



NTNU – Trondheim
Norwegian University of
Science and Technology

Evaluation of a stochastic model for short-term hydro power generation scheduling

Bendik Støa

Master of Science in Physics and Mathematics

Submission date: June 2012

Supervisor: Anne Kværnø, MATH

Co-supervisor: Tellef Juell Larsen, Statkraft Energi AS
Michael Martin Belsnes, Sintef Energi AS

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

The objective of this thesis has been to evaluate the utilization of a stochastic programming model for short-term hydro power scheduling. The focus has been on the real-life application of such a tool. By considering a prototype of the stochastic model SHARM, which is based on the widely used deterministic SHOP model, it has been possible to use the same degree of detail as in the current operative scheduling. The work has been carried out at Statkraft, using real plants and operational data.

The expected objective function value obtained from using stochastic and deterministic day-ahead plans have been compared. A cascaded system has been run for 24 days in the winter depletion season with price uncertainty, and the results show a very slight increase in profit. Two systems have been run for 6 days in the spring flooding season with inflow and price/inflow uncertainty. These tests show no significant benefits, in terms of the objective function, of considering uncertainty in the construction of day-ahead plans.

The reservoir handling of SHARM on a test system, consisting of one large and one small reservoir above a plant, have been evaluated in the spring flooding season with inflow uncertainty. The results show that a stochastic model performs well in this situation, producing a robust plan that avoids spillage for all inflow scenarios.

Finally, the thesis has examined the computational performance of the SHARM prototype and the supplementary scenario tree construction and reduction algorithms. It is shown that reducing the size of the input trees reduce the solution time significantly, while still retaining much of the original information. The reduction algorithm seems to have good stability properties when considering stochastic prices.

Preface

This thesis is the culmination of a scheme presented to me by Michael M. Belsnes at Sintef Energy Reserach in the spring of 2011. After studying applied physics and mathematics for four years, I had developed an interest in numerical mathematics and optimization. So when the time came to choose the topic for my project work and master's thesis, I wanted to find something within these areas. Since the department of mathematical sciences at NTNU only offered a single course in optimization, and I wanted a project focused on applications, it felt natural to approach other specialist environments.

At Sintef I got the opportunity to apply my skills to short-term hydro power scheduling in several ways. In a summer intern project in 2011, I was introduced to the field and worked on implementing a simulation functionality in the scheduling tool SHOP. During the fall term, I did a specialization project investigating the application of non-linear optimization methods to a sub-problem considered by SHOP. Both of these tasks I really enjoyed.

When accepting these projects, I mainly focused on the tasks for the summer and fall, and had no clear understanding of what I was going to this spring. Stochastic optimization was not something I was familiar with, so the first couple of months involved a lot of catching up and learning new things.

Carrying out the work at Statkraft was quite interesting, both in terms of experiencing a new working environment and by the fact that I felt as though I was working on a real project. Interacting with the production planners there, I got a new understanding of the tasks at hand and realized the importance of communicating with the end-users when working on research and development projects.

In hindsight, I would maybe have chosen a thesis problem which involved a bit more hands-on mathematical programming. This project may have been more suitable for a student with a background in power scheduling from a practical or financial perspective, who did not have to spend so much time on learning the basics. It has however been a really valuable experience for me. I have learned a lot about a new topic, and I have gotten an impression of how a real R&D project is carried out. This is knowledge that I know will be useful for me in the future.

Acknowledgements

There are a lot of people that have helped me during this spring. First I would like to thank Statkraft, for providing an apartment for me in Oslo and an office space in an exciting specialist environment. Tellef Juell Larsen has been my main supervisor, and have always been available for discussions and assistance. Fredd Kristiansen have provided helpful input to the project, and taught me about practical production planning and stock trading. I am also grateful to all the other people at Kraftsentralen that have helped me, talked to me and given me cake.

I thank Michael M. Belsnes at Sintef, for giving me the opportunity to work with applied optimization for over a year. Turid Follestad and Ingrid Honve at Sintef have also been very helpful. I have used their expertise a lot during this spring, for debugging, explaining stochastic programming and discussing evaluation methods. The method described in Section 4.3.3 is the main result of these discussions. I am grateful for all the input and feedback from the people at Sintef, which has been very important in this work.

Anne Kværnø has been my supervisor at NTNU, and I appreciate that she has given me the freedom to explore new areas and applications. Last fall she organized an optimization course just for me, and she has been helpful in reviewing my work.

Finally I would like to thank Nina and Mr. Cheeky for moral support (and bananas).

Contents

1	Introduction	1
1.1	Background and motivation	1
1.2	Purpose of the work	3
1.3	Structure of the report	4
2	Deterministic short-term hydro power scheduling	5
2.1	Hydro power scheduling	5
2.2	Short-term scheduling	6
2.3	SHOP	8
3	Stochastic short-term hydro power scheduling	13
3.1	Stochastic programming	13
3.2	Stochastic optimization in hydro power scheduling	15
3.3	Scenario tree generation and reduction	16
3.3.1	Scenario trees	16
3.3.2	Generation methods	18
3.3.3	Generation by scenario tree reduction	18
3.3.4	Evaluation of scenario tree quality	22
3.4	The SHARM prototype	24
3.5	Price and inflow modeling	26
4	Goals and strategy for the evaluation process	27
4.1	Test watercourses	28
4.2	Computational tests performed in the evaluation	30
4.3	Comparison against SHOP	32
4.3.1	Deterministic scheduling for the day-ahead market	32
4.3.2	Stochastic scheduling for the day-ahead market	33
4.3.3	Evaluation of day-ahead plans	35
4.3.4	Analysis of sensitivity to input	36
4.4	Evaluation of computation time in the new model	37
4.4.1	Performance of the scenario tree generation algorithm	37
4.4.2	Computational performance of SHARM	38

5	Results and discussion	39
5.1	Test 1: Comparison with stochastic price	39
5.2	Test 2: Run-time test, stochastic inflow and inflow/price . . .	43
5.3	Test 3 - 6: Comparison, stochastic inflow and inflow/price . .	48
5.4	Test 7: Stability tests, stochastic price	50
5.5	Test 8: Sensitivity to input and qualitative analysis	52
5.6	Similar investigations	56
6	Conclusion	59
6.1	Comparison against SHOP	59
6.2	Computational performance of SHARM	61
6.3	Final conclusions	62
7	Suggestions for further development	65
	Bibliography	67
A	List of symbols	71
B	Algorithms	73
B.1	Fast forward selection	73
B.2	Backwards construction	74
C	Additional results	76

List of Figures

2.1	A coarse depiction of the current hydro power scheduling hierarchy, as described in [4].	6
2.2	A typical timeline for the utilization of a short-term hydro power scheduling program in the Nordic market. Based on a figure included in [14].	9
2.3	Example topology of a hydro power system. Based on a figure included in [1].	10
3.1	A general five-stage scenario tree is shown on the left, and a scenario fan on the right.	17
4.1	Schematic description of the 2 watercourses that have been considered in the evaluations.	29
4.2	Showing 2 equivalent scenario fans where price is the only stochastic parameter. By applying the one on the right, a single plan can be calculated by SHARM while still taking uncertainty into account in the period $t \in [19, 42]$	34
5.1	The differences in objective function value between SHARM and the run with a deterministic day-ahead plan is shown for test system 1. A positive value means that SHARM gives a greater income. Each test case on the x-axis represents a specific day.	40
5.2	The difference in objective function value obtained with and without a deterministic day-ahead plan is plotted against the average variance in the input price scenarios. No obvious pattern is observed, suggesting that there is no significant correlation between the parameters in these cases.	42
5.3	Results from the run-time test with stochastic inflow on test system 1. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €13 873 471,12. The red line relates the CPU-time to the degree of reduction.	44

5.4	Results from the run-time test with stochastic inflow on test system 2. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €3 133 254,38. The red line relates the CPU-time to the degree of reduction.	44
5.5	Results from the run-time test with stochastic price and inflow on test system 1. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €13 831 700,09. The red line relates the CPU-time to the degree of reduction.	45
5.6	Results from the run-time test with stochastic price and inflow on test system 2. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €3 133 305,05. The red line relates the CPU-time to the degree of reduction.	46
5.7	The full trees with price/inflow uncertainty used in Test 2. Both are generated from an initial fan of 225 scenarios by specifying $\epsilon_{rel} = 0.10$	47
5.8	The original input scenarios are shown on the left, and the scenarios remaining after applying the suggested reduction strategy are shown on the right. Inflow scenarios are shown at the top, the units on the axis being hours and m^3/s . Price scenarios are found below, with units of hours and €/MWh.	48
5.9	The plot displays the difference in objective function value between SHARM and the run with a deterministic day-ahead plan for test system 1 with stochastic inflow. Positive values means that the stochastic plan gives in higher income. Each test case on the x-axis represents a specific day. Numerical values are found in Table C.1.	49
5.10	The plot displays the difference in objective function value between SHARM and the run with a deterministic day-ahead plan for test system 1 with stochastic inflow and price. Positive values means that the stochastic plan gives in higher income. Numerical values are found in Table C.2.	49
5.11	Results from the run-time test with stochastic price on test system 1. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €17 539 095,30. The red line relates the CPU-time to the number of scenarios included in the fan.	51
5.12	The inflow scenarios used in Test 8, with $\alpha = 1$. The expected value of the inflow from this tree is shown in red.	53

5.13	The figure shows the reservoir trajectories of R2 in test system 2 for the inflow scenarios shown in Figure 5.12, using $\alpha = 1$. The red line shows the expected reservoir trajectory, and the light blue bars represent the production discharge through P1.	53
5.14	The inflow scenarios used in Test 8, with $\alpha = 2$. The expected value of the inflow from this tree is shown in red.	54
5.15	The figure shows the reservoir trajectories of R2 in test system 2 for the inflow scenarios shown in Figure 5.14, using $\alpha = 2$. The red line shows the expected reservoir trajectory, and the light blue bars represent the production discharge through P1.	54
5.16	The inflow scenarios used in Test 8, with $\alpha = 5$. The expected value of the inflow from this tree is shown in red.	55
5.17	The figure shows the reservoir trajectories of R2 in test system 2 for the inflow scenarios shown in Figure 5.16, using $\alpha = 5$. The red line shows the expected reservoir trajectory, and the light blue bars represent the production discharge through P1.	55

List of Tables

3.1	Symbols and description of the variables used to discuss scenario reduction.	19
4.1	The table presents an overview of the computational tests conducted in the evaluation of SHARM.	30
5.1	The number of scenarios and corresponding CPU-times T are shown for different degrees of reduction ϵ_{rel} and test cases. . .	43
A.1	List of the sets used in the model example of Section 3.4. . .	72
A.2	List of the parameters used in the model example of Section 3.4.	72
A.3	The stochastic variable used in the model example of Section 3.4.	72
A.4	List of other variables used in the model example of Section 3.4.	72
C.1	The table shows detailed results from Test 3. The same results are presented in Figure 5.9. The deterministic objectives were in the range of €7 - 12 million.	76
C.2	The table shows detailed results from Test 4. The same results are presented in Figure 5.10. The deterministic objectives were in the range of €2 - 3 million.	76
C.3	The table shows detailed results from Test 1. The same results are presented in Figure 5.1. The deterministic objectives were in the range of €16 - 18 million.	77
C.4	The table shows detailed results from Test 5. These results are not presented anywhere else in the report. The deterministic objectives were in the range of €8 - 13 million.	78
C.5	The table shows detailed results from Test 6. These results are not presented anywhere else in the report. The deterministic objectives were in the range of €3 - 4 million.	78

Chapter 1

Introduction

1.1 Background and motivation

Hydro power is a valuable natural resource, and Norway's mountains, glaciers and fjords provide excellent conditions for its exploitation. It is the dominant form of power production in this country, constituting around 99% of the total production in 2012 [28]. Many power plants are connected to a reservoir where water can be stored. Along with low startup costs for the generating units, this means that hydro power has great flexibility for production scheduling.

However, there are several obstacles to overcome. A major problem with the northern climate is that in the winter, when the demand for electricity is at its highest, the inflow to the water reservoirs is almost completely absent due to the low temperature. Another problem is the unpredictable nature of hydrological inflow. It is difficult to foresee how much water will run into the reservoirs in the next week, and it becomes even harder as the time horizon is extended. Finally, the electricity produced in Norway is traded on the open Nordic power market. The question of what is most beneficial, trading and producing now or saving the water for later, arises. This decision is based on the uncertain future development in the electricity spot market.

These examples are only some of the topics that are assessed by *hydro power scheduling*. Adequate planning and scheduling routines are of vital importance when solving the problems mentioned above.

Following [4], the objective of hydro power scheduling is to “utilize the available water resources to satisfy the demand for electricity while obtaining the optimal result and satisfying all relevant constraints”. The term “optimal result” in this statement refers to the earnings of the power producers who carry out the scheduling. Its meaning has changed somewhat over the last years, as will be discussed below.

The objective can also be stated as the maximization of social welfare, where social welfare is defined as the sum of consumer and producer surplus.

In the context of hydro power scheduling, the most important constraints are the ones concerning the generation and transmission system, the environmental constraints and demand characteristics. These constraint poses limits to the amounts of electricity that can be produced in specific areas, how much can be transferred between areas, when the plants can run and many other factors.

A major shift in the Norwegian power system took place in 1991, when the power market was deregulated through the Energy Act. Before this, the price and size of production quotas was decided by the government to ensure a steady supply of electricity. In this situation the producers used hydro power scheduling to minimize their cost, as the income was predetermined. However, it was realized that an excess capacity had been built up in the system, both in the transmission grid and the production. By giving the producers access to the transmission grid, the available power could be utilized more effectively. This led to the introduction of an open Norwegian power market. Soon the whole Nordic area was included, and the goal of the producers shifted to focus on profit maximization. It is shown in [4] that this goal is in compliance with the paramount target of maximizing the social welfare.

A traditional distinction between long-term and short-term scheduling is made, as the methods used to deal with these areas differ significantly. The need for implementable operational plans for the next days means that short-term planning must have a high degree of detail. Presently, the time horizon for such scheduling operations is 1 to 2 weeks, and it is solved by a deterministic optimization procedure.

Problems such as deciding the reservoir levels at the start of the winter depletion season obviously require a longer time horizon. But when the planning period is extended, the assumption that inflow and spot prices are deterministic quantities is no longer valid. The use of stochastic optimization techniques and the longer planning horizon means that the degree of detail in the models must be drastically reduced to avoid unacceptable computation times.

Power scheduling is an active field of research, and the interest in stochastic modeling and programming has been a natural development in the past decades. Many of the governing parameters in the operation of a hydro power system are inherently stochastic, such as inflow, prices and demand. Inflow is dependent on both area-specific and more general hydrological conditions, and the size of these systems makes it difficult to foresee. With the introduction of deregulated power markets, the prices have become increasingly volatile and more coupled to demand. Another important factor is the inclusion of wind power, which has a limited, uncertain availability and no capacity for storage.

This uncertainty is present also in the short-term period, which in the Nordic market currently is considered by deterministic scheduling tools. The

thesis project will address this topic by evaluating a prototype of a stochastic model for short-term hydro power generation scheduling.

1.2 Purpose of the work

The background for this thesis is the development of the stochastic short-term scheduling tool SHARM. SHARM is an acronym for *Short-term Hydro Application with Risk Management*, and the prototype has been developed at Sintef Energy Research within the KMB-project “Optimal Short-term Scheduling of Wind and Hydro Resources”. It is based on the model SHOP¹, which is in operational use by many producers in the Nordic market.

Stochastic models for short-term scheduling have been proposed before, in articles such as [9], [14] and [18]. The focus of these publications is mainly on the model development, and the effect of the proposed model is investigated on one or more test cases. Such tests are typically constructed to highlight situations in which the positive effect of the new model is visible. It should also be noted that the test cases often contains simplifications compared to the real-life problem. These works conclude that there is a clear potential for increasing revenues by applying stochastic scheduling models, if high quality input data is available.

This thesis will not contribute a new model. The purpose of this work has been to evaluate the effect of using a stochastic scheduling model, compared to the current deterministic one, in a realistic setting. As opposed to the efforts mentioned above, the focus will be more practical and turned towards the real-life application of a model. Hopefully, this thesis can contribute to a clearer understanding of the differences between using a stochastic and a deterministic model in short-term hydro power scheduling problems. For a producer, some relevant questions include:

- What is the expected profit of replacing the current scheduling tool by a stochastic version?
- Is it necessary to consider uncertainty in the whole system, or does it suffice to focus on specific reservoirs or system states?
- What will this mean in terms of additional work, e.g. generation of stochastic input and CPU-time?

For the model developers, it is important to consider feedback and requests from the users. As an example, does the current prototype provide the desired output for operational use? These are some of the questions that will be assessed in this work.

¹Short-term Hydro Optimization Program, developed by SINTEF Energy Research.

The advantages of taking uncertainty into account will depend on the quality of the stochastic input, but a stochastic model will presumably be more robust. As a side benefit it could also provide realistic results for longer periods. It is thus natural to consider using a stochastic short-term model for tasks beyond the scope of the currently used tools. Examples of such tasks could include the providing of better and more robust water values, or a more direct link to a long-term model. This thesis will not address these questions directly, but some ideas for further application areas are mentioned in Chapter 7.

1.3 Structure of the report

This section has presented the background and motivation for considering the problem at hand. A brief overview of similar work on stochastic short-term scheduling has been given, along with a description of the approach taken in this thesis. In the following text, different aspects of the stochastic approach to short-term scheduling will be considered, discussed and tested.

In Chapter 2, background information on hydro power scheduling will be presented. A short general introduction will be given in Section 2.1, but the emphasis will be on short-term scheduling. The scheduling tools developed by Sintef Energy Research will be described, as well as the application of such models.

Chapter 3 provides an introduction to stochastic programming, focusing on applications to hydro power scheduling. A description of the SHARM prototype is included, along with a theoretical foundation for the supplementary scenario tree generation algorithms.

A concretization of the goals for the evaluations is given in Chapter 4. Here the areas of investigation and methods of comparison will be discussed. The chapter will motivate and present the and tests that have been carried out in this work.

The results of the comparisons and evaluations are presented in Chapter 5, along with a discussion of the findings. Finally, a conclusion and suggestions for further work will be given in Chapter 6 and Chapter 7, respectively. Supplementary information, such as lists of parameters and detailed results, is included in appendices.

Chapter 2

Deterministic short-term hydro power scheduling

This chapter will give a short introduction to hydro power scheduling in general, and a more elaborate presentation of short-term scheduling. Section 2.1 describes hydro power scheduling in Norway, with mention of the most widely used models. Short-term scheduling is discussed in Section 2.2, with focus on application areas in the Nordic power market. Finally, a description of the scheduling software SHOP is included in Section 2.3.

2.1 Hydro power scheduling

The traditional partitioning of the scheduling tasks is still applied by most producers in the Nordic market. The planning horizon is decided by the scheduling objectives, and typically spans from the next day up to around 5 years. Today, the uncertainty is taken into account through the long-term and seasonal models. The division of this period into subtasks can be seen in Figure 2.1.

Long-term scheduling represents the strategic management of the resources belonging to the producer, in interaction with the whole power system. In the models EMPS¹ and EOPS², which have been developed by Sintef Energy Research, this task is solved by a two-step process. First, stochastic dynamic programming is used to find expected marginal water values for an aggregated model of the total hydro energy system, resulting in an optimal strategy. When this is done, the hydro system operation is simulated for different price and inflow scenarios using a more detailed hydro model [27].

The seasonal scheduling acts as a coupling stage between the long- and

¹EFI's Multi-area Power market Simulator, known as Samkjøringsmodellen in Norwegian. EFI is an acronym for Energiforsyningens ForskningsInstitutt, a former name for SINTEF Energy Research.

²EFI's One-area Power market Simulator.

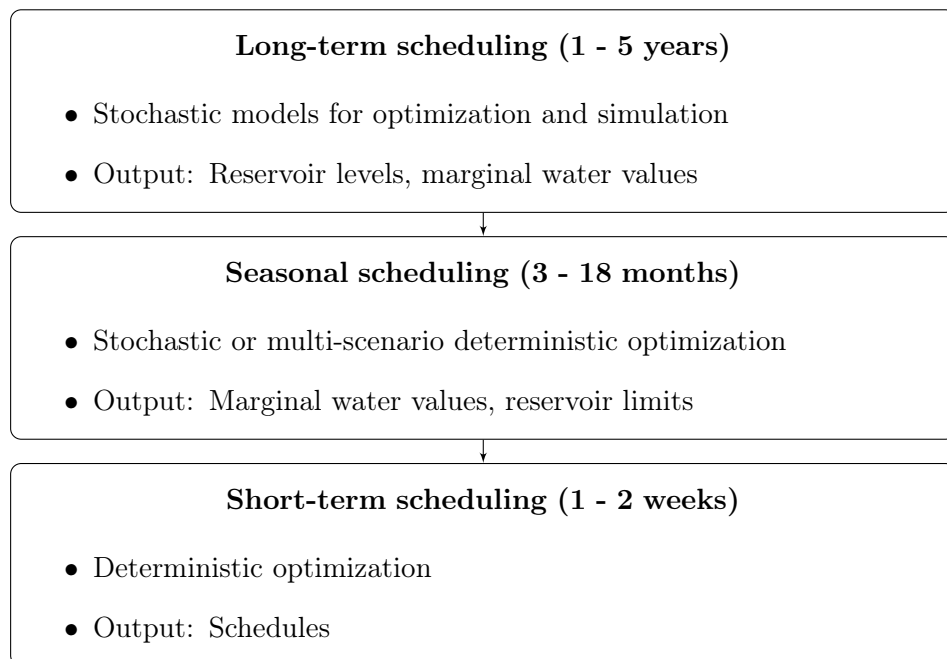


Figure 2.1: A coarse depiction of the current hydro power scheduling hierarchy, as described in [4].

short-term models, and provides *water values* and reservoir limits as boundary condition for the short-term optimization. The water value is a measure of the expected marginal value of the energy stored in the reservoir, and is used to quantify the revenue of storing water for later use. In the framework for hydro power scheduling developed by Sintef Energy Research, this task is solved by multistage deterministic optimization.

