s,t}, && \forall\ s \in S, t \in T, \\
0 &\leq q_{b,r_s,t} \leq \bar{Q}_{b,r_s,t} - \bar{Q}_{b-1,r_s,t}, && \forall\ b \in B, s \in S, t \in T.
\end{aligned}
$$

4.3 Handling binary variables

As mentioned in Section 4.1.6, $\omega^{Start}_{s,t}$ is a binary decision variable that says, for period t, if the generator at power plant s is turned on. SHOP's solution algorithm is described in [2], and it is divided into two modes: *Unit Commitment* (UC) and *Unit Load Dispatch* (ULD), which are illustrated in Figure 15. In UC mode, SHOP solves the scheduling problem, as well as the question if the generator unit at power plant s should be turned on in period t. This is solved as a MILP-problem, due to the previously mentioned binary decision variable $\omega^{Start}_{s,t}$, within a SLP-loop. After this is done for a given number of SLP-iterations, or after the convergence criterion is met, SHOP continues in ULD mode.

The SLP-iterations are performed to stabilize the head variation, in the reservoirs. For each SLP-iteration, the volume and water level of the reservoirs are updated. This is then used to calculate the gross head for the next SLP-

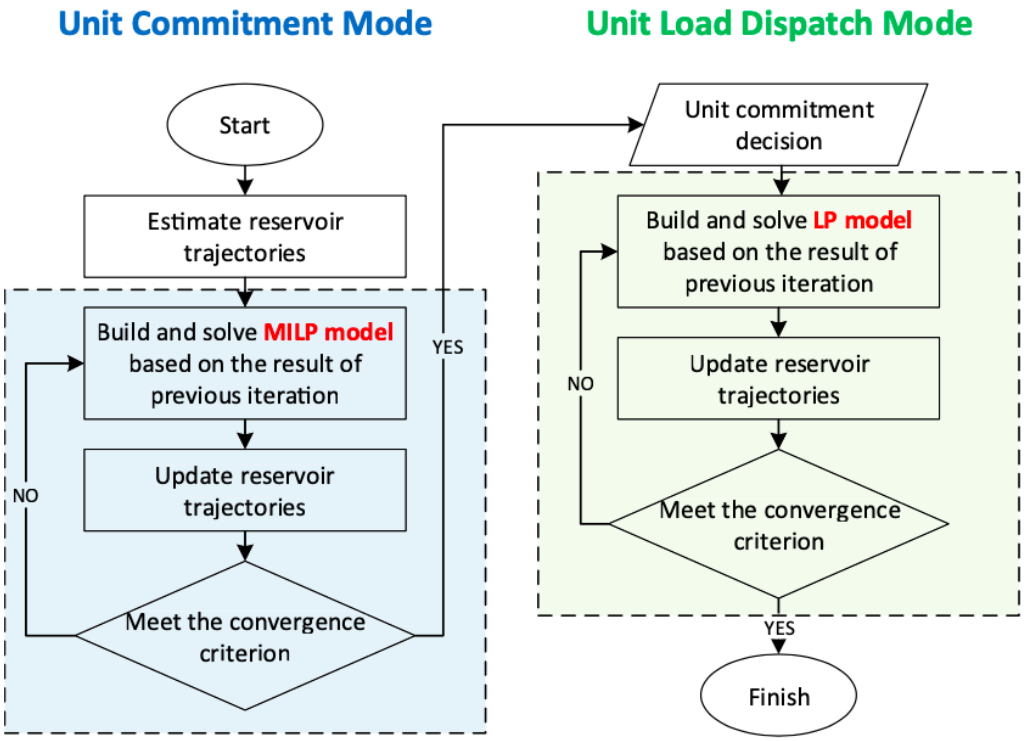

Figure 15: Solution strategy of SHOP [2].

iteration. The mentioned convergence criterion can be chosen as either of the following:

(i) The highest mismatch of the water level for each period before and after the reservoirs gets updated.

(ii) The maximum imbalance between the optimal power production that is found by solving the optimization model, and the power production, calculated based on the original HPF.

(iii) The change in objective function value between two consecutive iterations.

If the value is smaller than a given tolerance, the iterative process stops.

An alternative, fourth approach, is to define a stopping criteria after a finite number of iterations. The iterative process (SLP-loop) would then complete this given number of iterations, and then stop, whether the convergence has occurred or not. This eliminates the possibility that the iterative process could continue endlessly if the convergence criterion is never met.

The ULD mode builds on the results from the last SLP-iteration of the UC mode. Since the question, regarding which generator unit in power plant s, should be used for each time period t, has been answered in UC mode, the ULD mode's only focus is to calculate an optimal production schedule that fits with the "generator status"-plan, found in UC mode. Here, only committed generator units are included in ULD mode. Then the model turns into a LP-problem, in the absence of binary decision variables regarding overflow. In our case the model remains as a MILP-problem, since such variables are present.

Usually the convergence criterion, chosen from (i)–(iii), is met after three to five iterations for the UC mode, while the ULD mode needs two to three iterations. In our case, a stopping criterion of three iterations are used for both UC- and ULD mode.

4.4 Calibrating modelling parameter

As mentioned in Section 4.1.3, the modelling parameter $M_{r,t}$, in constraint (7), can be calibrated in an attempt to reduce the solution time of SHOP. The default value of $M_{r,t}$ is defined as ζ_r.

4.4.1 Defining the modelling parameter as small as possible

In order to obtain a strong LP-relaxation of the model in Section 4.2.2, the modelling parameter $M_{r,t}$, which occurs in constraint (7), should be kept as small as possible without cutting off feasible solutions.

Depending on the design and size of the dam in reservoir r, the maximum volume of water in r is denoted $\bar{x}_r$. It is limited by certain physical and hydrological properties, and is given as input to SHOP as part of a flood curve $o(x, r)$ which describes the amount of overflow from reservoir r, given volume x. The largest value that $xs_{r,t}$ can be assigned, in theory, is defined as

$$\theta_r = \bar{x}_r - x_r^{Max}, \qquad \forall\ r \in R. \tag{29}$$

When $M_{r,t} = \theta_r$, it is as small as possible, without risking to cut off solutions that fulfill constraint (6). Constraint (7) can hence be rewritten as

$$xs_{r,t} \leq \theta_r \cdot \delta_{r,t}, \qquad \forall\ r \in R, t \in T. \tag{30}$$

4.4.2 Heuristic for determining modelling parameters

Letting $M_{r,t} = \bar{x}_r - x_r^{Max}$ does not necessarily yield a strong relaxation to make a great impact on the solution time of the problem. There might be a substantial gap between the realistic volume of overflow $xs_{r,t}$, and the maximum volume now given by θ_r, for reservoir r in time period t.

