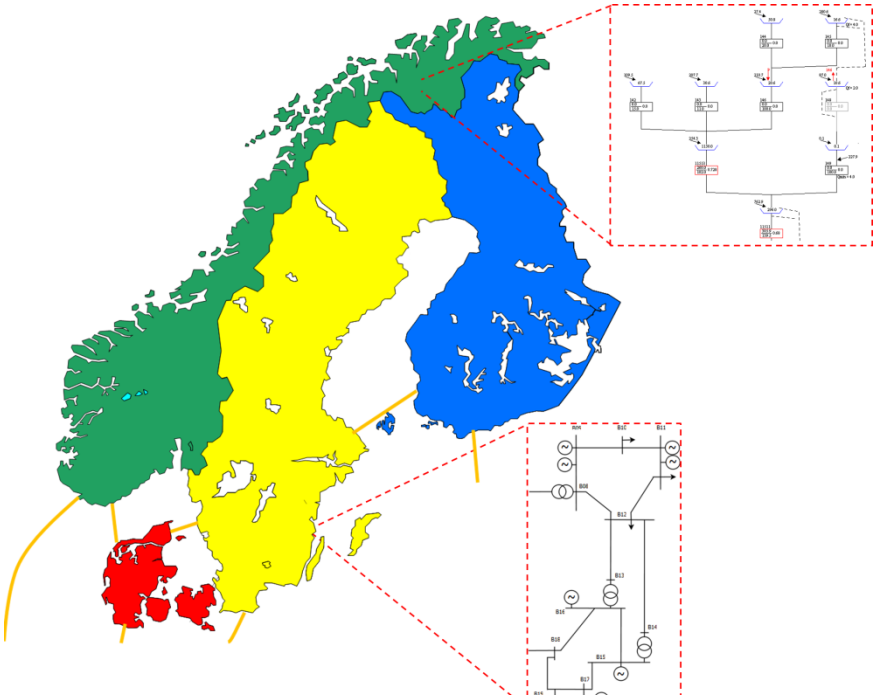
TR A7375- Unrestricted

Report

Stochastic optimization model with individual water values and power flow constraints

Choice of method

Authors
Arild Helseth
Birger Mo
Knut S Gjerden
Ove Wolfgang



SINTEF Energy AS
Production Planning
2014-01-10

SINTEF

SINTEF Energi AS
SINTEF Energy AS

Address:
Postboks 4761 Sluppen
NO-7465 Trondheim
NORWAY

Telephone:+47 73597200
Telefax:+47 73597250

energy.research@sintef.no
www.sintef.no/energi
Enterprise /VAT No:
NO 939 350 675 MVA

KEYWORDS:
Market Modelling
Hydro Scheduling
Inflow Models

Report

Stochastic optimization model with individual water values and power flow constraints

VERSION
1.0

DATE
2014-01-10

AUTHORS
Arild Helseth
Birger Mo
Knut S Gjerden
Ove Wolfgang

CLIENTS
Statnett
Statkraft
NVE
BKK

CLIENT'S REF.
Ivar Døskeland
Arve Tveter Iversen
Eirik Øyslebø
Frode Haga

PROJECT NO.
12X851

NUMBER OF PAGES/APPENDICES:
79

ABSTRACT

The aim of the SOVN ("Stochastic optimisation model for Scandinavia with individual water values and grid restrictions") project is to develop a computer program suitable for simulating the Northern European power supply market using individual water values in Scandinavia and power flow constraints.

This report compares two alternative methods suitable for simulating the Northern European power market with individual water values and power flow constraints. Both methods are based on stochastic optimization without hydro system model aggregation. The presented comparison formalises the decision basis for the choice of method in the SOVN research project.

The basic theoretical foundations of the two methods are presented. Previous experiences with these methods are discussed, as well as expected strengths and weaknesses. Some basic testing was performed and documented to gain experience with realistic problem sizes and computation times.

PREPARED BY
Arild Helseth

SIGNATURE
Arild Helseth

CHECKED BY
Michael M. Belsnes

SIGNATURE
Michael Belsnes

APPROVED BY
Magnus Korpås

SIGNATURE
Magnus Korpås

REPORT NO.	ISBN	CLASSIFICATION	CLASSIFICATION THIS PAGE
TR A7375	978-82-594-3578-1	Unrestricted	Unrestricted

Document history

VERSION	DATE	VERSION DESCRIPTION
1	2013-12-09	A first draft sent to consortium participants
2	2014-01-10	Revised version, primarily with updates on discussions and conclusion

SINTEF

Table of contents

1 Introduction

In this introductory part, we present the background and motivation for the SOVN ("Stochastic optimisation model with individual water values and grid restrictions") research project. We proceed by describing the context in which existing long-term hydrothermal market models are used today, and the expected use of the new model.

1.1 Background and motivation

In the future, the Nordic power system will have closer connections with Europe and an increasing proportion of intermittent renewable generation from, for example, wind and unregulated hydroelectric systems. Rapid and unpredictable fluctuations in intermittent generation will offer new possibilities for controllable generation to be able to respond to these fluctuations. Flexible and fast-responding power plants able to produce at demand peaks will therefore see a higher revenue potential. It will become increasingly important for hydropower producers with access to controllable production to correctly estimate the value of flexibility so that the water is scheduled optimally. Under these circumstances, investment decisions in hydropower systems will call for tools that are able to correctly predict the value of flexibility.

Nordic market players and system operators are faced with major investment decisions connected with possibilities for improved flexibility of the generation system (power output expansions and pumping schemes) and an increase in the number of cable connections to Europe.

From a Norwegian perspective, it is important to make sustainable decisions as to whether to invest in additional cables to the Continent and improvements in the flexibility of the hydroelectric generating system. Co-ordinated development of flexibility in the generating system and cables to other countries will call for accurate and verifiable modelling which documents the profitability and consequences of investments.

It is therefore a need to develop a new stochastic optimization model where the hydropower system is represented without aggregation. Such as model will give better and more robust investment decisions than what can be achieved with the currently available market models.

In the SOVN research project, the aim is to develop a computer program to simulate the Northern European power supply market using individual water values in Scandinavia. The operation of each individual hydroelectric reservoir shall be based on the result of formal stochastic optimisation in which all the relevant physical attributes of the market are represented. In the project description we confined the search for a method to the two known methods we find the most promising; the scenario fan simulator and stochastic dual dynamic programming.

More information about the project is available at the web page http://www.sintef.no/Projectweb/SOVN1/ .

1.2 Hydro scheduling in the Nordic power market

The use of computer tools for hydro scheduling has a long tradition in in the Nordic countries. The water value method was presented in the early 60s [1], and EFI (Energiforsyningens Forskningsinstitutt) started working with computer-based scheduling tools in the same decade.

Since then, computer-based scheduling tools have evolved gradually, adapting to the rapid increase in available computing power and new market structures. Practically all market players in today's power market use (or get information from) some sort of computer-based scheduling tool. Still, these tools are often based on simplifications and heuristics due to the complexity of the problem. The two major complicating factors in the scheduling problem are:

1) **Dynamic couplings**. Reservoir storages provide dynamic couplings between the stage-wise decisions in the scheduling problem. Decisions taken today will affect the opportunities (reservoir levels) tomorrow.
2) **Uncertainty.** Uncertainty about the future state of the system will affect the decisions made today. Normally uncertainties in the long-term scheduling problem are related to weather (inflow, wind, temperature), and inflow is the single most important uncertainty to consider in the Nordic system.

The two methods compared in this report treat these two complicating factors differently, as will be elaborated on in sections 2 and 0. Before we start discussing these methods in detail, it is important to discuss the framework in which the chosen method is going to be applied. In particular, the following points are important:

- **Representation of uncertainty:** Using historical inflow records to represent future uncertainty is the preferred procedure by the players in the Nordic market. Thus, the standard use of scheduling models developed at SINTEF involves direct use of historical inflow records. The major motivation for working with 'observed' scenarios is to conserve correlations in both time and space which are not easily incorporated in stochastic models.
- **Stochastic time-resolution:** Long-and medium-term scheduling models developed at SINTEF have a basic *stochastic time resolution* of one week. That is, at the beginning of the week all stochastic variables are known for that week, but not further ahead. The total planning period is normally several years. A weekly stochastic time resolution has been found adequate for representing the mix of reservoir sizes and regulation capability in the Nordic system.
- **Market structure:** The scheduling model being developed in this project will only refer to the spot market of electricity. Balancing markets (intra-day, reserve, etc.) will not be addressed.

In this project we do not plan to challenge this framework. That is, we admit that historical inflow records will be used in building and testing the model, we will use a stochastic time-resolution of one week and we will relate the model to the spot market.

1.3 The weekly decision problem

The model development in the SOVN project will adapt the stochastic time resolution of one week. Thus, independently of the choice of method, the natural modelling "building block" in a stochastic programming context is the *weekly decision problem*. Before discussing the two methods in detail, we can therefore introduce the weekly decision problem. All functional relationships are linear or linearized, so that the problem can be formulated as a linear programming (LP) problem.

The weekly decision LP problem aims at minimizing system costs within the given week in (1) and with the given realization of stochastic variables, subject to system constraints in (2) and variable constraints in (3).

$$\mathrm{Z} = \min c^T x \qquad \textbf{(1)}$$

$$Ax = b \qquad \textbf{(2)}$$

$$x^{\min} \leq x \leq x^{\max} \qquad \textbf{(3)}$$

We will divide the week into *load periods* for which we allow a minimum time resolution of one hour. The cost components in (1) typically comprise the marginal cost of thermal generation, the value of demand and exogenous prices. The constraints in (2) will normally refer to load periods, and will comprise a set of basic constraints:

- Reservoir balances, for all reservoirs
- Power balances, for all price areas/nodes

Several other types of constraints may be added to the weekly decision problem, e.g.:

- Power flow constraints
- Constraints on start-up costs on production units
- Ramping constraints

2 Scenario Fan Simulator

In this section the scenario fan simulator (SFS) methodology is described. Furthermore, SINTEF's experiences with this and similar methods are discussed, and relevant international literature is reviewed. Finally, we present some computational tests and estimates on the expected computational performance and convergence properties.

The scenario fan simulator (SFS) simulates sequences of decision problems, where each decision problem is formulated as a scenario fan. In this report we use the term scenario fan for a two-stage stochastic linear programming (SLP) problem. A scenario fan can be seen as a special case of a scenario tree, having only two decision stages. We want to emphasize the difference between a decision stage and a time stage in that the SLP's second decision stage will comprise many time-stages.

We start by defining the scenario fan problem (SFP) and the decomposition technique appropriate for solving this problem, and continue by describing the overall simulation logic.

2.1 The scenario fan problem

In this report, a scenario fan is referred to as a SLP problem with two decision stages, and where the second decision stage comprises multiple time stages. The first stage refers to a given week with a given realization of stochastic variables (the weekly decision problem). In the second stage, covering the remaining planning period, the stochastic variables can take values according to S predefined scenarios.

The extensive form SLP problem can be formulated as:

$$\mathrm{Z} = \min c_1^T x_1 + \sum_{s=1}^{S} p_s c_{2,s}^T x_{2,s} \qquad \textbf{(4)}$$

$$A_1 x_1 = b_1 \qquad \textbf{(5)}$$

$$T x_1 + A_2 x_{2,s} = b_{2,s} \; \forall s \in S \qquad \textbf{(6)}$$

Where the first term in the objective function Z is the cost associated with the first-stage decisions (x_1) and the second term refers to the cost associated with the S different second-stage decisions ($x_{2,s}$), where S is the number of scenarios and p_s is the probability of occurrence for each scenario. Note that we for simplicity omit the summation over all time stages in the second stage decision, so that $x_{2,s}$ and $c_{2,s}$ can be interpreted as vectors covering multiple time stages.

The shape of the SFP is illustrated in Figure 1, where the filled circles are decision points and branches are transitions. The first-stage decision is scenario-invariant and is taken at time t_1, and the second-stage decisions are related to one of the five scenarios covering time stage 2-N.

By solving the two-stage SLP we seek a robust first-stage decision that can be implemented in the simulator. The second-stage decisions are not used in the SFS. Equation (5) holds the set of first-stage constraints, and

equation (6) the second-stage constraints for each scenario. In our case, the first-stage state variables (reservoir levels in x_1) will enter the second-stage constraints by letting the technology matrix (T) contain ones associated with state variables.

If using price as a stochastic variable, it will enter as objective function coefficients $c_{2,s}$. The second-stage constraints are scenario dependent in that the stochastic variables (inflow, demand) enter the right-hand sides ($b_{2,s}$).

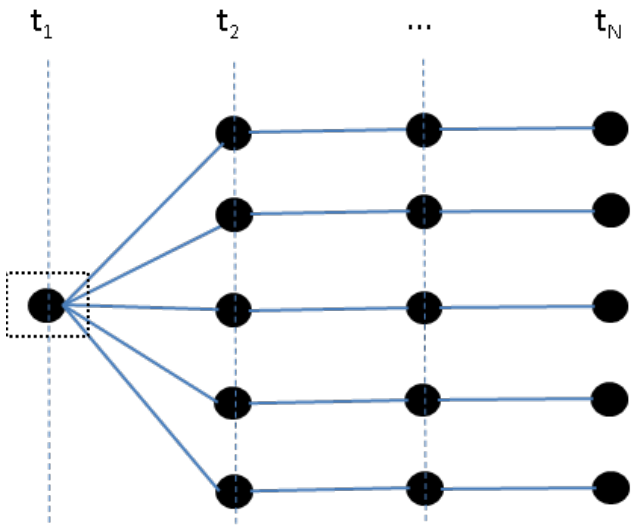


Figure 1: Illustration of SFP. Only the first-stage decision is used and stored.

2.2 Problem structure

The structure of the A-matrix for a weekly decision problem is illustrated in Table 1 below. Each row corresponds to a set of constraints, either reservoir balances or power balances. Each column corresponds to the set of variables for a given load period. There are 4 load periods within the week. As an example, consider row no. 3 reading "2 Reservoir". This is the set of reservoir balances for load period 2. It is indicated (by an X) that this set of constraints involves variables for load period 1 (reservoir filling in previous load period defines the starting condition for current load period) and for load period 2. We have assumed that power balances have no time couplings. Introducing additional constraints, such as start-stop for production units and ramping on e.g. plant discharge and flow on HVDC cables, these balances would also have time couplings. The key point here is the diagonal structure and the sparsity of the A matrix. The degree of diagonal structure and sparsity will impact the performance of different LP solution algorithms, which is further discussed in section 2.7.

Table 1 Matrix structure, weekly decision problem.

Load periods	1	2	3	4
1 Reservoir	X			
1 Power	X			
2 Reservoir	X	X		
2 Power		X		
3 Reservoir		X	X	
3 Power			X	
4 Reservoir			X	X
4 Power				X

The structure of the A-matrix for a scenario fan is illustrated in Table 2 below. We assume no uncertainty in the first stage and that the random variables may take values according to 3 different scenarios in decision-stage 2. There is only one time stage in each decision stage in this example. If formulating this problem as one (extensive form) SLP problem, the diagonal structure of the A matrix is no longer present. The A matrix now has a block-angular form, as illustrated in Table 2. All second stage realizations will have ties to the first-stage decision, and the structure looks more like an "L". When solving large-scale instances of such problems, significant computational savings may be gained by using a suitable decomposition method. The basic principle of decomposition is to transform the problem structure in Table 2 in to smaller problems with the structure in Table 1, and solve these in an iterative scheme.

Table 2 Matrix structure, scenario fan.

\Decision stages	**1**	**2-1**	**2-2**	**2-3**
1 Reservoir	X			
1 Power	X			
2-1 Reservoir	X	X		
2-1 Power		X		
2-2 Reservoir	X		X	
2-2 Power			X	
2-3 Reservoir	X			X
2-3 Power				X

2.3 Decomposition technique

One may obtain significant computational speed-up when decomposing the two-stage SLP rather than solving its extensive form in Equations (4)-(6). This problem may be decomposed by stage or by scenario; in this report we focus on the stage-wise decomposition. We create a master problem to represent the first-stage decision:

$$Z_{\text{master}} = \min c_1^T x_1 + \alpha \qquad \textbf{(7)}$$

$$A_1 x_1 = b_{11} \qquad \textbf{(8)}$$

$$\alpha + \pi^T x_1 \geq b_{12} \qquad \textbf{(9)}$$

After solving the master problem, the state variable solution (reservoir at the end of the first week) is passed to the sub-problem. A sub-problem represents the decision problem along one of the second-stage scenarios. The first-stage decisions variables (reservoir levels) are now passed as parameters to the left hand-side of the second-stage constraints as a trial solution:

$$Z_{\text{sub}}^{s} = \min c_{2,s}^{T} x_{2,s} \qquad \textbf{(10)}$$

$$A_2 x_2 = b_2 - x_1 \leftarrow \pi_s \qquad \textbf{(11)}$$

From the solution of a single sub-problem we obtain simplex multipliers (π_s) on the reservoir balances for the first load period in the second-stage. When all S second-stage sub-problems have been solved, we find the average multipliers (π) and right-hand side (b_{12}) to be used when constructing a new linear constraint (cut) for the master problem:

$$\pi = \sum_{s=1}^{S} p_s \pi_s \qquad \textbf{(12)}$$

$$b_{12} = \sum_{s=1}^{S} p_s (Z_{\text{sub}}^{s} + \pi_s^T x_1) \qquad \textbf{(13)}$$

The objective function value of the master problem will form a lower boundary. Cuts constraining the future-cost function will gradually increase the lower boundary.

$$Z_{\text{low}} = Z_{\text{master}} \qquad \textbf{(14)}$$

The upper boundary will be:

$$Z_{\text{up}} = c_1^T x_1 + \sum_{s=1}^{S} p_s c_{2,s}^T x_{2,s} \qquad \textbf{(15)}$$

The upper boundary is not necessarily strictly decreasing. We enforce a decreasing upper boundary by letting

$$Z_{\text{up}}^{i} = \min\left(Z_{\text{up}}^{i-1}, Z_{\text{up}}^{i}\right) \qquad \textbf{(16)}$$

Convergence is defined when the difference between the lower and upper boundaries is within a predefined tolerance, i.e., when:

$$Z_{\text{up}} - Z_{\text{low}} \leq \varepsilon \qquad \textbf{(17)}$$

This decomposition algorithm is often referred to as the L-shaped method, or more generally as Benders decomposition. There is another variant of this algorithm, known as the "multicut version". Rather than

building one average cut as outlined above, the "multicut" version builds one cut per scenario evaluation. This version may improve the convergence characteristics, but comes at the cost of a heavier master problem (one future-cost variable per scenario, and S cuts per iteration). A detailed discussion on solution algorithms is found, e.g., in [2].

2.4 Scenario fan simulator logic

The SFS repeatedly solves sequences of SFPs as described in pseudo code below:

```
for all scenarios s from 1 to S do
   for all stochastic time-steps t from 1 to T do
      Build and solve the SFP problem SFP(s,t)
      Store results from first-week decision, sol(s,t)
      Pass on state decision from sol(s,t) to SFP(s,t+1)
```

The procedure is illustrated in Figure 2, where the SFP is built for a given scenario s_1 and for time-steps t_1 and t_2. The first SFP is built with stochastic variables according to scenario s_1 in the first time step t_1. In the second decision stage (comprising time steps $t_2 - t_T$), stochastic variables may take values from any of the S scenarios with equal probability[1]. The solution $sol(s_1,t_1)$ is recorded, and the values of the state variables in $sol(s_1,t_1)$ are passed on as a starting point to the next time-step t_2, as illustrated in Figure 2. Subsequently, a new SFP is built with stochastic variables according to scenario s_1 in the first time step t_2. In the second decision stage (comprising time steps $t_3 - t_{T+1}$), stochastic variables may take values from any of the S scenarios with equal probability. This sequence is continued until a first-stage solution has been found for all time stages in the time horizon (t_1,t_N) for the particular scenario (s_1). The same procedure is carried out for scenarios s_2-s_S.

[1] To account for auto-correlations in stochastic variables, the transition between stages should be 'smoothed', as discussed in section 2.6.

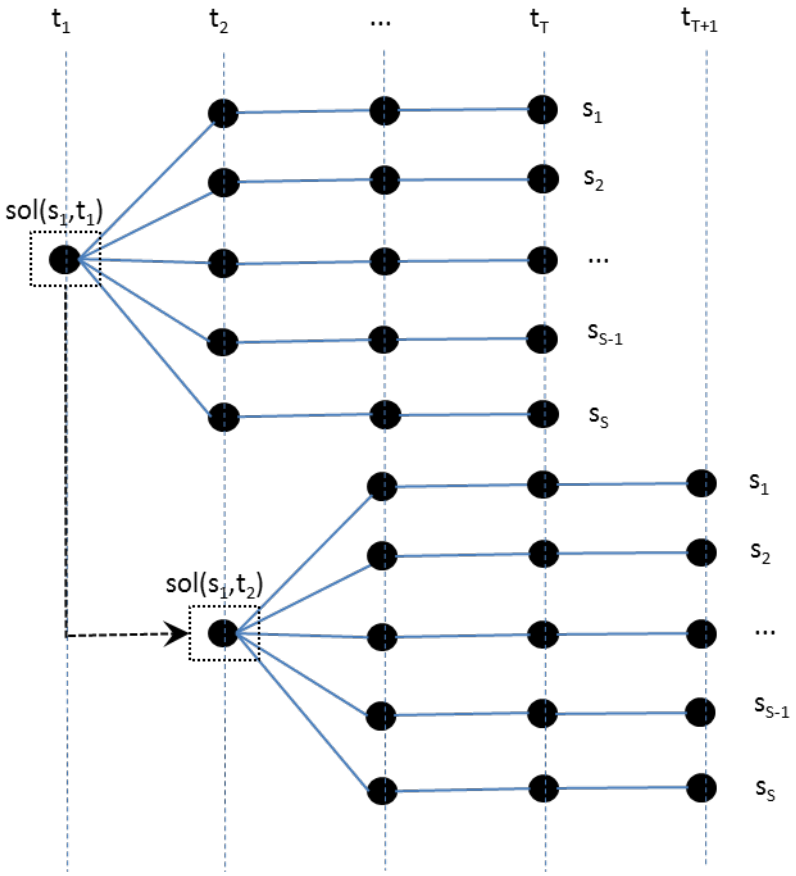


Figure 2: Illustration of SFS logic.

In principle, the second-stage scenarios could cover a planning period long enough to eliminate the impact of the end-value setting, but in practice one would need an explicit end-value setting. The rolling horizon illustrated in Figure 2, calls for a specific end value setting for each simulated stage. These values can e.g. be obtained from the EMPS model.

2.5 Relevant experiences

2.5.1 Models and experiences at SINTEF

The general SFS idea is based upon is rooted in previous activities at SINTEF. A similar scheme was briefly described in general terms in [3], but did not lead to a tested and documented prototype. Furthermore, the operative mid-term scheduling model termed 'sesongmodellen' works with scenarios in a similar way as in the SFS. In 'sesongmodellen', the aim is to obtain individual water values to be used either directly in the preparation of bids or as input to short-term scheduling models. This model is typically run on a specific watercourse, where power prices and inflows are defined as stochastic variables through scenarios. All scenarios are solved for pre-defined set of reservoir levels providing aggregate linear constraints (or cuts) to describe the future cost function for a given week [4]. That is, for each of the pre-defined reservoir levels all scenarios are solved to provide one aggregate constraint, as described in section 2.3. Thus, one can argue that

the SFS resembles the use of 'sesongmodellen' in a simulator with a rolling horizon. Similar to the SFP, all uncertainty is revealed in time step t+1, giving deterministic scenarios for later weeks. However, in contrast with the SFP decomposition described in section 2.3, there is no iteration between the deterministic first-stage (short-term problem) and the scenarios.

The SFS methodology was implemented in a prototype in the research project "Transmission planning in a changing environment" (2008-2012, led by SINTEF). This prototype was tested on a dataset of the Nordic power system, but with aggregate description of the hydropower, as documented in [5]. A brief summary of this test is given below. The test was carried out on a 13-area system containing realistic data for the Nordic system. A scheduling horizon of 156 weeks was chosen. The first year formulated in each SLP problem had a time resolution of one week with four load periods within each week, whereas the last two years had a resolution of four weeks with one load period. A total of 50 historical inflow records were used in the simulation, and all of these were assigned an equal probability of occurrence. The scenarios were simulated in parallel, using an initial reservoir filling of 60% for all reservoirs. The results from the SFS model was compared with those obtained by Samplan, an SDDP model developed at SINTEF. Both models used a predefined future cost function for evaluation of the final time stage in the scheduling horizon. This function was generated by the EMPS model. The case study results verified that the SFS method is competitive with Samplan for market simulation purposes. In terms of socioeconomic surplus, the SFS model provided better results than the SDDP model. Furthermore, the SFS handled extreme inflow scenarios slightly better than Samplan, as both the amount of spillage and the frequency of load curtailment turned out to be lower. It was observed that the average power prices were subject to less seasonal variations in the SFS model.

2.5.2 Literature review

In reviewing international literature we found a few approaches that have similarities to the SFS approach. All of these use terminology from field of control theory. Thus, this section is started by introducing some relevant terms.

We start by informally relating the principles of open loop and closed loop feedback control to the scheduling problem being discussed in this report. In *open loop feedback control*[2], only the current state (reservoir, inflow in previous time stage, etc.) will impact the solution of the weekly decision problem. At a given stage one computes the optimal decision for the entire time horizon by taking a deterministic view on future uncertainty. The decision related to the current stage is then implemented, and one proceeds to the next stage. In this scheme, one does not use the knowledge of the fact that the decisions in future stages are going to be re-computed. Conversely, in the *closed loop feedback control* the current stage decision problem is informed by the decisions made in future stages. The closed loop feedback control is also referred to as optimal feedback control, and this principle is used in dynamic programming. Thus, both stochastic dynamic programming (SDP) and stochastic dual dynamic programming (SDDP) are based on the closed loop feedback control principle.

For certain classes of stochastic control problems (linear quadratic Gaussian problems), the optimal strategy can be obtained by replacing stochastic outcomes by their expected values. This is often referred to as the *certainty equivalence principle* [6]. For such problems, the open loop feedback control scheme will provide identical results as the closed loop scheme. However, the general hydrothermal scheduling problem does not easily fit in to this category, e.g., due to constraints on reservoir, cf. [7] chapter 4.2 and [8, 9].

[2] Also referred to as open-loop-optimal feedback control [7] or deterministic feedback control [9].

In between the open loop and closed loop schemes, there are several variants normally being classified as approximate dynamic programming. One variant is the *partial open-loop feedback control* (POLFC) [6], which was applied to the scheduling problem by Martinez and Soares in [10]. In this scheme, for a given state and stage the average future inflows are forecasted and a deterministic nonlinear optimization model is solved. The first-stage decision is implemented before moving to the next stage. Feedback is provided through updated inflow forecasts and state descriptions for each simulated case. The article addresses the important task of finding the best suited forecasting horizon; a longer horizon will give less accurate forecasts, and a shorter horizon will give more weight to the terminal condition. The POLFC scheme was compared with SDP in a case study for a single reservoir system, using both historical and synthetic inflow series. The case study results and conclusions put emphasis on the superiority of the POLFC scheme over the SDP method when dealing with extremely dry periods occurring in the historical inflow series.

A similar approach, based on model predictive control (MPC) was presented by Zambelli et al. in [11]. It differs from the POLFC scheme in that it uses a rolling time horizon, ensuring that the inflow forecasts cover between 13 to 24 months. The rolling horizon makes the scheme more akin to MPC than POLFC, according to definitions in [6]. In a case study MPC is compared with SDDP on the Brazilian interconnected system. The case study shows that the system is operated at a lower cost, produces more power and maintains reservoirs at higher levels when using the MPC approach. The authors underline that the MPC method, which is based on a nonlinear optimization, is better suited to capture the influence of water head on hydropower efficiency than the SDDP method. The results presented in this article extend the findings in [10] to the multi-reservoir case. Both [10] and [11] uses averages of historical inflow records as forecasting values. Nolde et al. [12] suggest a stochastic version of the MPC approach. Decisions are simulated according to a known first-stage and an uncertain future is represented in a scenario tree.

A scheme similar to the SFS is briefly discussed by Labadie [8], describing the sequential solution of a two-stage stochastic linear programming problem where uncertainty is represented through predefined inflow scenarios.

In summary, methods similar to the SFS have been proposed in the literature, primarily by using expected values in the second-stage. These seem to outperform the performance of SDP [10] and SDDP [11] in the presented case studies. Conceptually, the stochastic MPC method in [12] seems to be comparable to the SFS method. The SFS method presented here and in [5] is unique in the representation of uncertainty (scenario fan) and in the suitability for large-scale application.

2.6 Treatment of stochastic variables

As discussed previously, the SFS uses a predefined set of scenarios to define the uncertain future. The standard approach will be to use historical records directly, with adjusted transitions between the first- and second-stage, as described below. We focus this discussion on treatment of inflow, but foresee similar treatment of other stochastic variables, e.g., price and temperature-dependent demand.

Inflows are correlated both in space and time, and both types of correlations are explicitly represented in the scenarios (historical records). There is a need to 'smooth' the transition of stochastic variables between the first-stage and second-stage in the SFP. Recall that the first stage refers to a given week t having stochastic variables (inflow, price, temperature) according to a given historical scenario m. If going directly from these values in week t to a week j in the second stage with historical values from a different scenario k, one may propose unrealistic SFP scenarios by not taking the correlation in time into account.

