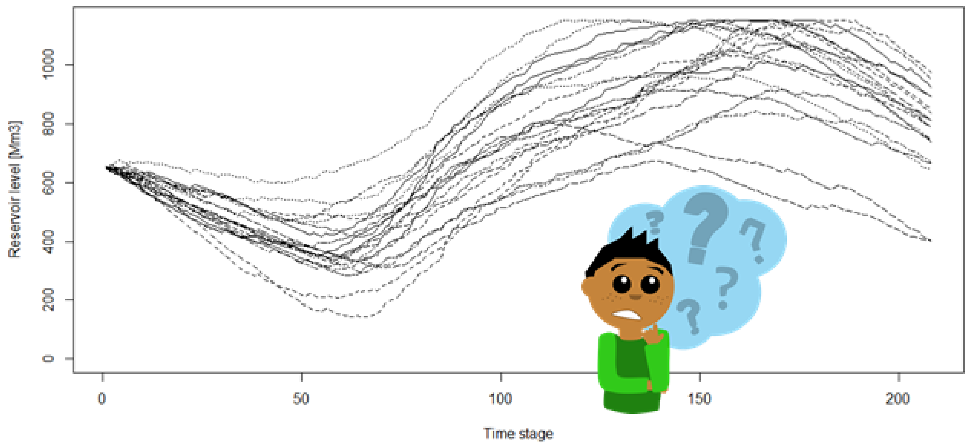
TR A7618- Unrestricted

Report

SOVN Model Implementation

Method, functionality and details

Author(s)
Arild Helseth
Birger Mo
Arild L Henden
Geir Warland



SINTEF Energy Research
Production Planning

2017-03-16

SINTEF

SINTEF Energi AS
SINTEF Energy Research
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

Report

SOVN Model Implementation

Method, functionality and details

KEYWORDS:
Hydro-thermal Scheduling, Power Market Modeling, Stochastic Optimization, Linear Programming

VERSION
1.0

DATE
2017-03-16

AUTHOR(S)
Arild Helseth
Birger Mo
Arild L Henden
Geir Warland

CLIENT(S)
Client(s)

CLIENT'S REF.
Client's reference

PROJECT NO.
225873

NUMBER OF PAGES/APPENDICES:
47 + Appendices

ABSTRACT
The SOVN project ("Stokastisk optimaliseringsmodell for Norden med individuelle vannverdier og nettrestriksjoner") aimed at creating a new fundamental hydro-thermal market model with detailed representation of the hydropower system. The model is based on a combination of optimization and simulation, and does not rely on aggregation of the hydropower system. It is suitable for both operational planning and expansion planning studies.

This report documents the theoretical foundation, implementation and functionality available in the SOVN model at the end of the project.

PREPARED BY
Arild Helseth

SIGNATURE
Arild Helseth

CHECKED BY
Michael Belsnes

SIGNATURE
Michael Belsnes

APPROVED BY
Knut Samdal

SIGNATURE
Knut Samdal

REPORT NO.	**ISBN**	**CLASSIFICATION**	**CLASSIFICATION THIS PAGE**
TR A7618	978-82-594-3680-1	Unrestricted	Unrestricted

SINTEF

Document history

VERSION	DATE	VERSION DESCRIPTION
1.0	2017-04-18	First version

Table of contents

APPENDICES

[List appendices here]

1 Introduction

This report describes a new model called SOVN, developed in the research project "Stokastisk optimaliseringsmodell for Norden med individuelle vannverdier og nettrestriksjoner". The project was funded by the Research Council of Norway, Statnett, Statkraft, BKK and NVE. The project goal was to develop a new fundamental market optimization and simulation model able to solve the hydro-thermal scheduling problem with detailed description of all relevant constraints, including constraints given by individual hydro storages and plants using a formal optimization method. New tools are needed since existing models, such as the EMPS model [1], include too many simplifications in important parts of the solutions procedure. Existing models therefore heavily rely on calibration in order to show the real value of e.g. pumped-storage plants in the future Nordic power system with more new renewables and stronger couplings to Europe.

The project started in 2013 and will end in early 2017. The first part of the project focused on choosing the most suited method. The presentation and evaluation of relevant methods and the selection criteria was documented in a technical report [1]. An important selection criteria was that the model will be used with historical records for inflow representing future uncertainty. The method of choice was a concept combining optimization and simulation, which we refer to as a *scenario fan simulator* (SFS). The SFS logic was documented in [13, 12]. In the remainder of this section we describe the basic SFS logic together with the basic structure of the repeated optimization problems and the decomposition technique used for solving those.

For practical use of the SOVN model, we refer to the user manual provided as a separate document. Some preliminary result obtained from the model was presented in [19]. Moreover, a research article documenting application of the SOVN model on Statnett's 2020 datasett has been submitted for review. This report will not deal with results obtained from the model.

1.1 Simulator Fan Simulator (SFS) Logic

For each time stage we solve a *scenario fan problem* (SFP) and pass the solution from the first-stage decision on to the next time stage. We start by describing the simulator logic, before going into the basic formulation of a SFP and how it can be decomposed to reduce computation time.

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

```
1: for all scenarios s from 1 to S do
2:    for all decision stages t from 1 to T do
3:       Build and solve the SFP problem SFP(s,t)
4:       Store results from first-week decision, sol(s,t)
5:       Pass on state decision from sol(s,t) to SFP(s,t+1)
```

The procedure is illustrated in Figure 1, where the SFP is built for a given scenario s_1 and for time-steps t_1 and t_2. The first problem, SFP(s_1,t_1), 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. 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 1. 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 decision 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.