By inspecting the data given as input, which typically includes starting volume, inflow, and a flood curve for reservoir r, one can use that to approximate the volume of the reservoirs, before the first SLP-iteration. This can in turn be used to approximate a value of $M_{r,t}$, which is tighter than θ_r, without taking a great risk of cutting off any valid solutions. This is done by defining the modelling parameter, not only for each reservoir r, but also for each

period t. It is computed by Algorithm 1, where g is a user-specified percentage that is used to increase the estimate of the volume $x_{r,t}$, of reservoir r in time period t. The values of $x_{r,t}$, for the first SLP-iteration, is as mentioned, approximated based on the input data. For the rest of the iterations, $x_{r,t}$ is equal to the reservoir volume $x^*_{r,t}$, from the previous SLP-iteration. As long as $\sigma_{r,t}(g)$ is not larger than θ_r, the solution time of the problem should decrease. Figure 17 illustrates example values of $\sigma_{r,t}(g)$ for Gråsidevatn, when the added percentage g is equal to 2.0, i.e. $\sigma_{r,t}(2.0)$.

Algorithm 1: Determining $\sigma_{r,t}(g)$

Input : Start volume x_r^{Start}, reservoir volume limit x_r^{Max}, inflow $f_{r,t}$, flood curve $o(x, r)$, added percentages g^{First} and g^{Later}, current SLP-iteration i

Output: Modelling parameter $\sigma_{r,t}(g)$

if $i = 1$ **then**

$x^0_{r,t} = x_r^{Start} + f_{r,t}$

$x_{r,t} = x^0_{r,t} - o(x^0_{r,t}, r)$

$g = g^{First}$

end

else

$x_{r,t} = x^*_{r,t}$

$g = g^{Later}$

end

$\sigma_{r,t}(g) = \text{MAX}\left[0, \quad \left(1 + \frac{g}{100}\right) \cdot x_{r,t} - x_r^{Max}\right]$

Return $\sigma_{r,t}(g)$

4.4.3 Formulating a model to minimize $M_{r,t}$

In an attempt to minimize the modelling parameter $M_{r,t}$, for all instances, a new optimization problem is formulated. Here the goal is to maximize the volume of overflow $xs_{r,t}$, which is the variable that is constrained by $M_{r,t}$.

Thus, by maximizing $xs_{r,t}$, one finds the minimum value that $M_{r,t}$ can be assigned. The model needs to be solved for each period t and each reservoir r. The optimal objective function value is thus $\phi_{r,t}$, for reservoir r and time period t. The model is defined as

Maximize $xs_{r,t}$

Subject to

$$(1), (4) - (6), (10), (15) - (19), (25) - (28).$$

Since the problem needs to be solved for each period t and reservoir r, the total solution time would be too large to have any value for the end-users. Thus, the result $\phi_{r,t}$, will be used as a guideline of how good the values of $\sigma_{r,t}(g)$, i.e. the result of Algorithm 1, are.

5 Effective elimination of unphysical overflow

The problem regarding the occurrence of unphysical overflow, when the optimization problem is solved effectively by using continuous variables, was outlined in Chapter 3. It was also shown how SHOP eliminates unphysical overflow, by introducing binary decision variables. However, this results in a very long solution time and thus the question that arises is:

What can be done to eliminate unphysical overflow effectively?

The natural answer is to, if possible, formulate a new optimization model for SHOP, such that unphysical overflow is eliminated, without using binary decision variables, or to reduce the solution time of the model that already eliminates unphysical overflow.

We will now direct our focus at the latter answer, and Section 4.4 makes several suggestions of how manipulation of the modelling parameter $M_{r,t}$, can have this desired effect. The next sections present the result of the experiments that have been conducted. The results mainly consist of the solution time that is spent in CPLEX, when SHOP solves the optimization problem. Several figures that compare the solution time in SHOP given different values of $M_{r,t}$, are presented and discussed.

The experiments are solved in SINTEF's SHOP-lab, using SHOP-version 14.2.2.0. The data basis describing the Fossdal watercourse, which consists of the Fossmark power plant and the reservoirs Gråsidevatn and Fossdalsvatn, has been provided by Eviny. It includes:

- Inflow $f_{r,t}$ to both reservoirs, for each time period (hour), for the whole planning horizon of 14 days.
- Start head and start volume, HRL and LRL of both reservoirs.

- Electricity price E_t in each time period, for the whole planning horizon.

- Flood-curve $o(x_{r,t}, r)$ for each reservoir.

- Minimum and maximum discharge- ($Q_{s,t}^{Min}$, $Q_{s,t}^{Max}$) and production ($P_{s,t}^{Min}$,$P_{s,t}^{Max}$) limit for the power plant Fossmark.

Some predefined settings of SHOP is:

- MIP-gap = 0.01%. This means that an integer solution is only considered optimal if its objective O_1, is within an optimality gap of 0.01% from the objective O_2 of the best found lower bound on the optimal objective value. To be more precise, O_1 is within the optimality gap of 0.01%, relative to O_2 if $\frac{O_2 - O_1}{O_1} \leq 0.0001$.

- A time limit per SLP-iteration of 600 seconds. This means that if no integer solution fulfills the MIP-gap, before 600 seconds have passed, SHOP ends the SLP-iteration, and the best found integer solution of the iteration is used as input for the next SLP-iteration.

- Penalty cost values presented in Table 1.

5.1 Comparing different values of $M_{r,t}$

Figures 16 – 17 give an overview how the different values of $M_{r,t}$, presented in Sections 4.4.1 – 4.4.2, compare to each other as well as to the actual volume of overflow $xs_{r,t}$.