In [5] it was suggested to smooth the transition (for inflow) according to

$$\hat{v}_{j,k} = v_{j,k}\left[1 + \frac{v_{t,m} - v_{t,k}}{\tilde{v}_j}\frac{\sigma_j}{\sigma_t}a^{j-t}\right] \qquad \textbf{(18)}$$

where:

$\hat{v}_{j,k}$ corrected inflow in week *j* scenario *k*;
$v_{j,k}$ inflow in week *j* scenario *k*;
$v_{t,m}$ inflow in first-week *t* and first-week scenario *m*;
$\tilde{v}_j$ average inflow in week *j*;
a^{j-t} correlation coefficient to the power of *j−t*;
σ_t inflow standard deviation for week *t*.

Note that the second term on the right-hand side in the equation will approach zero as the exponent *j-t* increases, so that the corrected inflow $\hat{v}_{j,k}$ will eventually take the original scenario value $v_{j,k}$.

An important point here is that the second-stage scenarios could be adapted to the first-stage state information as described in equation (18), but should not incorporate foresight.

An example is shown in Figure 3 to illustrate the smoothed transition of inflow. The figure shows 3 scenarios (s1-s3 shown as black, solid-line) covering 10 time stages. Our first-stage decision is to be taken in time stage 1 provided the inflow of scenario s2. The smoothed transitions to be used in the SFP are illustrated by dotted lines. We see that second-stage scenario s'1 underestimates the transition to the original scenario s1, while second-stage scenario s'3 overestimates its transition to the original scenario s3. This happens because the term $v_{t,m} - v_{t,k}$ in equation (18) is negative in case of scenario 1 and positive for scenario 3. There is of course no smoothed transition for second-stage scenario s'2, since the first-stage is based on s2.

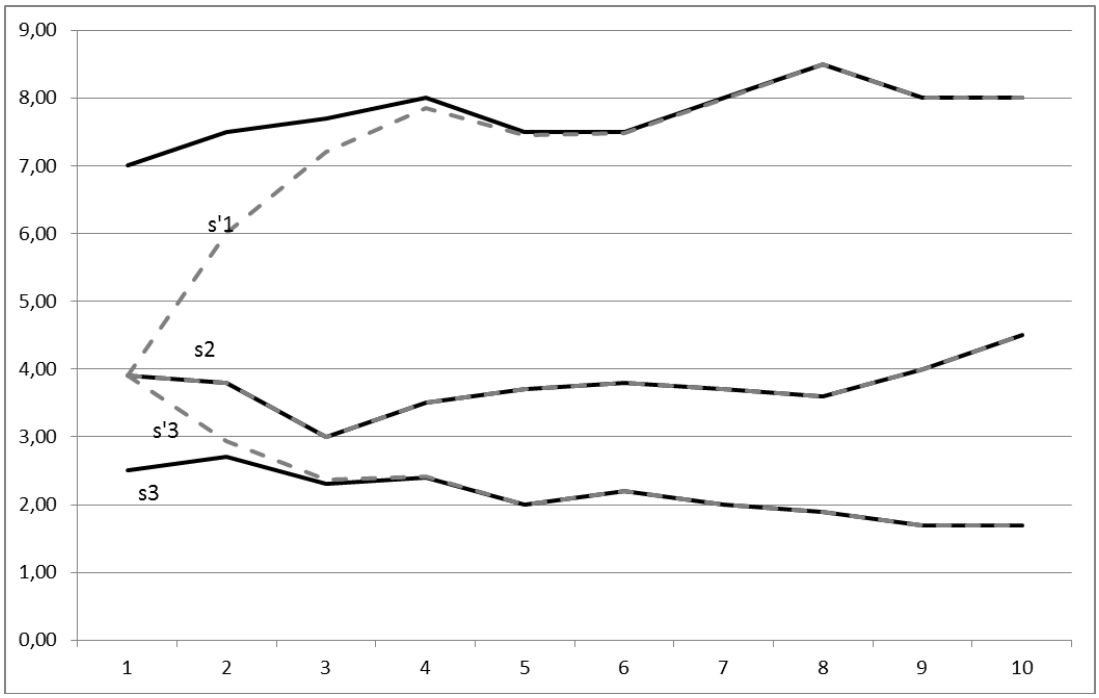


Figure 3: Transition of stochastic variables in SFP. 3 inflow scenarios are shown (black, solid-line). A scenario fan is built (grey, stapled line) with week 1 as the first-stage and weeks 2-10 as second stage, according to equation (18).

At this point it is worth discussing two special cases that can provide additional information regarding the value of information:

1) One could use the mean value of stochastic variables in the second stage. This option would provide optimal results if the certainty equivalence principle discussed in 2.5.2 holds true. When compared with the SFS approach, it will give an indication of the added value of stochastic information.
2) One could give the model perfect insight in the future by solving each scenario once with no uncertainty. This would give strategies that are tuned according to the scenarios that are going to be used, but not robust to future uncertainty. The solution would provide an upper boundary on model results, such as socio-economic surplus.

2.7 Testing

The SFS is clearly different from the SDDP method in that it solves less, but significantly larger optimization problems. Although the extensive form SFP can be decomposed, the second-stage scenarios easily fit in the category "large-scale linear programs". Through the testing in [5], we gained some experience in solving such problems. However, when going from aggregate area to detailed hydropower description, the problem size is considerably larger and the number of state variables roughly increases with a factor of 100. For these reasons, some basic tests were performed to obtain knowledge about solution times and convergence characteristics using state-of-the-art LP solvers and algorithms. In this work we used computer code from the research prototype ReOpt to build scenario problems with detailed hydropower representation. The ReOpt code was designed for building the weekly decision problem, as discussed in section 1.3, with detailed representation of hydropower modules. Weekly decision problems for a sequence of time-stages were linked together to scenarios by introducing reservoir couplings in time.

We start by looking at the characteristics and solution times for a single scenario in section 2.7.1 and 2.7.2. Then we proceed by solving the SFP, focusing on convergence characteristics in section 2.7.3.

2.7.1 Single scenario characteristics

The majority of computation time in the SFS will be spent solving the deterministic second-stage scenarios. Recall that the SFP can be split in a first-stage (or master) problem and several second-stage problems, glued together in a decomposition scheme. In this section we focus on the characteristics problems of the second-stage scenarios. Single scenarios were built for one year and then printed to file using the LP format, to be solved with stand-alone LP-solvers. The structure is illustrated in Figure 4. The use of stand-alone solvers was chosen to avoid programming related to the interface between model builder and solver[3].

Figure 4: Illustration of single scenarios, covering 52 weeks.

[3] Note that solution times may differ when comparing stand-alone solvers to library calls.

We used the following stand-alone solvers (all 64-bit versions):

- CPLEX – version 12.2 (latest version is 12.5)
- Gurobi – version 5.5.0 (latest version is 5.5.0)
- Clp – version 1.06.00 @ Windows (latest version is 1.15)
- Clp – version 1.11.00 @ Linux (latest version is 1.15)

In this section we report problem sizes and solution times for 2 different problems. Both problems are set up for the full Nordic system obtained from a current data set from Statnett (the Norwegian TSO), comprising 1235 hydropower modules. An LP problem comprises power balances for each time step and each area, and reservoir balances for each reservoir and each time step. The characteristics of the LP problems are outlined below. Note that 'aggregate' refers to aggregate load periods, and 'sequential' refers to sequential load periods. Note also the linear relationship between no. of time steps and no. of variables/constraints.

Aggregate – Full Nordic system (52 weeks, 5 load period: 260 time steps)

Variables:	$2{,}233 \times 10^6$
Constraints:	$0{,}331 \times 10^6$
Elements in matrix	$4{,}096 \times 10^6$

Sequential – Full Nordic system (52 weeks, 30 load periods: 1560 time steps)

Variables:	$13{,}399 \times 10^6$
Constraints:	$1{,}984 \times 10^6$
Elements in matrix:	$24{,}891 \times 10^6$

Tests were performed on 4 different machines:

- Windows 7 PC with Intel Core i7 CPU (4 cores at 1.73 GHz, 2 threads per core), 16 GB RAM
- SEFASPARA Windows server, 8 quad-core AMD processors at 2.4 GHz, 128 GB RAM
- SEFASHP Unix server, 8 Intel Itanium 2 cores at 1.6 GHz, 8 GB RAM
- Linux cluster with Intel Xeon CPUs, each node has 8 cores (dual-die quad core) with 2.93 GHz and 32 GB RAM dedicated.

Both CPLEX and Gurobi stand-alone solvers provide a tuning function testing all available algorithms with different settings. Based on some preliminary test, the most promising solution algorithm for was the dual simplex algorithm. We also tested barrier solver, which performs well on certain large-scale problems. The barrier solver runs in parallel for both CPLEX and Gurobi; we allowed it to use 8 threads.

Table 3 shows solution times for the first (aggregate) problem. Each test was run 3 times and the average solution time is reported. The solution times obtained from the PC were subject to less variation than for the two servers. This is most likely due to variable server load use during the period of testing. Independent of solver type, there is a significant increase in computation time when moving the problem from PC to servers. The 6th column compares solution times to the fastest solver/algorithm.

The large differences most likely indicate one of the following:

- Lack of dedicated physical memory for the servers.
- Differences in memory "quality", e.g., memory bandwidth and frequency, cache memory layout.

Table 3 Solution times (in seconds) for the aggregate LP problem.

Solver/machine	PC	SEFASPARA	SEFASHP	Cluster	Ratio (PC)	Objective
CPLEX – dual simplex	294	680	1065	-	1.48	-17041108
CPLEX – barrier	530	757	-	-	2.66	-17041108
Gurobi – dual simplex	199	-	-	-	1.00	-17041108
Gurobi – barrier	477	-	-	-	2.39	-17041108
Coin-Clp – dual simplex	669	2093	3217	998	3.36	-17041108

Table 4 shows solution times for the second (sequential) problem. CPLEX and Gurobi tests were run 2 times and the average solution time is reported. Clp was run one time only. Both Clp and CPLEX ran into memory problems on SEFASHP. The 6th column compares solution times to the fastest solver/algorithm.

Table 4 Solution times (in seconds) for the sequential LP problem.

Solver/machine	PC	SEFASPARA	SEFASHP	Cluster	Ratio (PC)	Objective
CPLEX – dual simplex	17216	50767	-	-	2.13	-47447564
CPLEX – barrier	14238	-	-	-	1.77	-47447564
Gurobi – dual simplex	8062	-	-	-	1.00	-47447564
Gurobi – barrier	11158	-	-	-	1.38	-47447564
Coin-Clp – dual simplex	39251	133732	-	53885	4.87	-47447564

2.7.2 Solution time as a function of problem size

It is clear from the above tests that the solution time of the LP problem increases more than linearly with the problem size. We tested this relationship using the dual simplex algorithm of the CPLEX solver, and the test results shown in Figure 5 were carried out on the PC (with specifications defined earlier). The algorithm is cold started. Recall that the solution time for the 52 week problem is 294 seconds (see Table 3).

It is interesting to see the dramatic shift in gradient when going from 8 to 10 coupled weeks. The rapid growth in computation time indicates that the second-stage scenarios themselves may be decomposed within the SFP. If scenarios are decomposed, a careful selection of decomposition point(s) should be made to balance the decrease in computation for small decomposed problems and increase in solution time due to communication between decomposed parts.

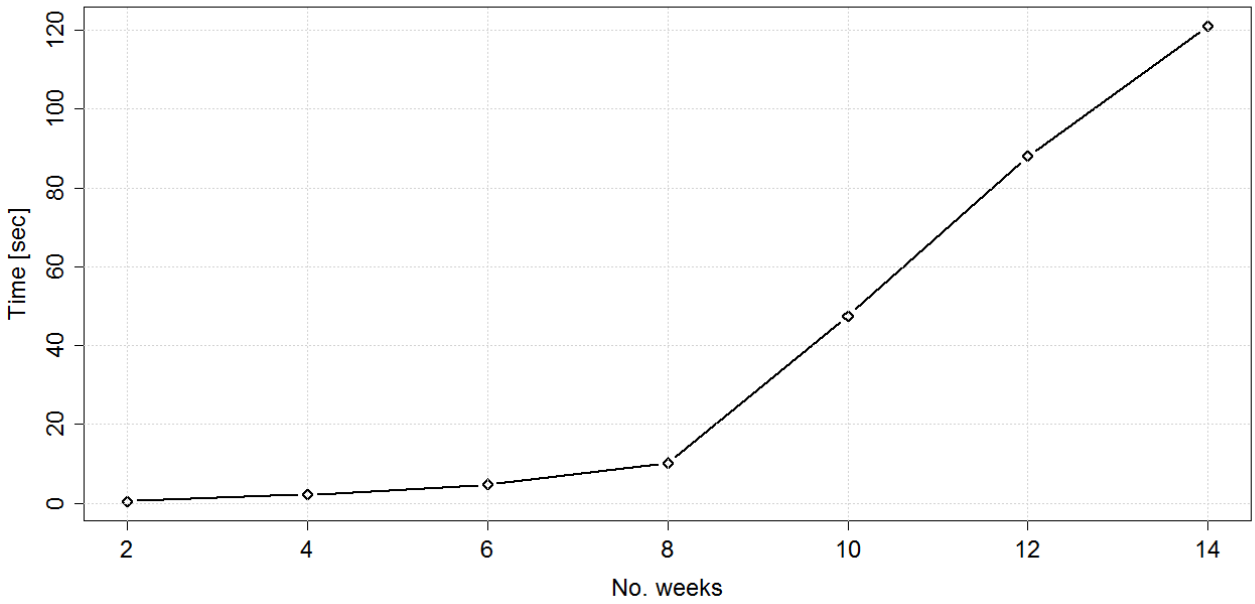


Figure 5: LP solution time as a function of problem size (number of weeks coupled).

2.7.3 Scenario fan problem (SFP) characteristics

Next we build the full SFP by coupling a common first-stage and several second-stage scenarios. The SFP is then solved by Benders decomposition, as outlined in section 2.3. The primary goal of testing Benders decomposition in this context is to learn more about the convergence characteristics when the number of reservoirs (state variables) is large and with reservoir sizes ranging from hourly/daily to yearly storages.

All implementation and testing has been done without parallel processing in mind. Thus, in order to reduce the test times for the decomposition, the basic tests were made on a scenario fan comprising 9 coupled weeks with aggregate time resolution. In the further presentation we assume that the convergence characteristics of the problem does not strongly depend on the number of coupled weeks considered. The SFP structure tested here is illustrated in Figure 6, where the stapled red line indicates the coupling between the first- and second-stage.

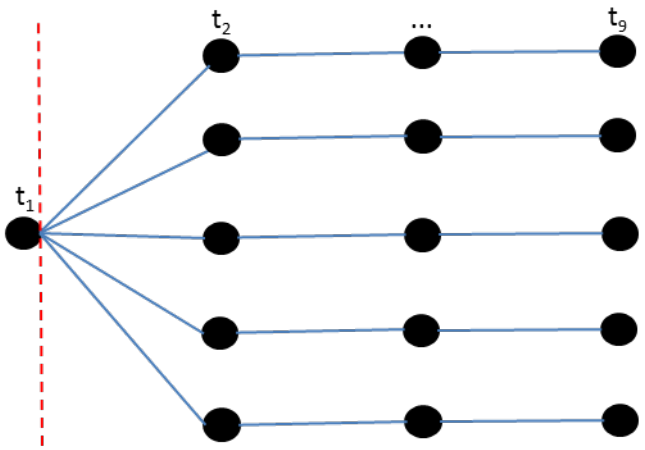


Figure 6: Illustration of SFP problem for 9 coupled weeks.

2.7.4 Convergence on cost

Figure 7 shows how the cost gap (difference between upper and lower bound) approaches zero as the number of iterations increases. Two different problems are solved here, with 10 (red curve) and 47 (blue curve) number of scenarios (inflow years) in the second stage. Both problems close the gap to a value of 3.0 around iteration number 30. The objective function value is $-8.5*10^5$, given a relative error of $3.5*10^{-6}$.

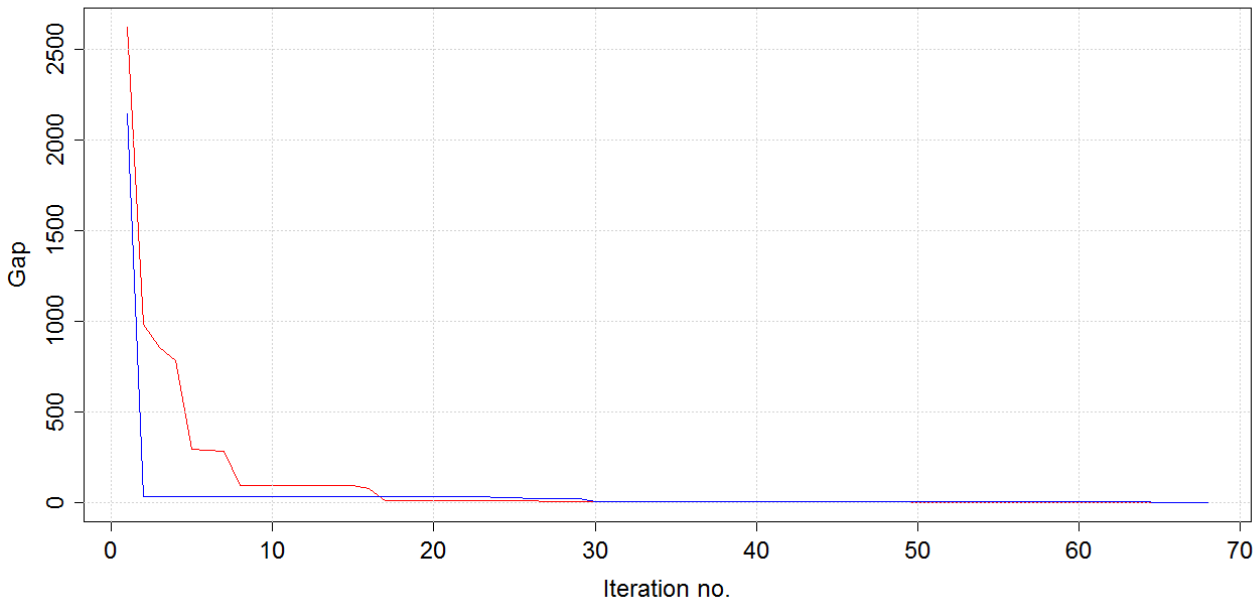


Figure 7: Cost gap (upper bound – lower bound) with number of iterations. Red curve refers to use of 10 inflow years, blue curve refers to 47 inflow years.

2.7.5 Convergence on reservoir operation

It is also of interest to study the convergence of the reservoir filling in each of the 1235 reservoirs considered. The reservoir capacities vary from negligible to yearly storages, as shown in Figure 8. Thus it is likely that some reservoirs will converge in operation faster than others. It is also clear that using a relative convergence criterion (percentage of capacity) can be misleading.

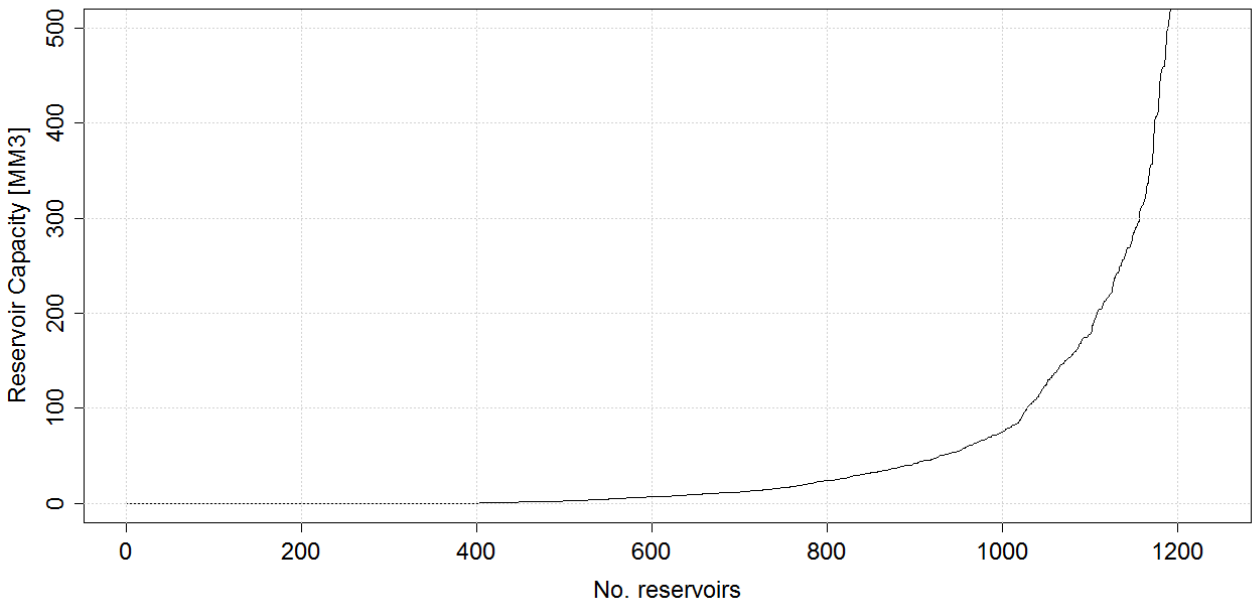


Figure 8: Reservoir capacities, sorted and in Mm^3, for all reservoirs in the system. (A few large ones exceed the scale on the y-axis.)

Figure 9 shows the number of reservoirs that see a content change larger than 10% from one iteration to the next.

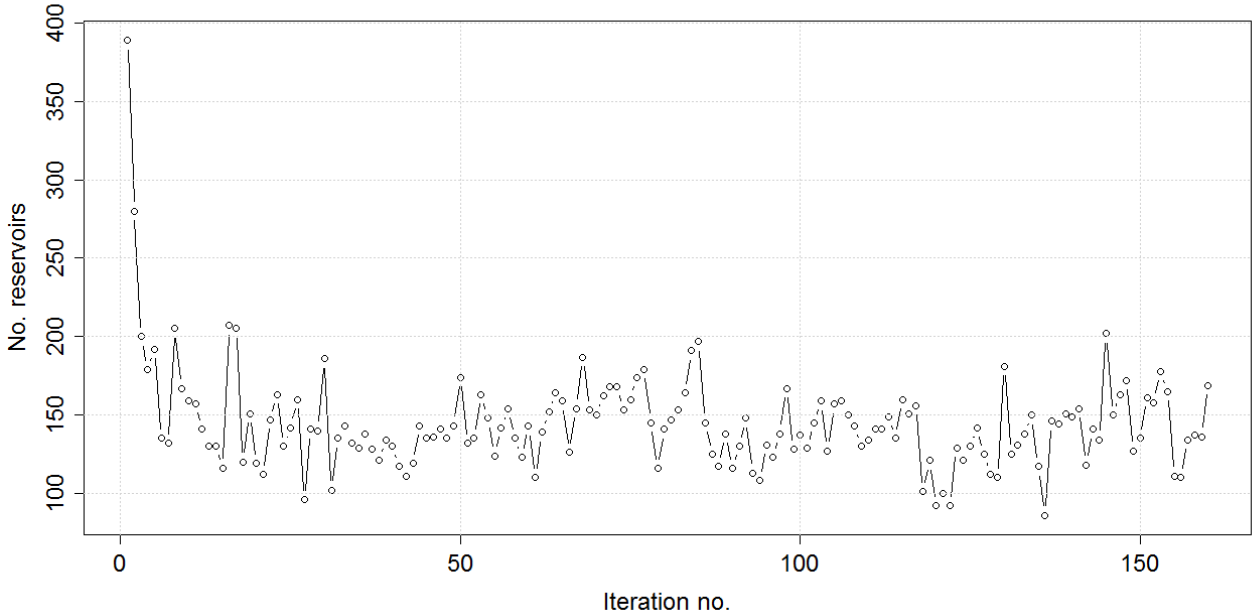


Figure 9: Number of reservoirs that sees more than 10% change from an iteration to the next (for the case with 10 inflow scenarios) .

We take a closer look at the type (size) of reservoirs change for some selected iterations in Figure 10.

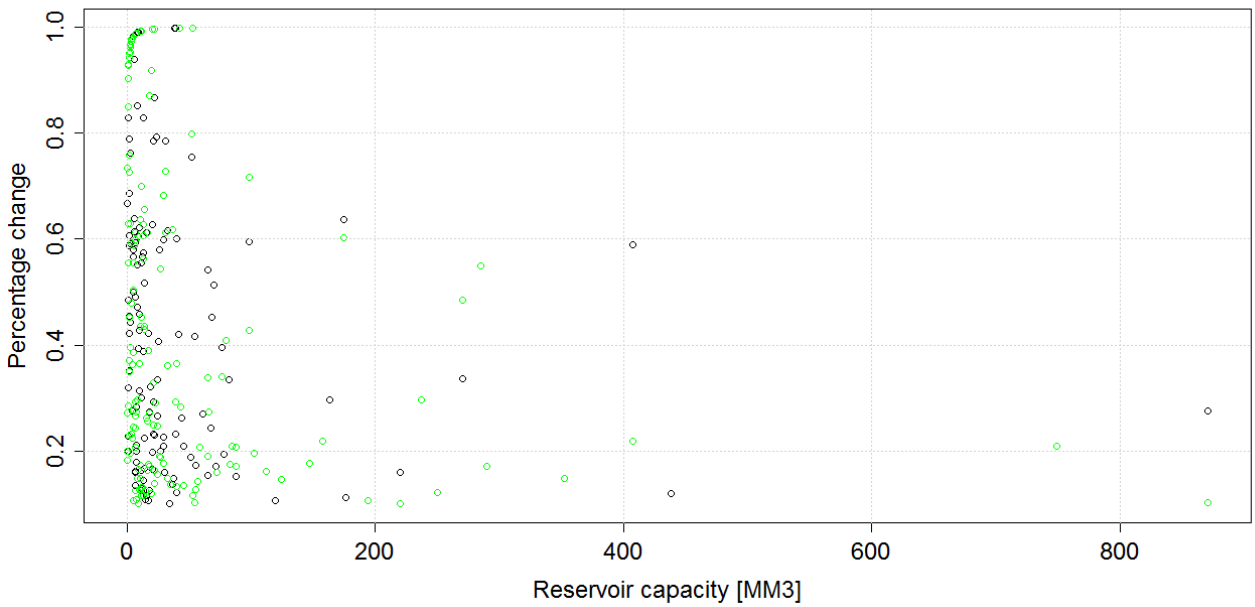


Figure 10: Reservoirs changing more than 10 % from an iteration to the next, as a function of capacity. The black circles are from iteration 5 and the green circles from iteration 50.

Looking at iteration 50 (green circles in Figure 10), we see that there are still some large reservoirs that have not converged in operation. The 5 largest of these reservoirs are listed in Table 5. Not surprisingly, some of these are closely hydrologically connected, and are cancelling out each other's changes. To illustrate this, the suggested reservoir fillings for 'Nore 1' and 'Pålsbu' are plotted as a function of iterations in Figure 12. An excerpt of the water course identifying these reservoirs is shown in Figure 11. A similar pattern was found when looking at 'Gardiken' and 'Abelvatnet'.

Table 5 The 5 largest reservoirs changing more than 10 % from iteration 49 to 50.

No.	Module no.	Name	Reservoir size [MM3]	Change from prev. iter. [%]
1	96244	Gardiken	871	10
2	07878	Songa	749	21
3	96241	Abelvatnet	407	22
4	07322	Nore 1	352	14
5	07323	Pålsbu	290	17

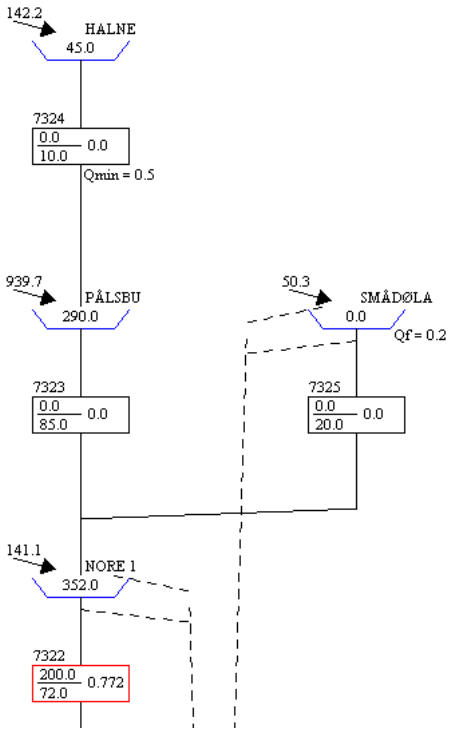


Figure 11: Excerpt of water course identifying 'Pålsbu' and 'Nore1'.