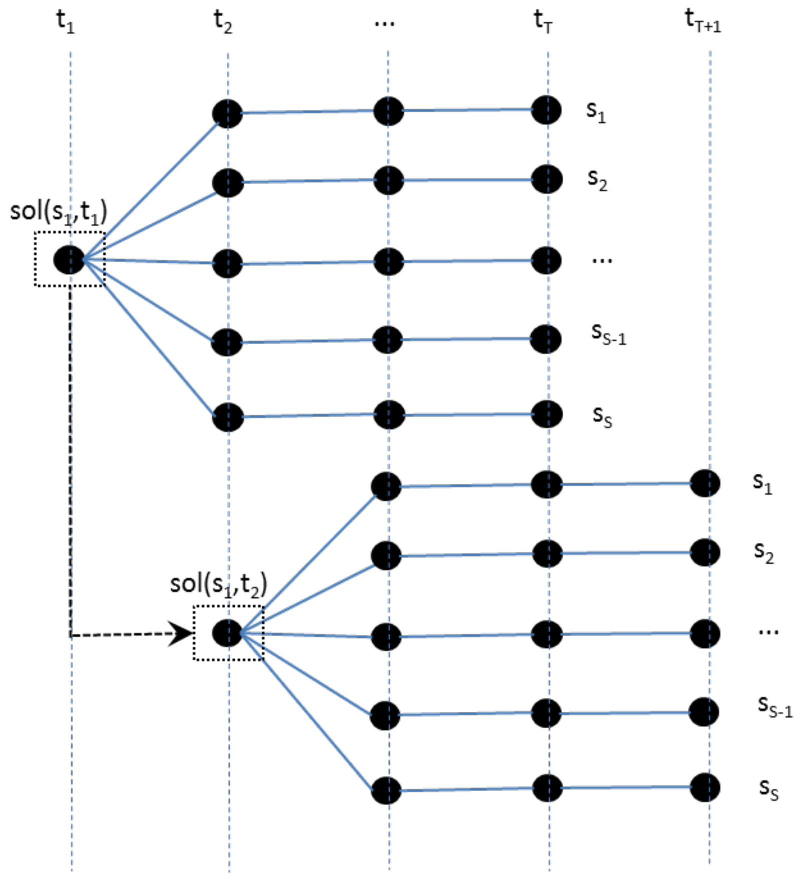


Figure 1 Illustration of SOVN logic.

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 1 and Figure 2, calls for a specific end-value setting for each simulated stage. These values can e.g. be obtained from the EMPS model.

1.2 The Scenario Fan Problem (SFP)

The first decision 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 generally be formulated as in (1).

$$Z = \min c_t^T x_t + \sum_{s=1}^{S} p_s \sum_{k=t+1}^{T+t} c_{k,s}^T x_{k,s} \quad \text{(1)}$$

$$A_t x_t = b_t \quad \text{(2)}$$

$$x_{k-1} + A_k x_{k,s} = b_{k,s} \; \forall s \in S \; \forall k \in K \quad \text{(3)}$$

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.

The shape of the SFP is illustrated in Figure 2, 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.

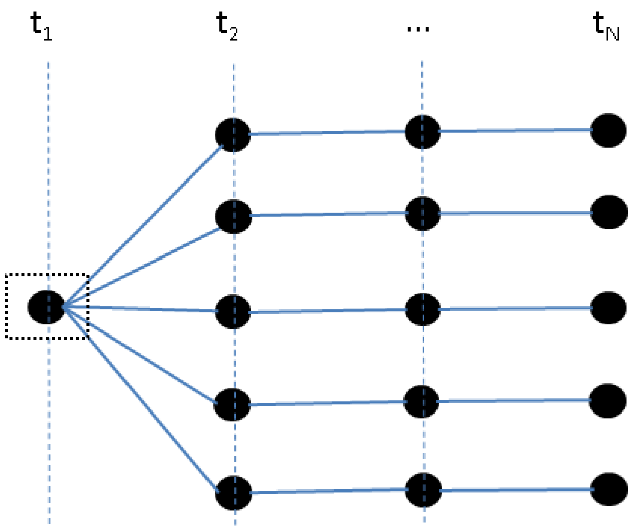


Figure 2 Illustration of scenario fan problem. Only the first-stage decision is used and stored.

1.3 Decomposition

Significant computational speed-up is obtained when decomposing the two-stage SLP rather than solving its extensive form in Equations (1)-(3). This problem may be decomposed by stage or by scenario; in this report we focus on the stage-wise decomposition. An example on scenario-based decomposition is presented in [11].

We create a first-stage problem[1] to represent the first-stage decision:

$$Z_{\text{first}} = \min c_1^T x_1 + \alpha \quad \textbf{(4)}$$

$$A_1 x_1 = b_{11} \quad \textbf{(5)}$$

$$\alpha + \pi^T x_1 \geq b_{12} \quad \textbf{(6)}$$

After solving the first-stage 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 right hand-side of the second-stage constraints as a trial solution:

$$Z_{\text{sub}}^s = \min c_{2,s}^T x_{2,s} \quad \textbf{(7)}$$

$$A_2 x_2 = b_2 - x_1 \ \leftarrow \pi_s \quad \textbf{(8)}$$

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 first-stage problem:

$$\pi = \sum_{s=1}^{S} p_s \pi_s \quad \textbf{(9)}$$

$$b_{12} = \sum_{s=1}^{S} p_s (Z_{\text{sub}}^s + \pi_s^T x_1) \quad \textbf{(10)}$$

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

$$Z_{\text{low}} = Z_{\text{first}} \quad \textbf{(11)}$$

The upper boundary will be:

$$Z_{\text{up}} = c_1^T x_1 + \sum_{s=1}^{S} p_s c_{2,s}^T x_{2,s} \quad \textbf{(12)}$$

[1] We use the term "first-stage problem" rather than "master problem" throughout this text to avoid   while master problem refers to the first week in the weekly decision problem where inflow is deterministic. To avoid this ambiguity, we use first-stage problem throughout this text.

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

$$Z_{\mathrm{up}}^{i} = \min\left(Z_{\mathrm{up}}^{i-1}, Z_{\mathrm{up}}^{i}\right) \qquad \textbf{(13)}$$

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

$$Z_{\mathrm{up}} - Z_{\mathrm{low}} \leq \varepsilon \qquad \textbf{(14)}$$

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 first-stage problem (one future-cost variable per scenario, and S cuts per iteration). This variant has not been tested in the SOVN model.