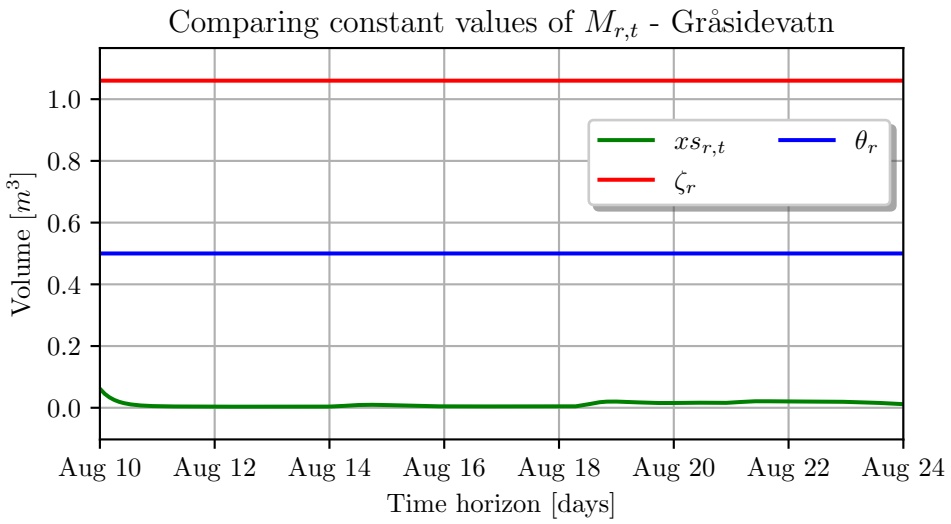

Figure 16: The values ζ_r (red) and θ_r (blue) of $M_{r,t}$, compared to the volume of overflow $xs_{r,t}$ for Gråsidevatn.

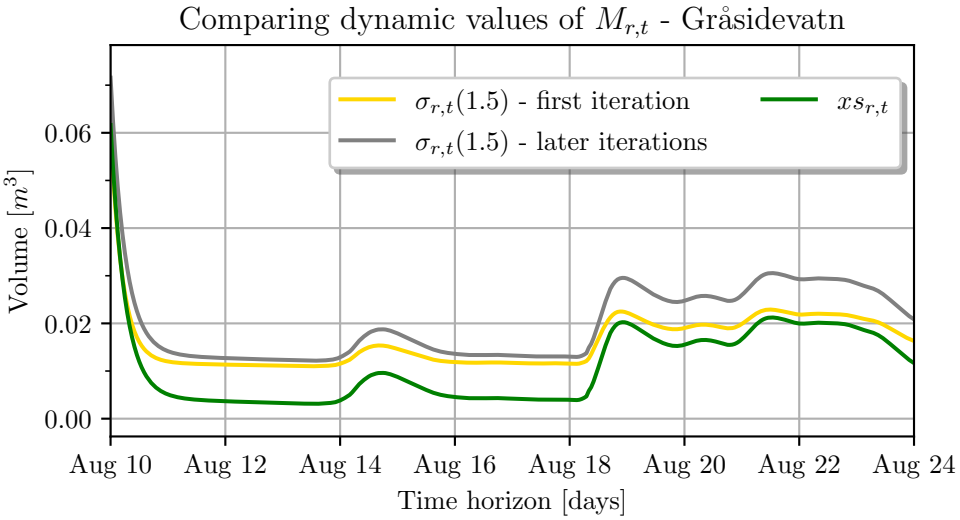

Figure 17: Dynamic values $\sigma_{r,t}(1.5)$, of the modelling parameter $M_{r,t}$, compared with the volume of overflow $xs_{r,t}$.

It is clear that the dynamic values $\sigma_{r,t}(g)$ of $M_{r,t}$ in Figure 17, is far tighter than the constant values ζ_r and θ_r in Figure 16. It also highlights how the values of $\sigma_{r,t}(g)$, used in the first SLP-iteration, are a bit tighter than the values of $\sigma_{r,t}(g)$, used in the later iterations. This suggests that it is possible to reduce the added percentage g for the later iterations, and hopefully reduce the solution time even more.

5.2 Experiments when $M_{r,t} = \theta_r$

The value θ_r for the modelling parameter $M_{r,t}$, defined in Section 4.4.1 as, as small as possible without risking to cut off any valid solutions, have been implemented and experimented with in SHOP. The results are presented in Figure 18. Here it becomes clear that this value assignment to $M_{r,t}$ in some instances reduces the solution time, whereas in other instances the solution time increases. It increases the solution time with nearly 50 % and 20 %, for the 5 and 10 day time horizons, respectively. While for both the 7- and 14-day time horizons, the solution time is decreased by almost 30 %.

Ideally, the solution time is decreased for all time horizons. However, since a time horizon of 14 days is the most preferred setting, and 7 days is the second most preferred setting, their results should be weighted more than the results of a time horizon of 5 or 10 days. The result shows that SHOP is sensitive to the value assigned to the modelling parameter, and that the value θ_r does not guarantee an improved solution time. Nevertheless, it is an important result, it shows a clear potential in reducing the solution time.

It is also worth mentioning that the Fossdal watercourse, with Fossmark power plant and the two reservoirs Fossdalsvatn and Gråsidevatn, is a small and quite simple hydro system. Thus, the result of using θ_r instead of ζ_r, might have an even larger impact on the solution time of a larger and more complex hydro system.

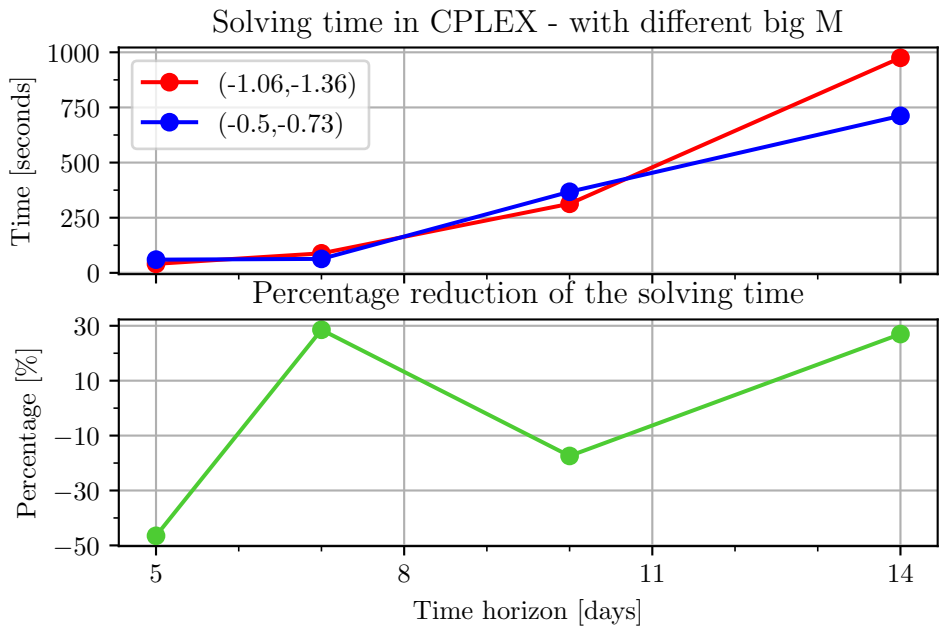

Figure 18: Comparing the solution time when $M_{r,t} = \zeta_r$ (red), with the solution time when $M_{r,t} = \theta_r$ (blue). The reduction of the solution time, in percentages is shown in green.