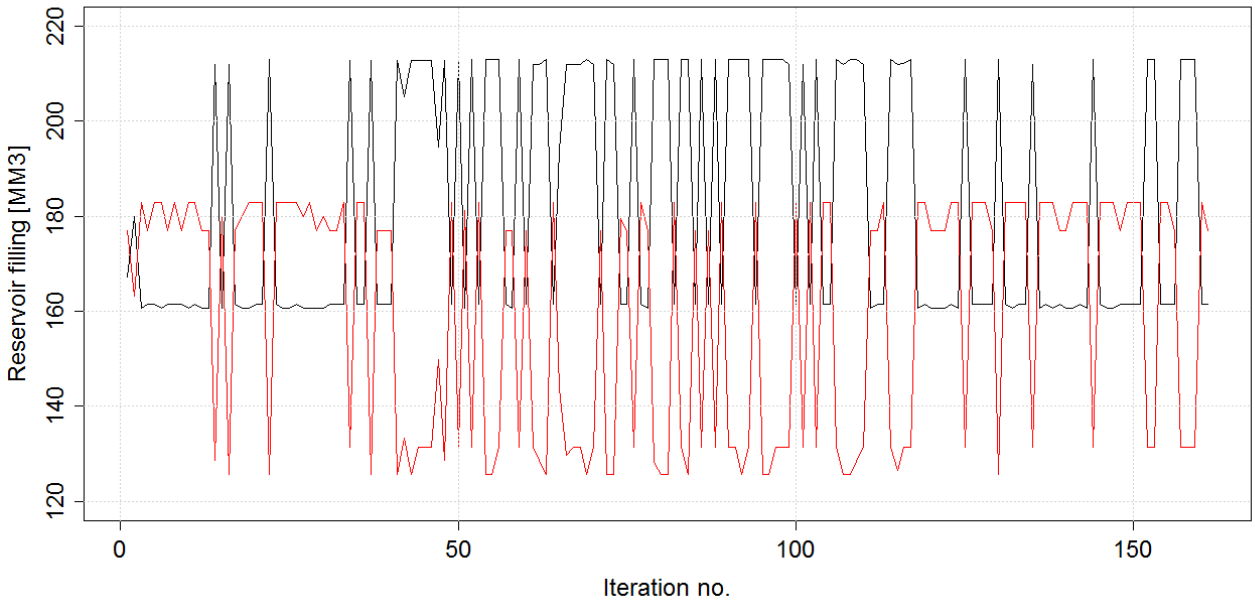


Figure 12: Reservoir content in first week (master problem) found in the different iterations for Pålsbu (red curve) and Nore1 (black curve).

Since the SFP sees no difference in economy discharging from one reservoir or the other in Figure 12, one should consider techniques for constraining or avoiding this type of behavior.

2.7.6 Second-stage problem solution times

The solution times for a single second-stage sub-problem depends on whether the problem is solved from scratch or with an updated right-hand side only. The decomposition scheme was carried out on a single processor in the tests reported here, and the master problem and sub-problem structures are built before starting the iterative decomposition algorithm. Once the master problem has passed a solution to the second-stage, we loop over all scenarios, update the right-hand sides of each scenario with the new first-stage solutions and the inflows corresponding to that scenario. Thus, only one scenario cold-start is needed; for the first scenario in the first iteration.

The cold-start solution time is approximately 10 seconds for the 8 week sub-problem considered in this section. This sub-problem covers weeks 2-9 in Figure 6. For a warm-started problem, the solution times are in the range of 2-5 seconds. It should be possible to reduce the variance in warm-start solution times by arranging similar scenarios in sequence, but this was not tested here.

If sufficient computational resources are available, it is better to allocate each scenario to a unique processor, and only update the master solution between iterations. The effect of minimal update is clearly demonstrated when running the 8 week sub-problem with one scenario only. The warm-started problems take solution times in the range of 0.2-0.6 seconds. The single scenario decomposition was tested for the full problem as well (51 week sub-problem). Cold-starting the problem with the dual simplex algorithm in CPLEX took

approximately 320 seconds (average of 3 runs), as is comparable[4] with the results in Table 3. Table 6 shows how sub-problem solution times generally decrease as the cost-gap is closing.

Table 6 Solution times for sub-problem.

Iteration no.	Solution time [sec]	Cost gap
1	320	Large
2	205	2305
3	80	48
4	44	48
5	27	48
6	25	10
7	32	10
8	45	7
9	23	0.7
10	23	0.7

2.8 Expected computational performance

The numerical tests reported in section 2.7 provide basic information and insight in computation times and convergence characteristics when solving deterministic scenarios and two-stage stochastic programs for the full Nordic system. This information can be used to provide rough estimates on the solution time for the SFS [5]. We present an example below, were expected computation time is roughly estimated. The numerous uncertain factors in this example and the underlying assumptions are discussed afterwards.

2.8.1 Example: estimating the computation time

The SFS has great potential for exploiting parallel processing. In the following we assume that there is enough computing capacity to run each SFS scenario independently (line 1 in pseudo-code in section 2.4), and that each second-stage scenario in the SFP can be allocated a processor and solved independently (in parallel with others). We also assume perfect scaling and neglect the solution time for the master problem. The assumptions are discussed later on in this section.

Based on these assumptions, we can set up a solution time estimate for each SFP:

$$T_{SFP} = T_C + (N_I - 1)T_W \approx T_C + N_I T_W \qquad \textbf{(19)}$$

Where N_I is the number of iterations used to solve the decomposed SFP, T_C the "cold start" solution time and T_W the "warm start" solution time. From the expression it is clear that the "warm start" solution time plays a crucial role for the overall solution time.

As an example, consider the deterministic aggregate scenario problem presented in section 2.7.1, with approximately 2 million variables and 300 000 constraints. The cold-start solution time for this problem

[4] Note that we do compare a library call with the use of stand-alone solver here.

using COIN Clp was 669 seconds on a PC and 998 seconds on the cluster, see Table 3. The warm-start solution time will vary between iterations, as shown in Table 6, but in this example we need the average value.

We assume the following problem characteristics:

- 50 inflow scenarios, giving
  - 50 SFS trajectories
  - 50 SFP second-stage scenarios
- Planning period of 156 weeks
- TC = 700 sec
- TW = 100 sec
- NI = 20

T_{SFP} = 700 sec + 20 iterations * 100 sec/iteration = 2700 sec = 45 min

The SFPs are solved in sequence for each week in the planning period. Thus, the first-stage solution in week two for a given inflow scenario will have to wait for a first-stage solution in week one for the same inflow scenario. The total computation time for one inflow scenario becomes:

T_{SFS} = 156 weeks*2700 sec/week= 421200 sec = 4.8 days

Since all inflow scenarios are run in parallel, 4.8 days will be the total computation time for the SFS in this example.

2.8.2 Assumptions and uncertainty in estimate

The example above is a rough estimate heavily depending on the assumptions and with uncertain input parameters. We start by discussing the assumptions being made:

1) <u>Access to "sufficient" computing capacity:</u>
Sufficient in this context means capacity to run each of the 50 inflow scenarios in parallel, and for each of these parallel executions to run the 50 scenarios in parallel. In other words, the example requires 50*50 = 2500 cores occupied for 4.8 days. In comparison, the Vilje supercomputer at NTNU has 22464 cores, the SINTEF-owned Linux cluster used for testing in this project has 768 cores. From this it is clear that the example has high requirements to the availability of computational resources, and that we should plan on running simulations with less computational resources available.

Assume that 100 cores were available for the example above. Most likely one would find that the optimal strategy is not to run each inflow scenario in parallel, but rather communicate cuts and warm starts between first-stage decisions at the same week, see [5] for further details.

2) <u>Perfect scaling:</u>

With perfect scaling we mean an implementation having an efficiency of 100%. The term 'scaling' has not been clearly defined; in the following we associate it with the definition of efficiency for a parallel program. We can define speedup as the ratio between the sequential run-time on one core and the parallel run-time on

p_c cores. Efficiency of the parallel implementation is defined as the ratio between the speedup achieved on p_c processors and p_c.

In general, perfect scaling is hard to achieve, as it is difficult to avoid overhead, e.g., due to differences in solution times. However, one could discuss the 'goodness' of the efficiency measure for this case. In the example, we do not allow the first-stage solutions along a scenario to be shared, and assume that one scenario requires 50 cores. Each of the 50 second-stage scenarios are stored in memory on a dedicated core, and the only change between two iterations within the same week will be an updated first-stage decision. If compared with a sequential solution, where each sub-problem has to be loaded into memory and then updated before solution, the parallel efficiency may even exceed 100%.

3) <u>Neglecting solution time for master:</u>
We believe that this assumption is fair and does not impact the example result much.

4) <u>Number of iterations:</u>
We have assumed that the decomposition algorithm converges in 20 iterations. As reported in sections 2.7.4 and 2.7.5, the SFP seems to converge in cost within 20 iterations[5], but not in reservoir operation. We should primarily focus on the cost convergence, since the lack of reservoir convergence can be explained by a 'flat optimum'. As shown in Table 6, the first iterations are expensive in terms of computation time, but this cost decreases as the first-stage decision stabilizes. Thus, the example result is not as sensitive to changes in the iteration counter N_I as equation (19) indicates.

5) <u>LP Solution times:</u>
As documented previously, the solution times significantly depends on the LP solver and on the hardware. Running the simulations with the Gurobi LP solver on fast processors with sufficient memory, the tests indicate that one could reduce the cold-start solution time with a factor of 3. However, commercial LP solvers are expensive to use, and one should therefore carefully evaluate these costs against the computational savings.

So far we have assessed the computation time for a fixed-size LP-problem. The SFP problem can be adjusted in several ways to add or remove details.

Increased LP dimension by:

- Introducing finer time-resolution
- Introducing additional constraints, such as:
    - Power flow constraints
    - Ramping constraints
    - Start-up costs on thermal units

Decreased LP dimension by:

- Aggregating time-steps far ahead, e.g., after year one in the second-stage problem
- Reducing or aggregating the second-stage scenarios
- Decomposing the second-stage scenario problems to take advantage of the solution-time characteristics shown in Figure 5

[5] Depending on the number of second-stage scenarios.

3 Stochastic Dual Dynamic Programming

In this section the stochastic dual dynamic programming (SDDP) method is described. First, the basic method is presented. Furthermore, we point out some challenges related to treatment of stochastic variables, and in particular inflow modelling. Finally, we present some computational tests and estimates on the expected computational performance and convergence properties.

3.1 Method

The SDDP algorithm was first introduced by Pereira in [13] and Pereira and Pinto [14]. Since then it has become the basis for several commercial scheduling tools, and has frequently been improved and analysed by researchers. There are numerous reports and scientific articles describing the various aspects and applications of the SDDP algorithm applied to hydrothermal scheduling problems, see e.g. [15, 16] and references therein. The SDDP algorithm, as implemented at SINTEF, is presented in brief in the following. Mathematical details are omitted here; these can be found in great detail in the literature.

The SDDP algorithm can be classified as a sampling-based Benders decomposition algorithm, where the overall problem is decomposed into weekly decision problems, as described in section 1.3. When solving a weekly decision problem, a future cost function represents the cost of using water in the current week. The future cost function is represented by linear constraints (or cuts). The algorithm is iterative and comprises two basic steps for each main iteration:

<u>Forward simulation:</u>
From an initial state defined by the initial reservoir levels, the system is simulated by solving a sequence of weekly decision problems along the scenarios of stochastic variables (e.g. historical inflows and price scenarios). These scenarios are indicated by s_1-s_3 in Figure 13. We assume that all stochastic variables for a given week are known at the beginning of that week. Cuts obtained so far are used to define a future cost function at the end of each week. The forward simulation generates an upper bound on total system cost, and reservoir trajectories for which cuts are calculated in the backward recursion.

<u>Backward recursion:</u>
Cuts at the end of the planning period can be obtained from the final value function (e.g. generated by the EOPS model). For each reservoir trajectory obtained in the forward simulation one starts from the state at the end of week *T–1*, and for each realisation of stochastic variables one computes the optimal operation for week *T*. This is illustrated for realisations ξ_1-ξ_3 in Figure 13. From the sensitivities of the objective function to the initial state values, new cuts at the end of week *T–1* are obtained. These cuts are constructed by averaging contributions over realisations of stochastic variables for a particular state. Thus, realisations ξ_1-ξ_3 in week *T* for scenario s_1 in Figure 13 will be used to construct one cut. Under certain assumptions we can assume that the cut constructed for s_1 in stage T-1 will be valid for scenarios s_2-s_3 as well. This is known as *cut sharing*, and is crucial for the computational performance of the SDDP algorithm. Subsequently, one carries out the same procedure for week T–1 and so on. The backward recursion provides an updated operating strategy, as the list of cuts corresponding to a time stage is extended. It also provides a lower bound for the expected cost.

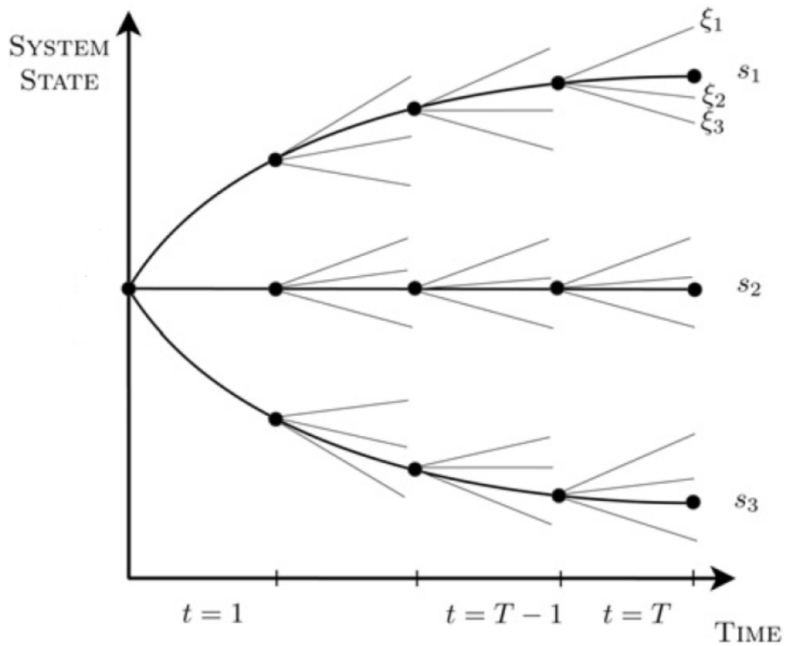


Figure 13: Illustration of a main iteration in the SDDP algorithm [16].

We assume that a set of initial trajectories is always available (e.g. from the EOPS model), so that the algorithm can start at the backward recursion step. Such a set of trajectories may for instance be obtained as percentage reservoir contents from aggregated models.

Some key points:

- Scenarios in the forward simulation: In the original version of SDDP, one re-samples scenarios in each forward simulation. Re-sampling is required to ensure perfect convergence of the algorithm. In practice however, we use historical inflow scenarios in the forward simulation. This is discussed further in section 3.4.3.

- Cut sharing: One of the key success factors of the SDDP method is the ability to share cuts among different states at a given stage, as mentioned in section 3.1. In principle, cut sharing allows collapsing of the scenario tree[6], and significantly speeds up computations. Cut sharing is admissible when the random variables are stage-wise independent or affine functions of the model error [17, 18]. That is, if the inflow model to be described in section 3.4 is perfectly valid for the normalised inflow, then the model error should be independent of the normalised inflows in the previous time stage, and cuts could be shared among states in the same stage.

- Convergence properties: Convergence is declared when the forward cost sample estimate and the lower bound from the backward recursion come close enough. This may be a slow process, or cost gaps exist, so in practice one often checks the change in reservoir trajectories from one iteration to the next, stopping after a fixed maximum number of main iterations.

[6] At least when scenarios are re-sampled in the forward simulation.

3.2 Literature review

There is a large amount of scientific literature on the SDDP method, see e.g. [15, 16] and references therein. The majority of this research emphasises on theoretical properties and model extensions; we will not discuss these here, but rather focus on reported large-scale case studies, potential for parallel processing and inflow modelling.

Large-scale case studies:
There are a few present studies on systems with many time stages and reservoirs. One of these is the study presented by Granville et al. in [19], where SDDP is applied to a Brazilian system model comprising 105 hydropower plants. To our knowledge, this is the system reported in the literature having highest number of reservoirs. In Tilmant and Kelman [20], SDDP is applied to the Turkish hydrothermal system comprising 40 hydropower plants.

Both case studies mentioned above applied monthly decision-stages. In terms of stages, it seems like the standard approach is to use either a monthly (Brazil) or a weekly (Norway) stochastic time resolution. In conclusion, we find no applications of the SDDP algorithm documented in the literature that comes close to the problem sizes that are going to be addressed in the SOVN project.

Parallel processing:
A recent article by Pinto et al. [21] demonstrates the potential for applying large-scale parallel processing to the SDDP algorithm. By using an optimized parallelization strategy, the authors demonstrate good scalability in their SDDP code for up to 128 cores. The optimized parallel strategy comprises efficient and asynchronous communication, load balancing and exploitation of hardware architecture. The documented scalability significantly depends on the number of scenarios used in the forward simulation. It is also worth noting, that the implementation in [21] has a synchronization point at each stage, although this is not needed from a mathematical point of view. Thus, there may be a potential good parallel performance using an even larger number of cores.

Inflow modelling:
Although SINTEF has worked with inflow modelling in the SDDP algorithm previously, this field clearly has potential for further exploration. In particular, more work can be done on grouping similar inflow series and prioritizing the ones being "most important". In a recent article by Penna et al. [22], a selective sampling technique was applied to an SDDP algorithm used in Brazil. Cluster analysis was used to group inflow series into homogenous groups to reduce the dimensionality of inflow data, clearly improving the convergence properties of the algorithm.

3.3 SDDP-based models at SINTEF

SINTEF has extensive experience in developing and applying the SDDP algorithm.

ProdRisk is a model for long- and medium-term scheduling in regional hydropower systems, based on a combination of SDDP and SDP described in [23]. The ProdRisk model is currently being used by several producers in the Nordic power market.

Samplan is a market model based on SDDP, and where the strategy is built for an aggregated representation of the hydro system. Samplan has been extensively tested against other market models, e.g., the EMPS and the SFS.

3.4 Stochastic inflow model

As discussed in section 3.1, the SDDP algorithm uses scenarios in the forward simulation, and a stochastic inflow model in the backward recursion. Ideally, to ensure good convergence properties, one should use scenarios obtained from the inflow model used in the backward recursion, and scenarios should be re-sampled in each major iteration [24]. Standard use of SDDP models developed at SINTEF involve direct use of historical inflow records. The major motivation for working with 'observed' scenarios (inflow and price) in the forward simulation is to conserve correlations not taken into account in the stochastic models.

In the remainder of this section we discuss the stochastic inflow model used in the backward recursion. Most of the material is a synthesis of previous work carried out at SINTEF [25, 26, 15].

SDDP is based on the principle of dynamic programming, and will therefore need a stochastic inflow model that represents the inflow distribution one week ahead, given some information about the current state of the system. This is the same type of inflow model needed in the traditional 'water value method' (SDP), but the SDDP method also require a convex problem formulation. In order to meet the convexity requirement, the stochastic model should be convex too. In practice so far, this requirement limits the stochastic inflow model to take the form of a linear model.

In short, a suitable stochastic inflow model should:

- Meet the convexity requirement
- Provide the correct expected value
- Provide a good estimate on higher order statistical moments

Below is a description of the linear inflow model used in ProdRisk and Samplan (see section 2.3 in [15]).

Inflow records tend to have strong sequential correlation; the inflow in a week *t* depends heavily on inflow in week *t-1*. Furthermore, the inflow has strong seasonal variations. One reason for this is the accumulation of snow during winter, with the following spring melt. Seasonal variations may be removed/reduced by normalizing the weekly inflow q_t, using the average inflow $\bar{q}_t$ and the standard deviation σ_t :

$$z_t = \frac{q_t - \bar{q}_t}{\sigma_t} \qquad \textbf{(20)}$$

We now assume that that the normalized inflow series comes from a weekly stationary multivariate stochastic process, and describe the process by a first-order auto-regressive model (AR1):

$$z_t = \varphi z_{t-1} + \varepsilon_t \qquad \textbf{(21)}$$

Where z_t now is a vector of different normalised inflow series, φ is the transition matrix and ε_t is the noise (model error), and these are estimated using a regression approach. A basic assumption is that the noise ε_t is

independent of z_{t-1}. In practice, it has been found that it is best to split the data into different seasons and to fit a separate model for each season.

From the fitting of (21) we have a multivariate sample distribution of the error term ε_t. For use in the SDDP model, we approximate this distribution by a discrete probability distribution with K discrete values ε_t^1- ε_t^K and corresponding probabilities, where K is not too large.

3.4.1 Principal components

One way of doing this is to carry out a model reduction by applying principal component analysis (PCA) to the sample ε_t for t=1, …,T . The principal components (PC) are transformed variables constructed so that they are independent, taken over the sample. Only the PCs that contribute most to the total variance are kept. After the PCA has been carried out, the distribution of each PC kept is approximated by a small number of discrete points. Finally the discrete points obtained are transformed back to the axes of the original normalized data and combined.

An inflow model based on PCA has become the standard in ProdRisk. When used in models with a large number of inflow series, it has been observed that the inflow model underestimates the variance. The most likely explanation to this is that the PCA does not cover the entire sample space.

3.4.2 Sampling

Another way is to obtain the ε_t^k by sampling from the collection of sample errors directly, as in a Monte Carlo simulation. Sampling seems to be the standard choice in SDDP applications used in Brazil, and has been tested at SINTEF Energy [26]. Under the assumption that the normalized inflow is a stationary process, one can sample in the group of all sample errors for the season in question. However, in SINTEFs implementation of the sampling, method sampling is done for innovation errors that are separate for each week.

3.4.3 Challenges with the stochastic inflow model

Underestimating variance:
The stochastic inflow model is a first-order model aiming at predicting the inflow one week ahead. Previous experiences has revealed that such models underestimates the probability of extreme dry and wet years compared to historical observations, see [27] section 7. This is also a problem in the EMPS and EOPS models, but with these models the user can to some extent calibrate to adapt to observations. For a stochastic optimization model to correctly predict inflows, all the statistical moments for all possible accumulated periods should be correct. The AR1 model outlined above will give correct expectation for all accumulated periods, whereas the standard deviation is only correct one week ahead. This problem depends on the stochastic time resolution; a coarser resolution (e.g., monthly) will reduce the impact of not treating higher order statistical moments.

Experiences with the sampling-based inflow model indicate that it provides a higher inflow variance than the model based on PCA [26]. However, the increased variance relates to the first-order model, not the ability to see higher-order moments.

Negative inflows:
Using the model in (21) may lead to negative inflows. This occurs more often with the sampling based inflow model. Besides being unphysical, negative inflows will also challenge the model in cases where reservoir levels are low/empty. As discussed in [26], one could in principle avoid negative inflows by working with the logarithm of the inflow. A previous study in [28] indicates that this variable transformation is favorable, but it can be shown that the transformation does not preserve convexity in the SDDP algorithm. This argumentation is supported by [18].

3.5 Stochastic exogenous price model

If exogenous price data are used, the price dimension can be treated in an SDP manner, as described in [23] and implemented in ProdRisk. Cut sharing is not possible between the discrete price points in each stage in the backward recursion; thus, a separate set of cuts is kept for each price point. We will not discuss the price model in detail here, but for the completeness of this report it is important to point out that including price as a stochastic variable as in [23] significantly increases computation time.

3.6 Testing

3.6.1 Background and motivation

As mentioned in section 3.4.3 our experience indicate that the inflow representation is the main concern regarding the applicability of the SDDP approach to the large scale problem addressed in this project.
Our previous experience from systems with many inflow series has been based on aggregated energy models and energy inflow series applied to market models of the Nordic market. We have not tested the model on a comparable physical system with natural inflow series before. This is the main purpose of the tests described in this section.

When the system is simulated for observed inflows, two types of problems may occur, both being possible sources of inconsistencies:

1) Sampling errors:
Model simulations of strategies calculated using stochastic optimization will contain sampling errors because the historical records consists of relatively few years of data.

2) Statistical properties:
Inflow statistics in time and space is very complicated and the models used in a stochastic optimization setting will not be equal to the statistics of the observed inflows used in the simulation. This is also the reason why observed inflows are used in the simulation part. However, using observed values in the simulations introduces the possibility of the strategy not being optimal, because it may be calculated for the wrong statistics.

The inflow model used in the SDDP model ProdRisk is given by equations (20) and (21). The *standard approach* is to use the inflow model with chosen parameterisation in the backward pass of the algorithm and the observations in simulation. As will be discussed in section, the standard approach does not provide satisfactory results. For this reason we have also tested the ProdRisk using artificial inflow series made from the statistical model that is used in the backward pass in the forward simulation. Thus, possible differences between observation and model statistics should be removed, and because we have made 500 different scenarios the sampling error is also reduced.

For the above-mentioned reasons, the testing related to the SDDP method primarily focuses on the inflow modelling.

3.6.2 Test set-up

The ProdRisk model has been tested on a dataset of the Norwegian system. The dataset was developed in the late 80's, but since the Norwegian hydro system has not been developed significantly since then it gives a good representation of the existing system. The dataset consists of 500 hydro reservoirs and uses 105 different inflow series. The market description consists of firm load obligations, some flexible load and production and import/export options to neighbouring countries. More specific information about demand and production is described in Table 7. We have used this dataset because it was made for the Vansimtap model and can therefore easily be adapted to ProdRisk. The hydro model has the same level of detail as a standard EMPS dataset.

Table 7 Supply and demand for the Norwegian dataset. Note that the table contains a mix of capacities (MW) and yearly energy consumption (GWh/year). Some of the capacities vary over the year, the ones shown in the table refers to week number 1. A curtailment function starts at a price of 56.7 øre/kWh with 3% curtailment and ends at maximum model price 364 øre/kWh.

Type of supply	**Capacity (MW)**	**Price (øre/kWh)**	**Type of demand**	**Capacity (MW)**	**Price (øre/kWh)**
Import Sweden	125	13	Export Sweden	101.2	14
Import Sweden	29.6	14	Export Sweden	101.2	12
Import Sweden	101.2	15	Export Sweden	101.2	10
Import Sweden	130.9	18	Export Sweden	89.2	9
Import Sweden	208.3	26	Export Sweden	303.5	8
Import Sweden	101.1	40	Export Sweden	577.4	7
Import Denmark	100	14	Export Sweden	630.9	5
Import Denmark	100	18	Export Denmark	100	13
Import Denmark	150	21	Export Denmark	100	11
Import Denmark	150	30	Export Denmark	100	9
Import Finland	59.6	15	Export Denmark	200	7
Import Russia	19	9.5	Export Finland	59.5	11
Thermal production	22.5	0	Sum price dependent load described by capacities	**2464.1 (MW)**	
Thermal production	25	1	**Other demand**	**Energy (GWh/year)**	**Price (øre/kWh)**
Thermal	34.5	9.6	Boiler	970	25

Type of supply	Capacity (MW)	Price (øre/kWh)	Type of demand	Capacity (MW)	Price (øre/kWh)
production					
Thermal production	20.7	14.3	Boiler	2583	17
Thermal production	51.1	17	Boiler	770	15
Thermal production	10	19.8	Boiler	367	13
Thermal production	35	45	Boiler	3514	8.5
Sum price dependent supply	1473		Flexible Statkraft contract	770	15
			Sum price dependent load describe by yearly energy consumption	**8974 (GWh/year)**	
Average hydro production	**109500 (GWh/year)**		**Price independent demand**	**108000 (GWh/year)**	-

All test runs are done for the same system. The planning horizon is 4 years with weekly time resolution. The only difference between the different runs is the parameterisation of the inflow model and the type of inflow series used in the simulation part of the algorithm. In many hydro-dominated systems it has been customary to simulate the system for records of observed inflow, this is also the case for the Nordic system. The inflow records we use in this test contain 50 years of observed data for the period 1931-1980.

The parameterisation of the inflow model deals with the following:

- Seasonal dependency of the parameters φ and model innovation ε_t. In all our tests the year is divided into 3 seasons with individual estimated parameters for each season. The seasons are week 1-17, 18-35 and week 36-52.
- Whether to use PCA or sampling from model estimation error (residual) to find representative discrete values of ε_t.
- The number of PCs to be used and the number of discrete values for each PC if this approach for modelling ε_t is used. The product of this will give the number of openings[7] used in the backward pass of the SDDP algorithm. Correspondingly the user also has to decide the number of openings to be used if the alternative sampling method is applied.

We have simulated 500 artificial inflow scenarios (years) with weekly time resolution for the complete set of inflow series to the Norwegian market model (105 different inflow series) for six different parameterisations of the inflow model:

1) 12 openings based on 3 PCs which can take respectively 3, 2, 2 values.
2) 12 openings based on sampling from the residuals.
3) 160 openings based on 5 PCs which can take respectively 5, 4, 2, 2 and 2 values.
4) 160 openings based on sampling from the residuals.
5) 1500 openings based on 6 PCs which can take respectively 5, 5, 5, 3, 2 and 2 values.
6) 1500 openings based on sampling from the residuals.

[7] The term openings is commonly used to describe the discrete values representing ε_t

The computation time for the SDDP algorithm is almost proportional to the number of openings and it is therefore desirable to use as few as possible.

3.6.3 Inflow statistics – individual series

We have selected 6 of the 105 inflow series as examples, see Table 8. Of these 6 series, 5 are used for large waterways in the model and 1 example from a smaller inflow series with atypical seasonal variation (555A). The 5 series with larger inflow are selected so as to represent the largest waterways with a good geographical spread.

Table 8 Overview of the example inflow series.

Identity	Measuring point	Represents (ex.)	Occurrence
483A	Møsvatn	Vinje	38
555A	Brådlandsvatn	Langfoss	10
581B	Hauge bru	Blåsjø	20
636C	Horgheim	Tafjord	34
894A	Kråkfjord	Tinnsjø	32
9687B	Namsvatn	Liningen	14

The following discussion uses the Møsvatn and Brådlandsvatn series as the prime examples. Corresponding figures for the other series can be found in Appendix A2. Statistical properties of the generated series are shown in Figure 14, Figure 15 and Figure 18 for Møsvatn and Figure 16, Figure 17 and Figure 19 for Brådlandsvatn. All figures show energy inflow from week 1 to week 52, averaged over all 500 years.

From the figures we can draw the following conclusions:

1. The PC-model shows much too small variation for each week of the year compared to the observed values. The variation is significantly better represented by the residual model. This is seen by comparing Figure 14 and Figure 15 for Møsvatn and/or Figure 16 and Figure 17 for Brådlandsvatn.
2. The weekly average inflow is about the same, as expected, for both models and equal to the observed values.
3. The residual model predicts prolonged dry or wet periods significantly better than the PC-model (when compared to the observed series). This is shown in the duration curves in Figure 18 and Figure 19.
4. The residual model predicts negative inflows more frequent than the PC-model.
5. The use of more openings in the model seems to better reproduce the variance in the model. This is particularly visible looking at the maximum value curves, but also the standard deviations, notably in Figure 16 and Figure 17. This reproduction is a good thing, but it also follows an increase in negative inflows, which is not so good.