2 Hydro System Modelling

Hydro modelling in SOVN generally allows for the same level of detail as other long-term models (EMPS, EOPS, ProdRisk [10], etc.). The descriptions of the basic variables included in modules (reservoir + power station) are assumed known and is documented in the EOPS/EMPS user manuals. These are listed below:

Variables:

- Reservoir, including separate variables for minimum and maximum levels
- Spillage
- Bypass
- Pumping
- Discharge per PQ-curve segment

For most of the variables one can specify time-dependent minimum and maximum boundaries. Most of these boundaries are "soft" in the sense that they can be violated at a given penalty cost. The user has great flexibility in specifying penalties to prioritize which boundaries and constraints that should be met.

In the remainder of this section we describe solutions and choices that are special for SOVN. Relationships, variables and constraints not commented are (most likely) implemented in a similar fashion as in SINTEF's seasonal model ("Sesongmodellen").

2.1 PQ description and head dependency

The relation between production P (MW) and discharge Q (m^3/sec) is modelled by a piece-by-piece linear relation often called PQ-curve.

Head dependency deals with the relations between head (reservoir volumes), production and discharge. This relationship is generally non-linear. It is therefore challenging to accurately represent these relations in a stochastic optimization model of a general serial watercourse and simplifications are therefore needed. The work in [6] elaborates on how to represent head more accurately in LP-based scheduling models.

In the long-term models this relation is modelled with production (P) being proportional to the actual head h. The discharge is divided into NS segments with successively decreasing relative efficiency (η). Relative efficiency refers to a nominal head h_0.

$$P = \sum_{s=1}^{NS} \eta_s Q_s \frac{h}{h_0} \qquad \textbf{(15)}$$

The immediate impact of higher head is a higher production for a given discharge, as seen from (15). In SOVN head dependent production is included in the optimization as follows: In the first-stage problem, which gives the simulated production for the whole week, P is scaled to the actual head at the beginning of the week. Remember that the reservoir level at the beginning of the week is a (known) state variable, and the head can therefore be found directly from the reservoir curve ("magasinkurve").

For each week in the second-stage scenario problem, P is scaled to actual head in the middle of the week calculated for the same scenario when previous week's scenario fan problem was solved. This procedure does involve a slight increase in computation time, since the very first week needs to be solved twice.

So far, we have only accounted for the immediate impact of head on the production. Since head enters the optimization problem as a parameter and not a variable. If one produce more hydropower here and now, one also leaves less water and lower head for the future. This relation is included in SOVN using the same simplified method as described in [7, 8]. The method is briefly summarized by the following. For a given time step, reservoir volume, production and market price, the "head value" of having one additional Mm^3 stored in the reservoir is calculated. This "head value" (a coefficient) is added to the objective function as an additional cost of using water. The "head value" is calculated individually for all time steps along each scenario. The input values needed to calculate the "head value" are taken from the previous weeks solution from the same scenario and time step.

2.2 Hydraulic Couplings

SINTEF Energy's long-term models allow for simplified modelling of a limited number of predefined hydraulic couplings. In SOVN these couplings are all included in both the first-stage problem and along the second-stage scenarios. The modelling include direct flow between reservoirs Code 300 and 200 single plant connected to several reservoirs (code 100/120/130). All couplings are modelled within the LP programing framework which limits somewhat the functionality. Some important aspects are:

- There is no limitations or cost connected to change of flow or hatch openings.
- A single plant can discharge from several reservoirs at the same time, but max. discharge is limited.
- The model does not limit flows towards a higher altitude, this will occur if economically preferable

2.3 Head dependent maximum discharge

Maximum discharge is always used independent of head variations, even if such a description is entered by the user. This relationship is non-linear and often non-convex. Introducing a head-dependent maximum discharge rate in SOVN would call for efficient linearization techniques, e.g. as described in [6].

2.4 Reservoir constraints

2.4.1 Maximum Reservoir

Time dependent maximum reservoir constraints can be defined by user to be either soft or hard. Hard constraints are absolute. Soft constraints can be violated at a user specified cost (monetary unit/m3).

2.4.2 Minimum Reservoir

Time dependent minimum reservoir constraints can also be soft or hard. Hard minimum constraints can be violated at cost defined by the user. The user-specified cost is multiplied by the energy equivalent to sea, only if this is larger than 1.0, before used in the objective function.

The soft constraint is transformed to a hard constraint within the model in a pre-processing. The pre-processing uses assumed known initial reservoir filling by the beginning of the week, known inflow and take into account minimum flow constraints to give a new hard minimum constraint by the end of the week that can be achieved if the plant is not discharging. This method is used week by week to give scenario dependent new minimum both for the first-stage problem and for the different scenarios in the scenario fan. Along the scenarios "initial filling" is given by the minimum filling calculated for the previous week.

2.5 Ramping on discharge

Ramping constraints is implemented on discharge and limits the change in flow between time periods.

The constraint on discharge $f_{m,t}$ for a given module *m* and time step *t* will take the form of (16) below. Note that implemented constraint in addition also include slack variables to ensure that a feasible solution always is found.

$$-\Delta F_m^{max} \leq f_{m,t} - f_{m,t-1} \leq \Delta F_m^{max} \qquad \textbf{(16)}$$

Ramping constraints are in the current version of SOVN only allowed in the first-stage problem. Ramping constraints is not included for the first time step because previous discharge is not assumed given.

2.6 Time-delay

If the SOVN model is used with fine time-resolution, including water travelling times may significantly constrain hydro scheduling flexibility and give a more realistic description. SOVN includes functionality for time delays on all water ways (discharge, spillage and bypass) both in the first- and second-stage problems.