More information on the modeling and solution methods for long-term and seasonal scheduling can be found in [27]. For a thorough introduction to all topics within hydro power scheduling, see [4].

2.2 Short-term scheduling

As mentioned in the introduction, the main reason for decomposing the scheduling problem is the contradictory requirements of detailed modeling and a long time horizon. In the short term, hydro producers need operational plans with a time resolution of hours or minutes, that demands a detailed description of all the system components. The different system elements, such as reservoirs and plants, may be arranged in complex, cascaded and often time dependent topologies. Reservoir storage capacities may differ significantly, and long water travel times means that the decisions are coupled over multiple time steps. Each plant can have several generation units that may

or may not be running, and the relation between discharge and production is nonlinear and often not convex. Combined, this amounts to a challenging task in its own right. To be able to solve such problems, the assumption that all input parameters are known is essential. This assumption can however not be considered valid for more than a few days, limiting the time horizon to a week or two.

So far, the incorporation of uncertainty has been considered too computationally costly in the short-term model [16], and the stochastic nature of inflow and prices is taken care of through the long-term and seasonal models. As described in Section 2.1, these provide boundary conditions for the short-term model, currently in the form of marginal water values for each reservoir at the end of the short-term period.

While having a limited time horizon, short-term scheduling covers many different tasks. A short description of the different tasks and the timeline for short-term scheduling is included in this section. For a more detailed description, see [30] and [4]. The main areas for which short-term scheduling tools are applied, as stated by e.g. [9], are listed below.

- Day-ahead bidding in the Elspot market.
- Establishing a production plan in accordance with the day-ahead commitments.
- Trading in the intraday Elbas market.
- Real-time balancing.

The Elspot market is the main market for electricity in the Nordic region. All participants in the market must submit a price-volume bid to Nord Pool Spot, stating how much electricity they will buy or sell at specific prices for every hour during the next day. The bids must be submitted before 12:00, and are valid for the next day from midnight to midnight. When all bids are received, the market operator determines the price for the next day as the intersection between aggregated demand and supply curves. Due to constraints in the transmission grid, this price may differ between different geographical areas. The plants and reservoirs controlled by a producer within such a price area are typically scheduled together.

Once the Elspot price is determined, it is compared with each participant's bid to decide the traded volume in every hour for that participant. The producers receive their load obligation around 13:00.

Short-term scheduling models, such as SHOP, are valuable tools in the preparation of day-ahead bids for most producers. However, the prices used in the bidding phase are not known and must be predicted by some model. The models are not perfect, and there is of course a possibility that the day-ahead commitments does not comply entirely with the planned production.

It is thus necessary to adjust the operational plans after the clearing of the Elspot market, and this is another important task for the short-term scheduling models. Again, the systems within a price area are rescheduled together.

Due to different factors, imbalances may occur for both a producer and the system after the settlement of the Elspot market. Two different markets are available to remedy this, and short-term scheduling models acts as decision support in both cases. Below follows a short description of the Elbas and regulating power market.

After the clearing of the Elspot market and the preparation of the production plans, the producers can trade in the Elbas market. The time span between the Elspot settlement and the actual delivery may be up to 36 hours, during which the consumption and production situation may change. Thus there may be a need for a market player to trade in this period, and this can be done in the Elbas market. The products are one-hour long power contracts, which can be traded continuously up to one hour before delivery. The purpose of this market is to act as a balancing for the Elspot market, creating an alternative to the real-time balancing market described below.

In each of the Nordic countries, a Transmission System Operator (TSO) is responsible for maintaining the stability of the electricity. Technically, this is achieved by holding the frequency in the transmission grid stable at 50 Hz. If imbalances between consumption and production occur, the frequency will deviate and there will be a need for balancing. The TSO must then buy or sell regulating power from the participants in what is called the regulating or balance power market. The market players can submit hourly bids for up- or down regulation, and the bid may or may not be accepted depending on the TSO's needs.

In addition to the markets described above, short-term scheduling is applied to several reserve markets, such as LFC, RKOM, RK, and FNR/FDR. Here the producers are paid not to utilize some of their capacity, so that it can be used by the TSO if needed.

Summing up, it is evident that a short-term scheduling model is a valuable tool for a power producer. It is used continuously in the course of the day, as decision support for many different planning tasks. A timeline for the utilization of short-term scheduling models is shown in Figure 2.2.

2.3 SHOP

SHOP is the short-term tool in the hydro scheduling framework developed by Sintef Energy Research. The underlying assumption is that inflows and prices are known, so that the problem becomes deterministic. It is solved by using a Successive Linear Programming (SLP) approach, and can incorporate use of mixed integer programming. As input, the model takes time series

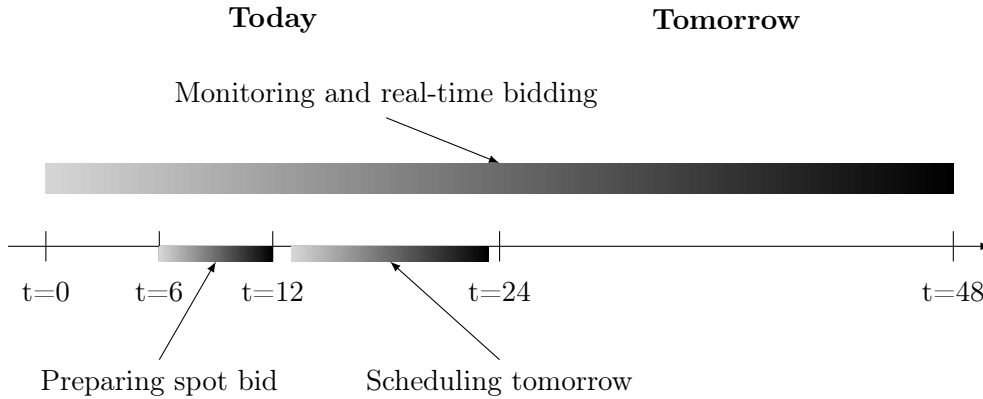


Figure 2.2: A typical timeline for the utilization of a short-term hydro power scheduling program in the Nordic market. Based on a figure included in [14].

for inflows and prices, end point value criterias from the seasonal model, a detailed description of all the system elements and connections, and any schedules or constraints that may apply during the planning period.

This section contains a description of the different elements and equations that constitute the SHOP model. Such a description is important in this context as the stochastic model SHARM, that is the main focus of this thesis, is heavily based on SHOP. The stochastic model will be further discussed in Section 3.4, where a specification of which elements are shared by the two models is included. A more in-depth description of the SHOP model is given in [2], [1], [16] and [15].

The SHOP model has two main characteristics. The first is the reservoir volume based on the reservoir balance equations (2.1), and the second is the modeling of the hydro power plants. In addition to reservoirs and plants, features such as junctions and spill-, flow- and bypass gates can also be included. Elements such as generators and pumps are specified within the plant description. The topology of the system is decided by specifying all the elements and their interconnections in a model file. An example topology of a cascaded hydro power system is shown in Figure 2.3.

Reservoir balance equations are the basic constraints in the model, introducing coupling in time as well as in space. All gates and plants must be associated with a reservoir, making the reservoirs the main connecting nodes of the system. Additional descriptions of the reservoirs are given through volume/head and flow descriptions, and minimum/maximum constraints on variables such as water level and discharge. The reservoir balance equations are based on the conservation of water in the system. The amount of water present in the reservoir at a time t is given by the water present at the previous time step plus the inflow to the reservoir minus the flow out from the

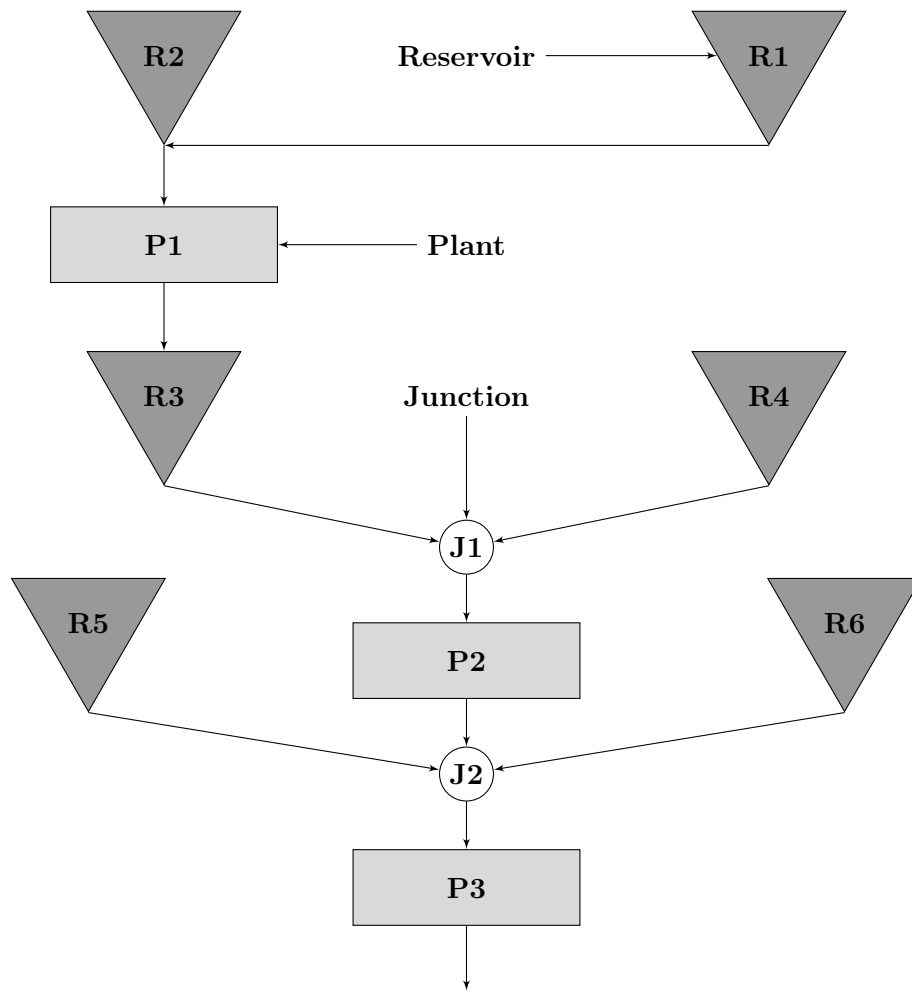


Figure 2.3: Example topology of a hydro power system. Based on a figure included in [1].

reservoir through gates or spill. Translated into equation form, this becomes

$$-X_i(t-1) + X_i(t) - \sum_{j=1}^{n_u} q_j^u(t - \tau_j) + \sum_{j=1}^{n_d} q_j^d(t) = 0. \quad (2.1)$$

Here, $X_i(t)$ denotes the volume stored in reservoir i at the end of time period t . The symbols n_u and n_d represents the number of upstream and downstream elements, respectively. In a similar fashion, q_j^u represents the inflow from upstream elements and q_j^d the flow from the reservoir to downstream elements. τ_j is the time delay in the flow from upstream.

Each power plant consists of one or more generating units. A generation unit consists of a turbine that is propelled by the falling water, and a generator that transforms the mechanical energy of the rotating turbine into electric energy. Each unit can be modeled with a separate efficiency curve along with minimum and maximum constraints on production and discharge. For a given unit j , the relation between production g_j and inflow q_j is

$$g_j(q_j) = k\rho g q_j h(q_j) \eta(q_j). \quad (2.2)$$

Here, k is a conversion factor, ρ is the density of water and g is the gravitational acceleration. h denotes the pressure head of the plant and η is the efficiency. The head dependency of the production will be discussed later in the report, especially in what is called Test 8.

The reason for choosing the plants as main elements instead of the individual units is that the hydraulic couplings within the plant must be accounted for. Water is transported from the reservoir through tunnels and penstocks to the generating units and further downstream. The flow through the tunnels and penstocks affects the net head, and hence the production, of the plant due to head losses, so the internal couplings are important factors. The model also includes the possibility of using mixed integer programming (MIP) to decide which units should be running at any given time. The term MIP is used for optimization problems where some, or all, of the decision variables are discrete. It typically refers to linear problems with the added constraints that some of the variables can only take integer values.

The optimization procedure in SHOP is based on successive linear programming. The solution is found through a series of main iterations, where in each iteration one of two different modeling modes is used [2]. The first mode is denoted *full description* or *unit commitment (UC) mode*, and the second mode is called *incremental description* or *Close-in mode*. Common for all iterations is the building and solving of a linear optimization model. This solution represents the optimal decisions based on the current system state approximation. Each of the iterations refines the model based on the results from the previous iteration. The refinement is performed in terms of new, linearized descriptions of reservoir levels, unit efficiency curves and

gate discharges. The reason for using an iterative approach is that some of the constraints and nonlinear elements in the model depend on production and discharge decisions. In the first iteration these decisions are unknown, and can not be taken into account.

The UC-mode aims at finding a commitment plan for the generating units in the system. This can be achieved either by using an aggregated plant description, or by applying MIP to model the unit startup costs. In either of the two approaches, a simplified description of the tunnel losses is used. If using discrete variables, the model is solved using a branch and bound technique. Other nonlinear elements, such as head optimization or reservoir level dependent spillage, can also be considered in this mode. An iteration in UC-mode results in a fixed unit commitment plan, and linearized descriptions of production levels and reservoir trajectories.

In Close-in mode it is known which units run at any given time, as the commitment plan is fixed from the UC-mode iterations. This enables the calculation of exact efficiency curves, resulting in a more accurate model. When the exact losses in the waterways can be accounted for, the true relation between discharge and production for each plant can be found.

It is usually advised to perform 2 or 3 iterations in each mode [16], but this can be decided by the user based on the task at hand. The full solution procedure is described in the following algorithm:

Algorithm 2.1 SHOP solution procedure

Initial input: system description, initial values, boundary conditions;
UC-mode
Input: Initial reservoir levels;
repeat
 Choose aggregated plant topology or MIP for startup cost modeling;
 Build and solve a model that is linearized around the previous solution;
 Update reservoir trajectories;
until Reservoir trajectories satisfactory
Close-in mode
Input: Fixed unit commitment plan, production levels and reservoir trajectories;
repeat
 Build and solve a model with linearization around the previous solution and exact efficiency curves;
 Update reservoir trajectories;
until Reservoir trajectories satisfactory

Chapter 3

Stochastic short-term hydro power scheduling

This purpose of this chapter is to introduce the concept of stochastic programming, and its application to short-term hydro power scheduling. First, a general description of stochastic programming is given in Section 3.1. Section 3.2 includes an overview of the utilization of stochastic programming models in the field of hydro power scheduling. A description of the scenario tree concept and an overview of scenario tree generation algorithms will be given in Section 3.3. Finally, the model used in the stochastic SHARM prototype will be presented in Section 3.4, and Section 3.5 will discuss the different models used to generate price and inflow scenarios.

3.1 Stochastic programming

The main difference between stochastic programming and its deterministic counterpart is that it includes some kind of uncertainty. According to [5], the term stochastic programming was introduced independently by several authors in the 1950's. It was observed that for many linear programs to be solvable, the values of the presumably known coefficients were not available. This led to the stochastic view of assuming that these parameters were random, and that their probability distribution was known and independent of the decision variables. Since then, stochastic programming has been applied in many different fields, such as finance, logistics, telecommunications and energy production and transmission. During the years, several types of standardized models have emerged. Among the best known are two- or multi-stage stochastic programs with recourse, models with probabilistic constraints, and integer stochastic programs.

The class of optimization models that will be considered in this thesis is two- and multi-stage stochastic programs with recourse. In broad terms, a first-stage decision is made and the expected utility of the consequences of

that decision is maximized. The result for the rest of the stages is an optimal strategy, dependent on the future realization of the stochastic variables. As opposed to deterministic optimization, which returns one optimal decision and one optimal value, the results from stochastic programs are generally not directly implementable.

As an example, a two-stage program can be considered. The current values, at time t_0 , of the stochastic variables are known, but their values at a certain future point t_1 are not. Solving such a problem will result in a decision for t_0 , and a strategy consisting of several decisions for t_1 that depends on the realization of the stochastic variables at this point. Multi-stage programs generalize this case to allow for realization of the stochastic variables at several future stages.

In the models considered in this work, the uncertainty will enter the optimization problem through stochastic variables in the objective function (spot prices) and the constraints (inflows). The stochastic variables are assumed to belong to some probability distribution, which in the context of this thesis is approximated by a discrete distribution. This is the case for all real-life applications, and only some trivial cases can be solved using a continuous distribution [24].

A widely used technique for discrete modeling of uncertainty is the construction of scenario trees. This is the form in which the probability information will be specified throughout the report. Scenario trees consist of a set of nodes for each time step in the model, and a branching structure connecting them. Each node contains a possible realization of the stochastic variables. A scenario consists of the nodes lying on a unique path connecting the start node with an end node. For a deterministic program the input is given by only one such scenario, which implies the loss of possibly significant information.

In hydro power scheduling, there are several problems that a deterministic model can not solve satisfactorily. To demonstrate that stochastic programming can handle one such situation, an example is included.

Example: *Consider a hydro power system consisting of 2 interconnected reservoirs and a plant. The topmost reservoir is large, and its water is discharged into the tiny second reservoir before passing through the plant. What makes deterministic programming unsuitable in this case, is that it will maintain the reservoir level in the small reservoir at the maximum. This is done to achieve the highest possible effect from the generating units, as can be seen from Equation (2.2). However, this will result in spillage if the real inflow is even a tad higher than prognosticated. Currently such cases are handled by operators manually specifying a lower maximum reservoir level in the small reservoir. This is done to maintain a safety margin in the reservoir, but is by no means optimal with respect to maximizing the profit*

from the plant.

Assume that good inflow scenarios are available, covering the possible outcomes for the following week. A stochastic model can then see the possibility of higher inflow, and will due to severe penalties on spillage keep a safety margin to the maximum reservoir level while still maximizing the expected profit. This example will be considered more thoroughly later on in the report, as it is identical to test system 2 described in Section 4.1. Several computational test will be performed on the watercourse, e.g. a qualitative analysis of the reservoir handling by the stochastic prototype.

It should be noted that, as the future is unknown, there is no guarantee that a stochastic model will yield a higher utility than a deterministic one. The single scenario may by chance be correct, providing perfect information for the deterministic model. It is however expected that a stochastic model will outperform a deterministic over time, assuming a high quality representation of uncertainty.

3.2 Stochastic optimization in hydro power scheduling

In Chapter 2, the programs for hydro power scheduling developed by Sintef Energy Research were described. Modeling of uncertainty in these programs is considered in the long-term and seasonal models, through stochastic dynamic and multi-scenario deterministic programming. While these models are prevailing in the Nordic market, there exists a variety of other methods and models for both long- and short-term scheduling. It is not the purpose of this work to compile a comprehensive summary of stochastic optimization models for power scheduling, but this section will present some ideas and refer to more detailed reviews from the literature. The focus will be on stochastic models for short-term scheduling.

In connection with the development of the SHARM prototype at Sintef Energy Research, a brief literature review was conducted by Follestad [12]. In addition to pointing out the most important models, this internal report includes references to more general reviews covering short-term, seasonal and long-term scheduling, such as [33]. A more recent review referred is [25], which focus on stochastic programming models for short-term scheduling. These are the main sources used in this thesis.

For short-term applications, several approaches are proposed. The main division is between stochastic scheduling models and models for determining optimal bidding strategies and bidding curves. Both two- and multi-stage models have been developed, as well as extensions to hydro-thermal systems and the Elbas market. Many models include unit commitment and risk management. Most of the models use a scenario tree and solve a deterministic

equivalent. A range of different methods for scenario tree construction have been developed, as will be discussed in Section 3.3.2.

An example of a multi-stage stochastic model for determining the day-ahead production plan is the one proposed in [9]. This model considers uncertainty in both spot prices and inflows, and is applied to an example topology with 2 reservoirs. The same authors have also published a two-stage stochastic model for determining optimal bidding curves under price uncertainty [8]. An extension of these models to include trading in the Elbas market is described in [7]. Some of these models have been investigated further in a student project [26] and master's thesis [31] at NTNU, which were carried out partly at Statkraft.