5.3 Experiments when $M_{r,t} = \sigma_{r,t}(g)$

The definition of $\sigma_{r,t}(g)$, i.e. a dynamic calculation of a minimal and case dependent value of the modelling parameter $M_{r,t}$ for each reservoir r and in each time period t, was given in Section 4.4.2. It has been implemented in SHOP, and experimented with. Figure 19 shows that the solution time varies a lot as the values of g are modified. With no added percentage, i.e. $g = 0$, SHOP does not manage to find an integer solution that is within a predefined MIP-gap of 0.01%. The solution time is substantially reduced when the user-specified added percentage is $g = 1.5\%$ or $g = 2.0\%$. These values of g will thus be further experimented with.

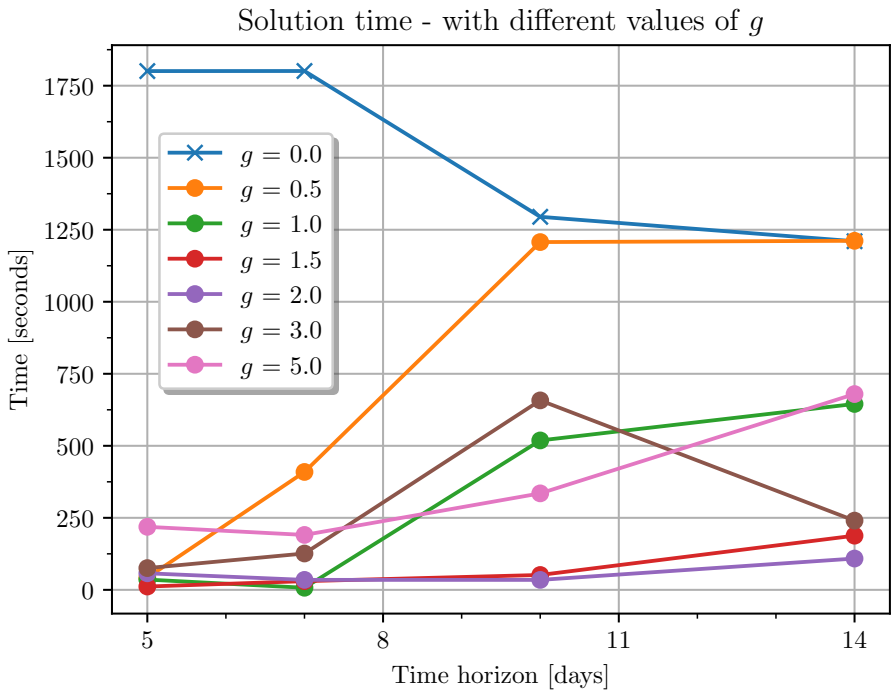

Figure 19: Solution time using $\sigma_{r,t}(g)$, for different values of g. The value of g is the same for all iterations. Points plotted as $\times$ indicate that no integer solution, fulfilling a MIP-gap of 0.01%, was found.

Figure 17 shows that $M_{r,t} = \sigma_{r,t}(1.5)$, plotted in yellow, which is based on an approximation of the overflow volume of reservoir r, for the first SLP-iteration, is barely larger than the actual overflow volume $xs_{r,t}$, plotted in green. This needs to be kept in mind when calibrating the values of g further, hence the value of g for the first SLP-iteration should be a bit larger than the value of g for the remaining iterations. This, and the result of it are shown in Figures 20 – 23.

Figures 20 – 21 show how the solution time varies with the added percentage g. As g gets smaller, so does the solution time. However with $g = 0.2$, the so-

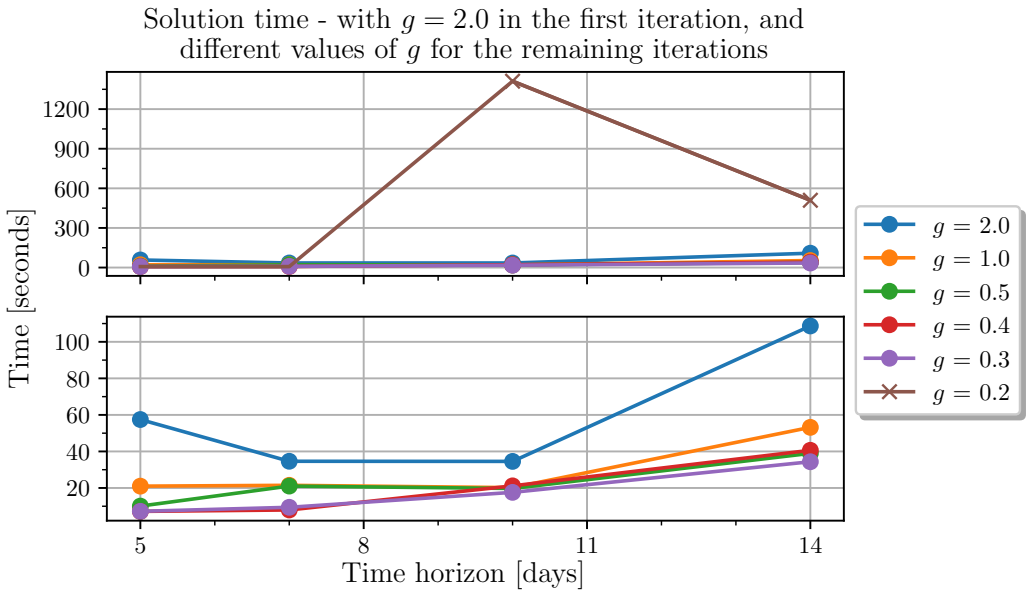

Figure 20: Solution time with $g = 2.0$ for the first SLP-iteration, and different values of g for the remaining iterations. Points plotted as $\times$ means that no integer solution, fulfilling a MIP-gap of 0.01%, was found.

lution time dramatically increases and no solution within the aforementioned MIP-gap is found. This could be explained by that the value $\sigma_{r,t}(0.2)$, for some time period t, is smaller than the real-life upper limit of the overflow. The heuristic uses reservoir volume from the previous SLP-iteration to approximate the volume of overflow. Since, for each SLP-iteration, it is possible that the volume of reservoir r has not yet converged, the added percentage g needs to be large enough to handle the possible gap in volume between the SLP-iterations. For this particular case we have experienced that g should not be smaller than 0.3.