When delays are modelled, the reservoir balance equations need to take water in transit into account. This additional book keeping is challenging in a decomposition scheme, since water flows decided in the first-stage will arrive in their target reservoirs in the second-stage, creating a time coupling. This effect was included in the Benders cuts, as discussed in e.g. [5, 4], but the impact of increasing the size of the cuts has not been studied in detail. Some experiments were carried out omitting the cut coupling and assigning a numerical value for water in transit in the last time step in the first-stage problem. A conclusion from these experiments was that it is difficult to give at a valuation principle that lead to consistent results for large and complex systems. Therefore, time delays should be included in the cuts.

2.7 End-value setting

In principle, SOVN should be run with a second-stage (scenario) covering a period of time long enough to strongly limit the impact of the end-valuation of reservoir content. However, the scenario length needs to be shortened in practical cases (large data sets with fine time resolution) due to computation times. Finding a balance between reasonable computation times and impact of end-valuation is one of the key challenges when setting up a SOVN run.

The end valuation is based on the water values from the EMPS model. These values are available for the aggregate reservoir. The basic method is a follows: For a given individual reservoir the end-valuation is defined by discretizing the reservoir volume in 51 segments (2 % intervals), and assigning a value to each segment i according to:

$$c_i = wv_i * E * R \qquad \textbf{(17)}$$

Where wv_i is the water value (e.g. in øre/kWh) calculated for the aggregate reservoir, $E$is the energy equivalent to sea from the reservoir (in kWh/m3) and R is the interest rate.

The method above describes the basic methodology. The method implemented in SOVN in addition utilize parts of the reservoir drawdown model in EOPS/EMPS, especially information about individual target reservoirs, to include individual differences related to overflow risk and discharge flexibility.

3 Market Modelling

Market modelling capabilities in SOVN is very similar to those in the EMPS model. The descriptions of the basic functionality included are assumed known and is documented in the EMPS use manuals. In this section we will focus on special solutions that are made for the SOVN model.

3.1 Dynamic end-user flexibility

Dynamic end-user flexibility are included in the master problem, but not in scenario-fan. In the scenarios this load is just modelled as regular price dependent load, no coupling between time steps. It is difficult to include dynamic end-user flexibility in the scenario-fan because load capacities depend on the solution the previous week. Along the scenarios the solution, i.e. the actual price dependent load, the previous week is not known. For the master problem the price dependent load in the previous week are known.

During testing we observed that this inconsistency between how dynamic end-user flexibility is modelled in master and scenarios may give unwanted consequences. The scenarios may see too high load flexibility for the extreme cases. This can e.g. result in too high price. The importance of this inconsistency depend on the size of the dynamic load.

3.2 Start-up costs

To correctly model start-up costs of thermal units will require binary variables. However, since the SOVN model is based on LP we linearize the start-up constraints, similar to what was done in [18].

$$p_{1,i}(t) \cdot P_{\min,i} + p_{2,i}(t)(P_{\max,i} - P_{\min,i}) - p_i(t) = 0 \qquad \textbf{(18)}$$

$$p_{1,i}(t) - p_{2,i}(t) \geq 0 \qquad \textbf{(19)}$$

$$p_{1,i}(t) - p_{1,i}(t-1) - \delta_i(t) \leq 0 \qquad \textbf{(20)}$$

Where all variables are continuous, and:

- *i* refers to the thermal unit, *t* is actual time step
- $P_{min,i}$ is the minimum production for thermal unit *i*
- $P_{max,i}$ is the maximum production for thermal unit *i*
- $p_i(t)$ is the actual production of the thermal unit
- $p_{1,i}(t)$ is a relative number indicating share of production below minimum production in time step t(ideally this is either 0 or 1)
- $p_{2,i}(t)$ is a relative number indicating share of production above $P_{min,i}$ (ideally this is zero as long as unit has not started, but with the linear approximation $p_{2,i}(t)$ may be positive even if $p_{1,i}(t)$ is below 1)

- $\delta_i(t)$ is a variable representing starting the unit in time step t (note that with linear approximation a partial start of the unit may occur). This variable is also added in the objective function with the cost given by the start-up cost of the unit.

All the variables $p_{1,i}(t)$, $p_{2,i}(t)$ and $\delta_i(t)$ takes values between 0 and 1. Equation (18) couples the relative values $p_{1,i}(t)$, $p_{2,i}(t)$ to the actual production of the unit. Equation (19) forces the unit to start before producing above $P_{min,i}$ (note that it does not guarantee a complete start of the unit). Equation (20) counts the number of startups (ideally either 0 or 1) in time step *t*. Equations (18)-(20) are added for all time steps.

The start up cost is in current version only included in master problem and thus not included in the cuts. Therefore, there is now no value for the unit to run at the end of the week.

3.3 Reserve capacity

The implementation of start-up costs can easily be extended to also include capacity reservation. This is done by adding an additional variable $p_{3,i}(t)$ representing the share of production above $P_{min,i}$ kept as capacity reserve. Equations (19) is then modified as follows:

$$p_{1,i}(t) - (p_{2,i}(t) + p_{3,i}(t)) \geq 0 \qquad \textbf{(21)}$$

Hydropower production can also provide reserve capacity. For hydropower it is not required that the unit has to run to provide reserve capacity (i.e. $P_{min,i,v} = 0$) . The constraint for reserve requirements becomes:

$$\sum_v p_{3,v}(t) \cdot P_{v,\max} + \sum_i p_{3,i}(t) \cdot (P_{i,\max} - P_{i,\min}) \geq R \qquad \textbf{(22)}$$

Where $p_{3,v}(t)$ is a relative number indicating the share of maximum hydro power production for unit *v* used as capacity reserve, $P_{v,max}$ is the maximum production capacity for unit *v*, and R is the required capacity reserve.