3.3 Scenario tree generation and reduction

The uncertain parameters introduced in a stochastic program must be described by distributions in the single-period case, or stochastic processes in the multi-period case. In this text, the term distribution will be used in both cases unless stated otherwise. As mentioned in the introduction to this chapter, stochastic programs can not be solved with continuous distributions except for in some trivial cases. When considering real-life applications, as is the case here, it is necessary to apply approximate discrete distributions to describe the stochastic parameters. The distributions must have finite support, i.e. a finite number of scenarios or outcomes. An especially common arrangement of the stochastic input is the scenario tree [5], [24], which will be described here. In addition to describing the concept of scenario trees, this section will give a brief overview of the most common techniques for scenario tree generation and present the algorithms that will be used later in the thesis.

3.3.1 Scenario trees

Scenario trees are based on the requirement that there is a one-to-one correspondence between the previous stages (from 1 to $t - 1$) and one of the nodes at stage t . A tree consists of a set of nodes for each time stage of the model, with only one parent node at the first stage. This stage, representing the present, is assumed known. Every node contains a possible realization of the stochastic parameters at that time stage. Each node is connected to only one parent node in the previous stage, but may have several descendants in the next stage. A *path* or *scenario* is a possible realization for the whole time period ($t \in [1, T]$), consisting of one node for each time step from the start node to an end node (node in stage T). This means that a scenario tree contains as many paths as there are end nodes. The connection between two nodes in succeeding stages is called a *branch*, and these are also parts of the scenario.

A transition probability is associated with each branch, and the sum of all transition probabilities from a node to connected nodes in the next stage is 1. The probability of an entire scenario is found by multiplying the transition probabilities of its branches. Naturally, the probabilities of all scenarios sum to 1 as well, as they describe the sample space of the probability distribution.

A special case of a scenario tree is the scenario fan. The fan only has branching at one stage, thus representing a two-stage stochastic program. An example of a scenario tree and a scenario fan is given in Figure 3.1. In many applications, real-life stochastic data comes as scenarios in the form of time-series where the first stage is known and equal. This means that the data is on scenario fan form already.

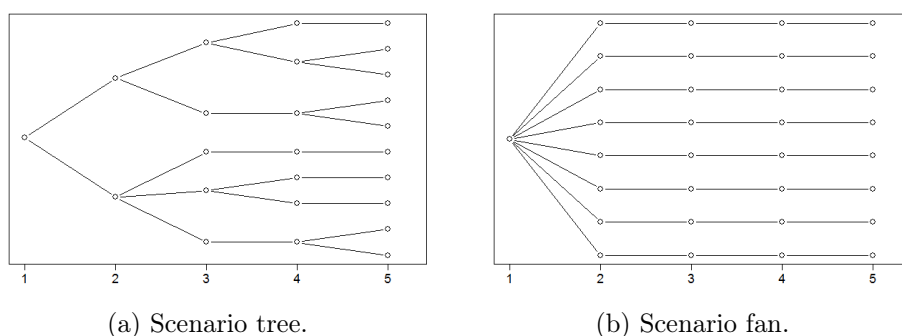


Figure 3.1: A general five-stage scenario tree is shown on the left, and a scenario fan on the right.

But why can not the scenario fan be used as input to the stochastic program directly? There are several reasons for this, but it should be noted that the fan is indeed a tree and can be used as input as it is. However, this will lead to a two-stage model as the fan does not include branching beyond the second stage. If a stage-wise decision process is considered, a tree structure with branching at more than one stage is needed. Another problem with the fan is the large number of nodes, which may lead to unnecessary high computation times.

High CPU times is a general problem in stochastic programming. To remedy this, reduction algorithms have been developed to decrease the number of nodes that are considered. Such algorithms apply to both trees and fans. It has been shown, e.g. in [6], that even with a large reduction of scenarios much of the information is intact. In a numerical example, a 50% reduction of the tree leads to a loss of approximately 10 % in the relative accuracy of the distribution representation. A similar investigation is performed in [14], where the effect of reduced trees on the optimization results is studied. It is found that close to optimal results can be obtained with a significant reduction of nodes and computation time.

3.3.2 Generation methods

The stochastic program under consideration is only an approximation to the real problem, and the quality of this approximation depends heavily on the quality of the scenario tree. Hence, much effort has been devoted to the development of scenario tree generation methods. In [24], a distinction between *pure* and *related* scenario generation methods is applied. The most important pure methods are considered to be *conditional sampling* [29], *matching statistical moments* [35] and *path-based methods based on probability metrics* [6], [19], [20]. Related methods cannot construct entire trees alone, but can be used either as a part of a pure method, such as *scenario reduction*, or as an incorporated part of a solution procedure, such as *internal sampling*.

According to [24], conditional sampling is the most common method for generating scenarios. The stochastic process describing the uncertain parameters is assumed known, and several values from the process are sampled at every node of the tree. This can be done by either sampling directly from the distribution, or by evolving the process according to some explicit formula. As traditional methods for sampling only works for univariate random variables, a separate sampling for every marginal (univariate component) must be performed to generate a random vector. The samples are usually combined all-against-all, which means that the size of the tree will grow exponentially. A more detailed discussion of this method is given in [29].

Moment matching is appropriate if the distribution functions of the stochastic variables are not known. In this method, the marginals are instead described by their moments, such as mean, variance, skewness and kurtosis. The correlation matrix must also be specified, possibly together with other properties such as percentiles or higher co-moments. A description of such a method can be found in [35].

The third class of methods, and the one that will be utilized in this project, is the path-based methods. Here the starting point is a set of scenarios with the same starting point, typically represented by a scenario fan. These scenarios are produced by for example a fundamental model or a model based on historical data. By clustering scenarios together at all stages except for the last one, a tree structure can be built from the original fan. The next subsection will provide a more thorough introduction to a certain variety of such methods, using probability metrics to decide which scenarios to group together.

3.3.3 Generation by scenario tree reduction

This section will present the method for scenario tree generation and reduction that is implemented as a supplement to the SHARM prototype. The method is an example of a path-based method based on probability metrics, as described in the previous section. It assumes that a set of scenarios in

the form of a scenario fan is available, and clusters these at different stages using probability metrics. The theoretical and practical foundation of the method has been developed by the group of Werner Römisch at Humboldt-University Berlin and their associates. A more complete description can be found in the papers written by the group. The theoretical foundation is given in [6], and a refinement of the algorithms is proposed in [19]. Applications to power scheduling and management are discussed in [20] and [18]. It can be noted that an implementation of this framework is available in the commercial optimization software GAMS [17].

The sample space of the distribution is represented by a set of scenarios. The generation of such scenarios will be discussed further in Section 3.5, but in this section they are considered given.

A key element of the method is the concept of *probability metrics*. In short, this is a measure of distance between two probability distributions. The main idea is to delete a set of scenarios and add the corresponding probabilities to the closest remaining ones, to create a reduced distribution. This should be done in such a way that the distance between the original and the reduced distribution is as small as possible with respect to the chosen metric. This approach can be used either to reduce the size of an existing tree, or recursively to transform a fan into a tree with branching at several stages. To clarify this concept, it is convenient to introduce the notation used by Römisch et al. in e.g. [18]. Table 3.1 presents the new variables needed to discuss the scenario reduction method.

Symbol	Description
$\xi, \{\xi_t\}_{t=1}^T, \tilde{\xi}, \{\tilde{\xi}_t\}_{t=1}^T$	s -dimensional stochastic processes with parameter set $\{1, \dots, T\}$.
$\xi^i, \tilde{\xi}^i$	Scenarios (sample paths of ξ and $\tilde{\xi}$).
p_i, q_j	Scenario probabilities. $p_i, q_j \geq 0, \sum_i p_i = \sum_j q_j = 1$.
P, Q	Probability distribution of the processes ξ and $\tilde{\xi}$ respectively.
N	Number of scenarios in the initial scenario set.
J	Index set of deleted scenarios.
$\#J$	Cardinality of the J , i.e. the number of deleted scenarios.
$n = N - \#J$	Number of preserved scenarios.
ϵ	Tolerance for the reduction.
$c_t(\xi^i, \xi^j)$	Distance between the scenarios $\{\xi^i\}_{\tau=1}^t$ and $\{\xi^j\}_{\tau=1}^t$.

Table 3.1: Symbols and description of the variables used to discuss scenario reduction.

In [6] and [19] the authors argue that the *Kantorovich distance* D_K is the most suitable measure for use in power management problems. It is shown that multi-stage stochastic programs with recourse are stable with

respect to small perturbations in terms of a Fortet-Mourier metric, which can be estimated from above by the Kantorovich functional or distance. For discrete distributions with finite support, D_K is the solution of a linear transportation problem known as a Monge-Kantorovich mass transportation problem. This is given by

$$D_K(P, Q) = \inf \left\{ \sum_{i=1}^N \sum_{j=1}^{\tilde{N}} \eta_{ij} c_t(\xi, \tilde{\xi}) : \right. \\ \left. \eta_{ij} \geq 0, \sum_{i=1}^N \eta_{ij} = q_j, \sum_{j=1}^{\tilde{N}} \eta_{ij} = p_i, \forall i, j \right\}. \quad (3.1)$$

The function c_t should satisfy a selection of properties listed in [6], and is typically chosen as

$$c_{t,r}(\xi, \tilde{\xi}) = \max(1, \|\xi - \xi_0\|, \|\tilde{\xi} - \xi_0\|)^{r-1} \|\xi - \tilde{\xi}\|. \quad (3.2)$$

Here, $\|\cdot\|$ is any norm in the euclidean space \mathbb{R}^s and ξ_0 is some fixed element in the same space. In the scenario tree construction algorithms presented below $r = 1$, so that the function c_t coincides with the metric induced by the norm on \mathbb{R}^s . In this work, s corresponds to the combined number of spot price and inflow scenarios.

By the assumption made earlier, the probability distribution P of the stochastic input data to our problem is approximated by finitely many scenarios. The goal is to reduce the number of scenarios while preserving as much of the original information as possible. A natural problem to consider at this point is to find the closest distribution Q to P , when the number of scenarios to be deleted, $\#J$, is known. For this problem, the minimal distance between the two distributions in terms of D_K , as well as an optimal redistribution rule for the probabilities, can be found explicitly. This is proved in [6]. The minimal distance $D_{K,J}$ and the probabilities q_j of the preserved scenarios $\xi^j, j \notin J$, of Q is given by

$$D_{K,J}(P, Q) = \sum_{i \in J} p_i \min_{j \notin J} c_t(\xi^i, \xi^j), \quad (3.3)$$

$$q_j = p_j + \sum_{i \in J(j)} p_i, \quad (3.4)$$

$$J(j) = \{i \in J : j = j(i)\}, j(i) \in \arg \min_{j \notin J} c_t(\xi^i, \xi^j), \forall i \in J.$$

The set $J(j)$ defined above is the set of all deleted scenarios to which the preserved scenario j is the closest preserved scenario. Now the optimal

choice of index set J , with $\#J$ fixed, is given by the solution of the *optimal reduction problem*

$$\min \left\{ \sum_{i \in J} D_{K,J} : J \subset \{1, \dots, N\}, \#J = N - n \right\}. \quad (3.5)$$

The problem (3.5) can be shown to be NP-hard. It is however not always the case that $\#J$ is given. A more realistic situation may be to find a distribution Q such that J has maximal cardinality. That is, delete as many scenarios as possible while keeping the distance $D_{K,J}(P, Q) \leq \epsilon$, with ϵ being some predetermined accuracy. This is the *maximal reduction strategy* formulated in [18].

In their papers, the group of Römisch suggests two fast heuristic algorithms to solve this problem, exploiting the structure of the objective. In the special cases of $\#J = 1$ (deleting one scenario) and $\#J = N - 1$ (keeping one scenario), the optimal reduction problem becomes more easily solvable. These cases form the basis for the two algorithms *Simultaneous backward reduction* and *Fast forward selection*.

The backward reduction strategy suggests repeating the optimal deletion of a single scenario recursively until the prescribed number of scenarios is removed. If a strong reduction is the goal, it may be advisable to repeat the optimal selection of a single scenario until the desired number of scenarios is reached. This is the basic concept of the forward selection strategy.

In [19], explicit expressions for the computational complexity of both algorithms are derived. To reduce a set of $N \in \mathbb{N}$ scenarios to a subset consisting of $n \in \{1, \dots, N\}$ scenarios requires $b_N(n)$ and $f_N(n)$ operations, when using simultaneous backward reduction and fast forward selection respectively.

$$b_N(n) = n^3 - n^2 \left(\frac{3}{2}N + \frac{1}{2} \right) - n \frac{3}{2}(N + 1) + \frac{N^3}{2} + \mathcal{O}(N^2 \log N) + 2N^2 + \frac{3}{2}N, \quad (3.6a)$$

$$f_N(n) = \frac{2}{3}n^3 - n^2(2N + 1) + n(2N^2 + 2N + \frac{1}{3}). \quad (3.6b)$$

It is evident that forward selection will be faster for smaller n , while backwards selection performs better for higher n . It can be shown that the number n^* such that $b_N(n^*) = f_N(n^*)$ is approximately given by $n^* \approx \frac{N}{4}$.

As they have been described above, the algorithms delete or select an entire scenario from the original tree or fan. Another important task is the generation of a tree structure from a set of separate scenarios, typically in the form of a fan. The tree construction strategies proposed in [20] makes

use of the two algorithms recursively to bundle scenarios at each stage of the time horizon.

As a supplement to the SHARM prototype, versions of the two algorithms have been implemented by Sintef Energy Research. For tree construction the *Backwards construction* algorithm (Algorithm 3 in [20]) is used. In the case of scenario reduction, the *Fast forward selection* algorithm (Algorithm 2 in [18]) is implemented. The two algorithms are presented as Algorithm B.2 and B.1 in Appendix B, respectively. More information on the implementation can be found in [13], but some central moments will be described here.

SHARM is on a prototype stage and the available algorithmic options in the supplementary tools are limited. As already stated, only one algorithm is available for each task. Only one probability metric is specified, the *Wasserstein* metric, defined as

$$c_t(\xi, \tilde{\xi}) = \|\xi - \tilde{\xi}\|^r. \quad (3.7)$$

Currently one can choose between the *manhattan*, *euclidean* or *max* norms. It is recommended that $r = 1$ if only one variable (inflow or price) is stochastic, while $r = 2$ should be used if both are uncertain. In the construction process, the algorithm removes scenarios and bundle them together so that new scenarios may be created by joining 2 original scenarios at a given stage. This process can destroy some of the time correlation in the original scenarios, and is therefore not optimal for multi-stage process as pointed out in [21], [22]. It is however deemed good enough for this purpose [14]. When reducing the size of a given tree, one is guaranteed that the reduced tree consists of a sub set of the scenarios that make up the original tree.

The form of the generated or reduced tree can be specified through different parameters. One option is to fix the number $S - \#J$ of remaining scenarios. It is also possible to set the number of nodes at each time stage. Another option is to specify a *degree of reduction* ϵ_{rel} , which is defined by

$$\epsilon_{rel} = \frac{\epsilon_{red}}{\epsilon_1}. \quad (3.8)$$

Here ϵ_{red} is the absolute probability distance between the full and reduced tree. ϵ_1 is the distance between the full tree and the optimal single scenario tree decided by forward selection, and does hence represent the maximum probability distance for the scenario reductions. The degree of reduction varies between 0 and 1, with a higher value giving a higher reduction. $\epsilon_{rel} = 0$ means no reduction and $\epsilon_{rel} = 1$ means full reduction, i.e. only one scenario remains.

3.3.4 Evaluation of scenario tree quality

The quality of the scenario generation method is an important question in stochastic programming. According to [24] it is imperative to also consider

the link to the model used, as no scenario generation method is optimal for all possible models. The article by Kaut and Wallace presents a practical approach to scenario tree evaluation, stating two properties that successful methods should satisfy. The authors argue that the most important criteria is the quality of the solution obtained with the scenario tree, not how well the distribution is approximated. The two requirements that scenario tree generation methods should satisfy are:

- *Stability*. Solving the optimization problem using several trees generated from the same input should result in very similar objective function values. Both *in-sample* and *out-of-sample* stability should be considered [24], [14].
- *Bias*. The scenario tree should not introduce any bias compared to the true solution.

By *in-sample* stability it is understood that the objective value obtained by using two different trees should be similar. Testing for in-sample stability is straightforward. *Out-of-sample* stability means that the solution obtained by two different trees, evaluated at the original objective function, should be similar. If considering a reduction algorithm, the decisions resulting from using the reduced tree should be applied to the full tree. For the reduction algorithm to have out-of-sample stability, the resulting objective value should be similar to the objective value obtained by solving with the full tree. Testing for out-of-sample stability is more complicated, but methods do exist.

To not introduce *bias* into the solution, a solution of the scenario-based problem should also be an approximately optimal solution to the original, continuous problem. Testing of this property is however practically difficult, as it involves solving the problem with the true continuous process. This is usually not possible, and if it was, scenario trees would not be necessary any more. For further descriptions of these issues, see [24].

A further discussion focusing on the scenario generation and reduction methods used in connection with the SHARM prototype is found in [14]. There, in-sample and out-of-sample stability is tested using a simplified hydro power scheduling model with stochastic inflow. The paper proposes a method for evaluating out-of-sample stability of scenario trees in multi-stage models, based on evaluating successive first stage decisions.

The idea is to simulate the decisions that will be taken for all outcomes, when these decisions are based on a reduced scenario tree. To do this, a subtree is found at each node of the original tree, using that node as root node. The tree reduction method is then applied to the subtree, and a first-stage decision is found by running the stochastic model on the reduced sub-tree. After all the scenarios of the original tree have been traversed, the objective function is found from the first-stage decisions and the node probabilities.

This objective can be compared to the solution of the original tree, if it is available. The method guarantees that the objective using the reduced tree will be less than or equal to the objective of the full tree.

The results suggest that the scenario reduction method has good in-sample and out-of-sample stability properties. An adaptation of this method will be used later in this report, in the run-time tests described in Section 4.4.1.

An alternative evaluation method that is suitable for multi-stage models, is the *rolling horizon* approach proposed in [10]. It is similar to the method described above, but the length of the optimization period is kept fixed and hence rolled forward as each stage is evaluated. The rolling horizon approach has not been used in this work.

3.4 The SHARM prototype

This section aims at introducing the major concepts of the SHARM prototype. A final version of the model may differ from the current implementation, but in the remainder of this section the prototype will be referred to as SHARM.

SHARM is built on the SHOP framework. The main characteristics, i.e. the reservoir balance equations and the modeling of the power plants, are the same in the two models. The model description, e. g. specification of the system topology, modeling of internal couplings in the plants and volume/head relations in the reservoirs, is identical. Boundary values are taken from the seasonal model as in SHOP. Other constraints, such as maximum limits on reservoir volume and generator production, are also specified in the same way. The solution procedure in SHARM is the same as described for SHOP in Section 2.3. The point where the models differ, is in the definition of the parameters inflow and price. In SHARM, these can be given as stochastic in the form of a scenario tree.

Stochastic data is typically available as scenarios on a time series format. To construct scenario trees from this material, SHARM comes with several supporting tools. First, there is a program that combines the inflow and price scenarios to a fan on XML-format. This fan forms the input to an implementation of the methods described in Section 3.3.3, called SCENTREEGEN. This program enables the user to construct and reduce scenario trees by specifying parameters such as the degree of reduction ϵ_{rel} or the number of scenarios at each stage.

As SHARM at the moment is on a prototype stage, there are several features of SHOP that are not available. Examples of such features include ramping constraints and delta meter flows. Ramping constraints pose limits to how fast changes in e.g. discharge can happen. Delta meter flows are connections between reservoirs. An example of a delta meter flow is the

waterway between reservoirs R1 and R2 in Figure 2.3. Some of the missing features are quite common in the Norway, so this limits the amount of hydro power systems that can be considered in this thesis.

The SHOP model, and thereby the SHARM model, is quite extensive, so a complete description is not included here. A more detailed description of the common features can be found in the SHOP literature. What will be presented is an example of a stochastic scheduling model, which can be seen as a simplification of SHARM. This is done to give the reader an overview of the objective and the most important constraints. The example is taken from [14], and is also described in [34].

The example consists of a simple topology with a single reservoir and a plant with only one generator below. Inflow is the only stochastic variable, and the model is solved as a deterministic linear problem. The objective function $\pi(g)$ is the expected future profit from the water in the system, and is maximized with respect to the generation g . In addition to the objective, the constraints imposed on the system are also included. The notation is similar to the one used in Equation (2.1), and the new symbols will be explained when they are used. A full description of the symbols is included in tables A.1 - A.4 in Appendix A.