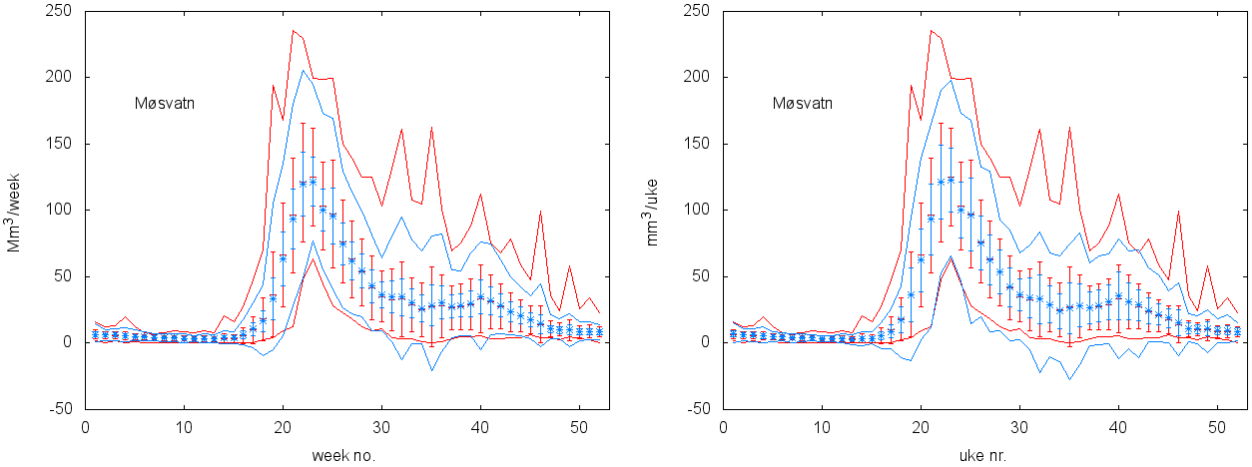


Figure 14: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Møsvatn. Observed data are in red and generated data from the PC-model in blue. Left figure: 12 openings. Right figure: 1500 openings.

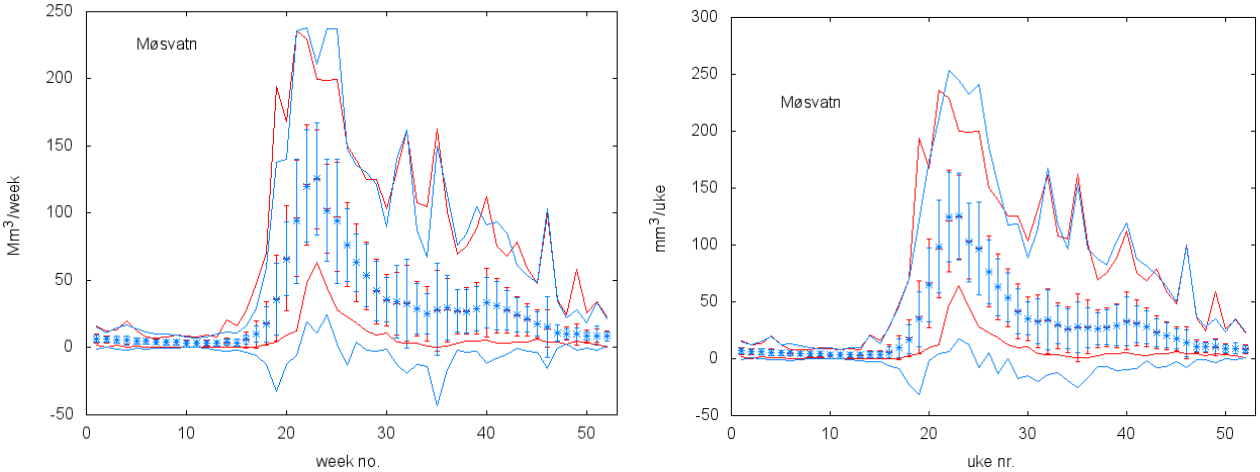


Figure 15: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Møsvatn. Observed data are in red and generated data from the residual model in blue. Left figure: 12 openings. Right figure: 1500 openings.

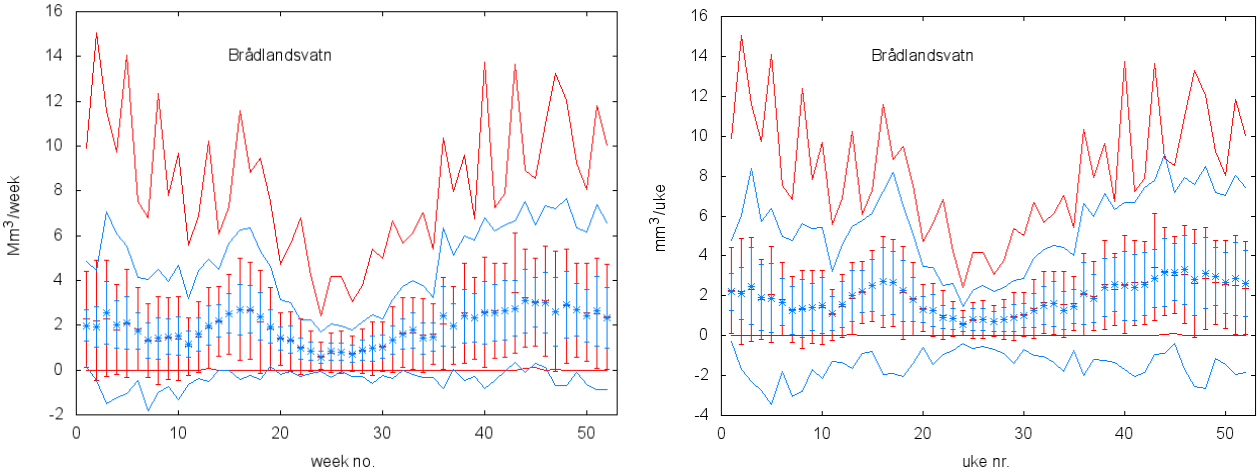


Figure 16: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Brådlandsvatn. Observed data are in red and generated data from the PC-model in blue. Left figure: 12 openings. Right figure: 1500 openings.

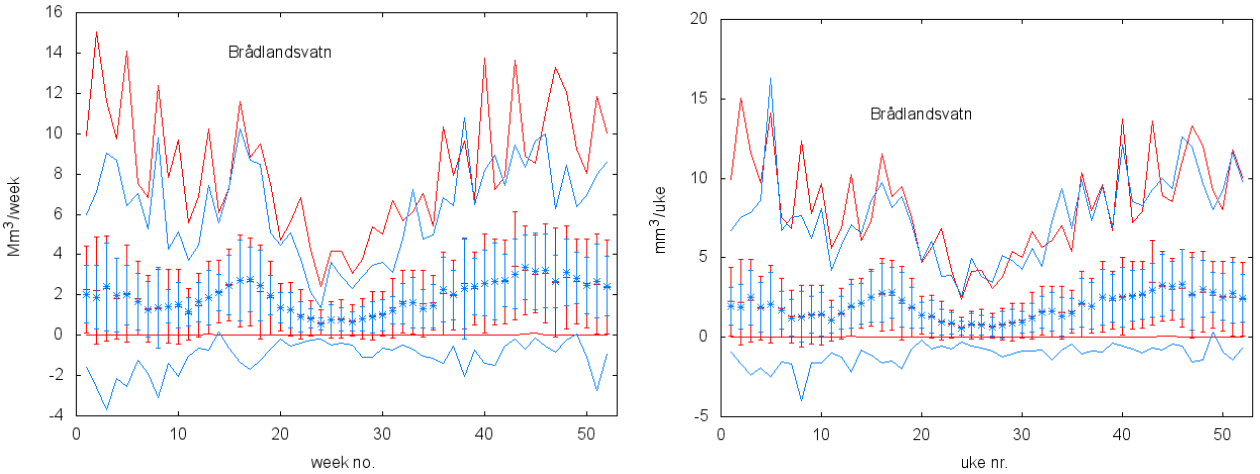


Figure 17: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Brådlandsvatn. Observed data are in red and generated data from the residual model in blue. Left figure: 12 openings. Right figure: 1500 openings.

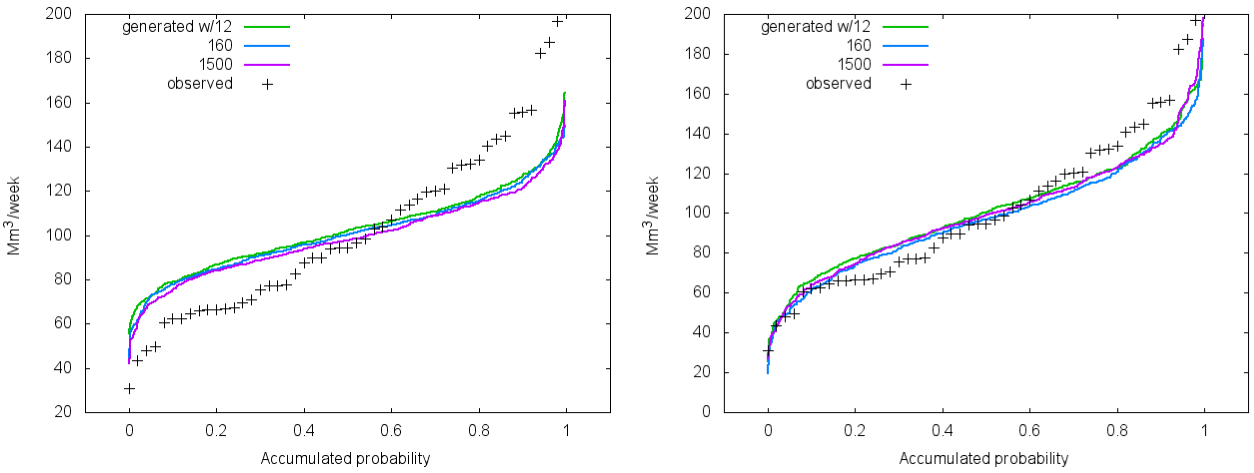


Figure 18: Sum of inflow for Møsvatn from week number 1 to 17 shown as function of accumulated probability. Generated series from the PC-model to the left and the residual model to the right.

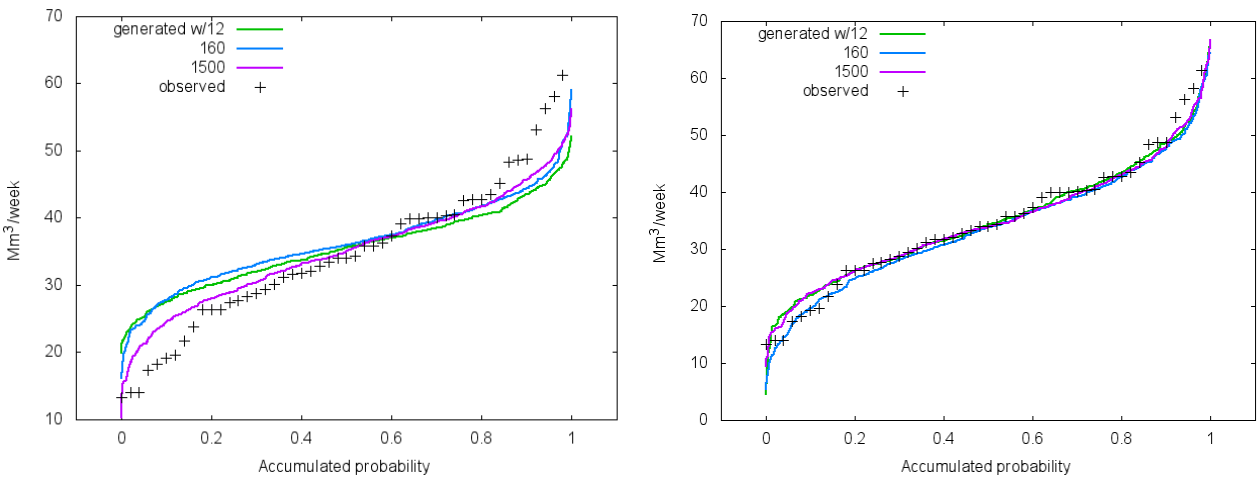


Figure 19: Sum of inflow for Brådlandsvatn from week number 1 to 17 shown as function of accumulated probability. Generated series from the PC-model to the left and the residual model to the right.

3.6.4 Inflow statistics – energy series

The purpose of this section is to present some "measures" that also includes aggregation of all time-series in time and space. To recapitulate, there are 105 different inflow series that are applied to 500 different hydro modules. This is possible because the same series are applied to several reservoirs with different scaling. The measures we have used to compare different inflow models are the storable and non-storable energy inflow series that are calculated by the EOPS model. These time series are calculated to represent inflows to a one-reservoir energy model of a physical system as good as possible.

Non-storable and storable inflows are calculated as follows and summarized for all reservoirs:

Non-storable inflow [GWh] =

Production caused by non-storable inflow to plant
+ Production caused by minimum discharge or bypass constraints
+ Production to avoid overflow
- Energy used for pumping to avoid overflow

Storable inflow [GWh] =

Sum production
- Non-storable inflow
- Energy used for pumping
+ Increase in reservoir storage (- decrease) (volume change multiplied by energy conversion factor to sea)

Possible overflow will not be included in the storable and non-storable inflow series. In the calculation, the actual production in each time step is controlled by the EOPS models general heuristics. The storable and non-storable inflow series aggregates all individual inflow series in the space dimension according to their importance for the hydro production. The split between storable and non-storable inflow may also say something about the accumulated effect in the time dimension because extreme high inflow in many consecutive time steps will increase forced production to avoid overflow for smaller reservoirs. Therefore these time series will also say something about accumulated time effects. Table 9 summarizes the different inflow models considered here.

Table 9 Different inflow models used for testing.

Abbreviation	Description
r-obs and u-obs	Storable (r) and non-storable (u) inflow generated from observed inflow series to individual reservoirs (time period 1931-1980)
r-12-500-prin, u-12-500-prin	Artificial inflow series, 500 years, generated from SDDP inflow model with 12 openings based on PCA
r-12-500-res, u-12-500-res	Artificial inflow series, 500 years, generated from SDDP inflow model with 12 openings drawn from model innovation
r-1500-500-prin, u-1500-500-prin	Artificial inflow series, 500 years, generated from SDDP inflow model with 1500 openings based on PCA
r-1500-500-res, u-1500-500-res	Artificial inflow series, 500 years, generated from SDDP inflow model with 1500 openings drawn from model innovation

We have also generated similar set of series based on 160 openings. All results show that these series have properties in between corresponding series based 12 and 1500 openings and are therefore not presented in this report.

Table 10 gives an overview of the average yearly storable and non-storable inflow. All numbers are in TWh.

Table 10 Overview of the average yearly storable and non-storable inflows, in TWh.

Type of series	Average yearly storable inflow	Average yearly non-storable inflow	Sum inflow
Observed	80.6	24.4	105.0
12-500-prin	83.1	24.0	107.1
12-500-res	81.5	24.5	106.1
1500-500-prin	82.7	23.4	106.6
1500-500-res	81.6	24.3	105.8

It is specified that the average individual inflow (Mm3/year) to each module is the same for all cases, but the calculated average inflow (TWh/year) to the whole system varies slightly as shown in Table 10. This is mainly because overflow is not part of the calculated storable and non-storable inflow series. Calculated energy inflow is also affected by head dependencies. The operation strategy used to calculate the energy inflow is the same for all cases[8], but differences between the inflow series give different storage levels and overflow.

The model of Norwegian system that is used for these calculations is old and does not completely represent the current system. This is also indicated by the estimated annual energy inflow (106-107 TWh) which is low compared to current estimates.

Figure 20 - Figure 28 present different statistical measures for the calculated storable and non-storable energy inflow series. All figures show energy inflow from week 53 to week 104. The energy inflow series are calculated through a simulation of the system and we have chosen to present the inflow for week 53 to 104 to make the results less affected by initial conditions (reservoir storage). The energy inflow is measured in GWh/week in all figures.

From the figures we can draw the following conclusions:

1. The weekly average inflow is about the same, as expected, for all models and equal to the observed values, see Figure 21 and Figure 26.
2. The PC-model shows much too small variation for each week of the year compared to the observed values. This can be seen when comparing the 25 and 75 percentiles between Figure 20 and Figure 22 (storable) and Figure 25 and Figure 27 (non-storable).
3. All statistical models show too little probability of prolonged dry or wet periods compared to the observed series. This is shown in the duration curves in Figure 24 and Figure 29.
4. The model based on sampling noise from the innovations (residual model) seems to be more precise than the model based on PCA.
5. The residual model with 12 openings is almost equally good as the model with 1500 openings. It should be noted that because the residual model is implemented with separate sets of residuals for each week and estimated from 50 years of observed data there are no more than 50 residuals to draw from. The residual model with 1500 openings is therefore overkill with no additional information.

[8] Obtained in what is referred to as "bunden simulering".

Figure 20: Observed storable inflow for each week of the year.

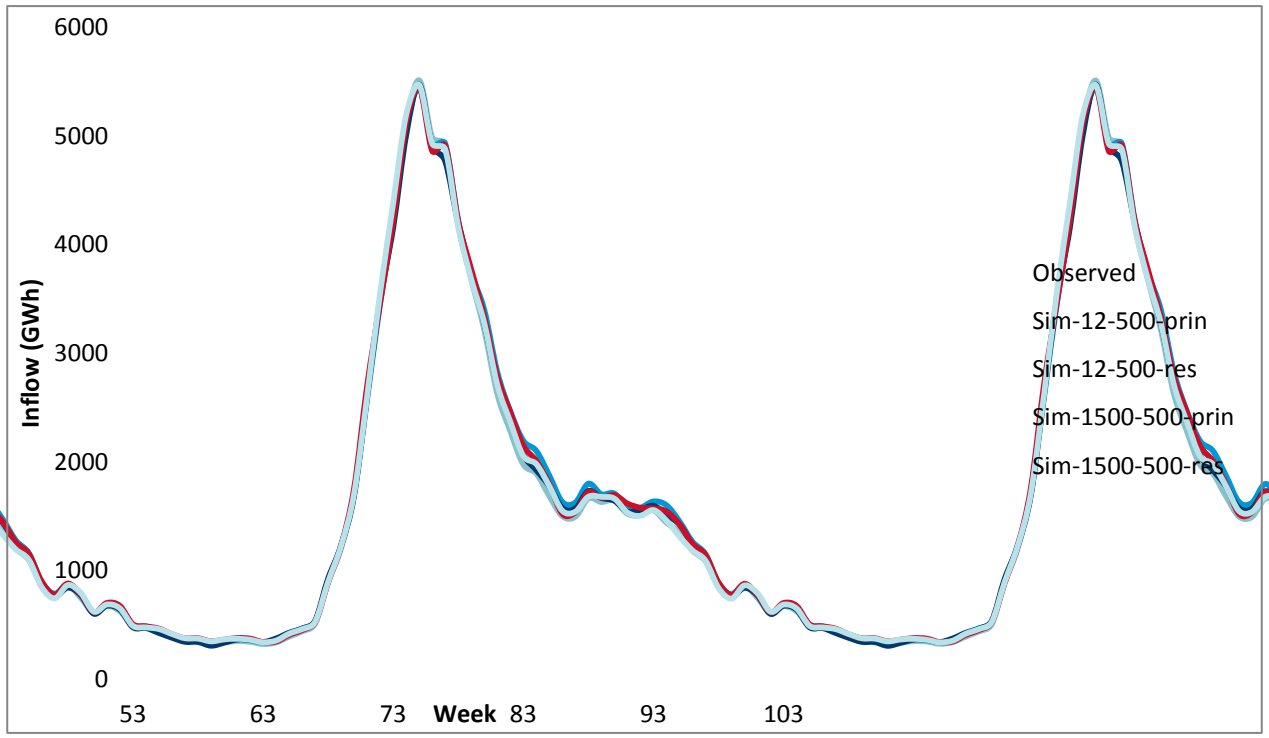


Figure 21: Weekly average storable inflows for the different inflow series.

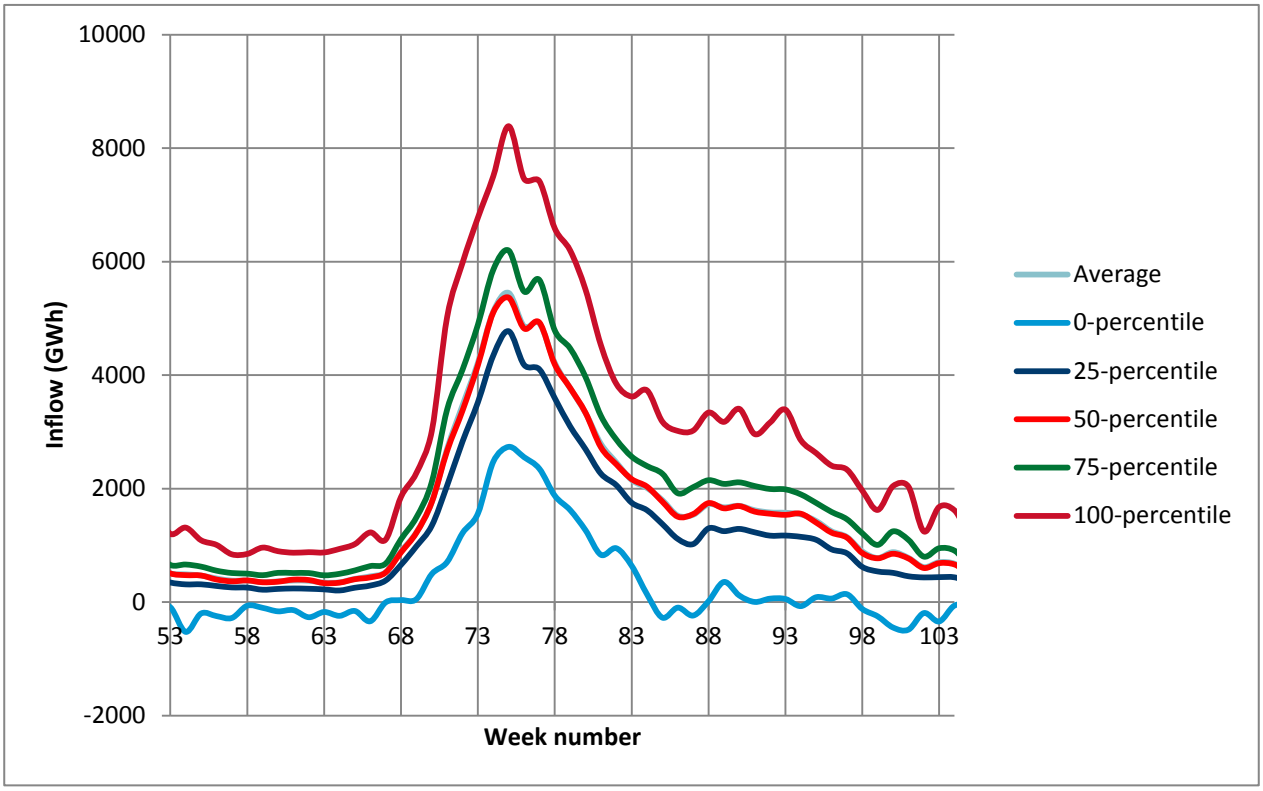


Figure 22: Storable inflows – series 1500-500-prin.

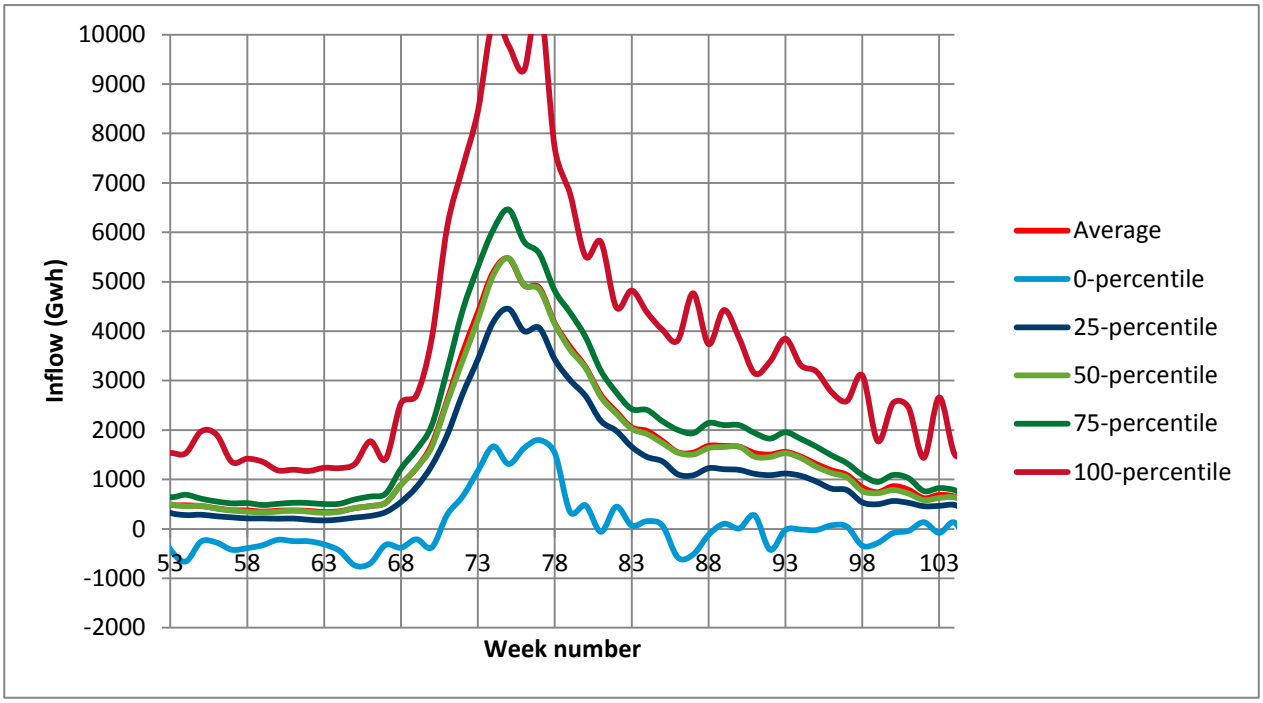


Figure 23: Storable inflows – series 1500-500-res.

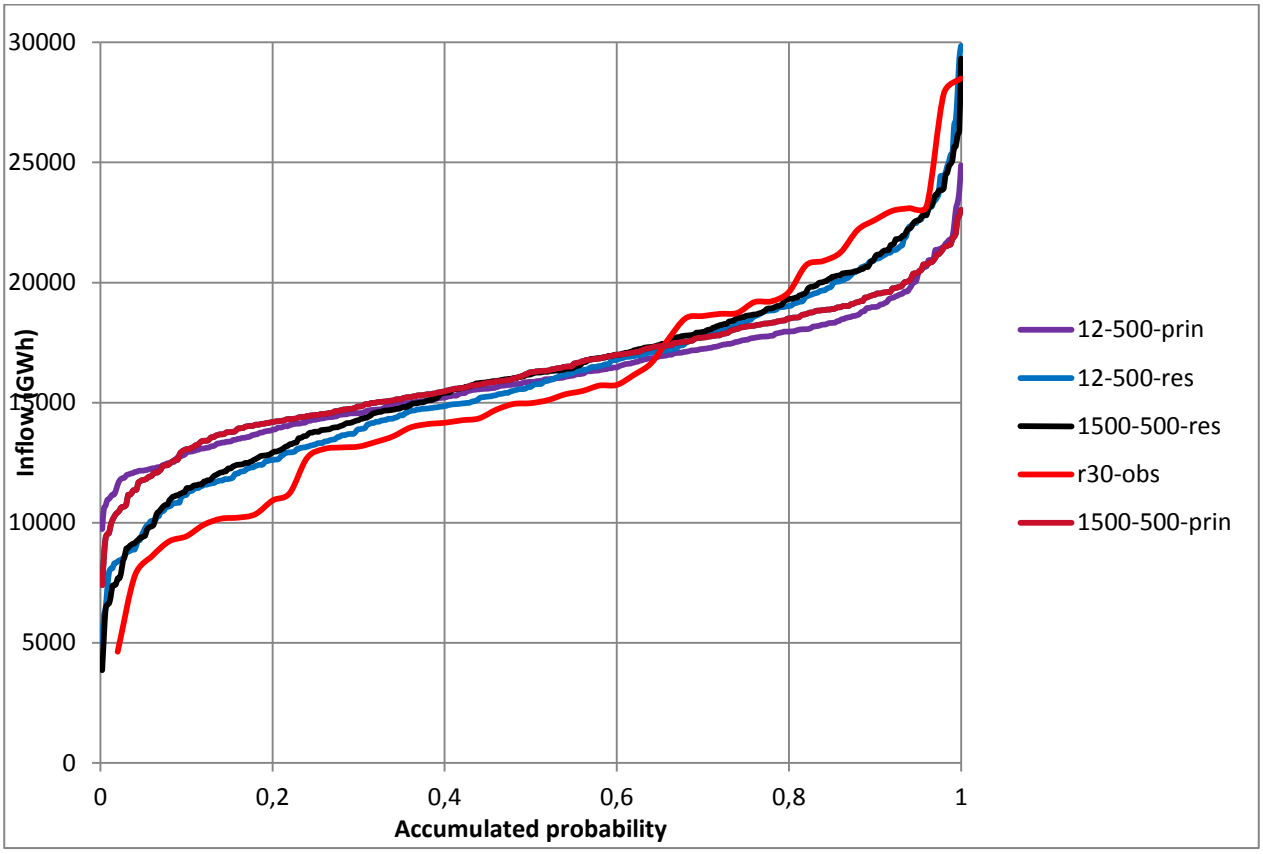


Figure 24: Sum of storable inflow from week number 1 to 20 shown as function of accumulated probability.