The sum of reserved capacity and production $p_v(t)$ equals maximum capacity:

$$p_{3,v}(t) \cdot P_{v,\max} + p_v(t) = P_{v,\max} \qquad \textbf{(23)}$$

By substituting for $p_{3,v}(t)$ $P_{max,v}$ in equation (22) we get:

$$-\sum_v p_v(t) + \sum_i p_{3,i}(t) \cdot (P_{i,\max} - P_{i,\min}) \geq R - \sum_v P_{v,\max} \qquad \textbf{(24)}$$

Thus, we do not need to add equation (23) for all of the hydropower units.

Reserve requirements can be added either for individual areas or for groups of areas. In the latter case equation (22) becomes:

$$\sum_k \left\langle \sum_i p_{3,i}(t) \cdot (P_{i,\max} - P_{i,\min}) + \sum_j p_{3,j}(t) \cdot P_{j,\max} \right\rangle \geq R \qquad \textbf{(25)}$$

where the index k runs over all the areas in the given group. In SOVN capacity reservation can be used without specifying start-up costs.

3.4 Ramping on transmission lines

Maximum allowed ramping (change in value from one time period to the next) is implemented on flow on transport model interconnections defined in MASKENETT.DATA.

The constraint limiting flow changes on a given cable l takes the form of (26) below. Note that this constraint also includes a penalty variable to ensure a feasible solution.

$$-\Delta F_l^{max} \leq f_{l,t} - f_{l,t-1} \leq \Delta F_l^{max} \qquad \textbf{(26)}$$

Ramping constraints are in the current version of SOVN only allowed in the first-stage problem. Note that no initial value is required as input to the model for t=0, therefore (26) is not included for that time step.

4 Modelling of Uncertainty

SOVN uses historical observations more or less directly to represent future uncertainty through scenarios. In the following we describe how these scenarios are created and how they are conditioned on known information.

4.1 Generating smoothed scenarios

The SOVN model is as mentioned previously built as a simulator that solves two stage stochastic problems. The second stage scenarios are based on historical observations and the different uncertain inputs are coupled through use of historical years. This method keeps the statistical variations between different stochastic variables in time and space. For example, one three-year long scenario may represent the historical sequence 1961-1963. Different scenarios are generated assuming that the different historical sequences may repeat itself and that each sequence have equal probabilities. Possible climate change adaptions will be part of the pre-processing of the observed input values.

Assume now as an example that our first-stage problem is solved for week 10 and year 1962. The second stage scenarios are built as described above assuming that available statistics may repeat itself from week 11 on with equal probability. If no special consideration are taken, there might be an unnatural change in the value of a given uncertain variable from the known value in week 10 and year 1962 to the scenario value in week 11. To avoid this, SOVN include a method that smooths the transition from the known value in the first stage to the closest in time unknown values in the second stage. Smoothing may be seen as short-term forecasting based on known information in the first stage.

The implemented smoothing is described by the following method and is done individually for each uncertain input.

Assume a time series of given uncertain input, e.g. a inflow series. The smoothing method consist of two main parts; an identification part and a smoothing part.

Identification part:

1. Normalisation of the time series (subtract weekly mean and divide by standard deviation)
2. Identify first order autocorrelation (might be seasonal)

Smoothing part:

1. Compute smoothed normalized values using equation 27

The identification part is done once for each uncertain time series (inflow series, wind power production series, exogenous prices etc) and the smoothing part is done for every series whenever a new scenario fan is generated. The smoothed values are calculated using equation (27). Possible negative values as results of the smoothing are reset to a small positive value.

$$T_j^{new}(t+k) = T_j(t+k) + \left[T_i(t) - T_j(t)\right]\frac{\sigma(t+k)}{\sigma(t)}a^k \qquad \textbf{(27)}$$

where:

$T_j^{new}(t+k)$	Smoothed value for scenario j, k time-steps ahead from the first stage
$T_j(t)$	Observed (unsmoothed) value for scenario j in time step t
$T_i(t)$	Known first stage value
$\sigma(t)$	Standard deviation in time period t
a	Estimated autocorrelation (a is typically in the rage 0.3-0.95 depending on the type of series inflow, price, wind)
k	Number of time steps forward from the first stage
t	Point in time for the first stage

Note that the second term on the right-hand side in the equation approach zero as the exponent k increases, so that the smoothed value will eventually take the original scenario value $T_j(t+k)$. Note also that the scenario values for the simulated year (i), that we now are solving the first stage for, will be unchanged for the whole scenario (because i=j).

Figure 3 and Figure 4 show an example of how the method works for an inflow series. The example are taken from the spring/summer period with large variations. Figure 3 shows the observed variations for a number of historical years. Figure 4 shows the first stage inflow values and the corresponding smoothed inflow scenarios assuming that that first stage is week 20 and that the inflow in that week is the lowest registered. Weeks 21 to 30 shows the smoothed values for the different scenarios. The example is one of the more extreme, but it illustrates how the method works.