The objective function is given as

$$\pi = \sum_{i \in I} p_i g_i y_i + \sum_{i \in I^{end}} p_i w_i, \quad (3.9)$$

where p_i , g_i and y_i is the probability, generation and spot price for node i , respectively. w_i represents the end-value of the remaining water stored in the reservoir at node i . The first constraint specifies the water balance in the reservoir, with water filling at the end of a period. It is similar to the reservoir balance equation (2.1), and is defined as

$$X_i = \begin{cases} X_{j:j \in I_i^{par}} + q_i - g_i - s_i, & i > 1 \\ X_{ini} + q_i - g_i - s_i, & i = 1 \end{cases}, \quad \forall i \in I. \quad (3.10)$$

In this equation, X_i is the reservoir content for node i , while q_i and s_i denotes the inflow and spillage, respectively. Both the reservoir and the generator have a limited capacity, as described by the inequalities

$$X_i \leq X_{max}, \quad \forall i \in I, \quad (3.11)$$

$$g_i \leq g_{max}, \quad \forall i \in I. \quad (3.12)$$

To describe the value of storing water for later use, an end-value function is defined. It is discretized into limited steps d_{ij} , and the sum of these is again limited by the available water at the end period. The end-value function w_i is defined by

$$w_i = \sum_{j \in E} c_j d_{ij}, \quad \forall i \in I, \quad (3.13a)$$

$$d_{ij} \leq d_j^{max}, \quad \forall i \in I, j \in E, \quad (3.13b)$$

$$\sum_{j \in E} d_{ij} \leq x_i, \quad i \in I^{end}. \quad (3.13c)$$

The final constraints in this simplified model are the non-negativity of the following variables

$$g_i, X_i, s_i, d_{ij}, q_i \geq 0, \quad \forall i \in I, \forall j \in E. \quad (3.14)$$

The SHARM model obviously is far more elaborate than this, but the main principles are the same. If e.g. price uncertainty is included, this enters into the objective function. More constraints can be added, and the modeling of more complex topologies must be considered.

3.5 Price and inflow modeling

The modeling of input parameters is important to the outcome of stochastic programs. Much research has been carried out to develop models for predicting both spot prices and inflows. This section will give a brief mention to the most important methods, but it should be pointed out that this is not a main focus of this thesis.

The tests conducted in this work are based on real systems and real data used in the daily scheduling at Statkraft. As the predictions of spot prices and inflows are used operationally, Statkraft does not wish to make the models public. Thus the stochastic input is taken as given throughout the tests.

In the literature review on short-term hydro power scheduling [11], a short description of price and inflow models is included. Important classes of statistical models used for modelling of spot prices include auto-regressive moving average (ARMA) models and auto-regressive integrated moving average (ARIMA) models. Another approach use generalized auto-regressive conditional heteroskedastic (GARCH) models. A recent master's thesis conducted at NTNU has also done some work on price modeling in cooperation with Sintef [32]. The model proposed in that thesis is based on a deterministic model, and a literature review is also included.

Stochastic inflow processes can also be modeled using time series fitted to historic data, e.g. using ARIMA models. A widely used alternative is to generate ensemble forecasts from hydrological model, relating inflow to parameters such as precipitation and temperature. An example of such an approach is the popular HBV-model [3].

Chapter 4

Goals and strategy for the evaluation process

The principal objective of this thesis is to evaluate the prototype of the stochastic short-term hydro power scheduling model SHARM. This statement will however need some major refinements and concretizations to act as a proper starting point for an investigation. The terms on which the model shall be evaluated must be decided. One approach is to consider it as a direct replacement of the current deterministic model. Another can be to consider alternate areas of utilization. As stated in the introduction, this work will focus on the first approach.

Such an investigation calls for an optimized plan from the stochastic model, which could be directly compared to a corresponding run of the deterministic model. Comparisons could be made in terms of expected profit, reservoir handling and starting and stopping of units. The expected profit of replacing the deterministic SHOP model by the stochastic SHARM model is important, as the users will not be interested if the new model does not perform better in terms of income than the current one. It is also interesting to know how much different factors will affect the impact of a stochastic model. Such factors may be the uncertain parameters, or the form of the stochastic input. This comparison against SHOP will be one main objective of the evaluation.

Another key aspect concerning the evaluation of the new model is the computation time. The stochastic model can not compete with the deterministic in this area, but to be useful as a scheduling tool in an operative setting the running times should not be excessively high. A way of reducing the CPU-time is to use stochastic input only in a few reservoirs, while the rest of the system is modeled as before. This would typically be smaller reservoirs, for which the limits may be reached within the optimization period. Another measure would be to reduce the size of the stochastic input by applying algorithms for scenario tree reduction. The CPU-time is anyhow

one of the major drawbacks of stochastic programming. An important area to consider is therefore the size and form of the stochastic input, and how it will affect the objective and the reservoir handling. This will be the second objective of the evaluation.

Unfortunately, the access to stochastic input for historical cases has been limited. As a consequence, all the tests have been run with data acquired during the months of March, April and May in 2012. The availability of stochastic inflow scenarios has also limited the number of hydro power systems that has been considered.

This section will discuss the topics mentioned above in more detail, and present the tests that will be conducted in the evaluation process. The test systems that have been considered will be described in Section 4.1. An overview of the computational tests that will be performed is given in Section 4.2. The reasons for choosing these tests will be discussed and a more detailed description is presented in the remaining sections: Comparisons against SHOP will be the focus of Section 4.3, while the issues concerning computation time will be considered in Section 4.4.

4.1 Test watercourses

The choice of watercourses to consider in the evaluations was limited by several factors. First, the test systems could not contain any of the features not included in the SHARM prototype, as described in Section 3.4. Second, stochastic scenarios for price and/or inflow had to be available on a daily basis. A third demand was that the watercourses should exhibit some feature that the current deterministic model did not handle satisfactory. Finally, the systems should be relatively small, to enable testing with large scenario trees within reasonable CPU-times.

To satisfy all these requirements proved to be more difficult than anticipated. Especially the need for stochastic inflow data was difficult to meet. Finally, 2 different watercourses were chosen. Both are real, and in operation by Statkraft. According to the wishes of the operator, the plants are anonymized, and will be referred to as test system 1 and 2. The topologies of the watercourses are shown in Figure 4.1. Both are examples of High Pressure Plant (HPP) systems, as opposed to run-of-river systems that have less capacity for regulation.

Test system 1 is a cascaded reservoir system. The watercourse consists of 2 reservoirs, R1 and R2, and 2 plants, P1 and P2. At the top, R1 is a large, regulated reservoir which actually represents several reservoirs. R1 has a maximum capacity of 2363 Mm³. The water from R1 goes to the plant P1, which consists of 3 identical generation units. P1 has a maximum total production of 159 MW. The discharge from P1 ends up in the small reservoir R2, which have a capacity of 24 Mm³. The last plant P2 has 6

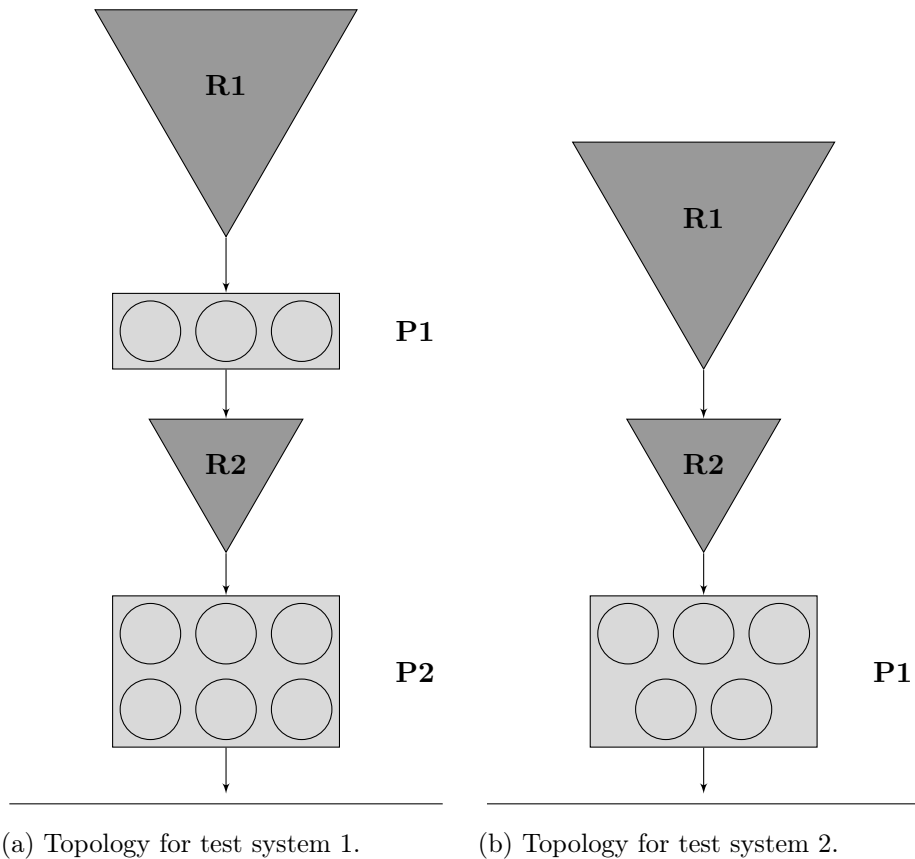


Figure 4.1: Schematic description of the 2 watercourses that have been considered in the evaluations.

generating units, of which 4 are identical and the 2 remaining units have a slightly higher capacity. The maximum total production of P2 is 259 MW.

This system is chosen because of the handling of reservoir R2. The reservoir is quite small compared to the capacity of the plant below, and may experience high inflow in certain periods. Combined, this gives the reservoir a potential for spillage if the scheduling does not anticipate a high inflow.

Test system 2 consists of 2 reservoirs, R1 and R2, and a plant P1. R1 has a maximum capacity of 256 Mm^3 , and the maximum volume of R2 below is 4 Mm^3 . The plant P1 has 5 identical generation units, with a total capacity of 315 MW. The reason for investigating this system is the handling of R2. Currently, the discharge from R1 to R2 is decided manually by the plant operators. A deterministic optimization by SHOP will maintain the reservoir level of R2 at maximum to achieve the highest plant head possible. This will however give a high risk of spillage in the case of unforeseen inflow, which is the reason for not using this result. It is assumed that a stochastic

scheduling tool such as SHARM will handle this situation in a more robust manner.

The systems belong to different price areas, so the stochastic input needed to perform tests are 2 sets of price scenarios and inflow scenarios for R2 in both systems. For both test systems, 21 price scenarios with hourly resolution and a time horizon of 7 days were available. 52 scenarios for each system were available for inflow. Initially, the plan was to use historical input data to investigate the effect of stochastic scheduling in different periods of the year. Unfortunately, this turned out to be impossible, so all input data had to be collected in the period between March and May 2012. Both price and inflow scenarios were available for test system 1 at the start of the project. It took more time to get inflow series for system 2, so the tests on this watercourse could not start until May.

4.2 Computational tests performed in the evaluation

This section will present the tests that were conducted during the course of this project. The reason for choosing these tests is a combination of two main factors. One is the availability of software and input data. The other is the priorities made by the author, in cooperation with personel from Statkraft and Sintef Energy Research. First an overview of the tests is given in Table 4.1. The duration of each test shown in the table is the actual number of tests conducted, not the planned duration of the tests. In the rest of this section, the purpose of each test will be discussed.

Test	System	Stochastic Input	Structure	Purpose	Duration
1	1	Price	Fan	Comparison	24 days
2	1/2	Price/Inflow	Tree	CPU/Reduction	1 day
3	1	Inflow	Tree	Comparison	7 days
4	1	Price /Inflow	Tree	Comparison	6 days
5	2	Inflow	Tree	Comparison	7 days
6	2	Price/Inflow	Tree	Comparison	6 days
7	1	Price	Fan	CPU/Reduction	1 day
8	2	Inflow	Tree	Sensitivity/Quality	1 day

Table 4.1: The table presents an overview of the computational tests conducted in the evaluation of SHARM.

Test 1: The main purpose of this test is to investigate the effect of considering price uncertainty in a cascaded system. Hence, comparisons with a deterministic plan will be made. Testing will be performed in March and April, when the uncertainty in inflow is low. The results can thus say some-

thing of the possible impact of stochastic scheduling in the winter depletion season.

This will be the first test to be conducted, and will thus be used to adjust the testing and evaluation procedures. It is expected that this part of the test may require both time and effort, but it is necessary to establish a solid framework before conducting the remaining tasks.

Test 2: An in-sample stability test of the reduction algorithm is conducted, using scenario trees for the cases of stochastic inflow and combined stochastic inflow and price. The concept and testing of in-sample stability is described in Section 3.3.4. The goal is to decide which degree of reduction should be applied to the scenario trees, weighting CPU-time against deviations in the objective function values. Another aspect that will be considered here is the computational behavior of SHARM with regard to different input. It is expected that a reduction in the size of the scenario tree will lead to significant reductions in the solution time.

The test is performed on both test system 1 and 2. The results from this test will be applied in tests 3 - 6.

Test 3 - 6: Using the reduction strategies decided in Test 2, these tests will examine the effect of considering stochastic inflow and combined stochastic inflow and price. Both test systems 1 and 2 will be considered. Apart from the different scenario tree structure, these tests are similar to Test 1. Corresponding deterministic plans will be found and compared to the stochastic ones. Testing will be carried out in May during the snowmelt period, in which the uncertainty in inflow is high. The results will be compared to those from Test 1, to see if the impact of considering uncertainty is greater in this period, and if it is more important to include stochastic inflow.

Test 7: This test focus on the performance of the reduction algorithm. The fan structure of the scenario tree is taken advantage of to perform tests of both in-sample and out-of-sample stability for the reduction algorithm. Both these concepts are discussed in Section 3.3.4. Similar tests from [14] indicate that the algorithms have good stability properties when considering inflow uncertainty. This test will consider stochastic prices in the form of a scenario fan on test system 1.

Test 8: Two aspects are considered in this test. First, the sensitivity of the scheduling with regards to the input is investigated by adjusting the amplitude of the scenarios. Second, a qualitative analysis of the handling of R2 in test system 2 is conducted. The hypothesis presented in the example of Section 3.1, that a stochastic model will be more careful in the handling of this reservoir, will be tested with inflow uncertainty.

4.3 Comparison against SHOP

This section describes the framework that will be used for comparing the results from SHARM with corresponding results from SHOP in tests 1 and 3 - 6.

As described in Section 2.2, short-term scheduling is a continuous process. The ideal strategy for comparing the two models would be to run them in parallel for an extended period of time. By performing with SHARM all the tasks that SHOP does now, a realistic measure for the differences in income and reservoir handling could be obtained. This task is considered beyond the scope of this thesis, as it would require either day-around operation together with SHOP, or the construction of an extensive simulation framework.

A different approach must therefore be taken. Scheduling of the day ahead is used as decision support in the preparation of bids for the Elspot market. This is arguably one of the most important tasks of a short-term scheduling program, and it is natural to consider such a task in an evaluation. It is also quite easily quantifiable in terms of objective function value. Hence, the tests described in this section will focus on the preparation of spot market bids for the day ahead. The construction of optimized stochastic and deterministic plans for the day ahead is discussed below.

4.3.1 Deterministic scheduling for the day-ahead market

The scheduling performed in the preparation of bids for the day-ahead market is run as follows. In the morning, SHOP is run with an optimization period of 162 hours. $T = 0$ is taken to be 6:00am the present day, and the day ahead is thus between $T = 18$ and $T = 42$. The remaining 5 days is included for coupling with the mid-term scheduling model, and to consider the future development of price and inflow. The price and production is known and equal for all scenarios for the first 18 hours, as the decisions for this period were taken by yesterdays scheduling.

The price forecasting department at Statkraft provides 22 price scenarios for this period, with hourly resolution, and updates them several times during the day. Inflow data is obtained in much the same fashion, but using a different model. The hydrology department use the fundamental HBV model [3] to produce the different scenarios. The quality and number of inflow scenarios vary between the reservoirs, as the need for good forecasts depends on the size and importance of each of them. Based on the available scenarios, one for price and one for inflow is chosen as the main prognosis and used as deterministic input to SHOP.

A corresponding deterministic plan must be specified for comparisons with the plan generated by SHARM. A deterministic plan in this context is understood to be the result from SHARM when specifying only one inflow and price scenario. In all comparisons, unless stated otherwise, the deter-

ministic scenarios will correspond to the main prognosis for price and inflow. The reason for using this approach, instead of comparing directly with SHOP, is that the prototype does not support all the features of the full operative deterministic program. It is imperative in the evaluation process that the models are competing on equal terms. A more detailed comparison with SHOP can be performed when a full implementation of the stochastic model is available.

The result of running SHARM with the main prognosis for price and inflow is one plan for the 162 hour period. The plan is optimal with respect to the chosen input.

4.3.2 Stochastic scheduling for the day-ahead market

As explained in Section 3.4, the result from SHARM will not be *one* optimal plan for the whole scheduling period. It will be an optimal first-stage decision, and an optimal strategy for each path of the scenario tree in the rest of the period. Thus, if several scenarios are to be considered at a given stage, the program will not return a single plan for this stage. This is a problem that must be resolved to produce a single, optimal plan for the day ahead.

First, a case where only price is stochastic is considered, corresponding to Test 1 of Table 4.1. The available stochastic input is 22 price scenarios that coincide in the 18 first hours. All scenarios are considered to be equally probable. As there are relatively few scenarios, a fan structure is used to maintain the structure of the scenarios. This means that the problem becomes a two-stage stochastic program with recourse.

One solution is to take advantage of some of the characteristics of the problem at hand. The goal is to produce a plan for the day ahead, i.e. $T \in [19, 42]$, which should take into account the stochastic input from $T = 19$ to the end of the period. A possible formulation of such a problem is to optimize using a fan with branching from $T = 19$, but with the additional constraint that production should be equal for all paths in the period $T \in [19, 42]$. As uncertain parameters are only present in the objective function through hourly prices, an equivalent formulation can be derived. The deterministic period is extended to 42 hours, with the first 18 hours as before. For the period $T \in [19, 42]$, the weighted average, or expected value $E(y_t)$, of the price scenarios is used instead of the main prognosis. In this case, running with stochastic price and one common decision is equivalent to running with expected price in the first 42 hours, as will be shown below.

Consider the objective function π given in equation (3.9), with production $g_{t,i} = g_t$. For the day ahead it becomes

$$\begin{aligned}
\pi &= \sum_{i=1}^N \sum_{t=18}^{42} g_{t,i} p_{t,i} y_{t,i} \\
&= \sum_{t=18}^{42} g_t \sum_{i=1}^N p_{t,i} y_{t,i} \\
&= \sum_{t=18}^{42} g_t E(y_t).
\end{aligned} \tag{4.1}$$

Using a fan as described above will thus be equal to imposing the additional constraint of a common production for the next day. This will enable the calculation of an optimal plan for the day ahead, that takes price uncertainty into account from $T = 19$ to $T = 162$. The 2 equivalent scenario fans are shown in Figure 4.2. This approach was used in the first tests conducted, namely Test 1 and 7. However, this strategy is no longer valid when considering stochastic inflow, which enters the problem through the reservoir balance equations (3.10) that is part of the constraints.

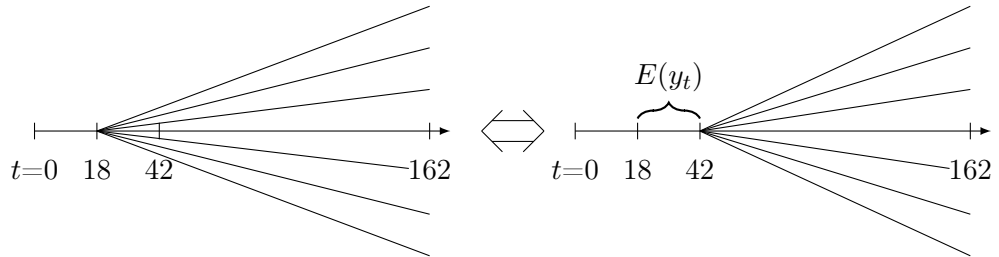


Figure 4.2: Showing 2 equivalent scenario fans where price is the only stochastic parameter. By applying the one on the right, a single plan can be calculated by SHARM while still taking uncertainty into account in the period $t \in [19, 42]$.

Several alternative evaluation approaches were considered, before settling on the one described above. These also considered a scenario fan, but used the main prognosis for price in the 42 hour deterministic period. Evaluation of the plans was conducted with an optimization period of 42 and 162 hours. The procedures were abandoned because of the lack of theoretical foundation concerning which plan would perform best. The work did however reveal some severe errors in the test framework, which were corrected before conducting further tests.

The question of how to produce a single plan with SHARM has been discussed with researchers at Sintef Energy Research during the course of work. It was suggested to include in SHARM the constraint that production

and gate discharge should be equal for all paths in a specified period in the start of the optimization. The developers at Sintef were able to implement a new version of the prototype so that this feature could be applied in this thesis. The new version enables the production of a single, optimal plan for the day ahead with stochastic representation of both inflow and combined inflow/price. This was used in Tests 2 - 6 and 8.

4.3.3 Evaluation of day-ahead plans

Following the approaches described in the previous section, stochastic plans for the day-ahead can be produced and compared to the deterministic plan. However, the question of what to compare is still not answered. A natural option would be a quantitative comparison, considering the value of the objective function. A qualitative approach investigating reservoir handling and unit commitment could also be interesting. An important point is that the objective function values can not be compared directly, as they are not based on the same input.

To overcome this problem, the approach for comparisons taken in this thesis is based on the assumption that the stochastic input is a good representation of reality. That is, the performance of the deterministic day-ahead plan in the situation described by the input scenario tree is evaluated. This strategy is similar to the one used for stability tests in [14], but here only the first first-stage decision is evaluated.

It would maybe seem natural to evaluate the 2 models in hindsight, based on the realized prices and inflow. There are several reasons not to do this. First, it is not the goal of this work to evaluate the price and inflow forecasts. It is also important to note that the realized price will depend on the real operational scheduling, which is performed deterministically by all producers. Finally, to perform such an evaluation fairly would require the scheduling of several watercourses together.