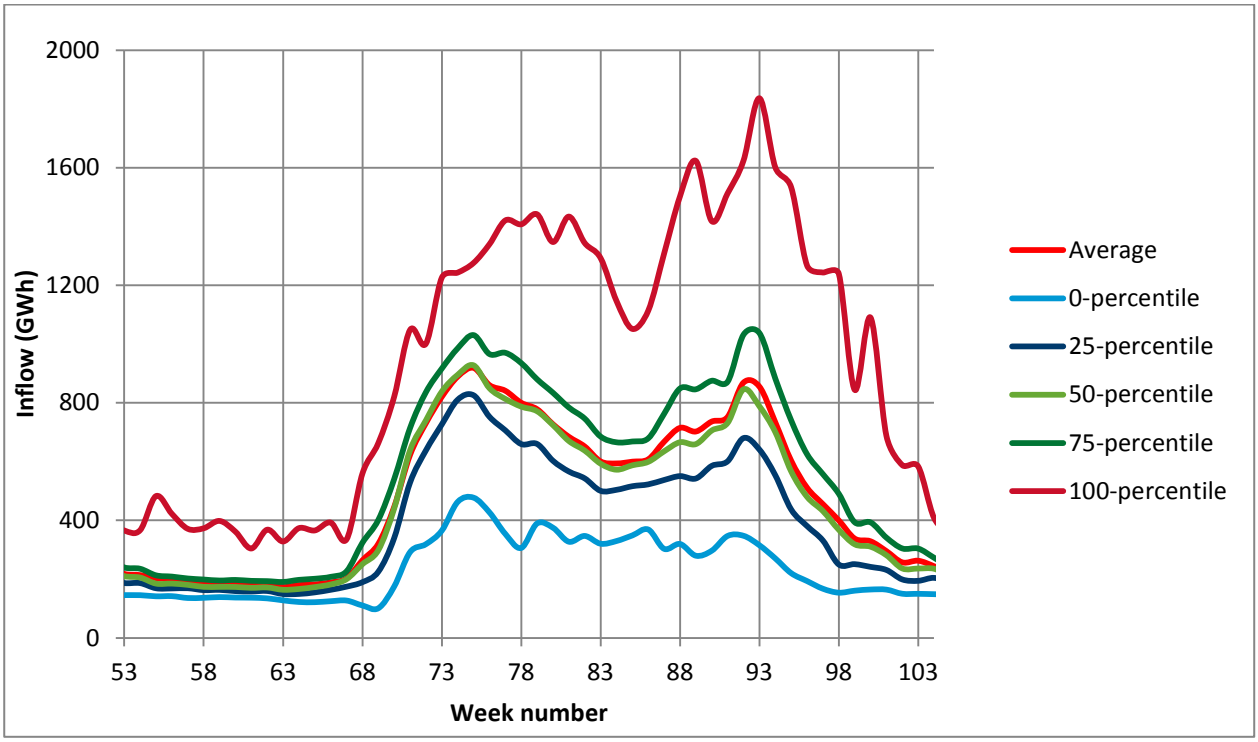


Figure 25: Observed non-storable inflows for each week of the year.

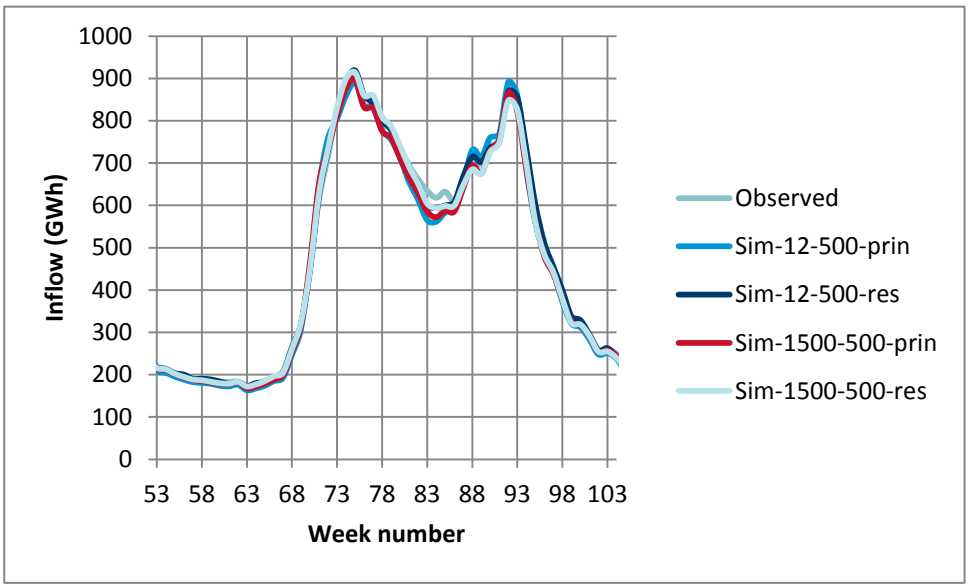


Figure 26: Weekly average non-storable inflows for the different series.

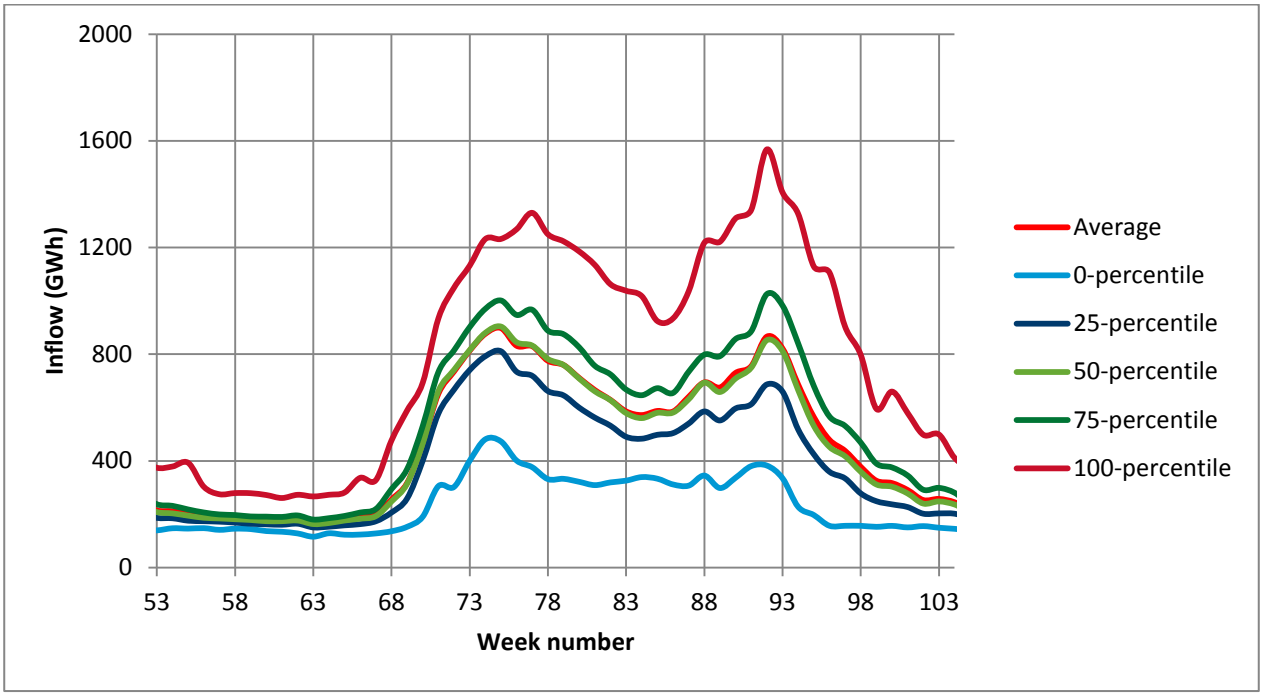


Figure 27: Non-storable inflows for series 15-500-prin.

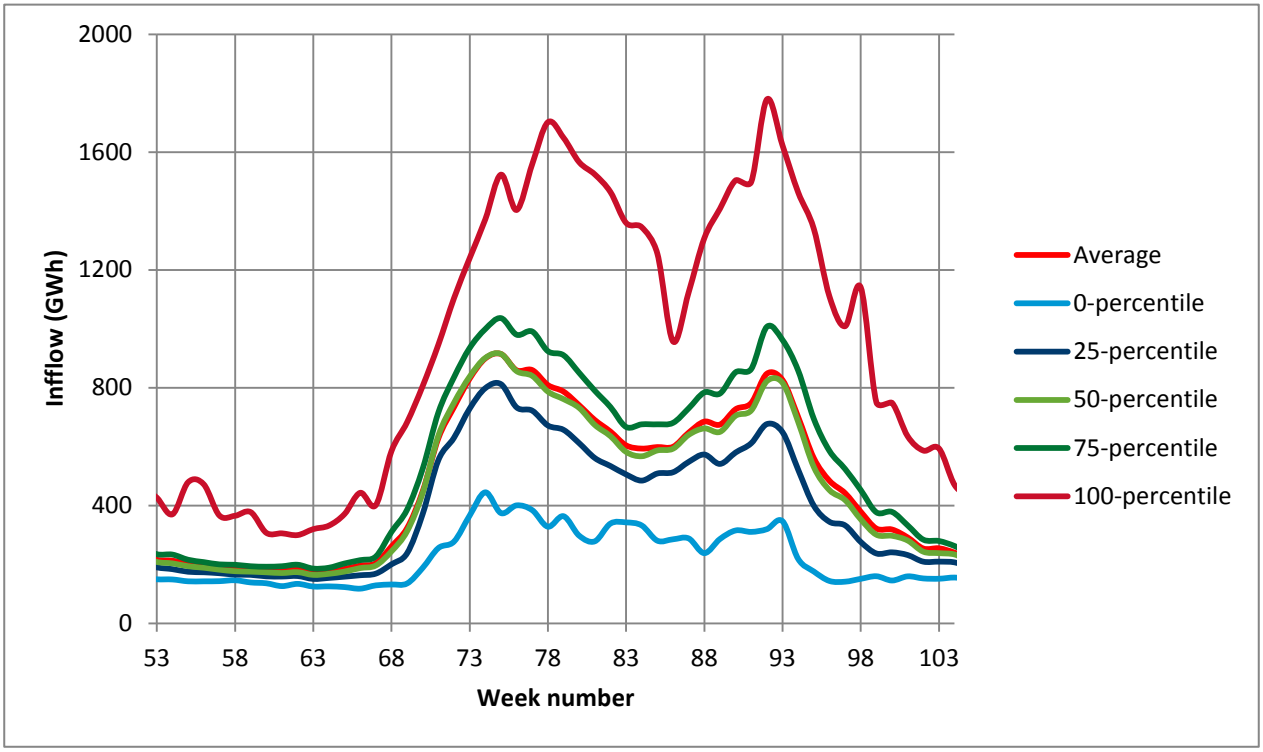


Figure 28: Non-storable inflows for series 1500-500-res.

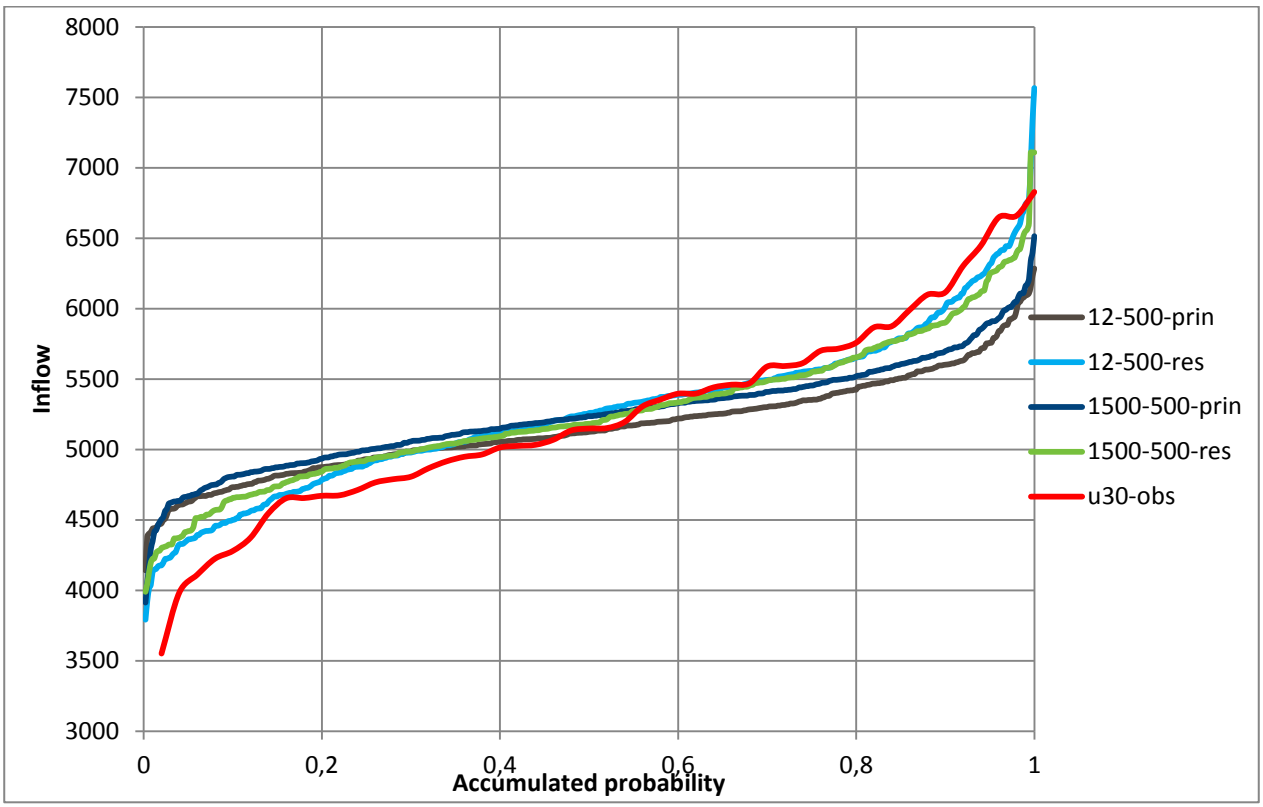


Figure 29: Sum of non-storable inflow from week number 1 to 20 shown as function of accumulated probability.

The main purpose of this comparison is to explain results from the ProdRisk model. The statistical models are used in the backward pass of the ProdRisk model and if they differ from the scenario that are used in the forward pass, simulated model results will not be optimal. The aggregated energy series are not used in the ProdRisk model itself, but they are measures for the sum consequences of using the individual series. That is, if the energy series for the statistical generated series show too little probability of sum storable inflow from January 1 to start of spring flood being low, ProdRisk will have too low water values January 1.

3.7 Expected computational performance

The ProdRisk tests reported here were performed on a Norwegian system model having 500 modules and 1 load period within the week. This problem is much smaller than the weekly decision problem used in the testing in section 2.7. The ProdRisk problem has the following characteristics (with the characteristics of the aggregate Nordic weekly decision problem in parenthesis).

Variables:	3600 (36618)
Constraints:	570 (6360)
Elements in matrix:	8374 (66149)

Roughly speaking, the ProdRisk problem is a factor of 10 smaller than the weekly decision problem used in section 2.7.

The test runs were performed on 2 different machines:

- SEFASPARA Windows server, 8 quad-core AMD processors at 2.4 GHz, 128 GB RAM
- Linux cluster with Intel Xeon CPUs, each node has 8 cores (dual-die quad core) with 2.93 GHz and 32 GB RAM dedicated.

In practice, we experienced various difficulties when running the tests on the Linux cluster. Thus, the results presented in the following are based on simulations performed on the SEFASPARA server.

Some run times are listed in Table 11. In practice, ProdRisk will usually not converge on cost or reservoir operation, but rather meet a predefined limit on iterations. The number of iterations allowed for the time estimates is stated in Table 11.

Table 11 Run times for ProdRisk.

Test run	Machine	No. processors	Forward scenarios	Backward openings	Number of iter.	Stored cuts	Computation time [hours]
1	SEFASPARA	12	50	12	200	1000	24
2	SEFASPARA	12	500	12	100	1000	102

By studying the differences between forward (upper bound) and backward (lower bound) cost estimates, one can gain insight in the converge process in ProdRisk. These results are plotted in Figure 30 for test run no. 1 and Figure 31 for test run no. 2.

In test run no. 1 we use observed inflows, and should therefore expect a cost gap, since the stochastic inflow model (used in the backward pass) will not represent historical records (used in the forward simulation). The convergence profile in Figure 30 is therefore typical for standard use of ProdRisk, and the cost gap seems to somewhat stabilize after 200 iterations.

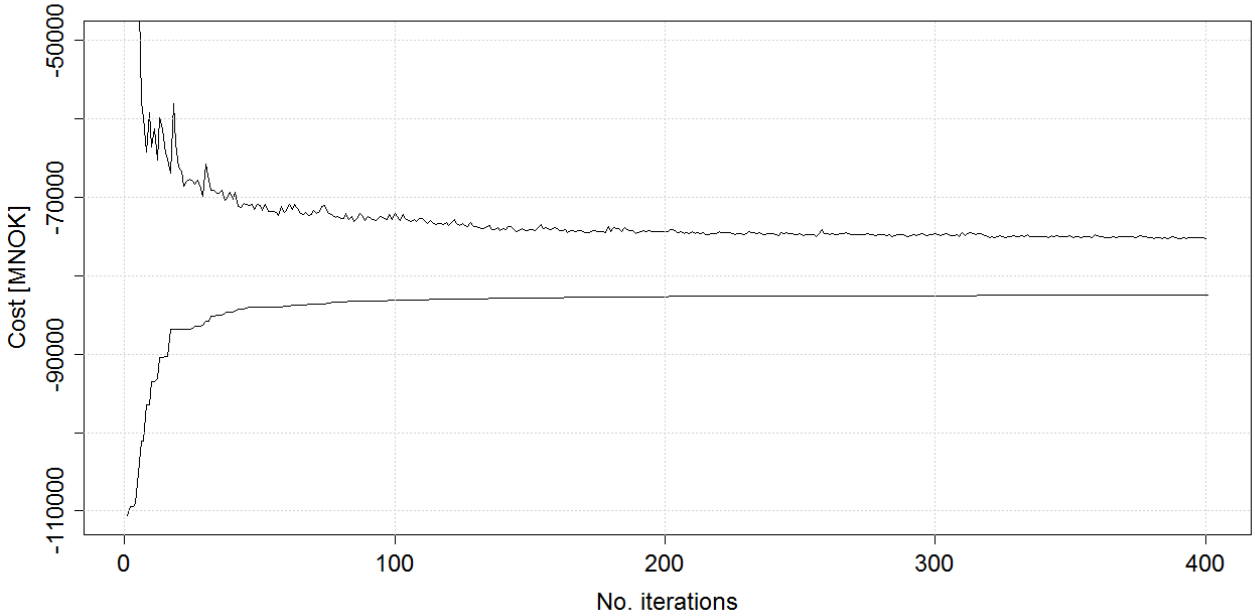


Figure 30: ProdRisk convergence on cost, 50 observed scenarios and 12 backward openings.

In test run no. 2 we have sampled 500 inflow series using the ProdRisk's stochastic inflow model. Thus, since the inflows used in the forward and backward iterations are obtained from the same model, we expected the cost gap to close. This is not the case in Figure 31, and we believe the explanation is the 'limited memory' we have given ProdRisk.

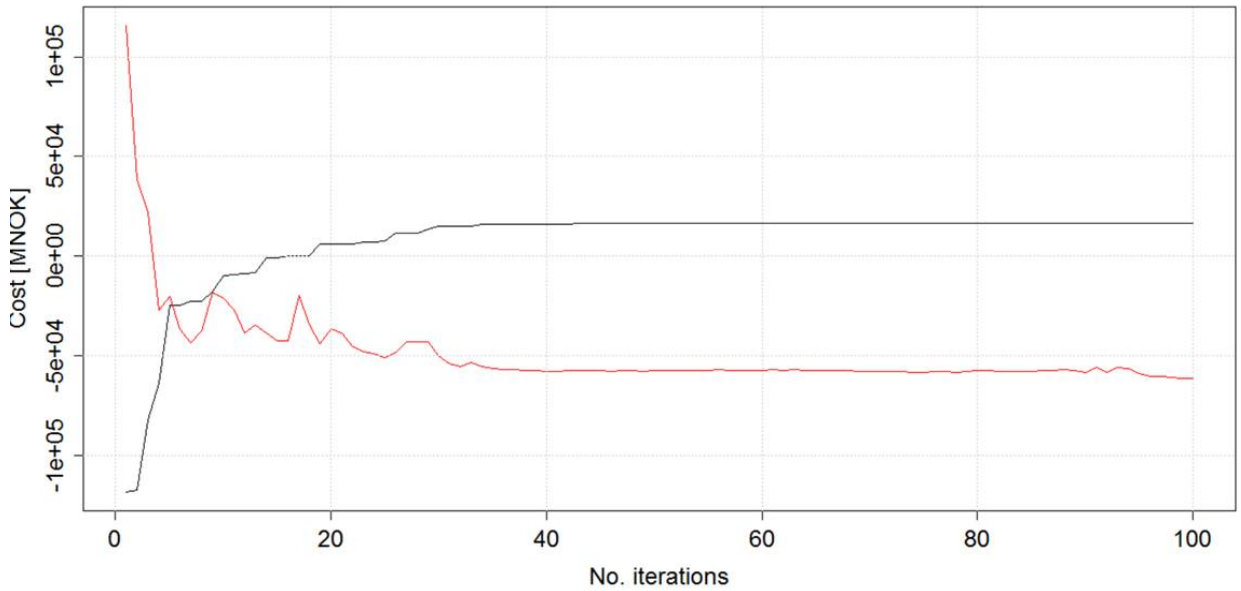


Figure 31: ProdRisk convergence on cost, 500 sampled scenarios and 12 backward openings.

3.8 ProdRisk test results

ProdRisk is run on the dataset described in section 3.6. This section shows some results from these tests and compare with results from the Vansimtap model. The results from the Vansimtap model are used as a reference. Vansimtap calculates water values for an aggregated hydro model of Norway and uses heuristics to distribute aggregated hydro production to the individual plants. When applied to the whole Norwegian system Vansimtap should give poorer result than a Samkjøringsmodell applied to the same system. This because in Samkjøringsmodellen aggregated water values can be calculated for different parts of the hydro system and therefore better take into account different system properties.

For applications to individual water courses ProdRisk typically gives 1-1.5 % better results than the Vansimtap model. This test is different because the number of reservoirs is about 20-50 times larger than for a typical water course. In typical user applications for long and seasonal hydro scheduling the market price is a stochastic exogenous parameter while in this case the market price is modelled endogenously. The SDDP method itself is in principle independent on how market price is modelled. However, when the market price is modelled endogenously the SDDP method is expanded with a discrete price state that is handled by traditional dynamic programming approach. Transition between discrete price states is modelled by a Markov chain type model.

All Vansimtap and ProdRisk results are from a parallel simulation starting in week 1 and ending in week 208. The results that are presented are picked out from week 105 to 156 but presented with x axis numbering from 1 to 52. It is possible to see that the results are not from a "real" serial simulation by comparing reservoir volume for week 1 with the volume in week 52. They should be symmetric for a serial simulation.

Figure 32 and Figure 33 show simulated reservoir volume for Svartevatn in the Sira-Kvina water course. Figure 34 and Figure 35 show the same for Møsvatn, a significant reservoir close to Rjukan. The simulated results show that for extreme dry years, indicated by the 0-percentile, both Møsvatn and Svartevatn are emptied earlier in the spring than what is the case for Vansimtap. On average both reservoirs have lower storage levels for all weeks in the ProdRisk simulations. Figure 36 and Figure 37 show the same results for the aggregated energy reservoir. In many cases load has to be curtailed long before the aggregated reservoir is emptied because it may not be possible to empty all reservoir at the same time. The simulated prices shown in Figure 38 and Figure 39 reflect the same results, maximum prices and average prices reflect a much higher probability of curtailment for the ProdRisk simulations. The weekly average (over all weeks and all inflow years) is 13.9 øre/kWh and 16.7 øre/kWh for Vansimtap and ProdRisk, respectively. Both the higher average and the significant increase in the simulated average price before spring flood is a clear indication that ProdRisk gives a much poorer simulation result than Vansimtap. Detailed energy balance results that are not shown here show the same; ProdRisk gives much poorer results that Vansimtap.

The ProdRisk and Vansimtap simulations discussed above are from simulations with 50 observed inflow years from the period 1931-1980. ProdRisk is run with 12 backward openings based on the residual model for the inflow, 1000 stored cuts and 400 iterations. The convergence, represented by the backward and forward costs, is shown in Figure 30. Ideally these costs should converge to the same value, i.e., zero gap. However, as discussed in section 3.6, we are using observed inflow series in the forward simulation with different statistics than what is given by the model used in the backward pass. Zero gap is only guarantied if the forward pass is a sample from the same model as used in the backward pass. The fact that we are using only 50 simulated scenarios results in sampling error which adds to the possibility of a gap between forward and backward costs.

Our initial hypothesis was that the differences between:
a) inflow model statistics and observed inflow and

b) sampling error
are the main causes for the poor ProdRisk simulation results and the convergence gap.

To test this hypothesis we have simulated with both ProdRisk and Vansimtap on the Norway dataset with artificial inflow series drawn from the inflow model used in the backward pass of SDDP model. The test are done for different parameterization of the inflow model but the results presented in Appendix A3 is from using 500 artificial inflow "years" drawn from an inflow model with 12 backward openings. The backward openings are based on residuals. The results presented earlier indicate that the residual model is better than a model based on principle components, and that it does not matter that much how many openings the model is based on. The calculation time for the SDDP model is increasing (almost linear) with the number backward openings and in practise we have to limit the number of openings. Because the number of historical years is increased to 500, sampling error should be reduced. We have not tested whether 500 years are enough to remove this effect, most likely it is not.

Running Vansimtap and ProdRisk with 500 scenarios has required some modifications of the program code mainly changing of dimensioning parameters. The inflow model used in the SDDP model can give negative inflows. Because we wanted to have full consistency between the forward pass and backward pass in the SDDP algorithm we chose to keep possible negative artificially generated inflows as seen in Figure 14 - Figure 17. This has generated some problems especially for the Vansimtap model and its heuristics which has never been applied with such an amount of negative inflows. It may also be that there are other consequences that we are not aware.

The simulated results shown in Appendix A3 are not as expected. The ProdRisk results are much poorer than Vansimtap results, more curtailment and higher prices. More or less the same results as when compared for observed inflows. The problem with these results is that they questions our understanding of why ProdRisk perform poor for a large market problem and very good for a smaller hydro scheduling problem with exogenous stochastic price. We believed ProdRisk should at least give as good results as Vansimtap when applied to artificial inflow series. When applied to artificial inflows the two main reasons for poor results are removed.

It feels more comfortable to decide between models when we understand why results are as they are. The test results shown in Appendix A3 to some extent undermine our understanding.

Below we discuss possible reasons for this:

A) Inflow model re-estimation
The inflow model parameters was estimated from observed inflow series. 500 artificial inflow years are then generated based on estimated model parameters for all 105 inflow series. But when ProdRisk is run with these artificial inflow series the inflow model parameters used in the backward pass is re-estimated from the 500 artificial inflow years. This may give unintended inconsistency between forward and backward pass. The ProdRisk code can be changed to avoid but this has not been prioritized so far.

B) Negative inflows.
As mentioned above to achieve full consistency between forward and backward pass we chose to allow for negative inflows when the artificial series was generated. However, ProdRisk has never been simulated with this amount of negative inflow before and it may be that there is something connected to this that we are not aware of.

C) Limited number of stored cuts for each time step.
To begin with we run ProdRisk with 1000 cuts but when we saw the simulations results we did a new run with 5000 cuts and the results improved, average simulated price was reduced by about 1 øre/kWh. It is the results from the run with 5000 cuts that are presented. If it is possible to run with 100000 cuts or more it may be that the results are coming close to the Vansimtap model. The limiting factor to increasing the number of cuts is memory requirements. When ProdRisk is applied to a local planning problem a typical above average size system consist of about 10 flexible reservoirs and 1000 cuts may be stored. The typical ratio between the number of cuts stored and the number reservoirs is then 100. In our case we may have 300 reservoirs with flexibility and should use 300000 cuts to keep the same ratio. We have not tested whether it is possible to run with 300000 cuts in a 64-bit application. For the cases studied here, roughly 1GB of memory is used per 1000 stored cuts.

D) Cut removal strategy
One new cut is generated for every backward pass and for every inflow year. If 500 scenarios are used, every backward pass generates 500 new cuts for each time step. When the limit to maximum stored cuts is reached old cuts are removed. This removal strategy is for the 500 scenario cases tested with very different ratio between maximum number of stored cuts and number of new cuts generated at each backward iteration. However, the backward costs shown Figure 31 is strictly increasing, indicating a sound removal strategy.

E) Real forward sampling
In our cases the same 500 artificial inflow years are used in all forwards passes. These 500 scenarios represent a sample of the statistical model. We do not believe that 500 scenarios is enough to remove all sampling uncertainty. Also, 500 scenarios are probably too small to fully explore all relevant combinations of reservoir fillings in the 500 reservoirs. Different combinations are reached in each forward run because the strategy is updated but will still be limited by the inflow scenarios. A statistically correct way to handle this is to draw different 500 scenarios in each forward run. If it is simulated with 500 scenarios the results will still include sampling uncertainty but the strategy will be based on a fully explored state space and therefore be more optimal. The reason this is not done so far is because when the end result is a simulation with observed inflows it is less obvious to use artificial inflows throughout the iteration process.