Figures 22 – 23 show only the best performing values of g, for the later iterations. All the values give similar results, with $g = 0.3$ performing slightly better overall. Figure 24 shows how much the value of g for the first SLP-

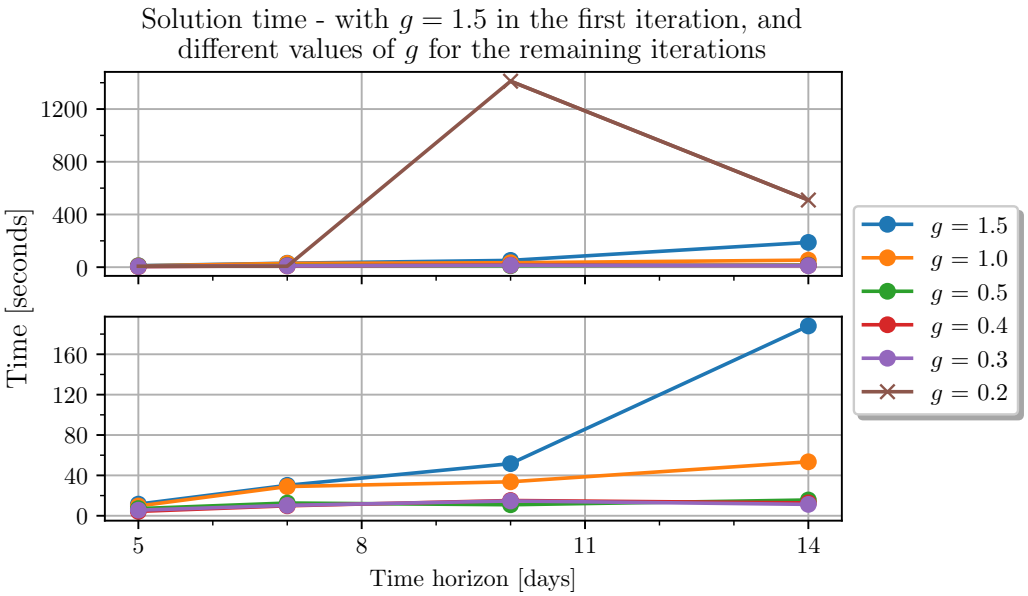

Figure 21: Solution time given $g = 1.5$ for the first SLP-iteration, and different values of g for the remaining iterations. Points plotted as $\times$ means that no integer solution, fulfilling a MIP-gap of 0.01%, was found.

iteration, impacts the solution time. Decreasing g by just 0.5, from 2.0 to 1.5, results in a 67% reduction in solution time, when solving the problem for the whole 14-day planning horizon.

The results presented in Figures 22 – 24 indicate that the added percentage g should be 1.5 in the first SLP-iteration, and 0.3 in later iterations, when solving the scheduling problem. Figure 25 compares the solution time of the problem when $M_{r,t} = \sigma_{r,t}(1.5)$ in the first SLP-iteration and $M_{r,t} = \sigma_{r,t}(0.3)$ in the later iterations, to the solution time when $M_{r,t} = \zeta_r$. It shows how the solution time is just beneath 1000 seconds when solving the problem with $M_{r,t} = \zeta_r$, whilst the proposed method, using $M_{r,t} = \sigma_{r,t}(g)$, reduces the solution time to 11 seconds.

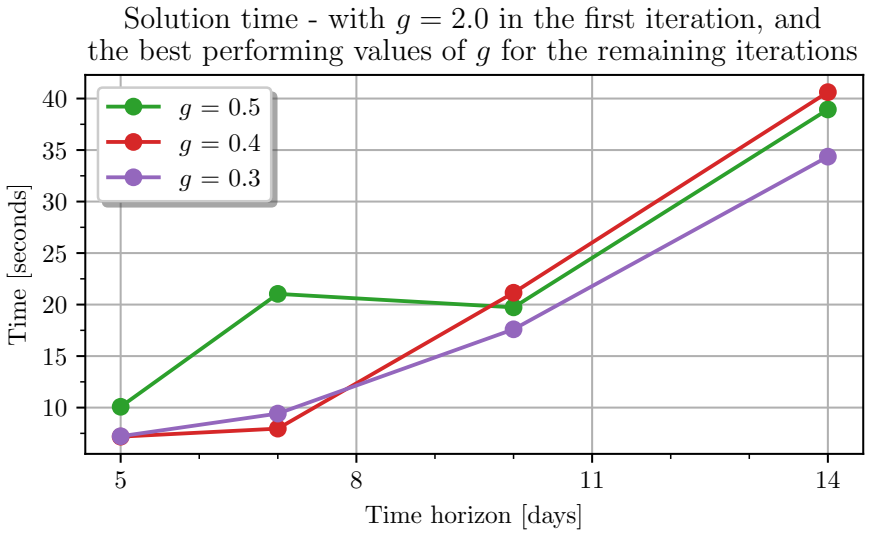

Figure 22: Solution time when $M_{r,t} = \sigma_{r,t}(g)$ for the best performing values of g, with $M_{r,t} = \sigma_{r,t}(2.0)$ in the first SLP-iteration.

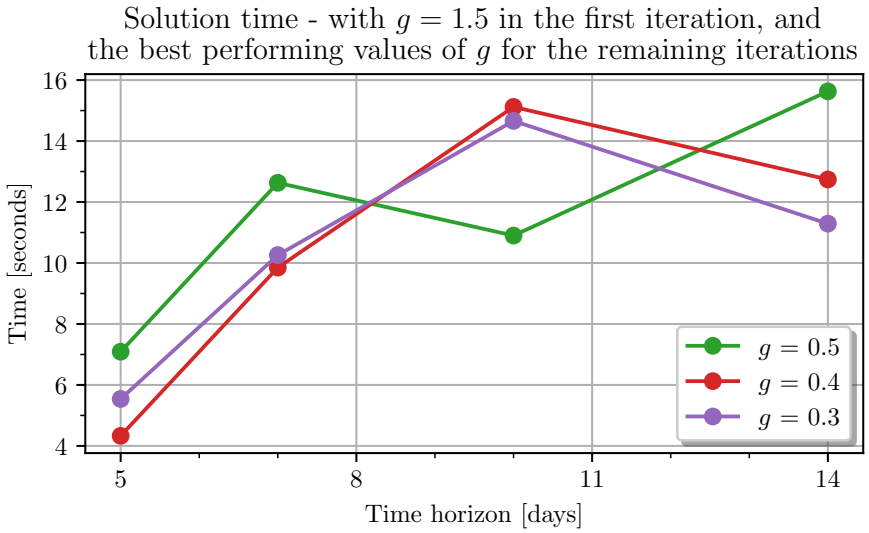

Figure 23: Solution time when $M_{r,t} = \sigma_{r,t}(g)$ for the best performing values of g, with $M_{r,t} = \sigma_{r,t}(1.5)$ in the first SLP-iteration.