To explain the strategy in more detail, consider Test 1 described in the previous section. First SHARM is run to obtain an optimal schedule with the scenario fan shown on the right in Figure 4.2. Then SHARM is run with the main prognosis to obtain a deterministic plan for the first 42 hours. The deterministic model is now evaluated by a two-stage process. At the first stage, the result of using the deterministic plan when the price actually is $E(y_t)$ for $t \in [19, 42]$ is found. At the second stage, starting at $T = 42$, new information is revealed. The model now receives the price development for every scenario, and because of the fan structure it can make a deterministic schedule for the rest of the period for each of them. Finally, the probability weighted sum of the objective function value for each branch is calculated. This sum could then be compared to the objective value from the first SHARM run.

Practically, the comparison is done as follows. In SHARM, production

can be fixed by specifying generator schedules. To evaluate the performance of the deterministic day-ahead plan, it is imposed as a generator schedule on a SHARM run with the original scenario tree. The objective function from this run is compared to the first SHARM run. In this way, the original SHARM run will always produce a greater, or equal, objective function value. The only difference is an additional constraint on the production level in the hours $T = 19, \dots, 42$.

The same approach is taken when the new version of SHARM is applied to produce day-ahead plans for more general scenario trees. As the fan structure is not enforced anymore, new information will be revealed at several stages. The comparison will therefore not consider the use of a deterministic model for all subtrees, but the effect of fixing the day-ahead plan can still be investigated.

As mentioned in the introduction to Section 4.3, the ideal method of evaluation would be to compare the performance of the 2 models over a longer period, performing tasks such as Elspot bidding, rescheduling and bidding in the secondary markets. The alternative considered in this thesis is to compare the day-ahead plans as described above, for as many days as possible. The average objective function values will be compared, as well as a more qualitative evaluation of the reservoir handling.

4.3.4 Analysis of sensitivity to input

Test 8 investigates how SHARM behaves when the amplitude of the input scenarios is changed. That is, the average value for each time step is found, and the deviation from this average for each scenario is multiplied by a factor α . If a value becomes negative, it is changed to zero. The test is performed for test system 2 with a stochastic representation of inflow. After the scenarios have been altered, a tree reduction is performed in the same way as for Test 5.

However, the main objective of this test is to find out how a stochastic model will handle the reservoir R2 of system 2. To examine this, SHARM is run with a common production and discharge plan for all the 162 hours of the optimization period. This is done to capture the uncertainty in the end of the period from an earlier stage, testing SHARM's ability to plan ahead. The differences in reservoir handling are then reviewed for different values of α . Comparisons to the deterministic handling will also be considered. As described in the example of Section 3.1, the deterministic schedule for this system is not implemented operationally, where instead tighter boundaries are specified manually.

It should be expected that SHARM will keep a safety margin by not filling R2 to its maximum level, and that the reservoir level will be lowered in advance if high income is forecasted later in the period.

4.4 Evaluation of computation time in the new model

SHOP is used extensively throughout the day by the power production companies. As described in Section 2.2, it is used for scheduling, spot bid preparation and rescheduling, as well as preparing bids for the balancing market. A significant factor contributing to the success of SHOP is the relatively low computation times, and for a stochastic model to prevail it should be able to compete also in this area.

An interesting aspects to consider in this context is whether the results of SHARM will justify a longer computation time. As described in Section 3.3, the size of the input scenario tree is important both to the quality of the solution and to the solution time. What is the best balance between these two goals, and is it affected by e.g. the topology of the system or the inflow characteristics?

The focus is split between two main areas when evaluating the computational performance of the stochastic model. First, the tree construction and reduction algorithms that have been chosen to accompany the SHARM prototype shall be evaluated in terms of stability and CPU-time. Secondly, the performance of the prototype itself. The question of how the CPU-time is affected by different inputs, i.e. scenario trees and system topologies, shall be investigated. Tests 2 and 7 described in Section 4.2 are conducted to assess this topic.

4.4.1 Performance of the scenario tree generation algorithm

When considering scheduling problems of realistic size, some form of scenario tree reduction must be performed. It is thus important for the user to have confidence in the performance of the reduction algorithm. This problem was discussed in Section 3.3.4.

The generation and reduction algorithms that accompany SHARM were described in Section 3.3.3, and their stability have been examined and found to be satisfactory in [14]. That paper considers a case with inflow uncertainty, and use the simplified stochastic model presented as an example in Section 3.4. Here, an in-sample and out-of-sample stability analysis will be performed with price uncertainty, using SHARM. Testing for out-of-sample stability is generally difficult, but can be simplified by using input in the form of a scenario fan. This is done in Test 7.

The starting point for the tests is an initial fan similar to the one on the right in Figure 4.2. The scenario reduction algorithm is applied to reduce the number of scenarios from 22 to 1, producing a total of 8 different fans. SHARM is run with each fan to produce objective function values for the in-sample analysis. The first-stage decision for each fan is then imposed as a generator schedule on the original fan, in the same way as the deterministic day-ahead plan in Test 1. The results from these runs are compared in the

out-of-sample analysis.

Testing for in-sample stability is easier, as the objective function values obtained with different trees are compared directly. This is thus carried out for more general scenario trees in Test 2. It is the in-sample stability results that are considered when recommending an appropriate degree of reduction for different inputs and test systems.

The CPU-time of the reduction algorithm itself could also be considered, and analytic expressions for the CPU-time are presented in Section 3.3. However, this has not been a main focus in this work. Experience has shown that the time used to generate and reduce trees is very short compared to the solution time in SHARM. This may be due to the limited number of scenarios considered.

Currently, constructing the initial scenario fan also takes significantly longer time than the CPU-time of the reduction algorithm. To generate fans on XML-format, an Excel spreadsheet written by Sintef Energy Research have been used. This tool limits the number of total scenarios to around 230, when using an hourly time resolution and an optimization period of one week.

4.4.2 Computational performance of SHARM

Test 2 investigates the time spent by SHARM to solve scheduling problems with different input for different watercourses. The goal of the tests is to get an idea of how the objective function value and the CPU-time change with the size of the scenario tree. Both test systems 1 and 2 have been considered, and uncertain inflow and combined price/inflow uncertainty is used.

As in Test 7, the starting point in Test 2 is an initial fan. In the case of inflow uncertainty the fan contains 52 scenarios. To introduce a tree structure, the tree generation algorithm is applied, with $\epsilon_{rel} = 0.1$. The resulting tree is then reduced with various degrees of reduction, ranging from $\epsilon_{rel} = 0.1$ to 1.0. SHARM is run with the different trees, and the objective function values are compared. The case of combined price and inflow uncertainty is treated in the same fashion. Here the starting point is a fan of 225 scenarios, combining 15 price and 15 inflow scenarios. This size is chosen due to limitations in the tree construction framework as described in Section 4.4.1.

Chapter 5

Results and discussion

The tests described in Chapter 4 have been carried out during the months of March, April and May 2012. Due to the limited availability of software and input data, only Test 1 and 7 could be carried out in the first phase. Arriving in the start of May, inflow scenarios for test system 2 and a new version of SHARM enabled the completion of tests 2 - 6 and 8. The time span of Tests 3 - 6 was originally planned to be similar to Test 1, but had to be cut short due to time limitations.

This chapter will present the results of the computational tests. One section for each test is included, apart from Tests 3 - 6 which are considered together. A discussion of the results will be given in the same section. The results will be considered in light of the expectations concerning the given test, and comments will be made on the general impact of the results with regards to short-term hydro power scheduling.

The tests were carried out on the 32 bit Statkraft test server. The solution times will vary with the computer used, but the important point here is the difference between the CPU-times when using different input.

5.1 Test 1: Comparison with stochastic price

As expressed in Section 4.2, the goal of this test was twofold. One side was to compare a deterministic and a stochastic day-ahead plan in the winter depletion season with stochastic prices. The other was to develop and refine the test procedure and the tools used for result processing and presentation. As expected, the latter part proved to be an extensive task, but will not be described in detail here.

When the test procedure and processing tools were settled upon, the comparisons commenced in the middle of March. Collecting data files and input scenarios from the operative scheduling with SHOP, 24 days of testing were completed.

Focusing on the immediate future, the operative settings use MIP to

model startup costs only in the first three days. This means that the optimization solver does not consider these costs in the remainder of the period, only adding them to the objective at the end. For a fair comparison, this represents a problem. Possible solutions include considering MIP in the whole period, or not at all. Due to computational considerations it was decided to not use MIP, and to exclude the startup costs from the objective functions used in the comparisons. Generally, there are few starts and stops in the cases considered in the test, so it is not expected that this will affect the results to much extent.

The results from the tests are given in Figure 5.1, showing the difference between the objective function values obtained with and without the deterministic day-ahead plan. Additional results are given in Table C.3.

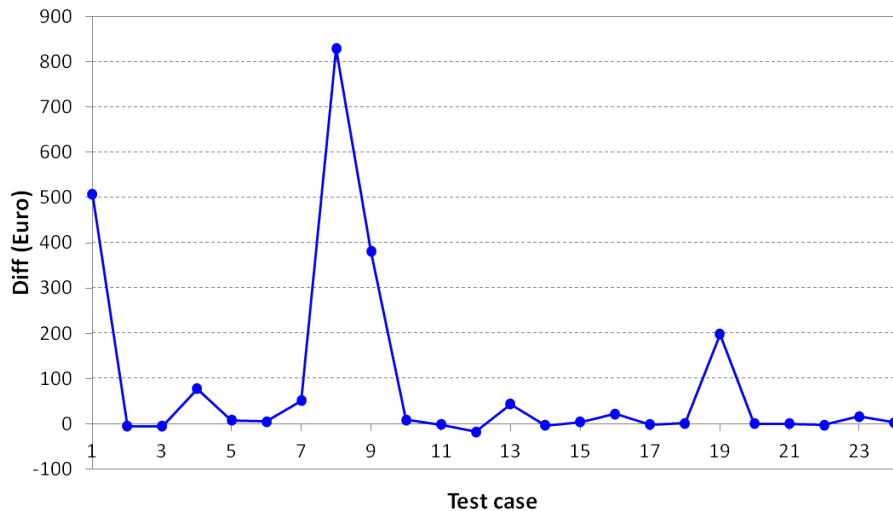


Figure 5.1: The differences in objective function value between SHARM and the run with a deterministic day-ahead plan is shown for test system 1. A positive value means that SHARM gives a greater income. Each test case on the x-axis represents a specific day.

These results indicate that there is not much to gain from using stochastic scheduling in these conditions. On average, SHARM increased the value of the objective function by €88 compared to the deterministic model. This corresponds to an income increase of 0.0005%. There are even some cases in which the deterministic plan performs better, something that should not be possible. An explanation of this may be found in the convergence of the SHARM iterations. Following the practice from the operational scheduling, the number of iterations in UC and Close-in mode is determined in advance. Thus the difference in objective value between iterations determines the ac-

curacy of the solution. In the tests conducted in this work, it is found to be in the range of €50 - €100.

As can be seen in Figure 5.1, there are only 4 cases in which the increased income exceeds this limit. These are cases 1, 8, 9 and 19. Some of them do in fact contribute a more substantial amount, with the maximum difference being €827 for test case 8.

It is necessary to see if the cause behind the increased income in these 4 cases can be found. If it can be related to some common feature of the input scenarios or the initial conditions, a recommendation can be made for the use of stochastic scheduling in similar cases.

One hypothesis is that the larger deviations can be explained by the variance of the input scenarios, i.e. larger input variance gives larger deviations. Other proposals of decisive factors include the initial reservoir level, average price and end-point description. It can also be informative to see where the deviations stems from. When not including MIP, the objective is given as the sum of buy and sale costs, end reservoir value and penalty costs. These hypotheses are investigated in the following.

The average variance of the input scenarios say something about how much the scenarios differ. It is possible that the effect of considering uncertainty is greater if the scenarios span a larger sample space. The average variance was calculated for each case as

$$Var(y) = \frac{1}{22} \sum_{i=1}^{22} \frac{1}{162} \sum_{t=1}^{162} (y_{i,t} - \bar{y}_t)^2, \quad (5.1)$$

where $y_{i,t}$ is the spot price in time stage t of scenario i . The correlation coefficient of this and the difference in income $\Delta\pi$ was found to be

$$r = \frac{\sum_{j=1}^{24} (Var(y)_j - \overline{Var(y)}) (\Delta\pi_j - \overline{\Delta\pi})}{\sqrt{\sum_{j=1}^{24} (Var(y)_j - \overline{Var(y)})^2} \sqrt{\sum_{j=1}^{24} (\Delta\pi_j - \overline{\Delta\pi})^2}} \approx 0.18. \quad (5.2)$$

The coefficient of determination was $r^2 \approx 0.03$, suggesting that about 3% of the increased incomes can be explained by the variance of the input. A test of the null-hypothesis that the correlation coefficient was zero could not be rejected with a significance level of 0.15. This means that there is little evidence that there is a linear relation between $Var(y)$ and $\Delta\pi$. To check whether there is another connection between the 2 parameters, they are plotted against each other in Figure 5.2. All these findings indicate that the increase in income can not be explained by the variance of the input scenarios alone.

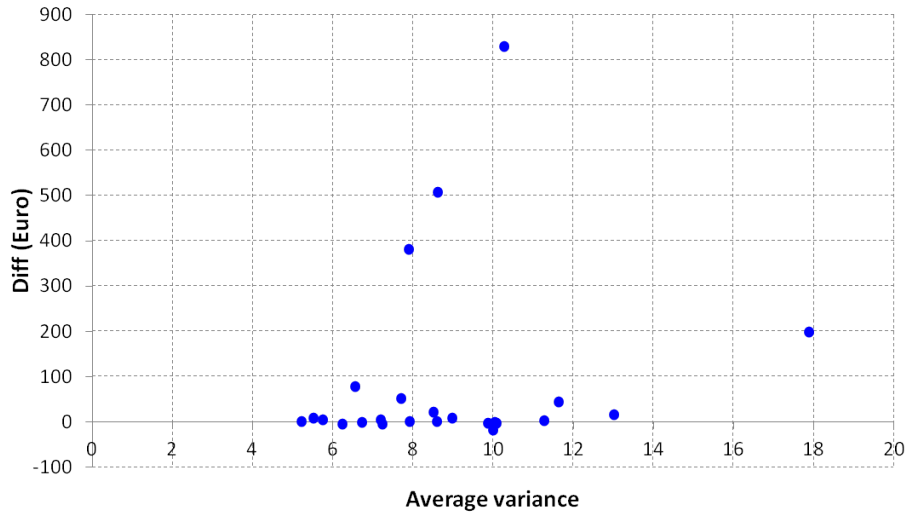


Figure 5.2: The difference in objective function value obtained with and without a deterministic day-ahead plan is plotted against the average variance in the input price scenarios. No obvious pattern is observed, suggesting that there is no significant correlation between the parameters in these cases.

The initial reservoir levels, average prices and end point descriptions in the various cases have been examined. No relevant differences have been found in these values between the 4 cases and the rest, as the overall conditions were quite similar in the test period.

Investigation of the distribution of the deviations between the different parts of the objective is somewhat more promising. First, there is no difference in the penalty costs between the stochastic and deterministic plans. It seems however that the greater part of the increased income comes from increased buy and sale costs. That is, the stochastic plan has a higher income from traded power in the optimization period. This may be due to a slightly higher production in the day ahead.

This is true for 3 of the 4 cases that gives an increased income above the accuracy of the iterations. In 2 of these cases, there is one scenario with a significantly higher price within the day-ahead. SHARM suggests a higher production in this period than the deterministic plan, and will thus benefit more from this slight peak in the price.

5.2 Test 2: Run-time test, stochastic inflow and inflow/price

In-sample stability analyses were performed for test systems 1 and 2 with stochastic inflow and stochastic price/inflow, as described in Section 4.4. The test will also be referred to as a *run-time test* in the following.

Besides assessing the stability of the scenario tree construction and reduction algorithms, the purpose of these tests was to investigate the relation between CPU-time and reduction, and to determine an appropriate degree of reduction for each case. These ϵ_{rel} were to be applied in Tests 3 - 6 to save time and consider the performance of SHARM with such reduced trees.

The starting point of the reductions was an initial fan. For inflow, 52 scenarios was used, while for price/inflow the fan consisted of 225 scenarios. To allow for more general tree structures, the tree construction algorithm was applied to the initial fans with a $\epsilon_{rel} = 0.10$. The argument for this was to keep a high number of paths while reducing the number of nodes, and hence altering the tree structure. The resulting tree was then used to represent the full sample space of the uncertain variables.

Different degrees of reduction were applied, and SHARM was run with the resulting trees. Tables and figures presenting the results are included below.

ϵ_{rel}	Stochastic inflow				Stochastic price/inflow			
	System 1		System 2		System 1		System 2	
	# Scen	$T[s]$	# Scen	$T[s]$	# Scen	$T[s]$	# Scen	$T[s]$
0.0	51	2553	52	1175	206	2008	222	-
0.1	32	825	37	875	25	423	142	2736
0.3	14	298	17	237	10	176	58	710
0.5	6	98	6	83	6	95	18	245
0.7	3	51	3	45	4	78	7	93
0.9	2	40	2	38	2	45	3	57
1.0	1	27	1	26	1	36	1	27

Table 5.1: The number of scenarios and corresponding CPU-times T are shown for different degrees of reduction ϵ_{rel} and test cases.

From Table 5.1, it is clear that the CPU-times of the full trees are unacceptable for operational use. It should be noted that these cases are small compared to e.g. scheduling against a given load for an entire price area. The case of test system 2 with a full tree and price/inflow uncertainty actually did not go through, as the solver ran out of memory. So there is indeed a need for reduction.

The results from the in-sample stability tests on system 1 and 2 are shown in Figures 5.3 and 5.4 respectively.

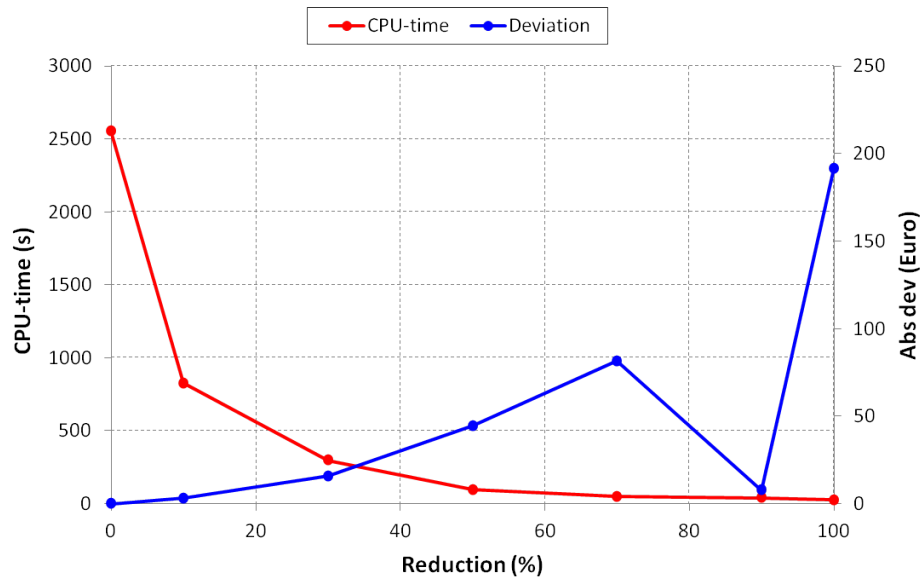


Figure 5.3: Results from the run-time test with stochastic inflow on test system 1. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €13 873 471,12. The red line relates the CPU-time to the degree of reduction.

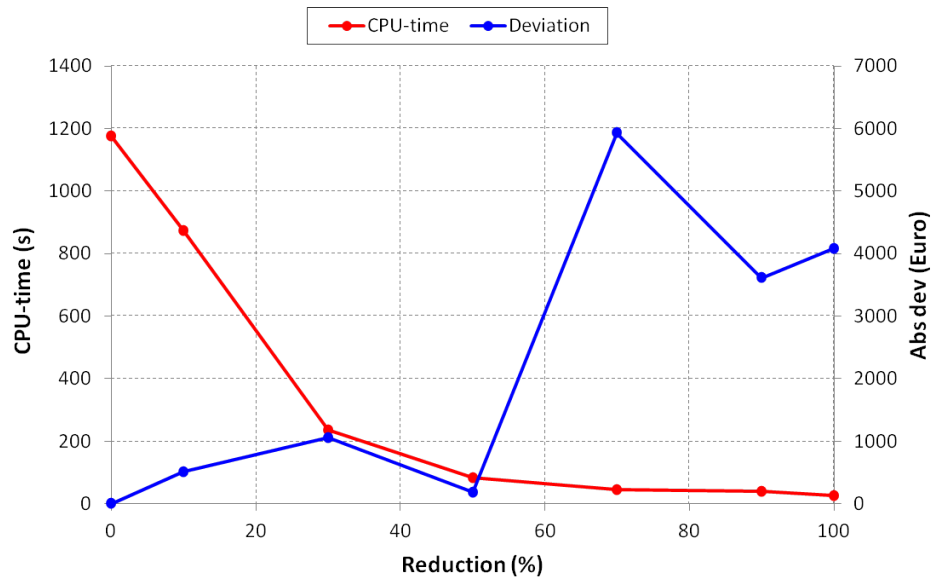


Figure 5.4: Results from the run-time test with stochastic inflow on test system 2. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €3 133 254,38. The red line relates the CPU-time to the degree of reduction.