A to E may explain why the simulations with 500 artificial inflow scenarios was not as expected. It is possible to test many of these assumptions with time. A limiting factor is the computation time. We believe only reason C and E may explain why ProdRisk with observed inflows performs that much poorer for our large market problem compared to a local scheduling problem.

Another important difference is that when ProdRisk is applied to a local planning problem with exogenous stochastic price, the dependency between local water values and the state (storage levels in all other reservoirs) of the system is represented by its dependency on current price level. Future prices and probabilities are given by a discrete Markov chain type price model with very strong autocorrelation. The price model is assumed to be independent of the local inflow model. For the endogenous market approach, each individual water value is in principle directly dependent on the storage level in all other reservoirs, provided enough cuts are used. The future value of water and indirectly future prices are given by the combined statistics of all the inflow models. Important are statistics in time and space and summed over all relevant time periods. Some examples of such statistics are presented in section 3.6.3 (sum for individual series along the time axis) and 3.6.4 (summed both in time and space).

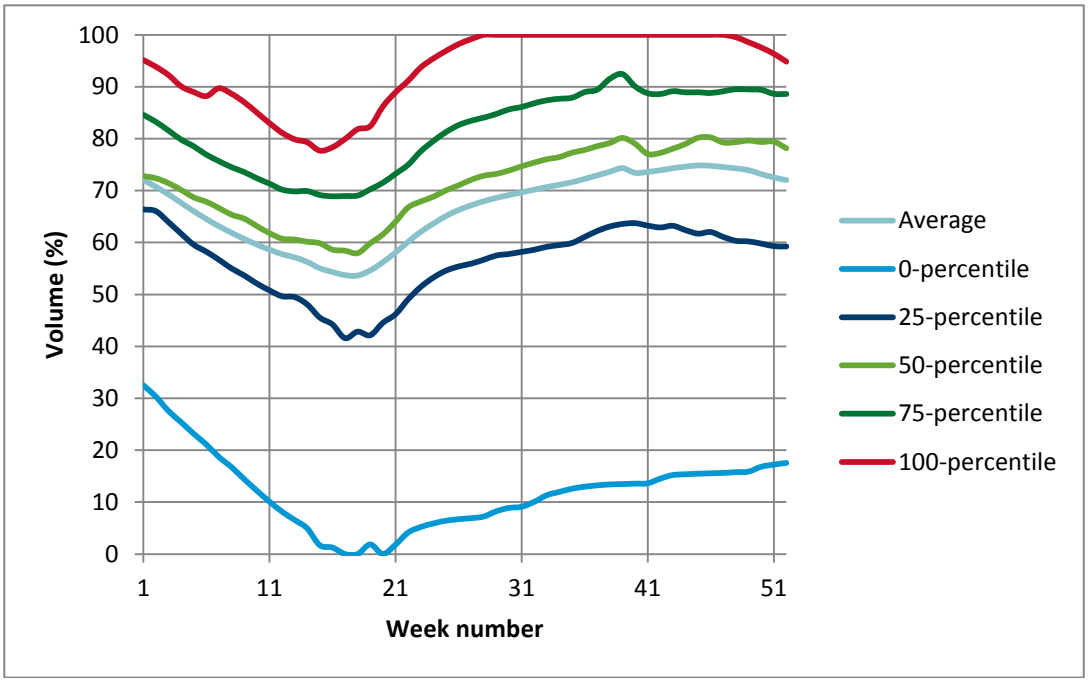


Figure 32: Vansimtap, simulated reservoir volume for Svartevatn.

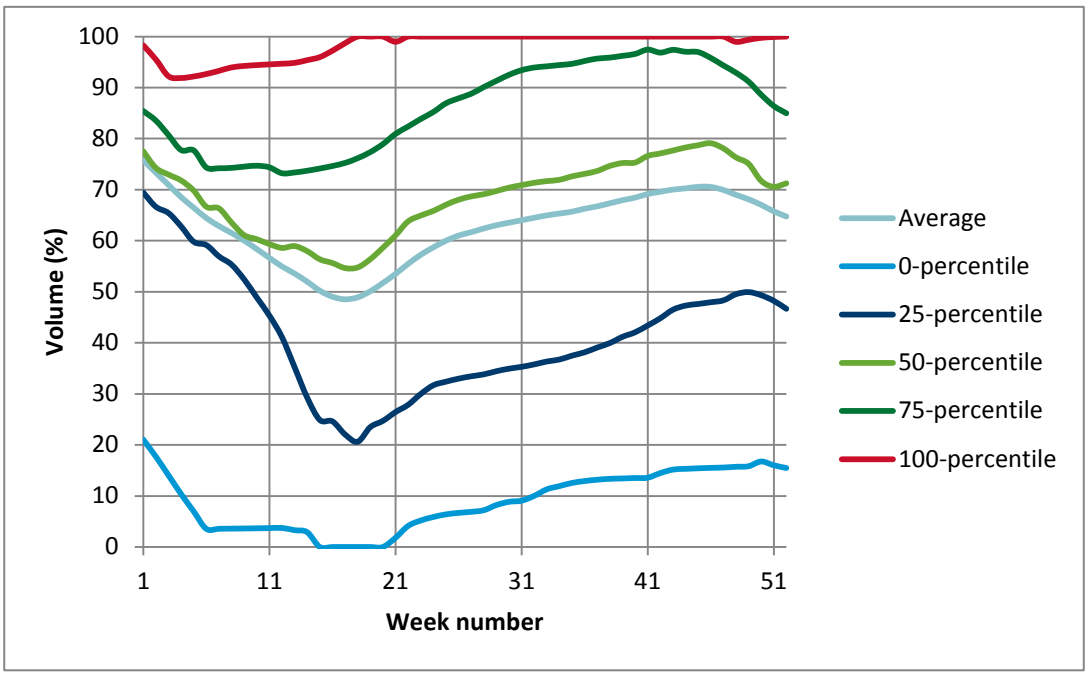


Figure 33: ProdRisk, simulated reservoir volume for Svartevatn.

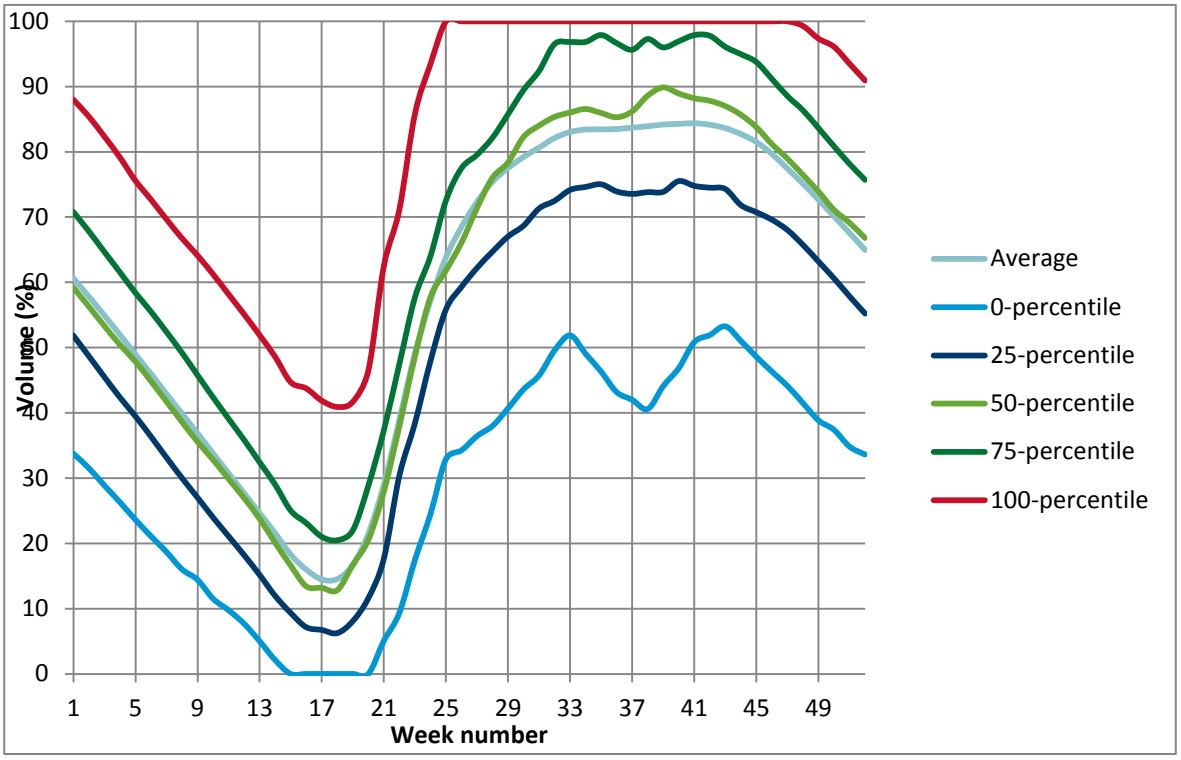


Figure 34: Vansimtap, simulated reservoir volume in Møsvatn.

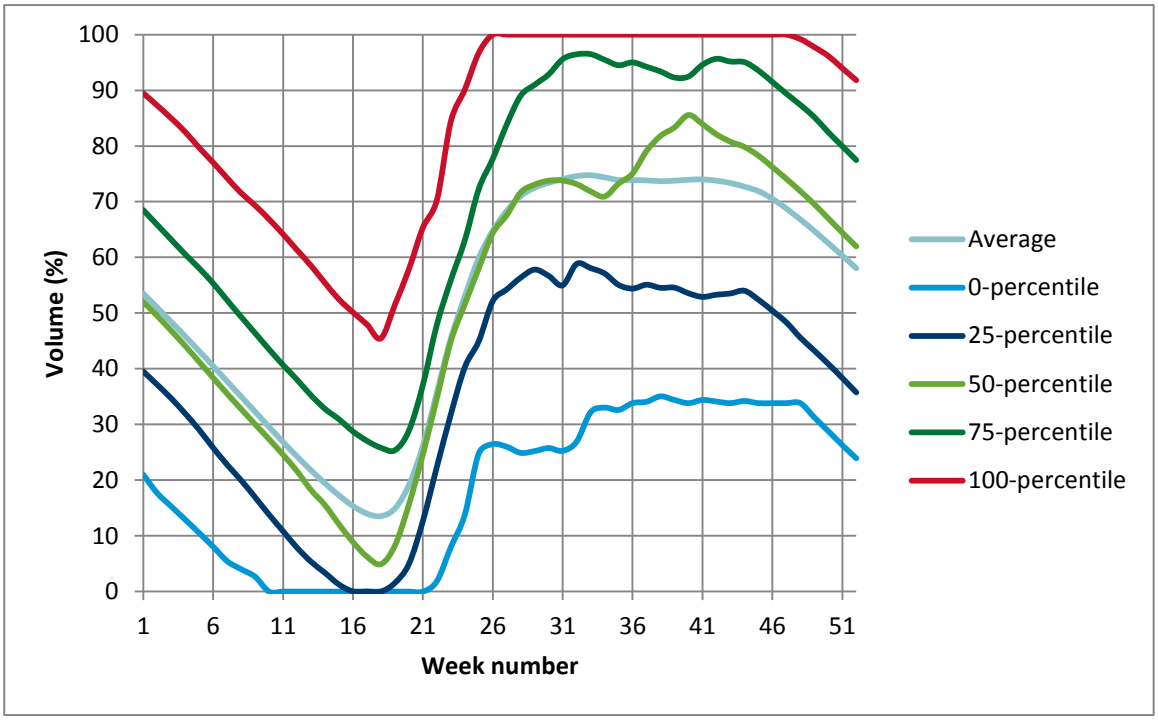


Figure 35: ProdRisk, simulated reservoir volume in Møsvatn.

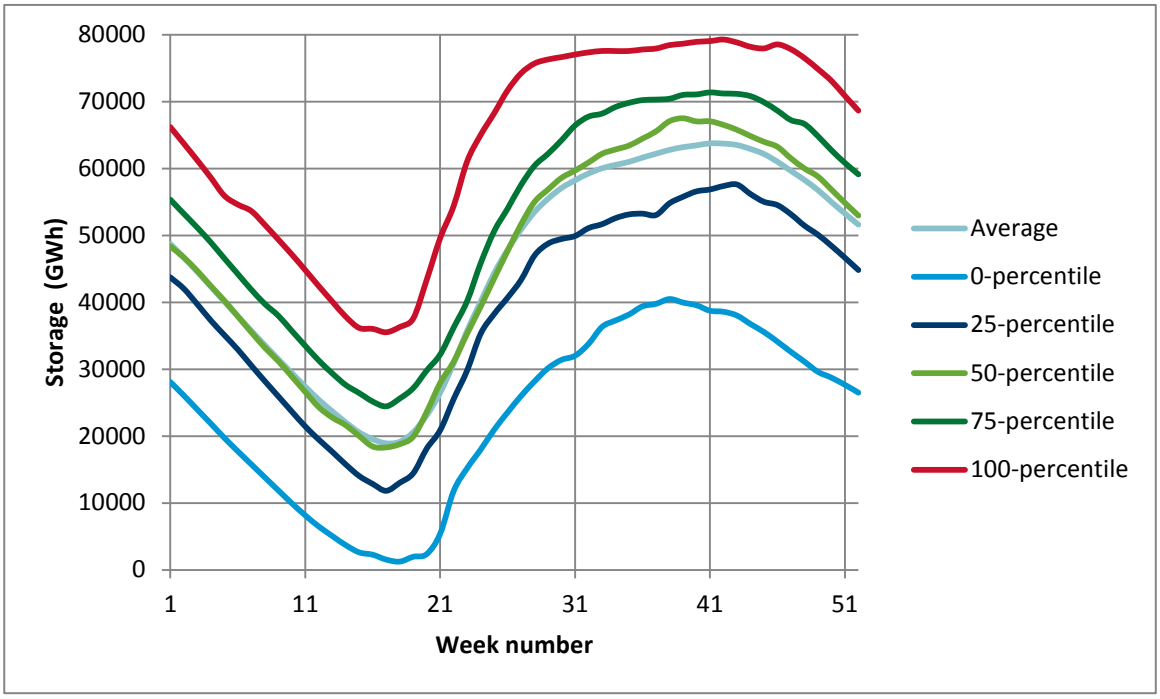


Figure 36: Vansimtap, simulated sum energy storage.

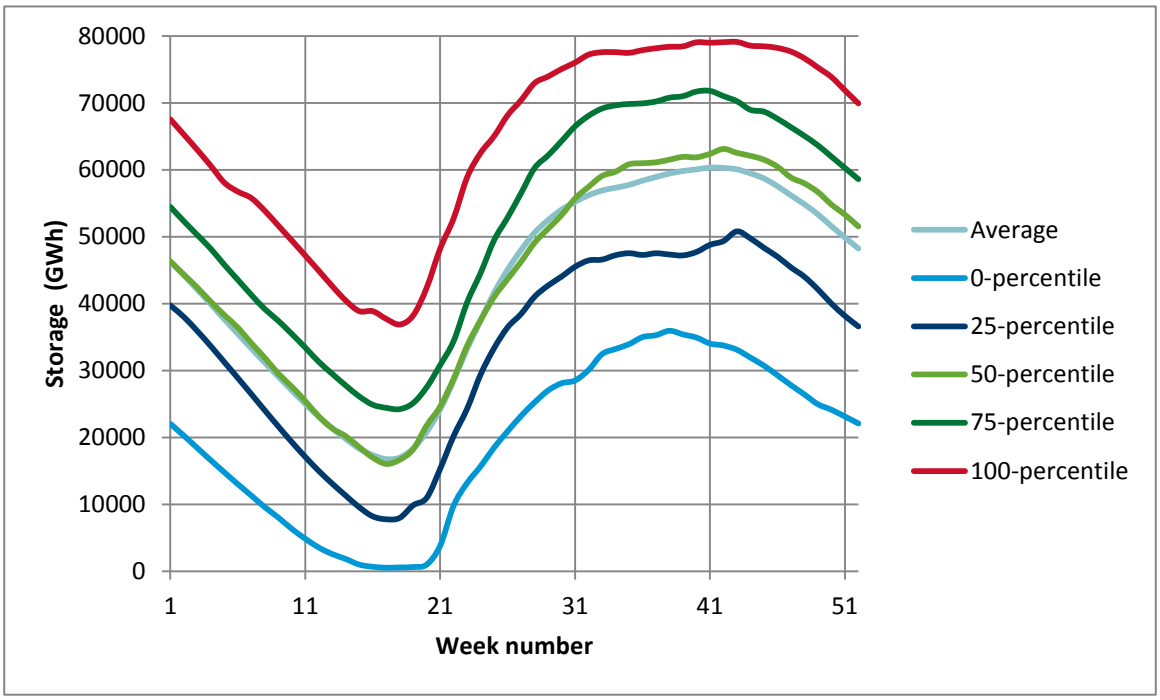


Figure 37: ProdRisk, simulated sum energy storage.

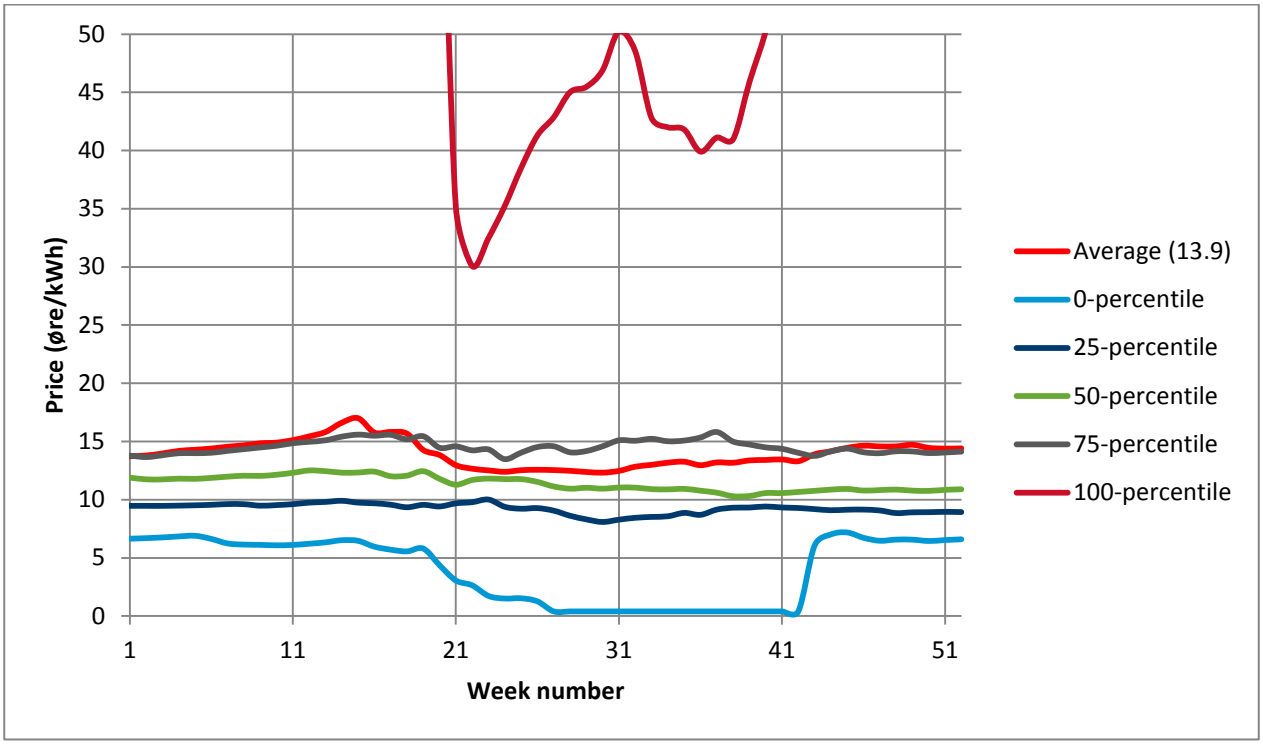


Figure 38: Vansimtap, simulated market price.

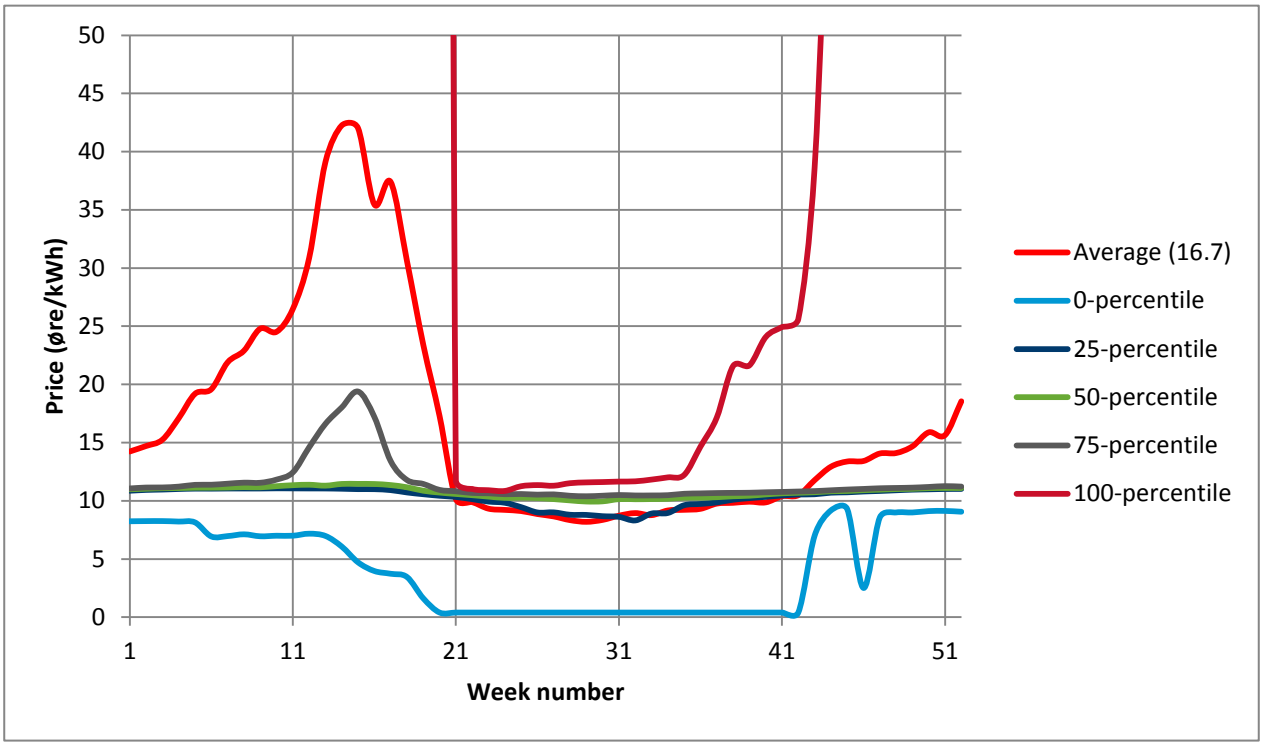


Figure 39: ProdRisk, simulated market price.

4 Comparison

4.1 General

The primary objective of this comparison can roughly be stated as follows: *Find the method that is expected to provide the best results in shortest time*. The overall indicator of goodness in this context is the estimated socio-economic surplus.

There are some major constraints to this objective:

- We limit the search to two known methods, namely the SFS and SDDP
- There is an upper bound on available computational resources. Although this project we will stretch this bound further than what has been done previously, it is still desired to limit the use of computational resources as much as possible.
- Strategies should be verified using historical records of inflow.

The two methods being compared both have the capability of modelling the system with detailed description of hydropower plants and with power flow constraints. Furthermore, both methods are expected to produce robust results. By robust in this context, we mean that if one, e.g., introduce an extra system constraint, we expect to consistently see the system-wide impact of this constraint, typically measured in terms of socio-economic surplus.

One can argue that the comparison of the SFS and SDDP methods is somewhat strange, since the SFS is a simulator and SDDP is an optimization method. Thus, the SFS could in principle[9] include SDDP to provide a first-stage decision.

4.2 How are they different?

We will present an example to illustrate a fundamental difference between the two methods. Consider Figure 40 as an excerpt of Figure 13 in section 3.1, illustrating the iterative SDDP scheme along one scenario. There are four stages and three realizations of stochastic variables at each stage in the backward recursion part of the algorithm. Dots indicate system states and branches indicate realizations of stochastic variables. Black dots indicate the system states obtained in the forward simulation, and the black line the historical inflow record followed for this particular scenario. Grey dots indicate system states obtained in the backward recursion when considering each of the three realizations.

Consider a system with a single reservoir. In a backward iteration we find that for one realization of inflow, the model cannot meet the reservoir constraint without adding artificial water[10], as indicated by the red dot in Figure 40. State 3 (black dot no. 3 from left) is the initial state leading to this particular solution. This situation will impact the cut generated to state 3, providing a signal for the model to increase reservoir level in state 3. This cut is seen when generating cuts in state 2, and so on. Thus, this signal is recursively passed down to state 1 in the backward iteration.

[9] This is not viable in practice due to the enormous computational capacity needed.
[10] The reservoir is emptied, and a penalty variable is used to ensure non-negative reservoir.

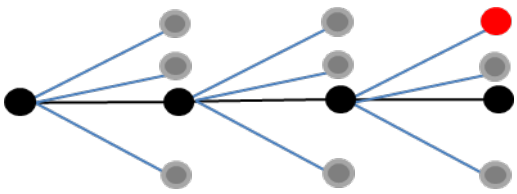


Figure 40: Impact of future extremes in the SDDP method.

Now we turn to the SFS method in the similar case, see Figure 41. If the model cannot meet the reservoir constraint without adding artificial water in stage 4, that will impact the first-stage decision, e.g., through an average cut defining the coupling between the first- and second-stage. However, in this case the second-stage scenario leading to the empty reservoir is a deterministic from time-stage 2 to 4. With perfect insight from time-stage 2 and onwards, it is easier for the model to optimize its way out of trouble in the final time-stage. For this reason, one may argue that it is less likely that the SFS first-stage decision will capture the impact of far-ahead extremes; and, since the SFS is a simulator, the first-stage decisions are not revisited. On the other hand, if such extremes are encountered by any of the second-stage scenarios, this information will be passed 'directly' to the first-stage decision (black dot in Figure 41). Conversely, in SDDP the information will be diluted (by multiplication of probabilities) and may easier 'disappear' in the backward recursion. Obviously, such effects are system dependent, and extremely difficult to generalize.

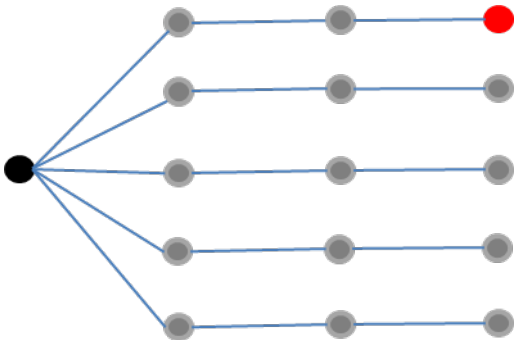


Figure 41: Impact of future extremes in the SFS.

4.3 Serial simulation

So far we have discussed use of the two methods in a setting where one starts at a known system state to schedule the system. This type of scheduling is referred to as running in 'parallel mode', and is normally used in scheduling where knowledge about the current system state is important.

Alternatively, one can arrange decisions in sequence according to the historical inflow records. This type of scheduling is referred to as running in 'serial mode', and is traditionally applied for expansion planning and systems studies which should not depend on the current system state.

The model to be developed in this project is primarily meant for expansion planning and system studies, and thus should be designed for running in serial mode.

4.3.1 SDDP in serial mode

SINTEF has experience with running the SDDP method in serial mode, and this functionality is implemented in ProdRisk. The concept is illustrated in Figure 42, with 3 scenarios in the forward simulation and 3 openings in the backward pass. The system is simulated for 2 years with 4 time-stages in each year. The initial reservoir levels are denoted Vs(1), Vs(2) and Vs(3) in Figure 42. Say that reservoir levels V_1 and V_2 are obtained in a forward simulation *i* for scenarios 1 and 2, correspondingly. In the next forward iteration, the initial reservoir levels are updated, so that Vs(2) = V1 and Vs(3) = V2.

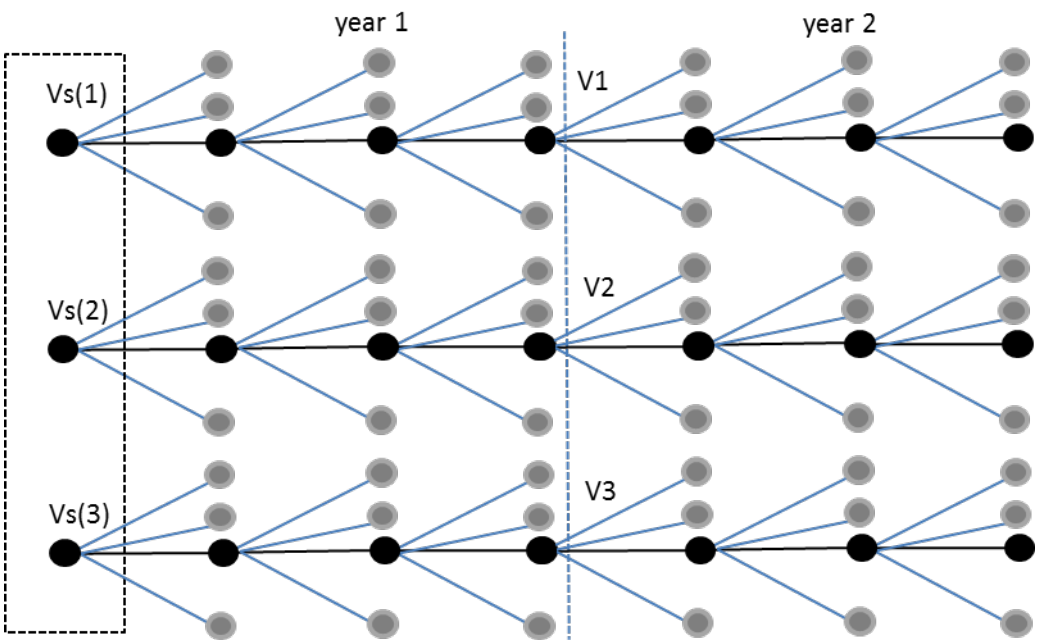


Figure 42: Illustration of serial mode in the SDDP method.