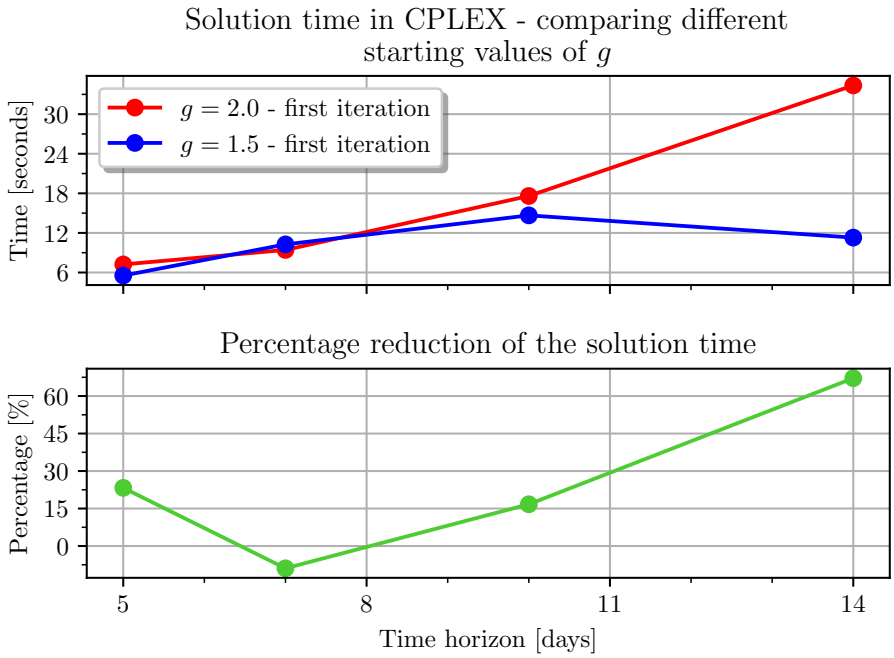

Figure 24: The problem is solved with $M_{r,t} = \sigma_{r,t}(0.3)$ in the later SLP-iterations, comparing the impact of different values of g in the first iteration.

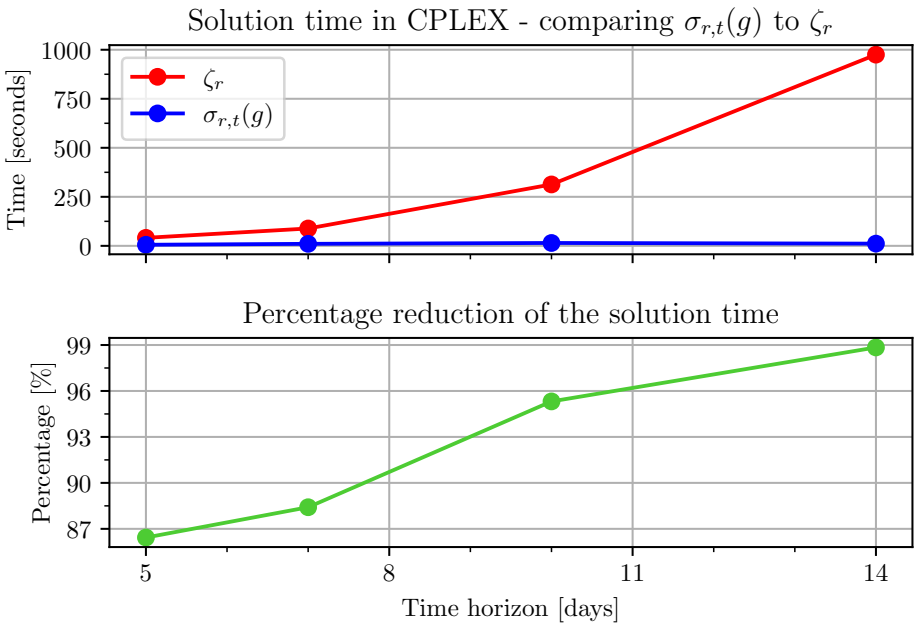

Figure 25: Comparing the solution time in CPLEX when $M_{r,t} = \zeta_r$, when $M_{r,t} = \sigma_{r,t}(1.5)$ in the first SLP-iteration, and when $M_{r,t} = \sigma_{r,t}(0.3)$ in the later iterations.

5.4 Comparing $\sigma_{r,t}(g)$ with $\phi_{r,t}$

As seen in Section 5.3, letting $M_{r,t} = \sigma_{r,t}(g)$ does give a substantial reduction in the solution time. However, Algorithm 1 which computes $\sigma_{r,t}(g)$, does not guarantee a great result in every case, or even a feasible solution, as seen in Figures 20 – 21. Nor can it guarantee that a solution is optimal.

Figure 26 compares the performance of Algorithm 1, to the maximum volume of overflow $\phi_{r,t}$, i.e. the minimum value the modelling parameter $M_{r,t}$ can have, at Gråsidevatn. The figure shows that $\phi_{r,t}$ at Gråsidevatn, coincides perfectly with the actual volume of overflow $xs_{r,t}$. This is logical since the gate regarding scheduled water from Gråsidevatn is closed, and the volume of the reservoir is thus only dependent of inflow and overflow.

This is the idea behind Algorithm 1, and it is clear from Figure 26 that the values of $\sigma_{r,t}(g)$, with $g^{First} = 1.5$ and $g^{Later} = 0.3$, are close to the values of $\phi_{r,t}$. Intuitively one would like $\sigma_{r,t}(g)$ to be equal to $\phi_{r,t}$. This would then mean that in the later SLP-iterations, where convergence is likely to have occurred, it would be possible to reduce g^{Later} even more. However, our observations from Figures 19 – 21 suggest otherwise, where $g < 0.3$ in the later iterations causes the solving time to increase dramatically. This could possibly be explained by that convergence is yet to occur, or that some of the constraints in the model are too strict. However, we have verified that convergence occurs, and that none of the constraints are overly strict. We therefore do not know, at this moment, why g cannot be smaller than 0.3.