Clearly the deviations increase with the reduction, but not strictly. A similar behavior is observed in both figures for the objective deviations, although in the latter case they are much higher. It is obvious from these results that there is much to gain in terms of CPU-time from reducing the size of the input tree. It should be noted that the deviations, even for the deterministic plan, is very small compared to the objective of the full tree.

In both these cases, the CPU-time drops dramatically from the full tree to the reduced tree with $\epsilon_{rel} = 0.50$. Higher reductions will lead to a further decrease in the solution time, but these cases are all run within a solution time of 100 seconds. As will be discussed in Section 5.4, the relation between CPU-time and the number of scenarios, or paths, in the trees seems to be almost linear for CPU-times below approximately 200 seconds. This can also be seen in these tests, e.g. from Table 5.1, although a plot is not included here.

Figures 5.5 and 5.6 show the results of the tests run with both price and inflow uncertainty. When running SHARM with the full tree for test system 2, the solver ran out of memory. Thus, no result was obtained for this case. The deviations reported in Figure 5.6 are relative to the tree with $\epsilon_{rel} = 0.10$.

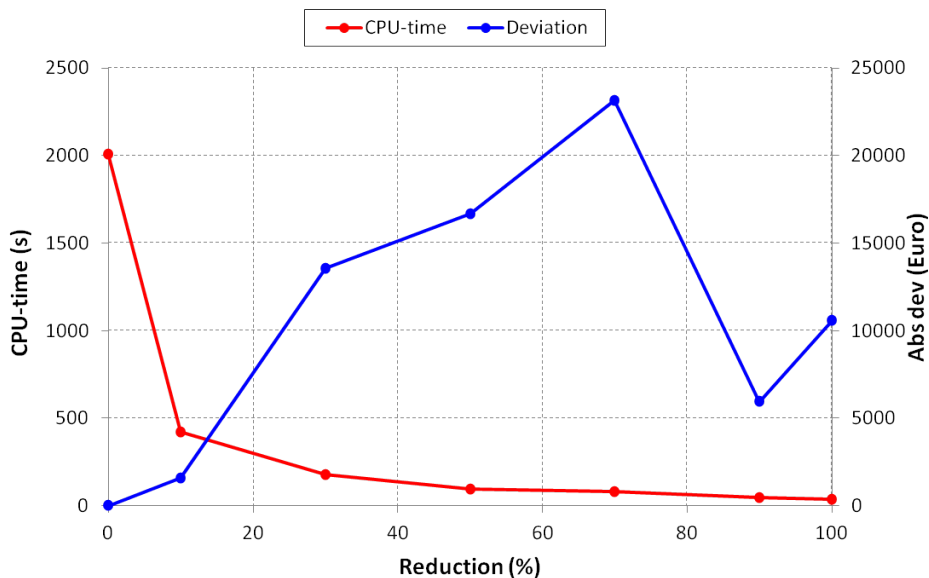


Figure 5.5: Results from the run-time test with stochastic price and inflow on test system 1. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €13 831 700,09. The red line relates the CPU-time to the degree of reduction.

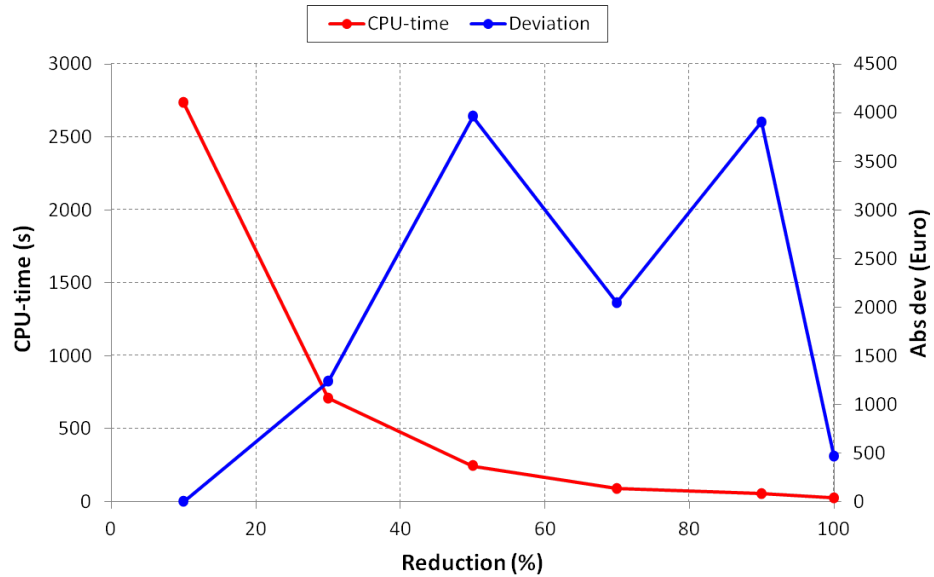


Figure 5.6: Results from the run-time test with stochastic price and inflow on test system 2. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €3 133 305,05. The red line relates the CPU-time to the degree of reduction.

The same reduction procedure was used for both systems, and the number of initial scenarios was equal. Nevertheless, the size of the full tree, obtained by construction with $\epsilon_{rel} = 0.10$ from the initial fan, differed significantly. For test system the number of nodes was 8728, while the full tree for test system 2 contained 21236 nodes. From Figure 5.7 it is evident that the majority of the branching for test system 1 happens at a later stage than for system 2, which explains the size difference. This may in turn be due to the fact that the inflow scenarios for system 2 have a much higher variability, especially in the start of the optimization period. The average variance for this case was almost twice as high for test system 2.

In terms of deviations in the objective, the pattern is the same as for stochastic inflow. Increasing the reduction leads to a decrease in solution time, and generally to higher deviations. The relation between the degree of reduction and CPU-time is also similar, especially for test system 1. As the size of the full tree is much larger for test system 2, the CPU-times are also higher, but the pattern is the same.

Based on the results shown above, together with an examination of the reduced trees and the remaining scenarios, a reduction strategy have been chosen. The strategy is to first construct a tree from the initial fan using the *backwards construction* strategy described in Algorithm B.2 with $\epsilon_{rel} = 0.10$. The tree is then reduced by the *fast forward selection* strategy displayed

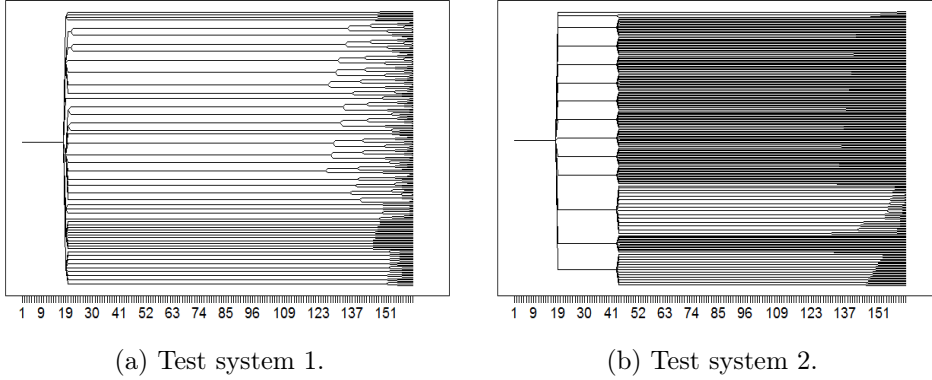


Figure 5.7: The full trees with price/inflow uncertainty used in Test 2. Both are generated from an initial fan of 225 scenarios by specifying $\epsilon_{rel} = 0.10$.

in Algorithm B.1, using $\epsilon_{rel} = 0.50$. This strategy is applied to all the 4 considered cases.

The reasoning behind this decision is that $\epsilon_{rel} = 0.50$ is the lowest degree of reduction that keeps the CPU-time within reasonable limits. At the same time, it is found that enough scenarios remain at this point to represent the original sample space in a satisfactory manner. It also seems that the jumps between scenarios no longer are that prominent at this level of reduction.

As an example, the original and reduced scenarios for inflow and price for test system 2 is shown in Figure 5.8. The example considers case 5 from Test 6, showing that the reduction strategy translates well to other input data.

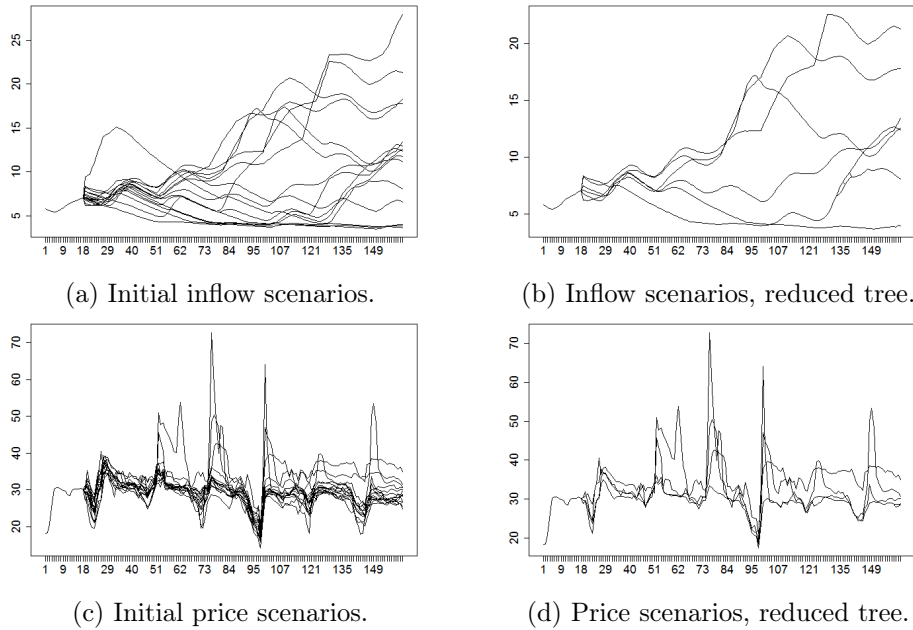


Figure 5.8: The original input scenarios are shown on the left, and the scenarios remaining after applying the suggested reduction strategy are shown on the right. Inflow scenarios are shown at the top, the units on the axis being hours and m^3/s . Price scenarios are found below, with units of hours and $\text{€}/\text{MWh}$.

5.3 Test 3 - 6: Comparison, stochastic inflow and inflow/price

Similar to Test 1, the purpose of these tests was to quantify the expected profit of considering uncertainty in inflow and price/inflow for the 2 test systems in the spring flooding season. Comparisons with the deterministic model in terms of day-ahead plans were performed.

These tests were performed in the beginning of May, but due to time limitations the number of cases considered was less than originally planned. Despite this fact the tests may provide some interesting information as they were conducted during a period with quite high inflow uncertainty. As a rule of thumb, the melting of snow starts around week 17 each spring. However, it is difficult to predict exactly when it will happen, and together with the chance of heavy rain this means great variability in the inflow scenarios in the period around this week.

The results from Tests 3 and 4 are shown in Figures 5.9 and 5.10. Additional results for Tests 3 - 6 are given in the tables of Appendix C.

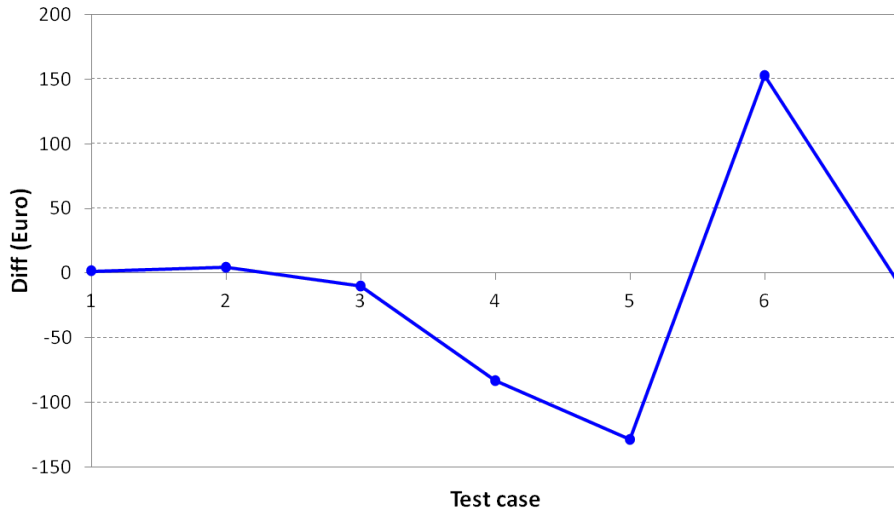


Figure 5.9: The plot displays the difference in objective function value between SHARM and the run with a deterministic day-ahead plan for test system 1 with stochastic inflow. Positive values means that the stochastic plan gives in higher income. Each test case on the x-axis represents a specific day. Numerical values are found in Table C.1.

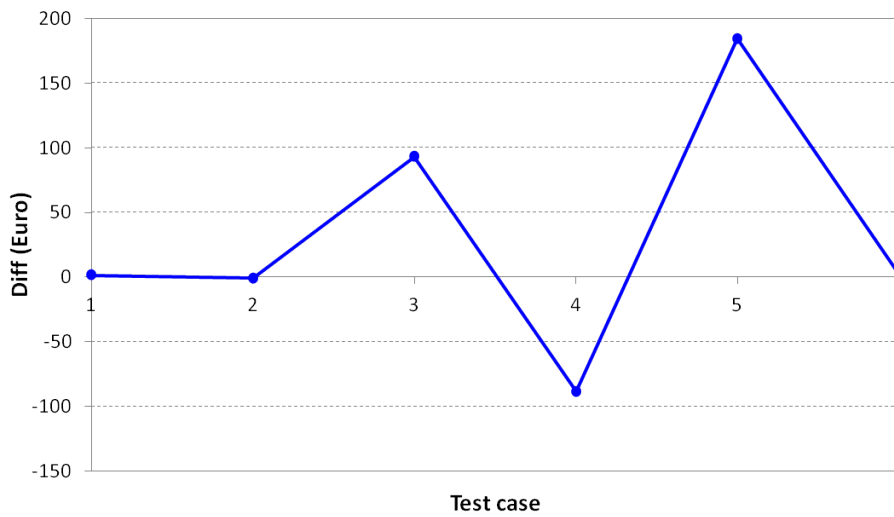


Figure 5.10: The plot displays the difference in objective function value between SHARM and the run with a deterministic day-ahead plan for test system 1 with stochastic inflow and price. Positive values means that the stochastic plan gives in higher income. Numerical values are found in Table C.2.

The differences in objective for test system 1 are not very significant. Considering the estimate of the iteration accuracy from Section 5.1, there is only one case in both Test 3 and 4 in which the stochastic plan outperforms the deterministic one. In Test 3, a negative value of more than €100 can be observed, indicating that the accuracy of the iterations may be even lower.

The results of Tests 5 and 6, performed on test system 2, are not included, because the differences in the objective with and without a deterministic day-ahead plan were negligible in both these tests. Numerical results from these tests can be found in Tables C.4 and C.5. A possible explanation may be found in the topology and state of the system. As there is enough water in R1 and the spring thaw is imminent, the availability of water is not the main issue. As will be discussed more closely in Section 5.5, the handling of R2 to avoid spillage is more important. There are restrictions on the discharge gate between R1 and R2, which means that R2 is just filled when the day-ahead period is over. As both models increase the R2 level in the beginning of the period to achieve better effect on the plant, the day-ahead plans become almost identical.

A conclusion based on these results would be that there are no significant benefits of using stochastic scheduling in these situations, at least not in terms of increased objective function value. The results are so similar that the method of comparison may be challenged. A possible way of assessing the differences is to take a more qualitative approach. This is done for Test 8 in Section 5.5.

5.4 Test 7: Stability tests, stochastic price

Test 7 investigated the stability of the reduction algorithm presented in Section 3.3.3 and Algorithm B.1. The procedure followed in this test is described in Section 4.4.1. This section will present the results of the tests and, based on them, give a brief discussion of the stability of the reduction algorithm. Some of the results from the in-sample stability test can be seen in Figure 5.11.

First, the in-sample stability was investigated by comparing the objective function values for different scenario fans. The full tree in this case is the same scenario fan as in Test 1, depicted on the right in Figure 4.2. From this case, the fan is reduced by 3 scenarios until only 1 remains. As expected, the CPU-time decrease with the number of scenarios included. An approximately linear relation between these parameters can be observed.

The CPU-time results from this case can be viewed together with those from Test 2 in 5.2. Apart from the largest trees, the tests, and indeed the results, are similar. Test 2 also suggests a linear-like relation between the number of scenarios in the trees and the CPU-time for the lower solution times.

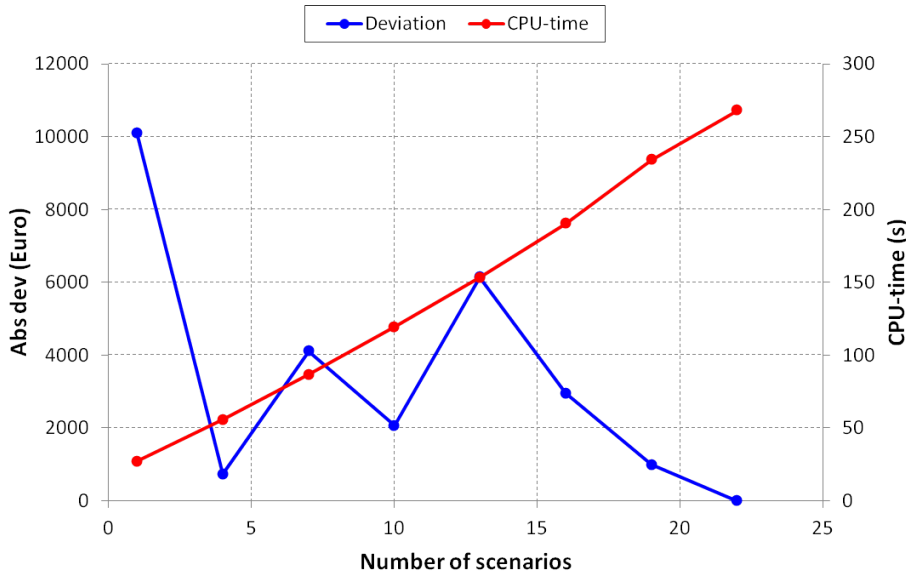


Figure 5.11: Results from the run-time test with stochastic price on test system 1. The blue line shows the absolute deviation from the objective function value of the full tree, which in this case was €17 539 095,30. The red line relates the CPU-time to the number of scenarios included in the fan.

As can be seen in Figure 5.11, the absolute deviation from the objective of the full fan is generally increasing with the decrease in CPU-time. This also follows the trend from Test 2. The difference is not strictly increasing with the reduction, but is clearly growing. However, these differences are relatively small compared to the total value of the objective function. A maximum deviation of 0.06% is observed for the case of a single scenario.

The observation of small differences continue in the out-of-sample analysis. Day-ahead plans resulting from each fan are imposed as generator schedules on the full fan, and the corresponding objective function values are compared. For this test case, the day-ahead plans were almost identical, resulting in negligible deviations. That is, the plan from the single scenario fan gave a decrease in the objective of €3,74 when applied to the full fan.

Summing up, the results of this test support the findings of [14], that the reduction algorithm have good stability properties. It should be noted that the considered test case, in combination with the fan structure necessary to perform the out-of-sample analysis, provides limited information as the differences are so small.

The results shows, as expected, that the CPU-time can be reduced by reducing the number of scenarios in the scenario tree. For small trees an approximately linear relation is observed. For this case, scheduling for test system 1 with hourly resolution and a horizon of 7 days, reducing the tree

by 1 scenario leads to around 12 seconds reduction of the solution time.

5.5 Test 8: Sensitivity to input and qualitative analysis

The situation considered here was first mentioned in the example in Section 3.1. Test system 2 was used in the example to showcase a problem with the current deterministic model, and it was predicted that a stochastic model would be able to handle this case better. This hypothesis is put to the test, and the results will be evaluated in a more qualitative manner than the previous tests.

Tests 5 and 6 showed that there was very little to gain for this test system by using SHARM to generate a day-ahead plan for production. Bearing this in mind, and focusing more on reservoir handling than on production, this test will take an alternative approach. Instead of generating a day-ahead plan, a common production and discharge plan is specified for the entire optimization period.

When investigating the results, perhaps the most interesting is the reservoir trajectory of R2. But since there are several inflow scenarios, the reservoir level will be different for each of these, even though the production and discharge between R1 and R2 are the same. An appropriate result to consider may be the expected reservoir level, based on the probability for each scenario.

The inflow scenarios for each case will be presented as in Figure 5.12, while the reservoir trajectories, including the expected reservoir level, will be shown as in Figure 5.13.

The reservoir trajectory from SHOP is similar to the topmost blue line in Figure 5.13, filling the reservoir as fast as possible and stabilizing on the maximum reservoir level. The reason R2 is not filled instantly is the presence of a maximum constraint on the gate connecting R1 and R2. A slightly rounded shape is observed for the head in the first days, which is due to the volume/head relation of the reservoir.