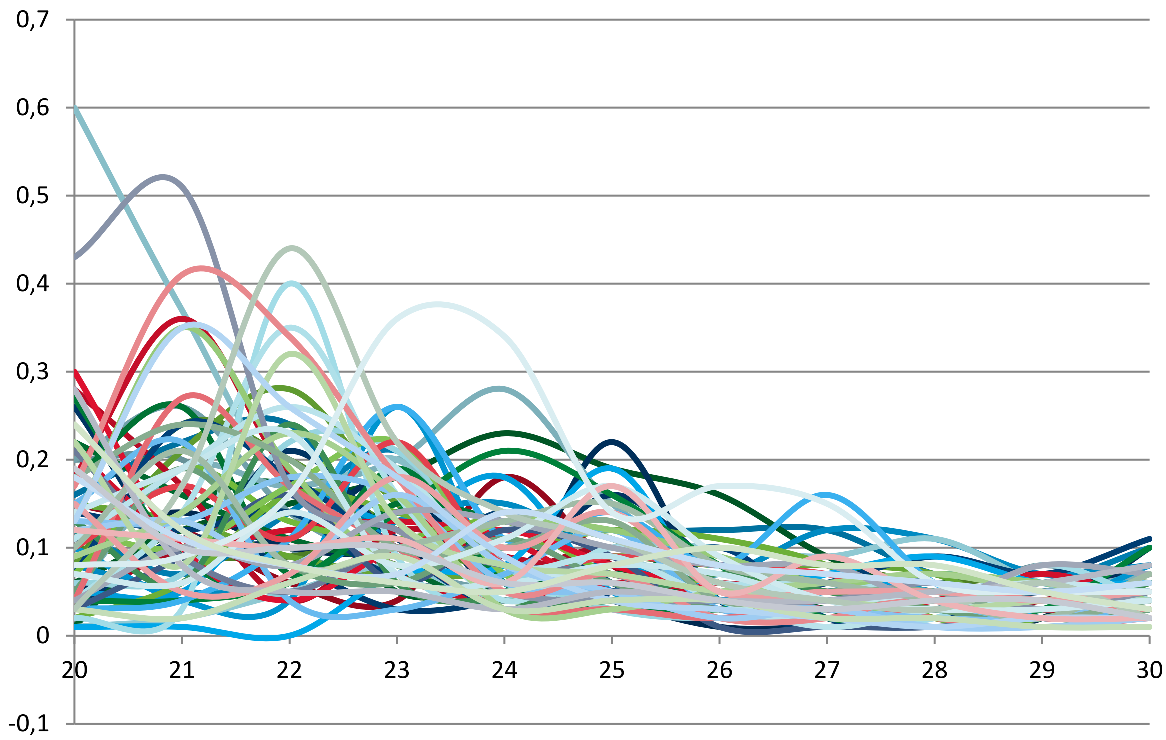


Figure 3 Observed inflows for different historical years shown from week 20 to week 30.

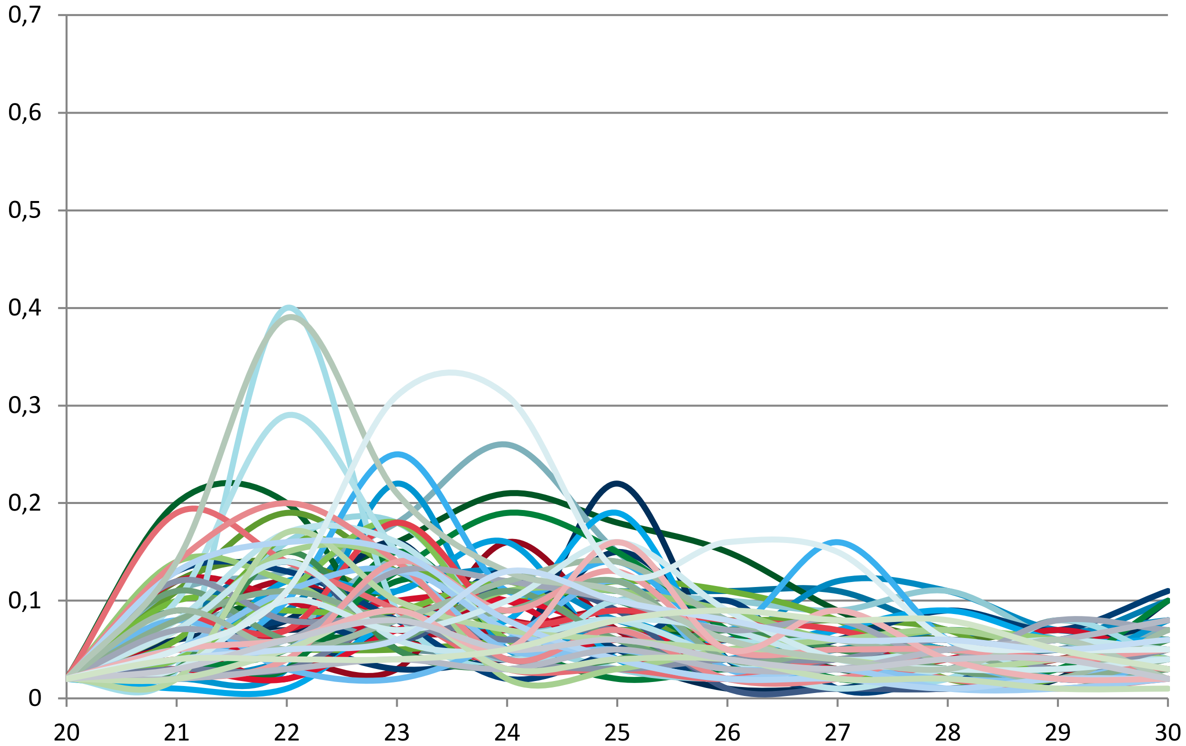


Figure 4 Smoothed inflow scenarios (week 21-30) corresponding known inflow in week 20.

Originally, the smoothing algorithm was implemented for log-transformed values before transformation back to real values. The main purpose of this was to avoid negative values. However, this transformation gave unrealistic high smoothed values for the first scenario weeks for some cases with very low values in the first-stage week. We therefore chose to skip the log transformation and use a simple resetting to a small positive value if the smoothing gives a negative value. The same smoothing algorithm is used for all types of uncertain input: inflow, exogenous prices, solar and wind power production. It might be that the log-transformation can be used if more effort is put into handling of the special cases that give problems.

4.2 Inflow

Inflow uncertainty is represented by the variation given historical time series, with weekly time resolution. In its current state, the SOVN model does not allow daily time resolution, but this is rather straightforward to open for.

For the first 52 weeks historical inflows may be correct based on known snow storages and short-term weather forecasts. Corrected inflows are thereafter smoothed in the transition from first-stage week to scenarios fan as described in section 4.1

4.3 Exogenous Power Price

Exogenous prices are given by the user for the whole planning horizon. There is one or more price scenario for each inflow scenario in the simulation.

4.4 Wind and solar power

Wind and solar power production inputs are given with one value for each weather year for each time period within the year. The time resolution can be hourly, daily or weekly.