This scheme takes advantage of the iterative SDDP scheme to update starting points. Thus, only minor changes are needed to the algorithm presented in 3.1 to run in serial mode.

The serial-mode scheme discussed above will not possess the convergence properties of the original SDDP scheme, due to the iterative starting point.

4.3.2 SFS in serial mode

Unlike the SDDP method, the SFS is not iterative in the sense that the algorithm does not step backwards in time to re-evaluate previous decisions. Thus the serial-mode SDDP scheme in Figure 42 cannot easily be adapted to use with the SFS.

The SFS can be designed to simulate a historical period in sequence, e.g., 1950 – 2000. However, being used this way, one undermines the SFS's ability to parallelize on scenario.

4.4 Computational performance

This report documents test results that can be used to estimate expected computation times for solving the Nordic market problem using the two methods. This was done in section 2.8 for the SFS. Computation times for the SDDP method are reported in section 3.7, but for a much smaller problem. A thorough comparison of the expected computation time two methods will depend on numerous assumptions, and will not be undertaken here.

As pointed out initially, both methods essentially use the same building block; the weekly decision problem stated in section 1.3. An interesting measure is therefore how many times each of the two methods need to solve the weekly decision problem. Note that the SFS is not actually solving the individual weekly decision problem, but the second-stage scenarios can be decomposed to such. One should therefore consider the measure for the 'functional' evaluation of a least common multiple.

Consider an example with a planning period of 156 weeks and 50 historical inflow records.

a) SDDP: For the SDDP method we assume 200 main iterations and 50 openings in the backward pass. The weekly decision problem is solved with cut relaxation, we assume on average 3 iterations in the relaxation loop. The number of weekly decision problems to be solved is therefore $2.4*10^8$.
b) SFS: For the SFS method we assume 50 second-stage scenarios, and 20 iterations in the Benders decomposition between the first- and second-stage when solving each SFP. The number of weekly decision problems to be solved is $1.21*10^9$.

For this particular problem, the SFS needs to evaluate a factor of 5 more weekly problems than the SDDP. Obviously, we need to make many assumptions to arrive at the numbers above, and the final answers are sensitive to these assumptions.

The numbers indicate that the SFS method is computationally more demanding than the SDDP method. An important difference between the two methods is that the SDDP method maintains a set of linear constraints (cuts) to each weekly decision problem which is initially relaxed. In the SFS case, practically all weekly decision problems are associated with the second-stage evaluations, and thus no additional constraints are needed. Furthermore, the SFS seems to be extremely well suited for exploiting the computational benefits of warm starting scenario problems.

If run with price as a stochastic parameter in addition to inflow, one incorporates the prices in the historical scenarios used in the SFS without increasing computation time. In the SDDP case, price could be treated as a discrete state as in ProdRisk[23], but this will significantly increase computation time.

However, we point to some important differences between the two

- Serial simulation
- Price dimension
- Uncertainties, possible compromises

4.5 Representation of stochastic variables

The testing of the SDDP implementation in ProdRisk revealed that both the PCA and residual models show too little probability of prolonged dry or wet periods compared to the observed series. This is in line with previous testing of SDDP implementations used on systems with many and geographically widespread inflow time-series.

Although there is room for improvement in the statistical inflow model used in ProdRisk, the improvement potential seems to be limited by two important factors. First, the convexity requirement of the SDDP algorithm significantly limits the ability to incorporate accurate statistical models to be used in the backward pass. As an example, there are several previous studies pointing to the appropriateness of working with the logarithm of the inflows before normalizing (as elaborated in section 3.4) to avoid negative inflows. However, it can be shown that this transformation does not preserve convexity, and cannot be used in the SDDP scheme. Second, establishing a statistical model that captures all statistical moments represented in the historical time-series is a very complex task. SINTEF has an on-going project together with the Norwegian Computing Centre (NR) and model users aiming at making a statistical inflow model with as good properties at possible. Status so far is that the project has done most of the work in the univariate case, i.e. for one inflow series independent of others, and made a statistical model for inflow that is better than the model used in the SDDP algorithm. The new model has very good statistical properties for sum inflow up to about 13 weeks ahead but becomes poorer for summation of longer forecasting periods. The proposed model includes a logarithmic transformation that cannot be used in SDDP model. The multivariate (several inflow series) version of the new model is less improved compared to existing SDDP model. It is believed that further improvements of the model should include one or more states that describe, e.g., snow reservoir and or accumulate inflow so far for a given period.

In contrast, the intention with the SFS is to use historical inflows 'directly' when constructing the second-stage scenarios. In case there are other stochastic variables, e.g., wind power, temperature-dependent demand and/or power prices, these will be organized in scenarios along with the inflow, and will not increase the problem size and computation time. There may be a need for reducing the number of second-stage scenarios to decrease computation times. This can be done by grouping scenarios with similar properties, or by drawing a reduced set of scenarios to use.

4.6 Potential for including additional modelling details

When developing a new market model, one should also try to foresee future possible needs and extensions to meet future modelling challenges. Some examples are discussed below.

<u>Modelling nonlinearities and binary variables:</u>
Some relations are difficult to describe using linear functions and continuous variables. Examples are head dependencies in hydropower generators (non-linear), start-stop costs on production units (binary) and AC power flows (both non-linear and binary). Current long- and medium-term models at SINTEF have addressed these modelling challenges by adopting iterative schemes (e.g., head dependencies) or linearizing functions (start-stop costs and power flows). Such techniques can also be included in the new market model being designed in this project. However, one may in the future want to take a step further; to capture more details by modelling non-linear or binary relations.

In SDDP this can be done in its simplest form by ignoring these relations when establishing the strategy (backward recursion) and including them in the simulation. This will add more details to the simulation, but will also lead to a larger cost gap between strategy and simulation, and will therefore give a less robust optimization scheme. Similarly, the first-stage decisions in the SFS may include nonlinearities and binary variables, and the single SFP will still converge according to standard theory of Benders decomposition. Also in the SFS one will have a mismatch between details seen in the 'simulation' (first-stage) and in the 'strategy' (second-stage), giving a less robust scheme.

Modelling state-dependent constraints:
State dependent constraints are often challenging to handle in an SDDP-scheme. Some examples of state-dependent relations are:

- maximum/minimum discharge depending on reservoir level or inflow
- maximum discharge depending on reservoir head
- latencies in water courses
- start-stop costs on production units
- ramping (reservoirs, discharge, flows on cables, etc.).

The simplest ones to handle are those describing linear relationships (i.e., the three last ones in the list above), but still some programming effort is needed to extend the cuts (state space) and update the treatment of these.
Linear state-dependent constraints are more easily handled in the SFS method, where one only needs to consider one coupling point between stages.

The nonlinear relationships (i.e., the first two types in the list above) are challenging to incorporate in both methodologies, but should be pointed to as a topic for future research, as they are increasingly important in the scheduling problem.

4.7 Possible new alternative

As a result of the discussions and experience in this project we have started to discuss a possible third new alternative market modelling approach. The proposed model concept is based on an iterative approach and consists of two main parts:

A: A market clearing simulator that calculates the balance between supply and demand for given marginal costs. Marginal costs for water are described by cuts that are dependent on market price.

B: A local hydro scheduling model that calculates individual water values for a given exogenous stochastic market price, i.e., exactly the problem ProdRisk is used to solve when applied locally by producers.

B is solved individually for each water course and price dependent cuts for the whole planning period are sent back to the market clearing algorithm.

A is also solved for the whole planning period and send simulated market prices back to each watercourse. The convergence criteria could be related to convergence of market price and or hydro production. A converged solution should in theory be the optimal solution.

Discussion

Pros

- Relative easy implementation
- Resembles how the system is operated, price forecasting and local scheduling.
- Applies a local hydro scheduling model that is shown to work well

Cons

- Uncertain about convergence properties of the method
    - Know the method has been used for deterministic unit commitment
- This method is most likely not as robust, (e.g. measured in terms of predicting the change in economy for a system change) as the other methods.

We believe this is good idea that should be pursued, but we also believe that there is much more uncertainty about whether his method will give a usable model by the end of the project period. We would therefore suggest this method to be more appropriate for a PhD or KPN-type research project.

5 Conclusion

Before the project started the following two methods was considered as the main contenders for implementation in the SOVN project.

- Based on stochastic dual dynamic programming (SDDP)
- Based on simulation of a stochastic two stage decision problem represented by a scenario tree called the Scenario Fan Simulator (SFS)

We had experience using both of these methods, but no experience with application to the problem size that are going be solved in the SOVN project. To apply the different methods to the large scale problem give different challenges and unknowns.

- How well does the SDDP method work when applied to a large market modelling problem?
- How is the convergence of the Benders decomposition, used within the SFS, for realistic problem sizes?
- What will be the computation time for the SFS method for a real problem size?

Before a final decision about SOVN implementation is also useful to have thorough literature review about possible new methods.

The report documents work that has been done to answer these questions and thereby support the decision about which method to implement in the SOVN project. The main conclusions from this work are the following:

- There is not described any new alternative methods in the literature that change our initial opinion about possible solution methods. Actually, it is not possible to find anything new about applications to the problem size we are trying to solve in this project. The literature report better scaling for parallel processing versions of the SDDP algorithm than we have achieved.
- Testing of the SDDP method, represented by the ProdRisk model, has confirmed earlier results from another SDDP-based model (Samplan) that show relatively poor simulation results especially for extreme inflow scenarios. We have so far not been able to verify that this is only caused by the statistical properties of the inflow model in the SDDP approach.
- Testing on real problem sizes has shown that the SFS approach is solvable and that the Benders decomposition approach, that has to be used, converges with respect to the objective function value. Simulated reservoir filling does not converge for all reservoirs.

Our main conclusion is therefore that the SOVN project should continue with implementation based on the SFS approach.

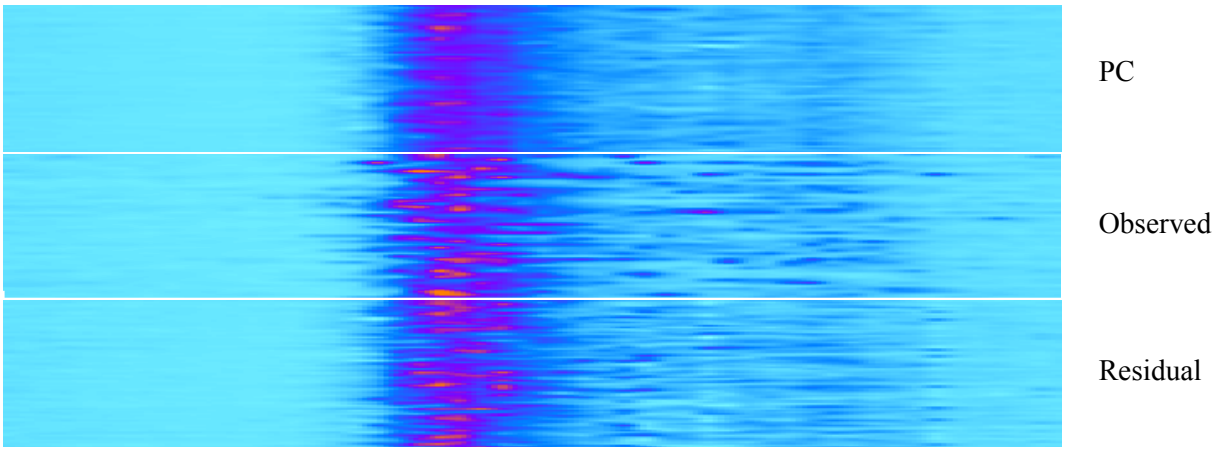
6 References

[1] S. Stage and Y. Larsson, "Incremental cost of water power," *Transactions of the American Institute of Electrical Engineers*, vol. 80, no. 3, pp. 361–364, 1961.
[2] J. R. Birge and F. Loveaux, *Introduction to Stochastic Programming*, 2nd ed. Springer, 2011.
[3] A. Gjelsvik, T. A. Røtting, and J. Røynstrand, "Long-term scheduling of hydro-thermal power systems," in *Hydropower 92*, E. Brock and D. Lysne, Eds., Lillehammer, 1992, pp. 539–546.
[4] B. Mo and A. Gjelsvik, "Sammenligning av metoder fro sesongplanlegging, kopling til kortidsplanlegging," SINTEF Energy Research, Tech. Rep. TR A4610, 1997.
[5] A. Helseth, B. Mo, and G.Warland, "Long-term scheduling of hydro-thermal power systems using scenario fans," *Energy Systems*, vol. 1, no. 4, pp. 377–91, 2010.
[6] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Athena Scientific, 2005.
[7] A. Gjelsvik, "Stochastic long-term optimization in hydroelectric power systems," SINTEF Energy Research, Tech. Rep. TR A2669, 1981.
[8] J. W. Labadie, "Optimal operation of multireservoir systems: State-of-the-art review," *Journal of Water Resources Planning and Management*, vol. 130, no. 2, pp. 93–111, 2004.
[9] R. Philbrick and P. Kitanidis, "Limitations of deterministic optimization applied to reservoir operations," *Journal of Water Resources Planning and Management*, vol. 125, no. 3, p. 135–142, 1999.
[10] L. Martinez and S. Soares, "Comparison between closed-loop and partial open-loop feedback control policies in long term hydrothermal scheduling," *IEEE Transactions on Power Systems*, vol. 17, pp. 330–336, 2002.
[11] M. Zambelli, A. E. Toscano, S. Soares, and E. dos Santos, "Newave versus Odin: Comparison of stochastic and deterministic models for the long term hydropower scheduling of the interconnected Brazilian system," *Revista Controle & Automação*, vol. 22, no. 6, pp. 598–609, 2011.
[12] K. Nolde, M. Uhr, and M. Morari, "Medium term scheduling of a hydro-thermal system using stochastic model predictive control," *Automatica*, vol. 44, p. 1585–1594, 2008.
[13] M. V. F. Pereira, "Optimal stochastic operations scheduling of large hydroelectric systems," *Electrical Power & Energy Systems*, vol. 11, no. 3, pp. 161–169, 1989.
[14] M. V. F. Pereira and L. M. V. G. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Mathematical Programming*, vol. 52, pp. 359–375, 1991.
[15] A. Gjelsvik, B. Mo, and A. Haugstad, *Handbook of Power Systems I*. Springer, 2010, ch. Long- and medium-term operations planning and stochastic modelling in hydro-dominated power systems based on stochastic dual dynamic programming, pp. 33–55.
[16] A. Helseth, A. Gjeslvik, B. Mo, and U. Linnet, "A model for optimal scheduling of hydro thermal systems including pumped-storage and wind power," *IET Generation, Transmission & Distribution*, vol. 7, no. 12, pp. 1426 – 1434, 2013.
[17] G. Infanger and D. P. Morton, "Cut sharing for multistage stochastic linear programs with interstage dependency," *Mathematical Programming*, vol. 75, no. 2, pp. 241–256, 1996.
[18] A. Philpott and V. L. de Matos, "Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion," *European Journal of Operational Research*, vol. 218, p. 470–483, 2012.
[19] S. Granville, G. C. Oliveira, L. M. Thomé, N. Campodónico, M. Latorre, M. Pereira, and L. A. Barroso, "Stochastic optimization of transmission constrained and large scale hydrothermal systems in a competitive framework," in *Proc. IEEE General Meeting*, Toronto, Canada, 2003.
[20] A. Tilmant and R. Kelman, "A stochastic approach to analyze trade-offs and risks associated with large-scale water resources systems," *Water Resources Research*, vol. 43, 2007.
[21] R. J. Pinto, C. L. T. Borges, and M. E. P. Maceira, "An efficient parallel algorithm for large scale hydrothermal system operation planning," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4888 – 4896, 2013.

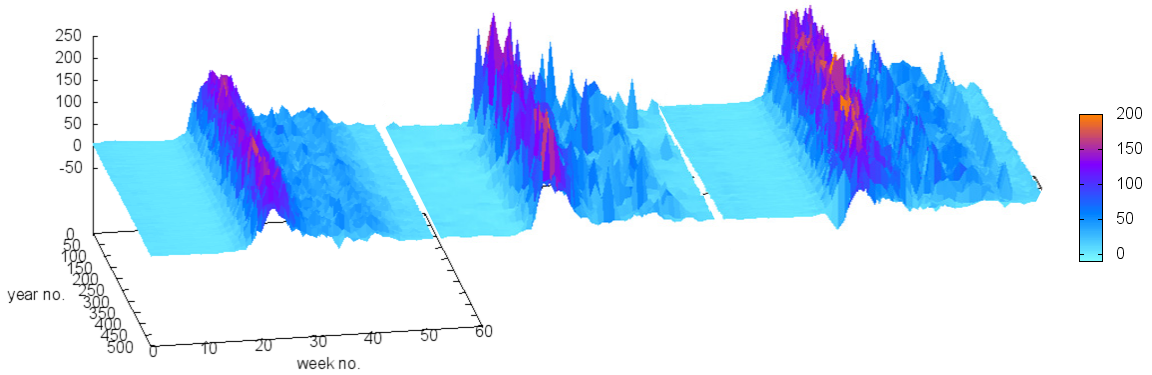
[22] D. D. J. Penna, M. E. P. Maceira, and J. M. Damázio, "Selective sampling applied to long-term hydrothermal generation planning," in *Proc. 17th Power System Computation Conference*, Stockholm, Sweden, 2011.
[23] A. Gjelsvik, M. M. Belsnes, and A. Haugstad, "An algorithm for stochastic medium-term hydrothermal scheduling under spot price uncertainty," in *Proc. 13th Power System Computation Conference*, Trondheim, Norway, 1999.
[24] A. Philpott and Z. Guan, "On the convergence of stochastic dual dynamic programming and related methods," *Operations Research Letters*, vol. 36, no. 4, pp. 450–455, 2008.
[25] A. Haugstad and et.al., "En samkjøringsmodell basert på stokastisk dual dynamisk programmering," SINTEF Energy Research, Tech. Rep. TR A 5496, 2001.
[26] A. Gjelsvik and I. Honve, "Tilsigsmodellering for SDDP-liknende modeller i driftsplanlegging i vasskraftverk," SINTEF Energy Research, Tech. Rep. TR A6259, 2005.
[27] O. Wolfgang and et.al., "Modelling quoatas and el-certificates in market simulation models," SINTEF Energy Research, Tech. Rep. TR F6260, 2005.
[28] S. Aam, "Undersøkelser av varians i resultater fra kraftverdiberegninger og driftssimuleringer for kombinerte vann-varmekraftsystemer ved bruk av en stokastisk dynamisk tilsigsmodell." SINTEF Energy Research, Tech. Rep. TR A2173, 1974.

A SDDP and inflow modelling

A.1 Individual inflow series



Figur A.1.1: Image plot of observed and generated data for Møsvatn with 12 openings. Only a tenth of the generated data is shown to make the figures comparable in size and scale. On the horizontal axis are weeks from 1 to 52 and on the vertical axis are individual years (50 observed years).



Figur A.1.2: Surface plot of the observed and generated series. PC-model to the left, observed data in the middle and residual model on the right. Note that there are only 50 observed years, so the middle sub plot contains less data in that dimension. The figure is meant to illustrate the large trends of the data, not individual points.

A.2 Energy inflow

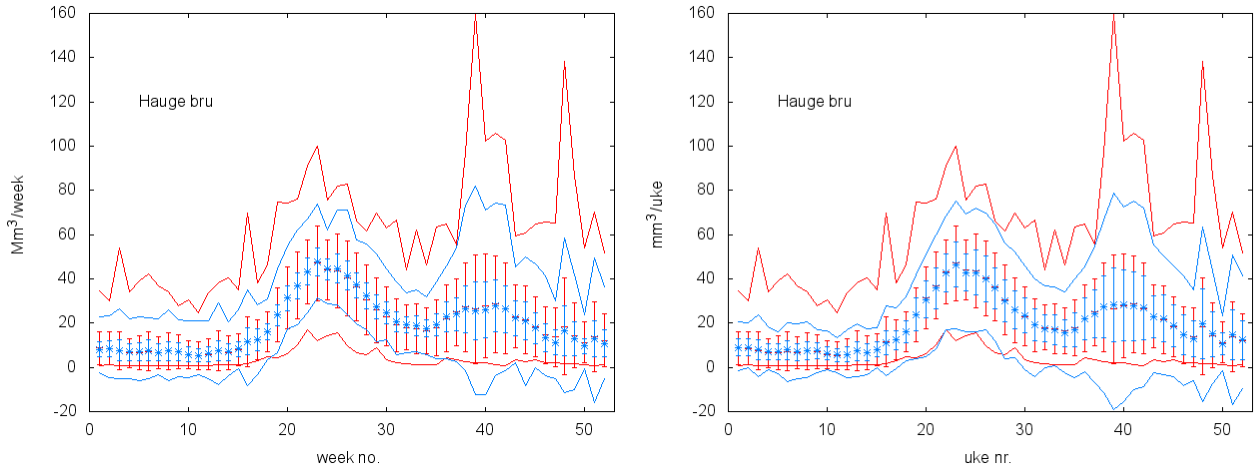


Figure A.2.1: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Hauge bru. Observed data are in red and generated data from the PC-model in blue. Left figure: 12 openings. Right figure: 1500 openings.

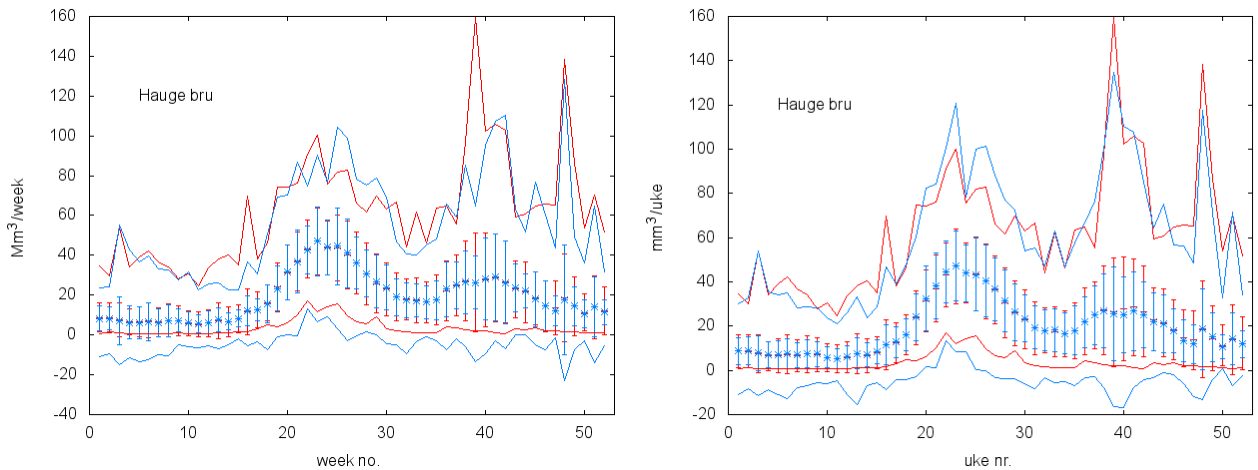


Figure A.2.2: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Hauge bru. Observed data are in red and generated data from the residual model in blue. Left figure: 12 openings. Right figure: 1500 openings.

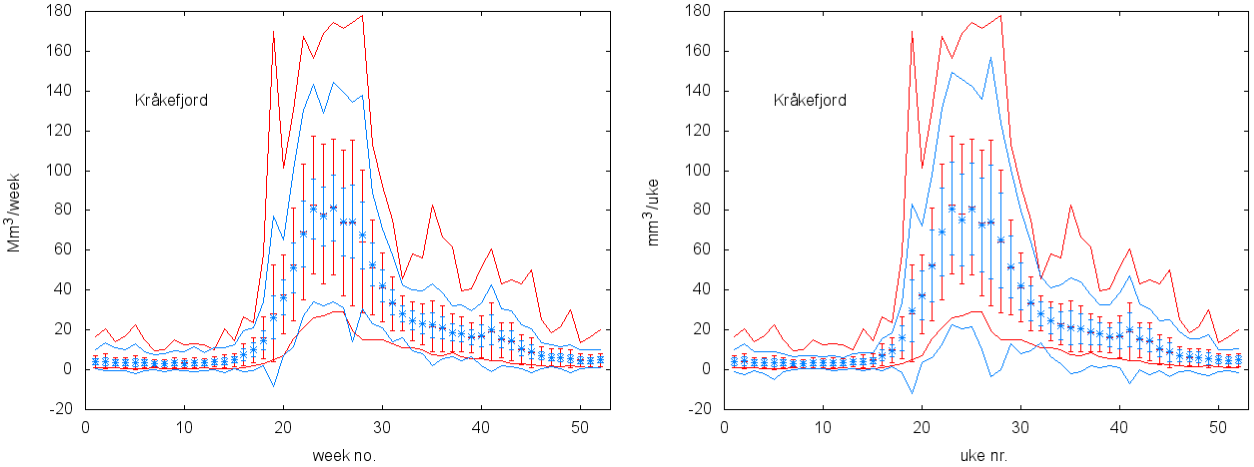


Figure A.2.3: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Kråkefjord. Observed data are in red and generated data from the PC-model in blue. Left figure: 12 openings. Right figure: 1500 openings.

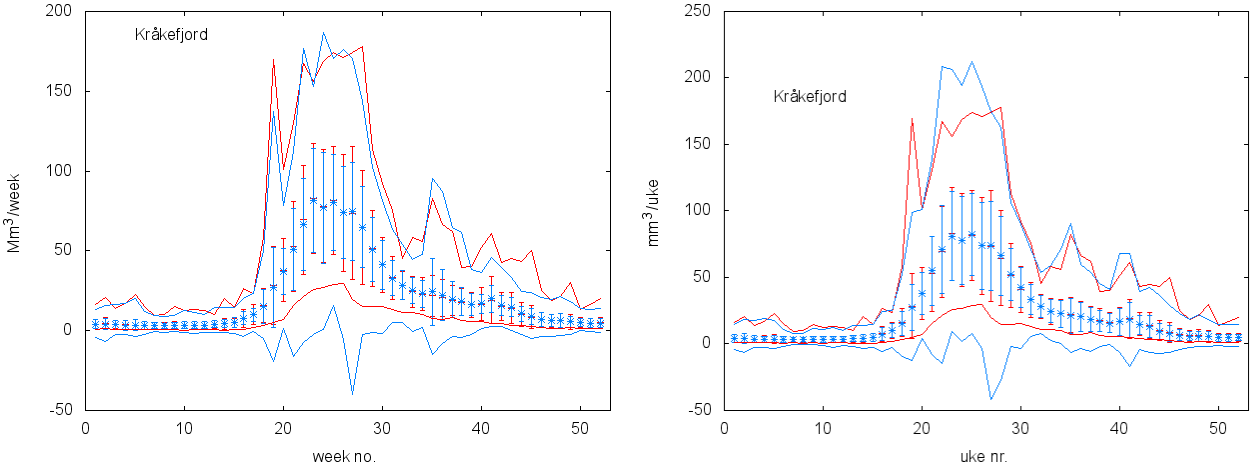


Figure A.2.4: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Kråkefjord. Observed data are in red and generated data from the residual model in blue. Left figure: 12 openings. Right figure: 1500 openings.

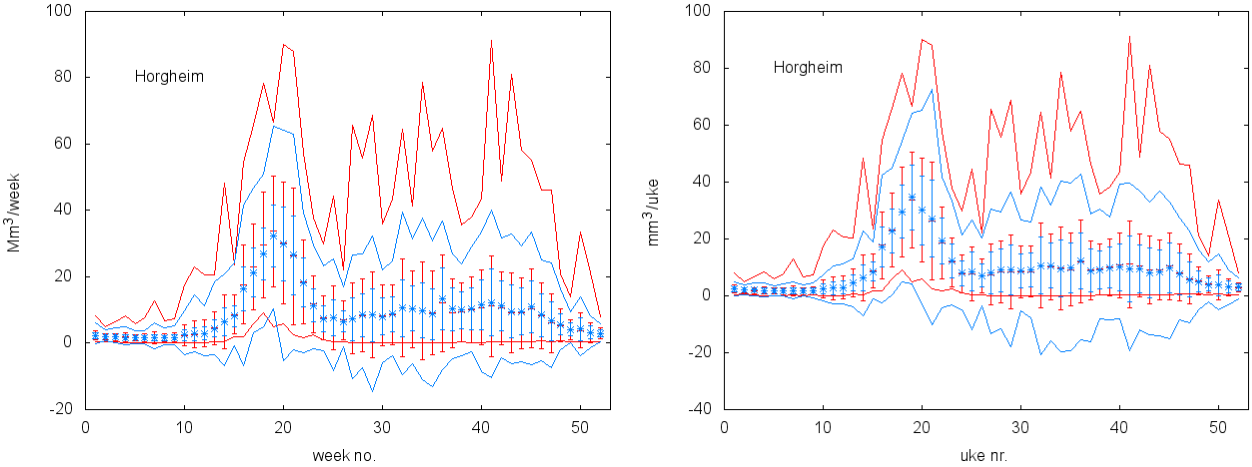


Figure A.2.5: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Horgheim. Observed data are in red and generated data from the PC-model in blue. Left figure: 12 openings. Right figure: 1500 openings.