Another point considered in this test is the sensitivity of the solutions with regards to the form of the input scenarios. The inflow scenarios have been manipulated by multiplying the deviation from the average with a factor α . Then the 52 manipulated scenarios have been used to generate a tree with $\epsilon_{rel} = 0.10$. This tree have been reduced using $\epsilon_{rel} = 0.50$, as in Tests 3 - 6, to achieve the tree used as input. The average variances for the manipulated inflow scenarios were 4.3, 16.8 and 78.4 for $\alpha = 1, 2$ and 5, respectively.

Figures 5.12 - 5.17 shows how the amplitude of the inflow scenarios affect the scheduling, for $\alpha = 1, 2, 5$.

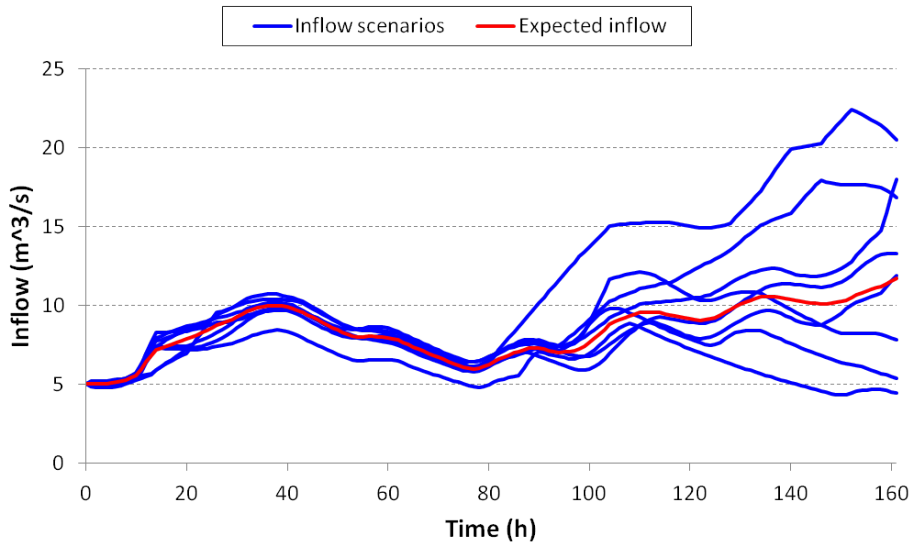


Figure 5.12: The inflow scenarios used in Test 8, with $\alpha = 1$. The expected value of the inflow from this tree is shown in red.

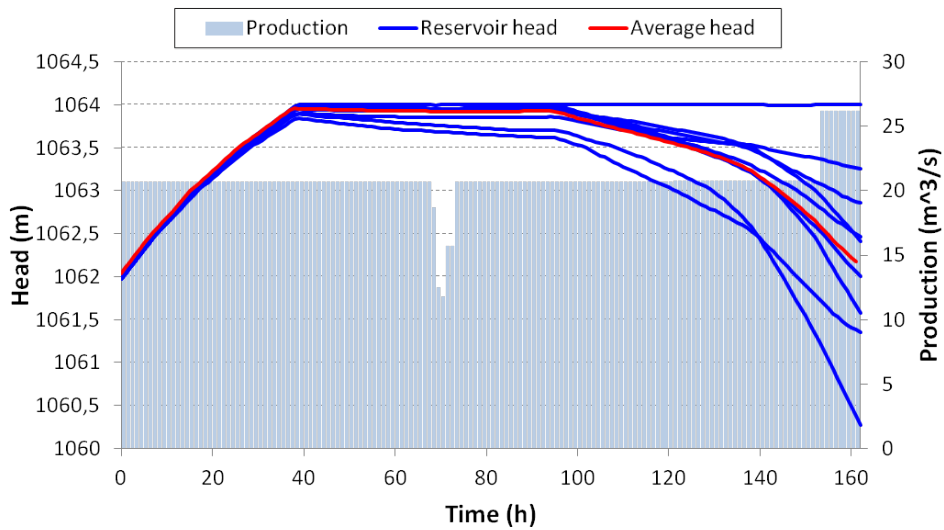


Figure 5.13: The figure shows the reservoir trajectories of R2 in test system 2 for the inflow scenarios shown in Figure 5.12, using $\alpha = 1$. The red line shows the expected reservoir trajectory, and the light blue bars represent the production discharge through P1.

The inflow scenarios in Figure 5.12 do not really branch out until the fourth day of the optimization period. This is reflected in the reservoir trajectory, which is kept quite high in the first 4 days before being lowered towards the end. By lowering the reservoir level in the final 3 days, spillage is indeed

avoided for all scenarios. It should be noted that the expected trajectory never touches the maximum level.

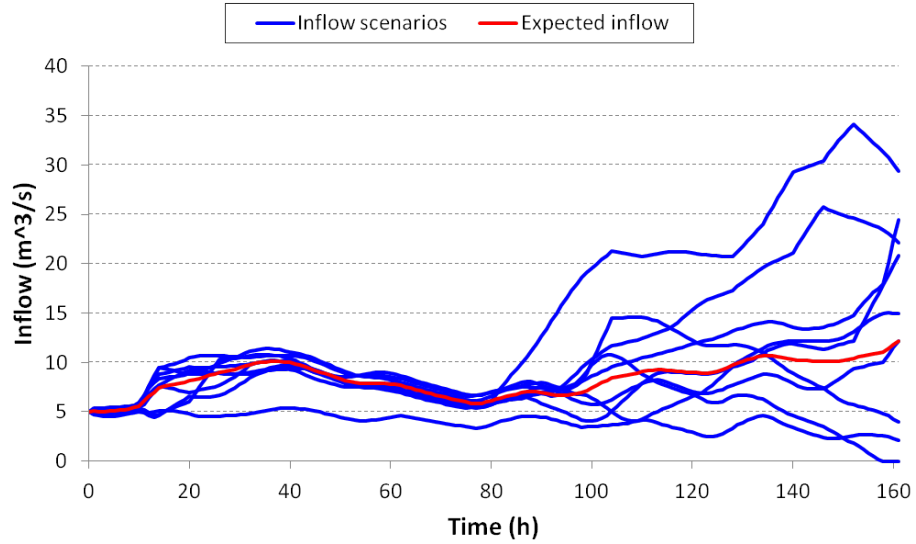


Figure 5.14: The inflow scenarios used in Test 8, with $\alpha = 2$. The expected value of the inflow from this tree is shown in red.

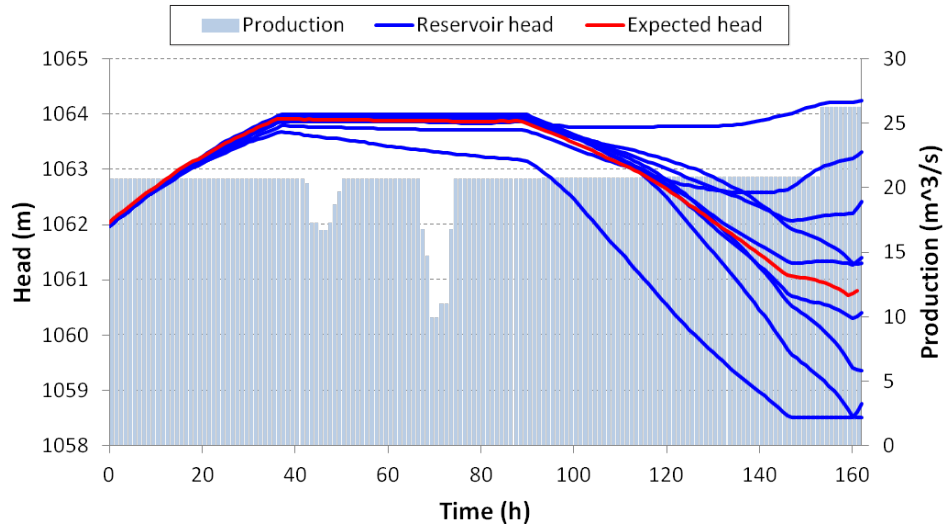


Figure 5.15: The figure shows the reservoir trajectories of R2 in test system 2 for the inflow scenarios shown in Figure 5.14, using $\alpha = 2$. The red line shows the expected reservoir trajectory, and the light blue bars represent the production discharge through P1.

In Figure 5.14, the deviations from the average in the inflow scenarios have

been multiplied by $\alpha = 2$. The expected reservoir trajectory seen in Figure 5.15 is similar to the previous case, but the level is drawn down about 1 meter further. In this case, the most extreme scenario leads to spillage from R2 in the final hours of the period.

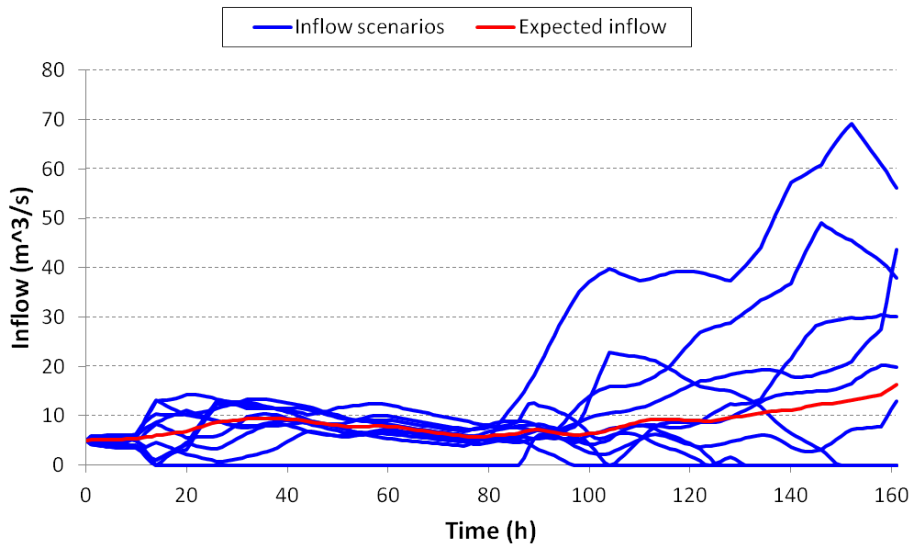


Figure 5.16: The inflow scenarios used in Test 8, with $\alpha = 5$. The expected value of the inflow from this tree is shown in red.

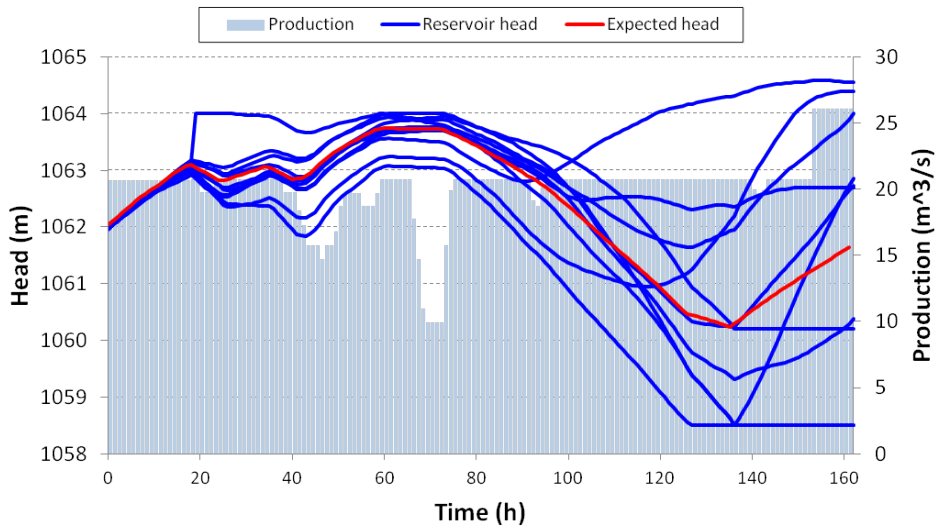


Figure 5.17: The figure shows the reservoir trajectories of R2 in test system 2 for the inflow scenarios shown in Figure 5.16, using $\alpha = 5$. The red line shows the expected reservoir trajectory, and the light blue bars represent the production discharge through P1.

When using $\alpha = 5$ the shape of the trajectory changes, exhibiting a more wavy form. The reservoir level is not taken as high as in the 2 previous cases. Spillage from R2 is observed for the 2 scenarios with highest inflow. Here, the behavior for low inflow can also be studied, as several scenarios have zero inflow in much of the period. The most drastic consequence of this is seen when the spill gate of R1 is used to obtain the common production for a low inflow scenario in hour 19.

From Figures 5.13, 5.15 and 5.17 it is observed that increasing the amplitude of the inflow leads to a small reduction in the production. Taking the reservoir level below the lowest regulated level (LRL) should be avoided, and is therefore associated with a severe penalty cost in both SHARM and SHOP. When α is increased, more scenarios with almost no inflow appear. As the production is common to all scenarios, it is reduced so that the reservoir should not be emptied in these cases.

From the results shown above, it is clear that the reservoir handling by SHARM display more robustness than that of SHOP. Using the original inflow scenarios, spillage was avoided for all cases with the given production and discharge schedule. The test supports the hypothesis put forward in the example in Section 3.1, that a stochastic model is suited to perform scheduling on this kind of topology.

Experienced production planners at Statkraft have verified that the expected reservoir trajectory from SHARM is satisfactory. It will also achieve a higher head, and hence give better effect, than the somewhat too risk averse manual plans used today.

With regards to the changes in input, SHARM reacts as expected. The schedules respond to the new input when it is possible, keeping the reservoir level lower when the amplitude is increased. It can be noted that the solution time increased with α . For $\alpha = 1, 2$ and 5 , the solution times were 210, 273 and 424 seconds, respectively. This may be explained in part by the spillage that occurred in the latter 2 cases, and the corresponding penalty cost calculations.

5.6 Similar investigations

To contextualize the results, especially those of Tests 1 and 3 - 6, it can be noted that earlier investigations of stochastic short-term scheduling, such as [26] and [31], also have found a slight increase in income when using a stochastic model compared to a deterministic. The former applies a simplified model based on the one described in [9] to a cascaded system with 3 plants, using stochastic price and inflow. Less than 1% increase in the objective value is achieved when scheduling the day ahead. The latter develops a model based on the one presented in [8], but with increased time horizon to overcome problems in the coupling to the long-term model. A simulation

framework is used to compare the performance of the stochastic and deterministic models under price and inflow uncertainty for a period of 20 weeks. For a system similar to test system 1, an increased income of around €40 000 is found for this period, corresponding to around 0.1% of the total revenue.

Chapter 6

Conclusion

The objective of this thesis has been to evaluate the utilization of a stochastic programming model for short-term hydro power scheduling. As opposed to some of the earlier work on this topic, the focus has been on the real-life application of such a tool. By considering a prototype of the stochastic model SHARM, which is based on the widely used SHOP model, it has been possible to use the same degree of detail as in the current operative scheduling.

The first chapters have presented a brief introduction to hydro power scheduling, and described stochastic programming and its applications to this field.

A range of different tests have been conducted to evaluate the stochastic model. The evaluation process has considered two main objectives. First, the results from the SHARM prototype have been compared to those from SHOP with price and inflow uncertainty. Both objective function values and reservoir handling have been considered. The second objective has been to examine the computational performance of the new model. Both these approaches have aimed at revealing the potential of using a stochastic model in operational hydro power scheduling, from a financial and practical point of view. The conclusions based on the work of this thesis are presented below.

6.1 Comparison against SHOP

The comparison between the 2 models have mainly focused on the construction of day-ahead plans. Using a week-long optimization period, the deterministic plans have been imposed as schedules on SHARM runs with stochastic scenario trees. The resulting objective function values have been compared to corresponding runs without a deterministic day-ahead plan, as described in Section 4.3.3.

The stochastic parameters considered in this work are inflows and prices. Scenarios have been provided by the hydrology and price forecasting depart-

ments at Statkraft, and corresponds to the ones used operationally. Evaluation of day-ahead plans have been carried out for the 2 real-life HPP hydro power systems described in Section 4.1.

Applying stochastic prices only, test system 1 was run for 24 days in the winter depletion season, using input from the operational scheduling. Deterministic day-ahead plans were produced and compared to the stochastic scheduling. An average gain of €88 was achieved by using stochastic scheduling. However, only 4 of the 24 cases contributed increased profit above the iteration accuracy. No definite explanation has been found, but it is suggested that these earnings appear as the result of peaks in some of the price scenarios for the day ahead. This seems logical, as SHARM in these cases sees information that the deterministic model does not. The majority of the increased income stems from an increase in the traded power during the optimization period.

Similar analyses were conducted for a week in the start of May, during the spring thaw. These tests considered stochastic inflows and a combination of stochastic inflows and prices for both test system 1 and 2. Even smaller differences were observed between deterministic and stochastic day-ahead plans. The results for test system 2 were practically identical, while there was one case for test system 1 that produced an increased income above the iteration accuracy. As the time span of these tests were just one fourth of the one considering price uncertainty, it is difficult to conclude on which of the uncertain parameters affect the scheduling the most.

Taking a more qualitative approach, the reservoir handling for the smaller reservoir in test system 2 was considered for different inputs. This topology is not handled satisfactorily by SHOP, so it is currently operated manually. A common plan for production and discharge for all scenarios in the whole optimization period was specified in SHARM. The resulting expected reservoir trajectory was found to meet the expectations of the system operators, and spillage was avoided for all the operational input scenarios. In such situations, the stochastic alternative outperforms the deterministic scheduling model. Using SHARM to create robust, week-long plans may be an approach that should be considered for other situations as well.

If such plans from SHARM can be trusted, this approach could be of great benefit for the producers. In addition to producing a more optimal plan based on the given input, the need for manual adjustments will be reduced. This will in turn mean that more time can be spent on other tasks, and the risk of human error is avoided. Manual specification of new reservoir limits, such as in test system 2, are based on experience, and can hence not be guaranteed to be optimal.

To summarize, the use of stochastic scheduling to produce day-ahead plans seems to have little effect on the expected income of the producer compared to the current deterministic model. In this work, 2 different water-courses have been considered, with price uncertainty in the winter depletion

season and price and price/inflow uncertainty in the spring flooding season. The largest differences are found when considering price uncertainty, usually if some of the scenarios deviate significantly from the rest in the day-ahead period.

It should be noted that this is based on the method of evaluation described in Section 4.3.3, which use the given input scenarios to represent all possible outcomes. This means that the form and distribution of the input scenarios will have a strong effect on the results. The range of systems considered has been limited by several factors, and may not represent situations where the effect of a stochastic model is expected to be most pronounced. Reviewing the results of the tests for increased income, it is evident that the slight difference does not justify replacing SHOP by SHARM in all cases, especially when considering the increased CPU-time.

The small differences may be explained in many ways. One factor is that the situations considered in Tests 1 and 3 - 6 did not include high risks of flooding or emptying reservoirs. It is also possible that comparing the reservoir handling and bidding by SHARM and SHOP over a longer period will give a different picture of the expected profit.

Using SHARM to generate a single production and discharge schedule for the week ahead results in a sensible expected reservoir trajectory for test system 2. A robust reservoir handling is observed, avoiding spillage for all scenarios while still maintaining a reasonably high head. Applying SHARM to this situation illustrates the effect of considering uncertainty more clearly than comparing the objective function values. In such cases, where manual scheduling is applied today, SHARM may be more beneficial.

6.2 Computational performance of SHARM

Supplementing the SHARM prototype is an implementation of algorithms for construction and reduction of scenario trees. The in-sample and out-of-sample stability of the reduction algorithm described in Section 3.3.3 and Appendix B have been investigated for a scenario fan with price uncertainty, and found to be very satisfactory. In-sample analyses for scenario trees with inflow and price/inflow uncertainty have also been conducted. The results are in accordance with those presented in [14] and suggests that the reduction algorithm has good stability properties.

The run-time of SHARM with scenario trees constructed using different degrees of reduction have been examined for different systems and inputs. For test systems 1 and 2, with both inflow and price/inflow uncertainty, a reduction strategy is proposed. The strategy is to first construct a tree from the original fan using a degree of reduction $\epsilon_{rel} = 0.10$ and then reducing this tree with $\epsilon_{rel} = 0.50$. It is found that the solution time for these cases using the resulting tree usually is below 100 seconds. Solving corresponding

deterministic cases in SHARM takes approximately 25 seconds, and using SHOP on the operational servers further reduce the CPU-time of these cases to a couple of seconds. At the same time, little of the information is lost. The most extreme scenarios remain and the deviations in objective value are generally well below 1%.

It is difficult to conclude on the computational performance of the stochastic model. In an operational setting it is necessary to optimize all the plants and reservoirs within a price area, which can consist of 10 watercourses or more. Due to the limited features of the prototype, it has not been possible to test SHARM on systems of this size in the course of this work. On single systems, it is shown that relatively low solution times can be achieved while still reflecting much of the original stochastic information, but it is not known how the CPU-time will scale with the system size.

6.3 Final conclusions

Testing for increased profit: Based on the results produced in this thesis, it seems there is little effect of using stochastic scheduling on the objective function. Two different watercourses have been tested on, and uncertainty in inflows, prices and both have been applied. The observed increase in income is mainly due to increased trading profit, usually in the presence of some extreme price scenario early in the optimization period. It should be noted that only a couple of situations and system states have been considered here, so there may very well be other cases where the effect of considering uncertainty is more significant.