4.5 Temperature

Temperature inputs are specified with one value for each weather year for each time period within the year. The time resolution can be weekly or daily. The temperatures are used for correction of load and for correction of production from CHP plants.

Smoothing is not implemented for temperatures. There is no specific reason for this, and most likely smoothing is more important for temperatures than for wind and solar because of higher autocorrelation.

4.6 Snow

4.6.1 Introduction

Snow reservoirs provide information about future inflow. In the winter and spring period there is information about future inflow in the snow reservoir. Information about current snow reservoirs are normally included into the spring flood forecast for the first year in hydro scheduling models. The issues discussed here does not deal with the specifics of how to use current snow pack information, but rather how in general the relation between snowpack and spring flood can be used to make better and more realistic simulation models.

The snow reservoir can be estimated or measured.

An example:
We are simulating operation of a given reservoir for the 3rd year in a five year a planning period. In standard models we calculate water value tables for each week that depend on the storage level in the reservoir. When we simulate week by week for the winter and spring in the third year we use this table to find the marginal value of water and calculate the corresponding production. The calculated production depends only on the reservoir storage and the market. In real operation the owner of the reservoir will from early in the winter have knowledge about the snow storage and modify his water values depending on this information. The modification comes through updating the spring flood forecast, as discussed in the first section. Snow reservoirs above normal lead to lower water values, and vice a versa. The importance of snow reservoir information is increasing until the spring flood starts.

In a market simulation model like SOVN we simulate for many different inflows, some have very high spring floods and some have very low. The point is that booth of these extremes will be known some time before they occur because of the snow reservoir information. The further discussion focuses on how this information can be included into the SOVN simulations.

One possible method is explained below:

- Assume existence of a calibrated hydrological model with available historical weather data for the complete historical period to be used in SOVN.
- The hydrological model is first used to simulate a time series for "historical" snow storages using historical weather input.
- For each week in the period January to August for each of the historical years an inflow forecast is made assuming that the initial snow storage is given by the simulated "historical" value for that date and assuming that all weather years are possible from that date on.
- The result of this is a complete and separate new set of inflow scenarios for each simulated year and week

For the part of the year where snow storage gives little information about spring inflow, i.e. typically week 35-52, historical inflow years are used directly to make future inflow scenarios.

The main drawbacks with this method are the number of data that must be pre-processed and made available to the SOVN model. Assume that there are 80 inflow years with daily time resolution, 30 weeks of the year with relevant snow storage information, 100 different inflow series and a 5 year planning period. The total number of inflow data will then be given by: 104*7*2*30*80*80*100 *4 = 111 GByte.

104*7 - The number days in the inflow forecast. It is assumed that the inflow forecasts for the third to fifth year in the planning period are equal to the second year, independent of the initial condition.
2 *30 - Number of different time steps that needs a separate forecast. 30 weeks in the two first years. Also here it is assumed that the inflow forecasts for the third to fifth year in the planning period are equal to the second year, independent on the initial condition.
80 - The number of scenarios in one forecast
80 - The number of initial conditions (snow storages) for a given time step
100 - Number of different inflow series
4 -Number of bytes

The detailed explanation of the data size is also intended to help explain the method. In SOVN separate inflow scenarios would be used in principle for almost each simulated week adapted to the known snow storage.

The rest of this section emphasize on the alternative statistical based and simpler method that needs less pre-processed input data. It is this simpler method that has been implemented in SOVN. It also assumes there is available one historical time series for snow storage for each inflow series.

4.6.2 Implemented Method

In SOVN future inflows are given by historical weather years:

Assume that we have available the following historical time series:
$T_j(t)$ - Inflow in week t for inflow year j
$S_j(t)$ - Snow reservoir in week t for inflow year j

Define also:
T_a - Last week of the melting season where the snow reservoir give significant information about future inflow. This week is calculated automatically to maximize the correlation between snow storage and future inflow for the whole snow accumulation period. T_a is fixed for a given time series and is typically around week number 33.

Further define a new time series for accumulated inflow:

$$\mathrm{T}_{\mathrm{acc}\ \mathrm{j}}(\mathrm{t}) = \sum_{k=t+1}^{T_a} T_j\ (k) \qquad \textbf{(28)}$$

$T_{acc\ j}(t)$ for time step t is sum inflow from time $t+1$ to time T_a. for inflow year j .

The method is based on the estimated correlation between the time series for snow storage $S_i(t)$ and accumulated future inflow $T_{acc\ j}(t)$. The details of the whole approach was first described in [16]. Correlations are estimated from normalized versions of $S_i(t)$ and $T_{acc\ j}(t)$. The normalized time series are calculated by subtracting the mean value and dividing by the standard deviation for each week t. An example of the estimated correlations for three different inflow and snow storage series are shown in Figure 5.