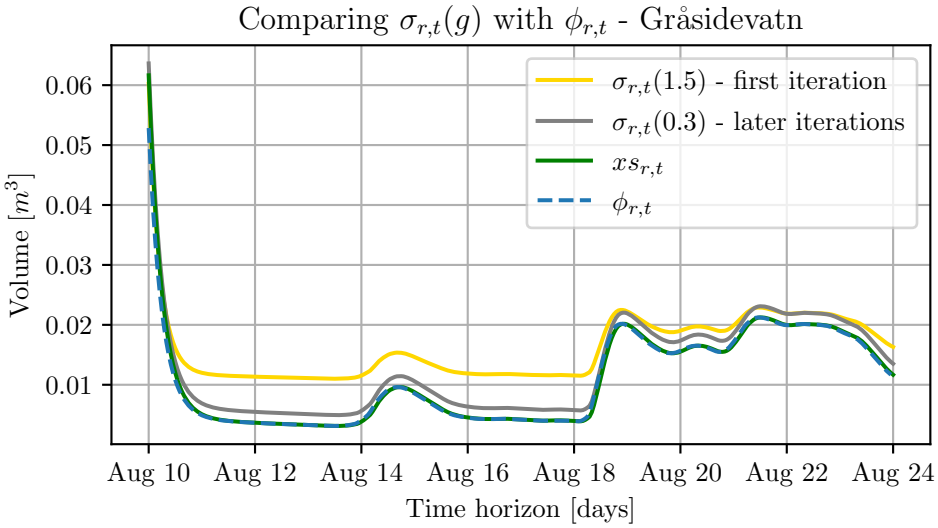

Figure 26: $\sigma_{r,t}(g)$, computed by Algorithm 1, compared to $\phi_{r,t}$, which is the maximum volume of overflow $xs_{r,t}$, i.e. the minimum value $M_{r,t}$ can have.

6 Conclusion

We started with the problem regarding the occurrence of unphysical overflow. A model, based on Eviny's power plant Fossmark, with adjacent reservoirs, was introduced. It then became clear that by introducing binary decision variables to model the relation between overflow and reservoir level, unphysical overflow is eliminated completely. However, it also increased the solution time dramatically.

Thus, in an attempt to reduce the solution time, an optimization method involving more precise formulations of the modelling parameter ζ_r, which acts as an upper limit on the volume of overflow, was suggested. By using the modelling parameter $\sigma_{r,t}(g)$, calculated by Algorithm 1, the solution time was reduced dramatically, from just beneath 1000 seconds, to around 11 seconds.

Although we were able to achieve some great results for this case, there is no guarantee that the aforementioned method performs well in every case. Observations indicate that the optimization problem is very sensitive to the value of g in $\sigma_{r,t}(g)$.

With this being said, the results show a potential for substantial reduction of the solution time of the problem. It should therefore be investigated whether other variable and constraint modifications result in a further reduction of the solution time, of scheduling problems for other hydro systems with flooding risks.

References

[1] Vaclav Smil. Global primary energy consumption by source, 2017. URL: https://ourworldindata.org/grapher/global-energy-substitution?country=~OWID_WRL.

[2] Hans Ivar Skjelbred, Jiehong Kong, and Olav Bjarte Fosso. Dynamic incorporation of nonlinearity into milp formulation for short-term hydro scheduling. *International Journal of Electrical Power & Energy Systems*, 116:105530, 2020. URL: https://www.sciencedirect.com/science/article/pii/S0142061519310105, doi:https://doi.org/10.1016/j.ijepes.2019.105530.

[3] Asbjørn Vinjar and Knut Hofstad. Vannkraft, Store norske leksikon, 2021. URL: https://snl.no/vannkraft.

[4] Asbjørn Vinjar and Knut Hofstad. Vannkraft i Norge, Store norske leksikon, 2021. URL: https://snl.no/vannkraft#-Vannkraft_i_Norge.

[5] Ali Thaeer Hammid, Omar I. Awad, Mohd Herwan Sulaiman, Saraswathy Shamini Gunasekaran, Salama A. Mostafa, Nallapaneni Manoj Kumar, Bashar Ahmad Khalaf, Yasir Amer Al-Jawhar, and Raed Abdulkareem Abdulhasan. A review of optimization algorithms in solving hydro generation scheduling problems. *Energies*, 13(11), 2020. URL: https://www.mdpi.com/1996-1073/13/11/2787, doi:10.3390/en13112787.

[6] Jiehong Kong, Hans Ivar Skjelbred, and Olav Bjarte Fosso. An overview on formulations and optimization methods for the unit-based short-term hydro scheduling problem. *Electric Power Systems Research*, 178:106027, 2020. URL: https://www.sciencedirect.com/science/article/pii/S0378779619303463, doi:https://doi.org/10.1016/j.epsr.2019.106027.

[7] Jarand Røynstrand and Ingelin Steinsland. Physical overflow mode in SHOP using MIP (internal report). *Internal note at SINTEF*, 2006.

[8] SINTEF. About SHOP, 2021. URL: https://www.sintef.no/programvare/shop/.

[9] Eviny. Fossmark powerplant, 2022. URL: https://www.eviny.no/om-eviny/vannkraft/andre-vassdrag.

[10] M.M Belsnes, O Wolfgang, T Follestad, and E.K Aasgård. Applying successive linear programming for stochastic short-term hydropower optimization. *Electric power systems research*, 130:167–180, 2016.

[11] Z.K Shawwash, T.K Siu, and S.O.D Russell. The b.c. hydro short term hydro scheduling optimization model. *IEEE transactions on power systems*, 15(3):1125–1131, 2000.

[12] M. Piekutowski, T. Litwinowicz, and R.J. Frowd. Optimal short-term scheduling for a large-scale cascaded hydro system. *IEEE Transactions on Power Systems*, 9(2):805–811, 1994. doi:10.1109/59.317636.

[13] D.A.G. Vieira, L.S.M. Guedes, A.C. Lisboa, and R.R. Saldanha. Formulations for hydroelectric energy production with optimality conditions. *Energy Conversion and Management*, 89:781–788, 2015. URL: https://www.sciencedirect.com/science/article/pii/S0196890414009297, doi:https://doi.org/10.1016/j.enconman.2014.10.048.

[14] Chun-tian Cheng, Sheng-li Liao, Zi-Tian Tang, and Ming-yan Zhao. Comparison of particle swarm optimization and dynamic programming for large scale hydro unit load dispatch. *Energy conversion and management*, 50(12):3007–3014, 2009.

[15] G Chang, M Aganagic, J Waight, J Medina, T Burton, S Reeves, and M Christoforidis. Experiences with mixed integer linear programming-

based approaches in short-term hydro scheduling. *IEEE power engineering review*, 21(9):63–63, 2001.

[16] R. E. Griffith and R. A. Stewart. A nonlinear programming technique for the optimization of continuous processing systems. *Management Science*, 7(4):379–392, 1961. URL: http://www.jstor.org/stable/2627058.