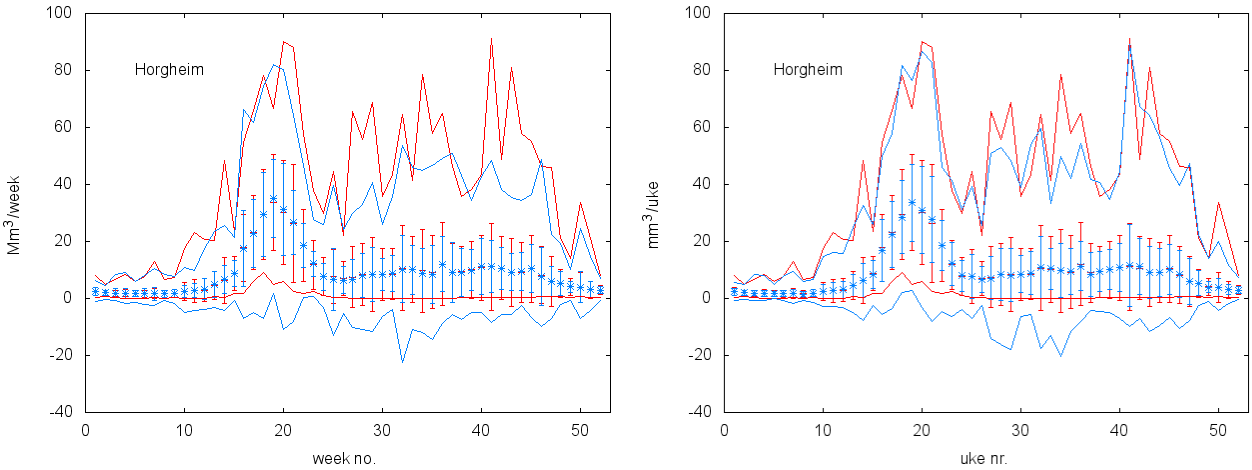


Figure A.2.6: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Horgheim. Observed data are in red and generated data from the residual model in blue. Left figure: 12 openings. Right figure: 1500 openings.

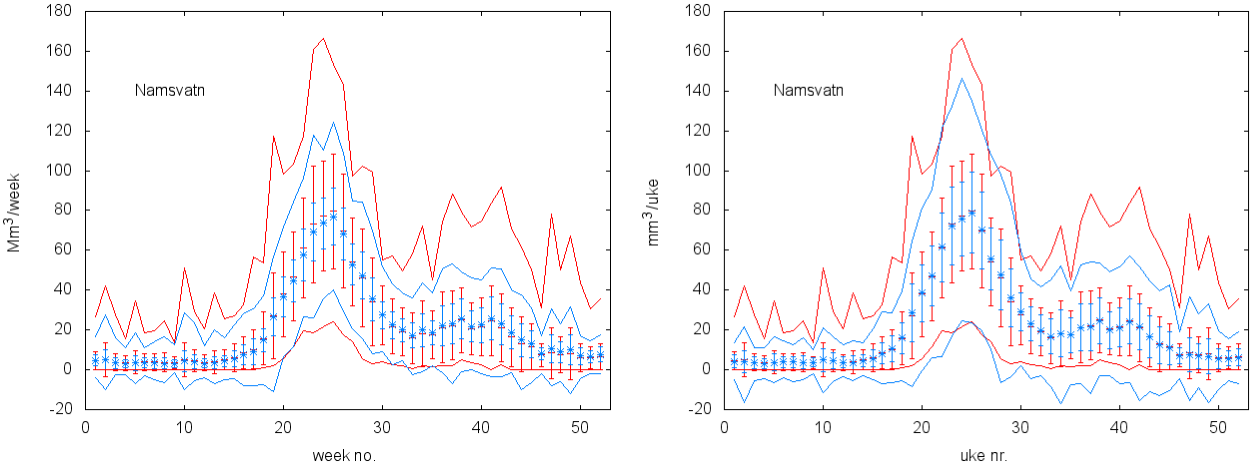


Figure A.2.7: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Namsvatn. Observed data are in red and generated data from the PC-model in blue. Left figure: 12 openings. Right figure: 1500 openings.

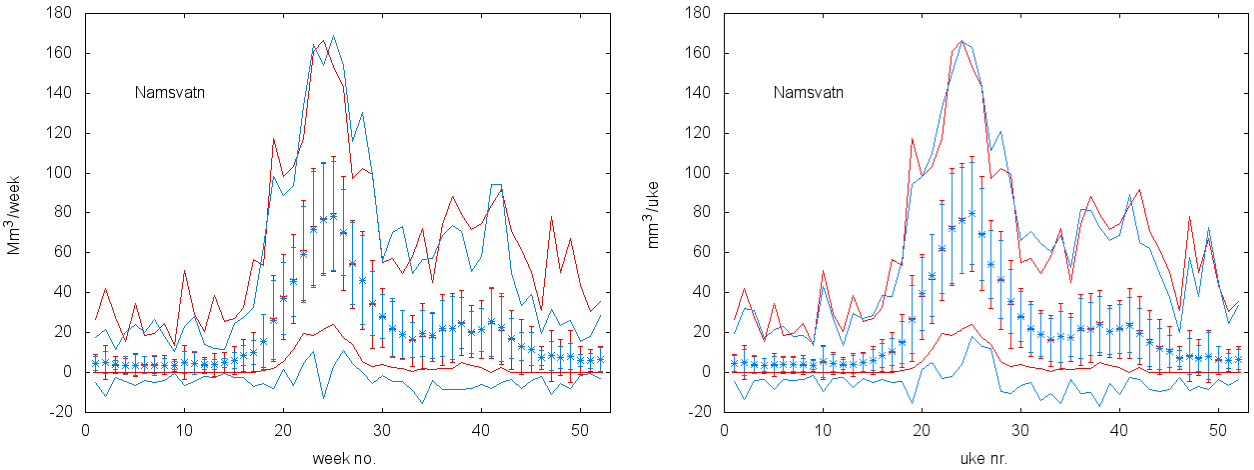


Figure A.2.8: Average values plus/minus estimated standard deviations with enclosing maximum value and minimum value curves for Namsvatn. Observed data are in red and generated data from the residual model in blue. Left figure: 12 openings. Right figure: 1500 openings.

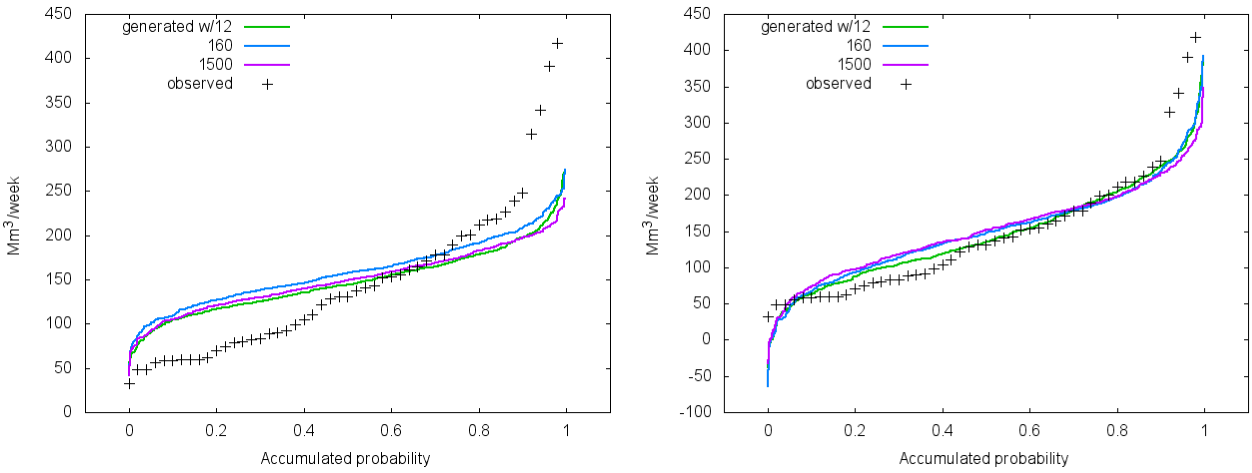


Figure A.2.9: Sum of inflow for Hauge bru from week number 1 to 17 shown as function of accumulated probability. Generated series from the PC-model to the left and the residual model to the right.

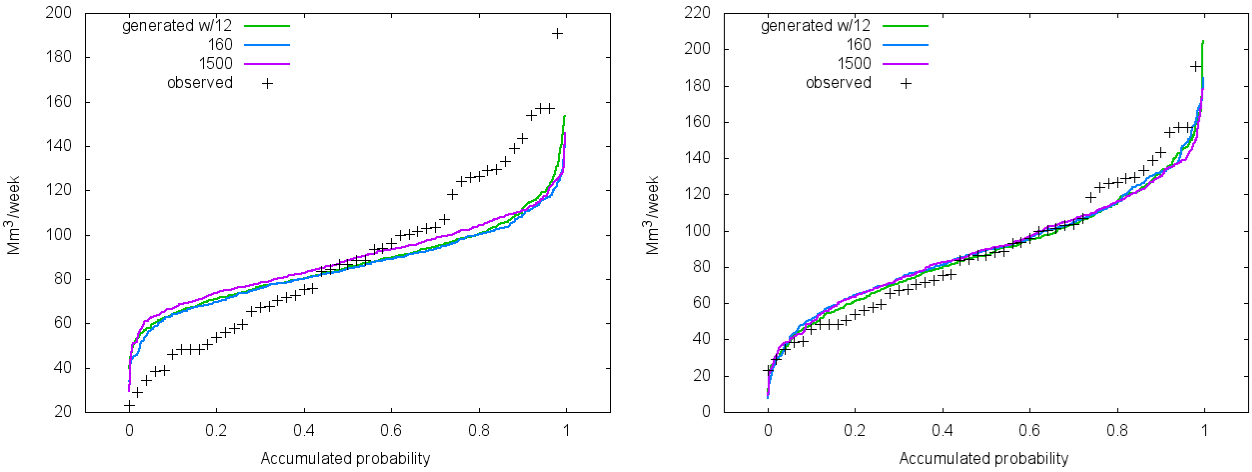


Figure A.2.10: Sum of inflow for Kråkefjord from week number 1 to 17 shown as function of accumulated probability. Generated series from the PC-model to the left and the residual model to the right.

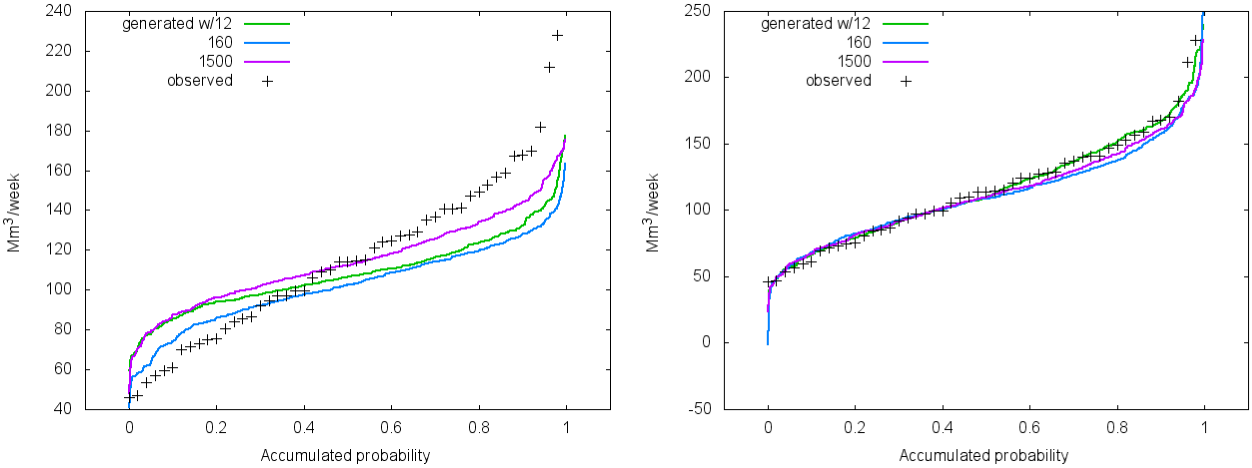


Figure A.2.11: Sum of inflow for Horgheim from week number 1 to 17 shown as function of accumulated probability. Generated series from the PC-model to the left and the residual model to the right.

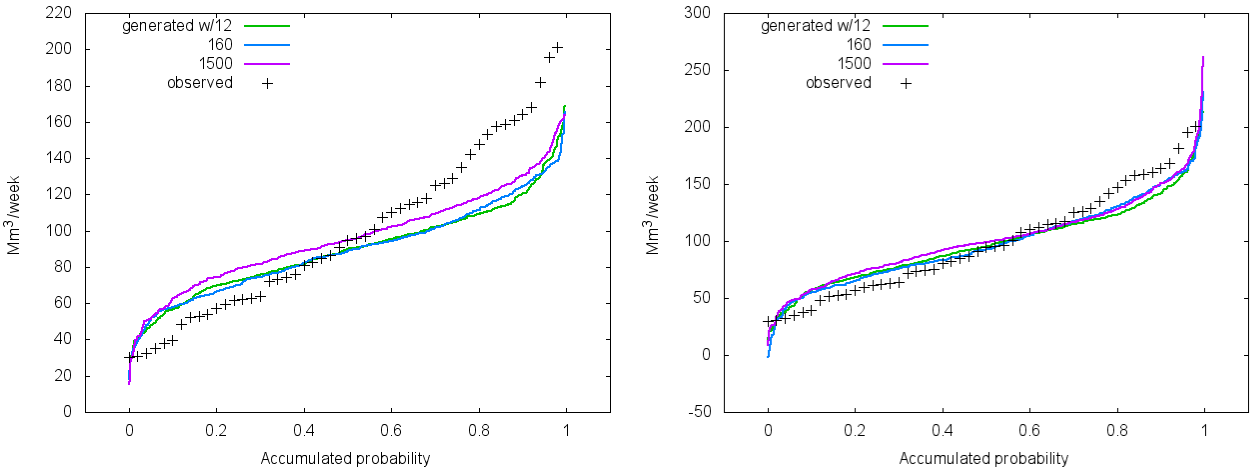


Figure A.2.12: Sum of inflow for Namsvatn from week number 1 to 17 shown as function of accumulated probability. Generated series from the PC-model to the left and the residual model to the right.

A.3 Vansimtap/Prodrisk comparison for artificial inflows

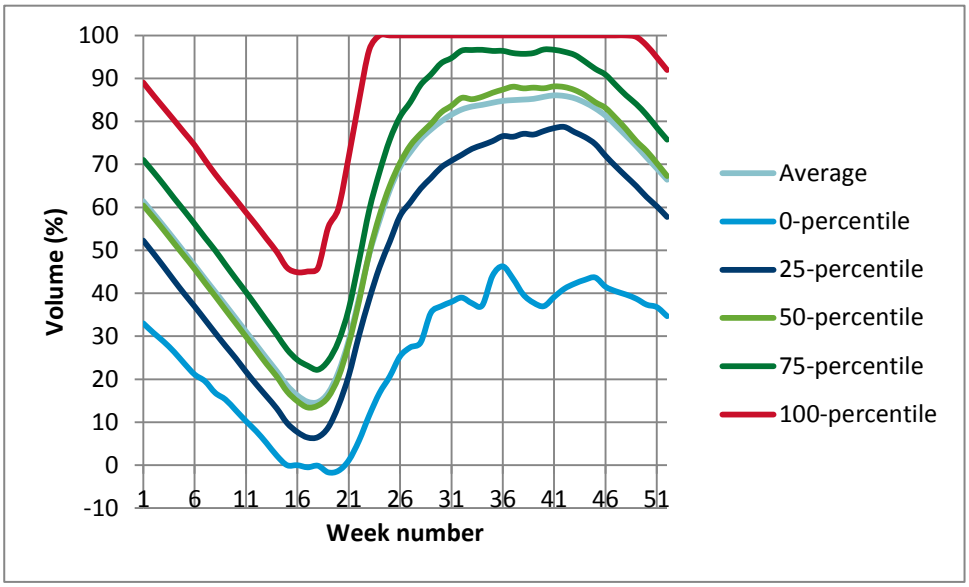


Figure A.3.1: Vansimtap, simulated volume in Møsvatn.

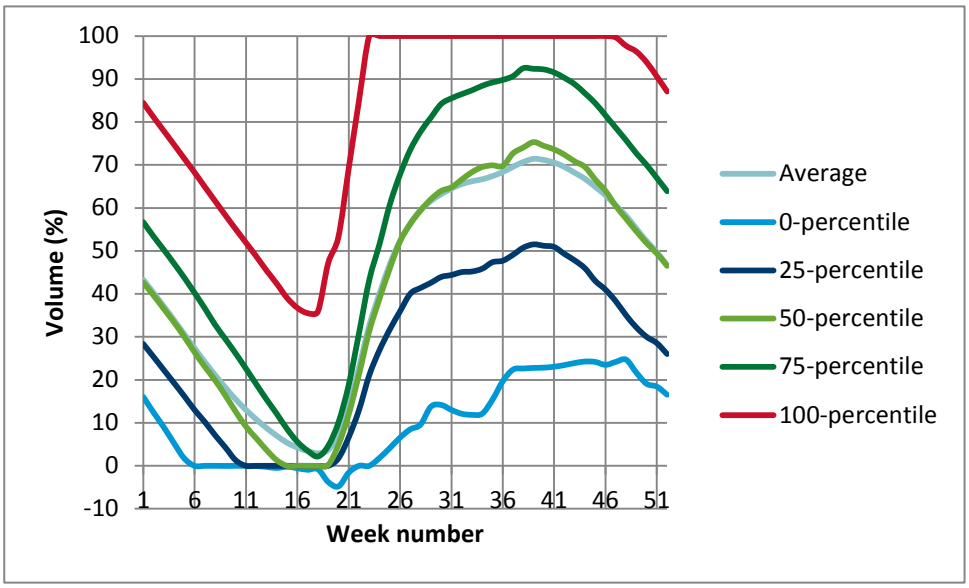


Figure A.3.2: ProdRisk, simulated volume in Møsvatn.

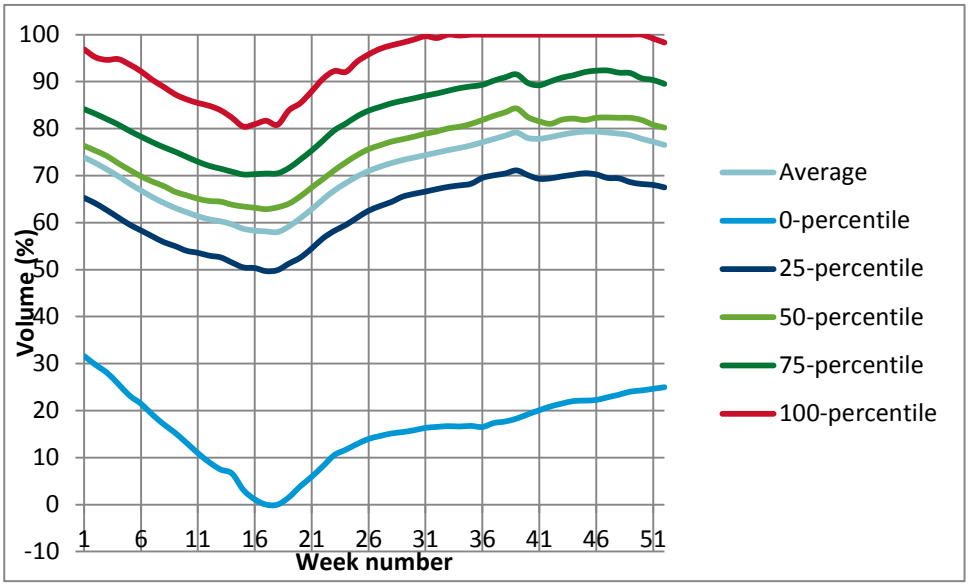


Figure A.3.3: Vansimtap, simulated volume in Svartevatn.

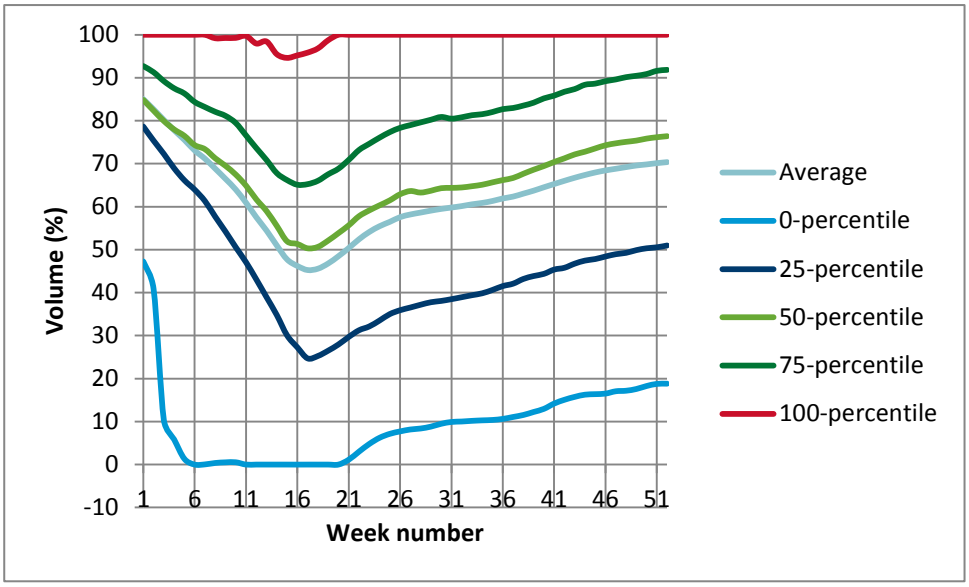


Figure A.3.4: ProdRisk, simulated volume in Svartevatn.

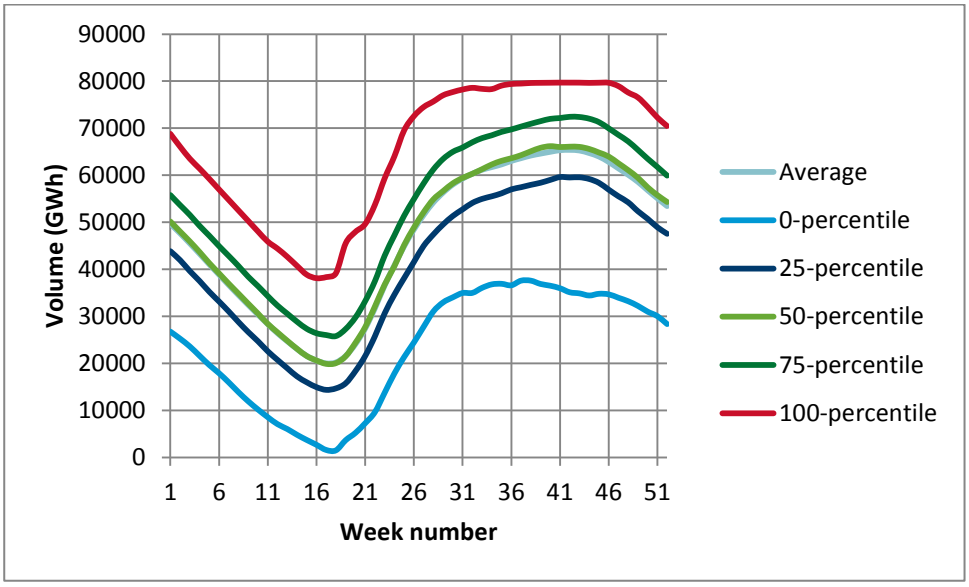


Figure A.3.5: Vansimtap, simulated sum storage.

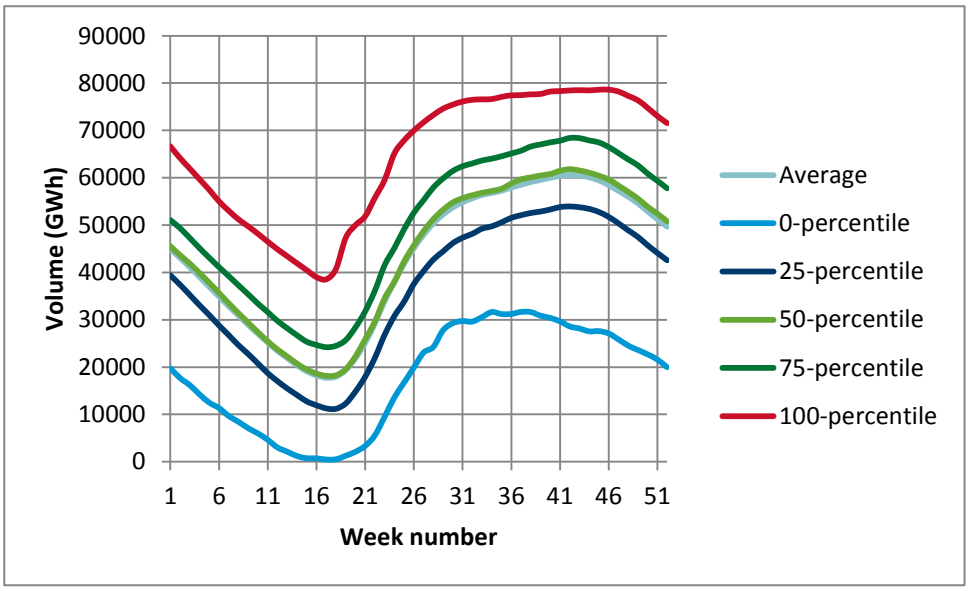
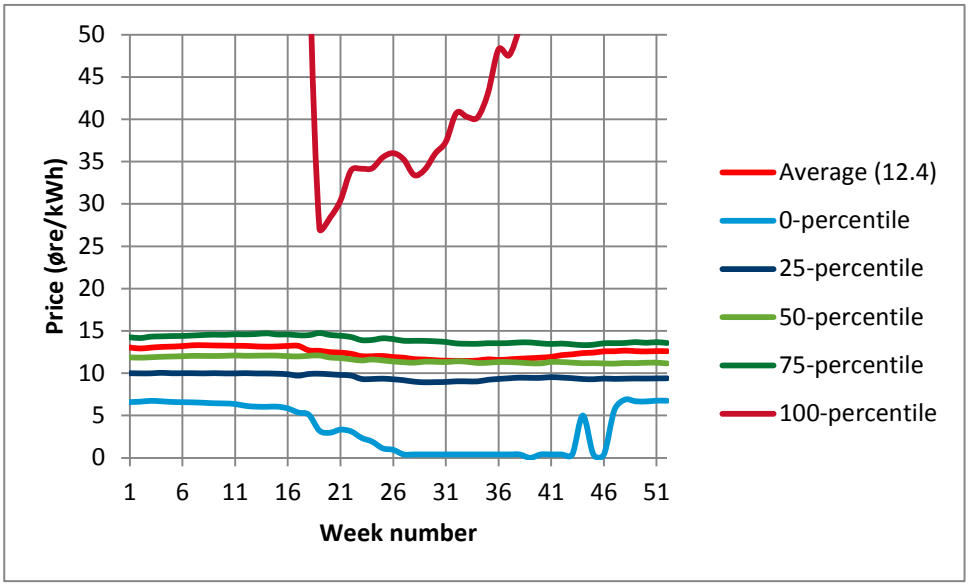
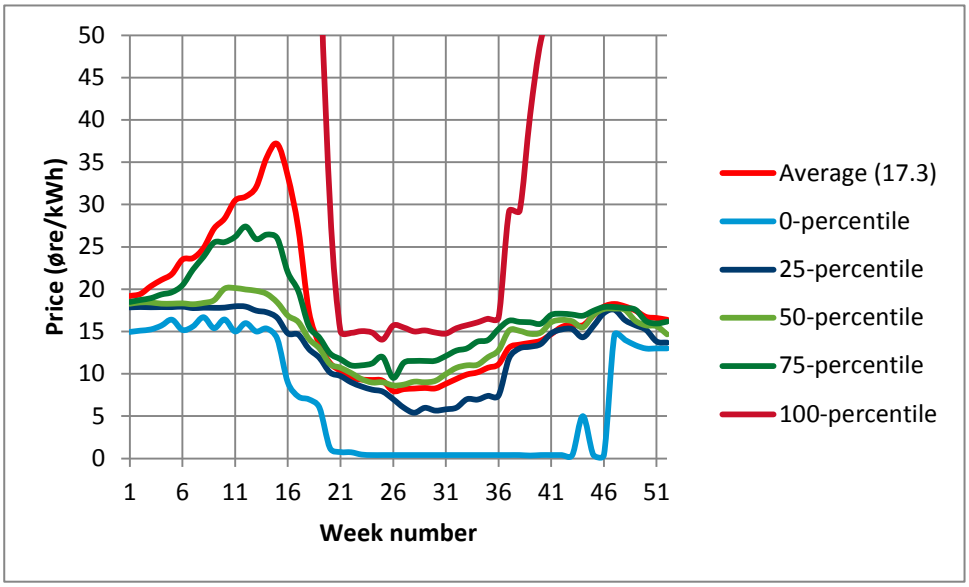


Figure A.3.5: ProdRisk, simulated sum storage.



A.3.6: Vansimtap, Simulated prices



A.3.7: ProdRisk, Simulated prices

SINTEF

Technology for a better society

www.sintef.no