Testing of reservoir handling: Using SHARM to handle the small intake reservoir R2 in test system 2 is more successful, resulting in a robust plan that avoids spillage. The results presented here suggests that running SHARM with stochastic parameters gives a better reservoir handling than SHOP. The changes in production are less pronounced, especially when considering day-ahead plans.

Testing the tree reduction: The scenario tree construction and reduction algorithms supplementing SHARM performs well in terms of stability. By reducing the size of the input, the solution times have been cut short significantly without too large deviations in the objective. This is promising, and necessary if SHARM should be used in the day-to-day operational scheduling.

Final recommendations: As mentioned in Section 5.6, the expected increase in profit from considering uncertainty is found to be limited also in other studies. One conclusion from this work is that there seems to be little

reason to replace SHOP with SHARM in all situations, since the expected increase in profit is low and the CPU-times are much higher. The results from Test 8 does however suggest that SHARM performs well in cases where the uncertainty is more significant, so focusing the SHARM effort on such situations may be advisable.

Chapter 7

Suggestions for further development

The development of SHARM has been a part of the KMB project “Optimal Short-term Scheduling of Wind and Hydro Resources”. Statkraft and the other participating production companies must decide, based on the performance of the prototype, whether they will finance a full implementation of SHARM or not. This means that the further work on this topic has different time horizons. First, some further testing should be done to find out if the expected profit of using stochastic scheduling is worth the investment. If it is decided to go ahead with SHARM, it is necessary to expand the current implementation in several areas. This section will present some of the most pressing tasks, both in the short and longer term.

In the immediate future, it is natural to focus on two main areas. The *method of evaluation*, and the *application of SHARM to other functions*.

Concerning the method of evaluation, alternatives to the one presented here should aim at investigating the long-term effects of using stochastic scheduling in more detail. From e.g. Test 8, it is evident that a more robust reservoir handling is one of the most desirable features of a stochastic model. This feature is not captured by the day-ahead approach taken here, and a scheduling framework that handles a reservoir over an extended period should be applied. The most practical way to do this will probably be to construct a simulation environment, and run SHOP and SHARM in parallel to see how they will manage a given system. Only in this way can a proper estimate of the expected profit of considering uncertainty be produced.

An extension of the method used here will be used in the future work [23] to evaluate the performance of SHARM in a lack-of-water situation and a situation with potentially high inflow late in the scheduling period. This method is similar to the one used in [14], but considers stochastic prices as well.

The qualitative approach taken for Test 8 in Section 5.5 may also be ap-

plied to other topologies. In situations where manual scheduling is performed because the SHOP plans are not optimal, there can be much to gain from applying stochastic scheduling. If these plans can be trusted, one gets the benefit of not needing any manual scheduling in addition to a more optimal plan.

Another aspect may be to assess the input scenarios. The scenarios used in this work have generally had a relatively low variability, especially in the first few days. Even though the forecasts may be quite reliable, it might be advisable to include some more extreme scenarios with low probability to account for unforeseen events. The effect of adding such scenarios could be considered, as it has not been tested specifically in this work. There is also the question of correlation between inflow and price, that could be investigated. Finally, run-time tests should be performed on larger systems. This may require either a further development of the SHARM prototype, or a simplification of some of the power system models.

Other applications of SHARM have not been discussed in detail in this work. Such possibilities should however be considered when deciding to go forward with the project or not. This will be left for the production companies to investigate for themselves, but areas such as trading in the balancing markets, calculation of water values and transition to seasonal or long-term models can be mentioned. If SHARM should be used for balance power trading, the price in this market has to be modeled as a stochastic variable.

If it is decided to go forward with the project, and implement a full version of SHARM, other challenges must be considered. Obviously, all the features of SHOP must be implemented in SHARM. Using a stochastic programming model will necessarily lead to longer solution times, so measures to limit this downside should be taken. One alternative could be to investigate the possibility of using parallel processing in the solution.

The current framework for constructing and reducing scenario trees should be reconsidered. Especially the current tool for creating scenario fans on XML-format has limitations, both in terms of CPU-time and the possible size of the fans. This should probably be carried out by each producer separately, to allow for coupling to the scenario generation method used. To succeed, it is important that the research and industry work together to find a solution that also respect the operational requirements.

Bibliography

- [1] M. M. Belsnes, O. B. Fosso, and J. Røystrand. Handling state dependent nonlinear tunnel flows in short term hydropower scheduling. In *Proceedings of the 2004 International Conference on Power System Technology, POWERCON 2004*, Singapore, 2004.
- [2] M. M. Belsnes, O. B. Fosso, J. Røystrand, T. Gjengedal, and E. Valhovd. Unit commitment in hydro power operation scheduling. In *Hydropower '03*, Stavanger, Norway, 2003.
- [3] S. Bergström. Development and application of a conceptual runoffmodel for Scandinavian catchments. Technical report, Norrköping, SMHI Report RHO 7, 1976.
- [4] G. L. Doorman. ELK15 Hydro Power Scheduling. Department of Electrical Power Engineering NTNU, Autumn 2010. Course syllabus.
- [5] J. Dupačová. Applications of stochastic programming: Achievements and questions. *European Journal of Operational Research*, 140(2):281–290, 2002.
- [6] J. Dupačová, N. Gröwe-Kuska, and W. Römisch. Scenario reduction in stochastic programming: An approach using probability metrics. *Mathematical Programming*, 511(3):493–511, 2003.
- [7] E. Faria and S.-E. Fleten. Day-ahead market bidding for a Nordic hydropower producer: taking the Elbas market into account. *Computational Management Science*, 8(1):75–101, 2011.
- [8] S.-E. Fleten and T. K. Kristoffersen. Stochastic programming for optimizing bidding strategies of a Nordic hydropower producer. *European Journal of Operational Research*, 181(2):916–928, 2007.
- [9] S.-E. Fleten and T. K. Kristoffersen. Short-term hydropower production planning by stochastic programming. *Computers and Operations Research*, 35(8):2656–2671, 2008.

- [10] S.-E. Fleten, K. Høyland, and S. W. Wallace. The performance of stochastic dynamic and fixed mix portfolio models. *European Journal of Operational Research*, 140(1):37–49, 2002.
- [11] Turid Follestad. Scenario generation and scenario reduction for short-term hydropower scheduling models. Technical report, Sintef Energy Research, 2009.
- [12] Turid Follestad. A summary of literature on stochastic short-term hydropower scheduling and bidding. Technical report, Sintef Energy Research, 2010.
- [13] Turid Follestad. Documentation of the C program Scentreegen for scenario tree generation and reduction. Technical report, Sintef Energy Research, 2012.
- [14] Turid Follestad, Ove Wolfgang, and Michael M. Belsnes. An approach for assessing the effect of scenario tree approximations in stochastic hydropower scheduling models. In *Power Systems Computation Conference (PSCC)*, Stockholm, Sweden, 2011.
- [15] O. B. Fosso, M. Belsnes, J. Røystrand, T. Gjengedal, and E. Valhovd. Hydro power short-term scheduling in an online environment. In *Med-Power 2002*, Athens, Greece, 2002.
- [16] O. B. Fosso and M. M. Belsnes. Short-term hydro scheduling in a liberalized power system. In *Proceedings of the 2004 International Conference on Power System Technology, POWERCON 2004*, Singapore, 2004.
- [17] GAMS. GAMS/SCENRED Documentation. <http://www.gams.com/dd/docs/solvers/scendred.pdf>, March 2012.
- [18] N. Gröwe-Kuska, H. Heitsch, and W. Römisch. Scenario reduction and scenario tree construction for power management problems. In *Proceedings of the IEEE Bologna Power Tech Conference*, Bologna, Italy, 2003.
- [19] H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24(2-3):187–206, 2003.
- [20] H. Heitsch and W. Römisch. Generation of multivariate scenario trees to model stochasticity in power management. In *Proceedings of the IEEE St. Petersburg Power Tech Conference*, St. Petersburg, Russia, 2005.
- [21] H. Heitsch and W. Römisch. A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, 35(6):731–738, 2007.

- [22] H. Heitsch and W. Römisch. Scenario tree reduction for multistage stochastic programs. *Computational Management Science*, 6(2):117–133, 2009.
- [23] Ingrid Honve, Turid Follestad, and Michael M. Belsnes. A model for operational stochastic short-term scheduling for hydropower systems - examples of use. To be presented at the 4th IEEE and Cigré International Workshop “Hydro scheduling in competitive markets”, Bergen, Norway, 2012.
- [24] M. Kaut and S. W. Wallace. Evaluation of scenario generation methods for stochastic programming. In W. Römisch and J. L. Hight, editors, *Stochastic Programming E-Print Series*. Humbolt-Universität Berlin, Institut für Mathematik, 2003.
- [25] T. K. Kristoffersen and S.-E. Fleten. The development of stochastic programming models for planning short-term power generation and bidding. In E. Bjørndal, M. Bjørndal, and M. Rönnquist, editors, *Energy, Natural Resource and Environmental Economics*. Springer, 2009.
- [26] H. K. Lumb and V. K. Weiss. Short-term hydropower scheduling - a stochastic approach. Norwegian University for Science and Technology, December 2006. Project report.
- [27] B. Mo, A. Haugstad, and O. B. Fosso. Integrating long and short-term models for hydro scheduling. In *Proceedings of Hydropower '97*, pages 279–286, Rotterdam, Netherlands, 1997.
- [28] Statistisk Sentralbyrå. Statistisk sentralbyrå: Energi - temaside, April 2012.
- [29] A. Shapiro. Inference of statistical bounds for multistage stochastic programming problems. *Mathematical Methods of Operations Research*, 58(1):57–68, 2003.
- [30] Nord Pool Spot. The Nordic electricity exchange and the Nordic model for a liberalized electricity market. <http://www.nordpoolspot.com/How-does-it-work/>, February 2012.
- [31] Jens Arne Steinsbø. Constructing bidding curves for a hydroower producer at NordPool. Master’s thesis, Norwegian University for Science and Technology, June 2008.
- [32] Kim Stenshorne. A framework for constructing and evaluating probabilistic forecasts of electricity prices: A case study of the Nord Pool market. Master’s thesis, Norwegian University for Science and Technology, June 2011.

- [33] S. W. Wallace and S.-E. Fleten. Stochastic programming models in energy. In A. Shapiro and A. Ruszczyński, editors, *Stochastic programming*. Elsevier, 2003.
- [34] Ove Wolfgang. Comparing simulation results for stochastic and deterministic strategies under uncertainty. Technical report, Sintef Energy Research, 2009.
- [35] K. Høyland, M. Kaut, and S. W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2-3):169–185, 2003.

Appendix A

List of symbols

Symbol	Description
I	Set of nodes, $i \in I = \{1, \dots, N\}$.
$I_i^{par} \subset I$	Set of parent nodes for every $i \in I$. Only one parent per node.
$I_{end} \subset I$	Set of end nodes in I , i.e. the nodes in the final period.
E	Set of steps in the discretized end-value function for stored water.

Table A.1: List of the sets used in the model example of Section 3.4.

Symbol	Unit	Description
X_{ini}	MWh	Initial reservoir content.
X_{max}	MWh	Reservoir capacity.
p_i	1	Probability for node i .
y_i	€/ MWh	Spot price in node i .
c_j	€/ MWh	Marginal value of stored water, step $j \in E$.
d_j^{max}	MWh	Capacity for step j in end-value function.
g_{max}	MWh	Generator capacity.
N	1	Number of nodes.
w_i	€	End-value of stored water.

Table A.2: List of the parameters used in the model example of Section 3.4.

Symbol	Unit	Description
q_i	Mwh	Inflow to the reservoir.

Table A.3: The stochastic variable used in the model example of Section 3.4.

Symbol	Unit	Description
g_i	MWh	Generation.
s_i	MWh	Spillage.
X_i	MWh	Reservoir contents at the end of the period.
d_{ij}	MWh	Use of step j in the discretized end-value function for node i .
π	€	Objective function. Expected future profit.

Table A.4: List of other variables used in the model example of Section 3.4.

Appendix B

Algorithms

This appendix presents the two algorithms mentioned in Section 3.3.3, and the nomenclature is the same as in that section. Along with a technical description of each algorithm, a short conceptual explanation is included.

B.1 Fast forward selection

Algorithm B.1 is used for scenario reduction. The starting point is a set of N scenarios, and the objective is to remove $N - n$ of these to end up with the reduced tree that is closest to the original one in terms of a probability metric. This is achieved by selecting one scenario in each of n stages. At every stage the distances between the remaining scenarios in terms of c_T is found, and the probability weighted distance from one scenario to all the others is calculated. The scenario closest to the other remaining ones is chosen, and its index removed from the set J of deleted scenarios.

Algorithm B.1 Fast forward selection

$$c_{ku}^{[1]} \leftarrow c_T(\xi^k, \xi^u), k, u = 1, \dots, N;$$
$$z_u^{[1]} \leftarrow \sum_{\substack{k=1 \\ k \neq u}}^N p_k c_{ku}^{[1]}, u = 1, \dots, N;$$
$$u_1 \in \arg \min_{u \in \{1, \dots, N\}} z_u^{[1]};$$
$$J^{[1]} \leftarrow \{1, \dots, N\} \setminus \{u_1\};$$

for $i = 2 \dots n$ **do**

$$c_{ku}^{[i]} \leftarrow \min\{c_{ku}^{[i-1]}, c_{ku_{i-1}}^{[i-1]}\}, k, u \in J^{[i-1]};$$
$$z_u^{[i]} \leftarrow \sum_{k \in J^{[i-1]} \setminus \{u_i\}} p_k c_{ku}^{[i]}, u \in J^{[i-1]};$$
$$u_i \in \arg \min_{u \in J^{[i-1]}} z_u^{[i]};$$
$$J^{[i]} \leftarrow J^{[i-1]} \setminus \{u_i\};$$

end for

Redistribute probabilities by (3.4);

The index set $J^{[n]}$ contains $N - n$ deleted scenarios.

B.2 Backwards construction

Algorithm B.2 is used for scenario tree construction. The starting point is a distribution P approximated by a set of N scenarios $\xi^i = (\xi_1^i, \dots, \xi_T^i)$ in the form of a scenario fan. This implies that all scenarios have the same starting point ξ_1^* at $t = 1$ and branching only at the first stage. P and the tolerance $\epsilon > 0$ is assumed given. The objective is to find a distribution P_ϵ whose scenarios form a tree with root node ξ_1^* , has fewer nodes than P , and satisfies

$$D_{K,J}(P, P_\epsilon) \leq \epsilon. \quad (\text{B.1})$$

Now a recursive scenario reduction is performed on the horizon $\{1, \dots, t\}$, where t is reduced recursively from $t = T$ to $t = 2$. For a given time horizon $\{1, \dots, t\}$, the following relative costs are considered:

$$c_t(\xi, \tilde{x}i) := \sum_{\tau=1}^t \|\xi_\tau - \tilde{\xi}_\tau\|. \quad (\text{B.2})$$

Equation (B.2) corresponds to Equation (3.2) for $r = 1$. It can be shown that Equation (B.1) holds for any scenario tree P_ϵ constructed by the following algorithm:

Algorithm B.2 Backwards construction

Set $\epsilon_t > 0, t = 2, \dots, T$, so that $\sum_{t=2}^T \epsilon_t \leq \epsilon$;
 $I_{T+1} \leftarrow \{1, \dots, N\}, p_{T+1}^i \leftarrow p^i, \forall i \in [1, N]$;
for $m = 1 \dots T - 1$ **do**
 $t \leftarrow T + 1 - m$;
 Determine $I_t \subset I_{t+1}$ by scenario reduction such that
 $\sum_{i \in I_{t+1} \setminus I_t} (p_{t+1}^i \min_{j \in I_t} c_t(\xi^i, \xi^j)) \leq \epsilon_t$;
 Update the probabilities p by $p_t^j = p_{t+1}^j + \sum_{i \in J_{tj}} p_{t+1}^i$, with
 $J_{tj} = \{i \in I_{t+1} \setminus I_t : j = j_t(i)\}$ and
 $j_t(i) = \arg \min_{j \in I_t} c_t(\xi^i, \xi^j), i \in I_{t+1} \setminus I_t$;
end for
Construct P_ϵ : Determine recursively mappings $\alpha_t : I_T \rightarrow I_t$ for $t = T, \dots, 2$, where $\alpha_T \leftarrow id|_{I_T}$ and such that

$$\alpha_t(i) = \begin{cases} j_t(\alpha_{t+1}(i)) & \text{if } \alpha_{t+1}(i) \in I_{t+1} \setminus I_t, \\ \alpha_{t+1}(i) & \text{else,} \end{cases} \text{ for } t = T - 1, \dots, 2;$$
Determine scenarios $\tilde{\xi}^j$ for $j \in I_T$ with $\tilde{\xi}_1^j = \xi_1^*$, and $\tilde{\xi}_t^j = \xi_1^{\alpha_t(j)}$ for $t = 2, \dots, T$;
 $\tilde{p}^j \leftarrow p_T^j$ and $P_\epsilon \leftarrow \sum_{j \in I_T} \delta_{\tilde{\xi}^j}$;

The tolerances are chosen recursively, as described in [20], to be

$$\epsilon_T = \epsilon(1 - q), \quad q \in (0, 1), \quad (\text{B.3})$$

$$\epsilon_t = q\epsilon_{t+1}, \quad t = T - 1, \dots, 2. \quad (\text{B.4})$$

Choosing q closer to 1 leads to more branching points and a higher number of remaining scenarios, while a q closer to 0 has the opposite effect. In the scenario tree reductions and constructions performed in this work, q is chosen to be 0.95.

Appendix C

Additional results

The results presented here are the same as in Chapter 5, given in numbers in stead of graphs. The total objective function value is calculated as the sum of buy/sale income and end-reservoir value, minus penalty costs.

Test case	Difference [€]			
	Buy/Sale	End-reservoir	Penalties	Total
1	1,23	0,30	0	1,53
2	5,30	-0,70	0	4,60
3	217,33	-227,60	0	-10,27
4	14,17	-97,36	0	-83,19
5	458,71	-587,63	0	-128,92
6	157,93	-5,21	0	152,72
7	-14,56	5,92	0	-8,64
Average	120,02	-130,33	0	-10,31

Table C.1: The table shows detailed results from Test 3. The same results are presented in Figure 5.9. The deterministic objectives were in the range of €7 - 12 million.

Test case	Difference [€]			
	Buy/Sale	End-reservoir	Penalties	Total
1	1,93	-0,34	0	1,59
2	122,60	-123,48	0	-0,88
3	-256,10	358,26	0	93,16
4	524,61	-613,33	0	-88,72
5	219,25	-34,66	0	184,59
6	1,11	0,56	0	1,67
Average	100,73	-68,83	0	31,90

Table C.2: The table shows detailed results from Test 4. The same results are presented in Figure 5.10. The deterministic objectives were in the range of €2 - 3 million.

Test case	Difference [€]			
	Buy/Sale	End-reservoir	Penalties	Total
1	303,88	203,09	0	506,97
2	-66,58	61,31	0	-5,27
3	-29,28	35,50	0	-6,22
4	104,77	-27,34	0	77,43
5	-15,86	22,85	0	6,99
6	3,57	0,87	0	4,44
7	58,53	-8,25	0	50,28
8	827,29	1,45	0	828,74
9	345,58	35,26	0	380,84
10	6,72	1,82	0	8,54
11	-1,69	-0,19	0	-1,88
12	-71,73	53,61	0	-18,12
13	-4,83	48,02	0	43,19
14	105,20	-109,41	0	-4,21
15	4,68	-0,60	0	4,08
16	19,11	2,55	0	21,66
17	-2,37	0,43	0	-1,94
18	0,79	0,38	0	1,17
19	0,03	198,18	0	198,21
20	-0,82	0,88	0	0,06
21	-0,55	0,19	0	-0,36
22	-3,91	1,37	0	-2,54
23	18,14	-1,92	0	16,22
24	-5,93	8,86	0	2,93
Average	66,45	22,04	0	87,97

Table C.3: The table shows detailed results from Test 1. The same results are presented in Figure 5.1. The deterministic objectives were in the range of €16 - 18 million.

Test case	Difference [€]			
	Buy/Sale	End-reservoir	Penalties	Total
1	33,53	-17,77	0	15,76
2	29,64	-15,94	0,01	13,70
3	28,83	-20,38	0	8,45
4	24,24	-18,95	0	5,29
5	25,71	-18,94	0	6,77
6	29,86	-18,95	0	10,91
7	28,64	-18,94	0	9,70
Average	28,64	.18,55	0,00	10,08

Table C.4: The table shows detailed results from Test 5. These results are not presented anywhere else in the report. The deterministic objectives were in the range of €8 - 13 million.

Test case	Difference [€]			
	Buy/Sale	End-reservoir	Penalties	Total
1	32,27	-17,30	0	14,97
2	29,79	-20,48	0,03	9,31
3	23,62	-18,95	0	4,67
4	273,72	-267,94	0	5,78
5	30,14	-18,96	0	11,18
6	30,42	-18,94	0	11,48
Average	69,99	-60,43	0,01	9,57

Table C.5: The table shows detailed results from Test 6. These results are not presented anywhere else in the report. The deterministic objectives were in the range of €3 - 4 million.