[17] H. I. Skjelbred. Unit-based short-term hydro scheduling in competitive electricity markets. *P.h.D. H. I. Skjelbred*, 2019. URL: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2644165.

[18] O.B Fosso and M.M Belsnes. Short-term hydro scheduling in a liberalized power system. In *2004 International Conference on Power System Technology, 2004. PowerCon 2004*, volume 2, pages 1321–1326 Vol.2. IEEE, 2004.

[19] O.B Fosso, A Gjelsvik, A Haugstad, B Mo, and I Wangensteen. Generation scheduling in a deregulated system. the norwegian case. *IEEE transactions on power systems*, 14(1):75–81, 1999.

Appendix

A Python code used to solve the model in SHOP

```
import pandas as pd
import numpy as np
from pyshop import ShopSession

tick_label = ["Aug 10","Aug 12","Aug 14","Aug 16",
              "Aug 18","Aug 20","Aug 22","Aug 24"]

# Create a new SHOP session.
shop = ShopSession(silent=False, log_file='')

# Setting the start time of the model
starttime = pd.Timestamp('2020-08-10 00:00:00')
# Setting the end time of the model
endtime = pd.Timestamp('2020-08-24')

# Setting the start and end time to the shop instance,
# and defining the time unit
index = [pd.Timestamp(2020, 8, 10, 0)]
resolution = [1]
time_resolution = pd.Series(index=index, data=resolution)
shop.set_time_resolution(starttime = starttime, endtime = endtime,
                         timeunit = 'hour',
                         timeresolution = time_resolution)

# Import file with the model and load model into ShopSession
```

```
from Fossmark import fossmark_LP as fossmark
fossmark.load(shop.shop_api)

def discharge_to_Mvolume(discharge):
    return discharge*3600/1000000

def volume_to_head(vol,reservoir):
    if reservoir == 'Gråsidevatn':
        vol_head = shop.model.reservoir.Graasidevatn.vol_head.get()
    else:
        vol_head = shop.model.reservoir.Fossdalsvatn.vol_head.get()
    if vol not in vol_head:
        vol_head.loc[vol] = np.nan
        vol_head = vol_head.sort_index().interpolate(method='index')
    return vol_head[vol]

def head_to_discharge(head,reservoir):
    if reservoir == 'Gråsidevatn':
        flow = shop.model.reservoir.Graasidevatn.flow_descr.get()
    else:
        flow = shop.model.reservoir.Fossdalsvatn.flow_descr.get()
    if head not in flow:
        flow.loc[head] = np.nan
        flow = flow.sort_index().interpolate(method='index')
    return flow[head]

def calculate_future_volume(reservoir):
    if reservoir == 'Gråsidevatn':
        start_vol = shop.model.reservoir.Graasidevatn.start_vol.get()
        inflow = shop.model.reservoir.Graasidevatn.inflow.get()
```

```
        else:
            start_vol = shop.model.reservoir.Fossdalsvatn.start_vol.get()
            inflow = shop.model.reservoir.Graasidevatn.inflow.get()
        volume = [start_vol]
        head = volume_to_head(start_vol,reservoir)
        flow = head_to_discharge(head,reservoir)
        volume[0] -= discharge_to_Mvolume(flow)
        for e,i in enumerate(inflow):
            vol_inflow = discharge_to_Mvolume(i)
            volume.append(volume[-1]+vol_inflow)
            head = volume_to_head(volume[-1]+vol_inflow,reservoir)
            flow = head_to_discharge(head,reservoir)
            volume[e+1] -= discharge_to_Mvolume(flow)
        return np.array(volume)

    def calculate_bigM(reservoir, percentage):
        vol = calculate_future_volume(reservoir)
        if reservoir == 'Gråsidevatn':
            max_volume = shop.model.reservoir.Graasidevatn.max_vol.get()
        else:
            max_volume = shop.model.reservoir.Fossdalsvatn.max_vol.get()
        return (1 + percentage/100) * vol - max_volume

    def update_bigM(reservoir, percentage):
        if reservoir == 'Gråsidevatn':
            volume = shop.model.reservoir.Graasidevatn.storage.get()
            max_volume = shop.model.reservoir.Graasidevatn.max_vol.get()
        else:
            volume = shop.model.reservoir.Fossdalsvatn.storage.get()
            max_volume = shop.model.reservoir.Fossdalsvatn.max_vol.get()
```

```
    return (1+percentage/100) * volume - max_volume

# Build and solve model:

percentage_start = 1.5
percentage = 0.3

BigM_G = -calculate_bigM('Gråsidevatn',percentage_start)
BigM_F = -calculate_bigM('Fossdalsvatn',percentage_start)

shop.model.reservoir.Fossdalsvatn.overflow_mip_flag.set(True)
shop.model.reservoir.Graasidevatn.overflow_mip_flag.set(True)

iterations = 6

for i in range(iterations):
    if i > 0:
        BigM_G = -update_bigM('Gråsidevatn',percentage)
        BigM_F = -update_bigM('Fossdalsvatn',percentage)
        BigM_G.loc[BigM_G>0] = 0
        BigM_F.loc[BigM_F>0] = 0
    if i == 3:
        shop.set_code(['incremental'],[])

    # Create a file with the LP-model
    model_name = 'printed_model_%s.lp'%(i+1)
    shop.print_model([],[model_name])

    # Set up the model in SHOP
    shop.lp_model.build()
```

```
shop.lp_model.load_model()

# Change the values of the "Big M" in the model:

# find the id number of the row (constraint) corresponding
# to "reservoir volume over maximum limit"
row_id = shop.lp_model.row_type['rsv volume over limit max'].id

# find every such row (constraint) in the model
ovfl = shop.lp_model.row.filter(row_type = row_id)

counter_G = 0
counter_F = 0

# for every such row, change the coefficient of
# the binary variable to BigM_G[counter_G] (for Gråsidevatn).
for j in ovfl:
    if abs(shop.lp_model.row[j].vars[1][1]) < 1.06:
        shop.lp_model.row[j].set_parameters(variables =
        [shop.lp_model.row[j].vars[1][0]], coefficients =
        [BigM_G[counter_G]])
        counter_G += 1
    if abs(shop.lp_model.row[j].vars[1][1]) > 1.35:
        shop.lp_model.row[j].set_parameters(variables =
        [shop.lp_model.row[j].vars[1][0]], coefficients =
        [BigM_G[counter_F]])
        counter_F += 1

# Solve the model
shop.lp_model.solve()
```