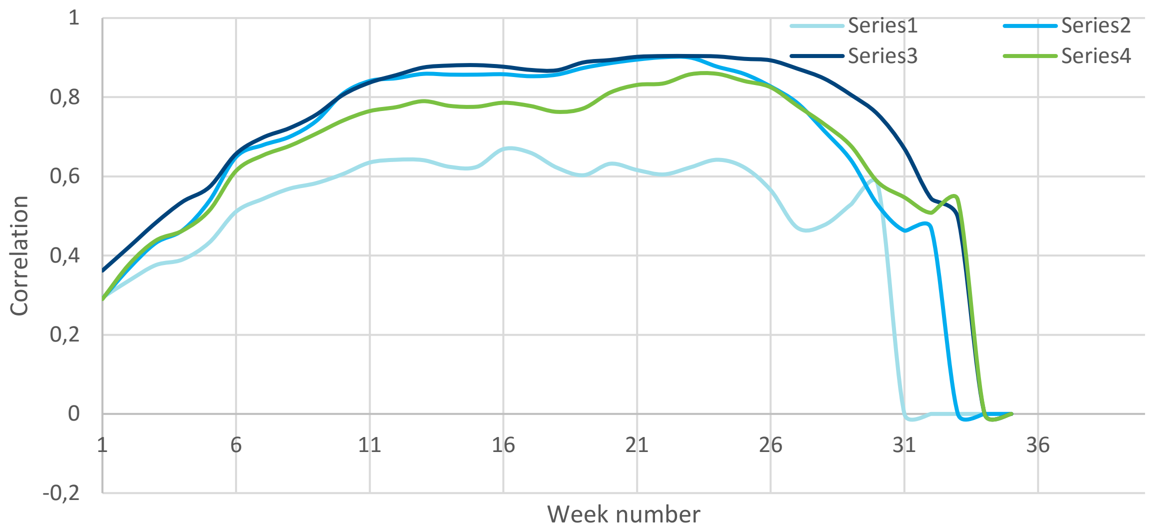


Figure 5 Estimated correlations between snow storage Sj(t) and sum future inflow Tacc j(t) for four different inflow and corresponding snow storage series.

For a given time step t the normalized snow storage is given by $S_j^N(t)$.

$S_j^N(t)$ - Normalized snow storage in week t for inflow year j. Values > 0 means snow storage above normal.

Based on the estimated correlation it is possible to calculated the expected future inflow conditioned on known snow storage in week t, i.e. $S_j^N(t)$.

It is more or less shown in [16] (some assumptions and knowledge of statistics is also needed) that the expected sum future inflow conditioned on the information that the snow storage in week t is $S_j^N(t)$ is given by :

$$\overline{T_{acc}(t)|S_j^N(t)} = \rho(t) * S_j^N(t) * \sigma_{Tacc}(t) + \overline{T_{acc}(t)} \tag{29}$$

where

$\overline{T_{acc}(t)}$ - Unconditional expected sum inflow from $t+1$ to T_a

$\sigma_{Tacc}(t)$- Standard deviation for sum inflow in week t

$\rho(t)$ - Estimated correlation between snow storage in week t and sum future inflow $T_{acc}(t)$

$\overline{T_{acc}(t)|S_j^N(t)}$ - expected sum future inflow conditioned on known snow storage $S_j^N(t)$ in week t

We see from (29) that snow storage above normal gives more than normal future inflow and that higher correlation gives higher dependency on the snow storage.

Equation (29) gives the average sum inflow of the scenarios that are going to be used when we solve the first-stage problem week t, inflow year j with known snow storage $S_j(t)$.

The uncertainty of the sum inflow scenarios can be estimated by equations (30) and (31) based on +/- 2 standard deviations

$$T_{acc,min}(t) = \overline{T_{acc}(t)|S_j^N(t)} - 2 * \sqrt{1 - \rho^2(t)} * \sigma_{Tacc}(t) \tag{30}$$

$$T_{acc,max}(t) = \overline{T_{acc}(t)|S_j^N(t)} + 2 * \sqrt{1 - \rho^2(t)} * \sigma_{Tacc}(t) \tag{31}$$

4.6.3 Implementation in SOVN

The implementation SOVN consist of the following main parts:

1. Calculate the correlation between snow storage and future inflow. Based on the correlation define the period of the year where the correlation is significant (e.g. $\rho(t)$ > 0.4).

2. If correlation is significant, pick the inflow scenarios that have sum inflow between maximum and minimum given by equations (30) and (31). This is done to remove scenarios that has a (melting) profile that is most inconsistent with the known snow storage.

3. Scale chosen scenarios to the estimated average given by equation (29).

4. Use these scenarios in SOVN.

In SOVN the inflow forecast (scenarios) must cover the length of the whole scenario period, not only the period up to week T_a. Beyond week T_a all historical inflow scenarios are equally probable, assuming no climate change. However, because we are using snow storage information we have picked out a subset for the beginning of the scenario period. The problem is how to connect the subset scenarios to unconditional statistics. We need to have the same number of scenarios for the whole length of the scenario problem because SOVN use deterministic scenarios. The number of scenarios in the subset depends on the time of year and the snow reservoir size. In some cases, close to the start of spring flood, with extreme snow reservoirs the number of scenarios in the subset will be small.

The problem is solved by duplication of the subset scenarios and coupling them to the original scenarios. The chosen subset scenarios already have a coupling because they represent a historical year.

Example:
Assume that we are using 10 inflow years (1951-1960) and a planning period of 3 years and we are now solving for week number 10, inflow year 1951. We use 94 weeks long scenarios. Snowpack information is significant from week number 2 to week number 33.

The unconditional inflow scenarios (existing SOVN) are built as follows:

Scenario 1: 1951, 1952
Scenario 2: 1952, 1953
.
.
Scenario 10: 1960, 1951

Assume further that based on the snow storage information in week 10 only scenarios 1, 3, 5 and 10 are within the limits given by equations (30) and (31). We therefore need to multiply these scenarios to keep the original number of scenarios from week 11 to week 33. A possible solution is shown in Table 1. We see that some years are used twice (e.g. 1951) and some only (1960) two times. The resulting scenarios are scaled to the average given by equation (26) for week 11 to 33. The chosen solution shown in Table 1 is just one of many possible solutions.

Table 1 An example of possible scenario duplication.

Scenario number	Week 11-33	week 33-52	Week 53-104
1	1951	1951	1952
2	1951	1952	1953
3	1953	1953	1954
4	1953	1954	1955
5	1955	1955	1956

6	1955	1956	1957
7	1960	1957	1958
8	1951	1958	1959
9	1953	1959	1960
10	1960	1960	1961

Possible scenario reduction is done after the above procedure.

We have tested the described method for snow storage correction on real data of the Nordic system. Figure 6 shows variation in expected future inflow depending on variation in known snow storage in week 12. The figure shows sum energy inflow to the whole system. These values are calculated using the method described above for individual inflow series and then aggregated to total inflow. The scale of the y-axis is hidden, because real user data is used, but still it shows that significant information about the future inflow is available conditioned on known snow storage in week 12.

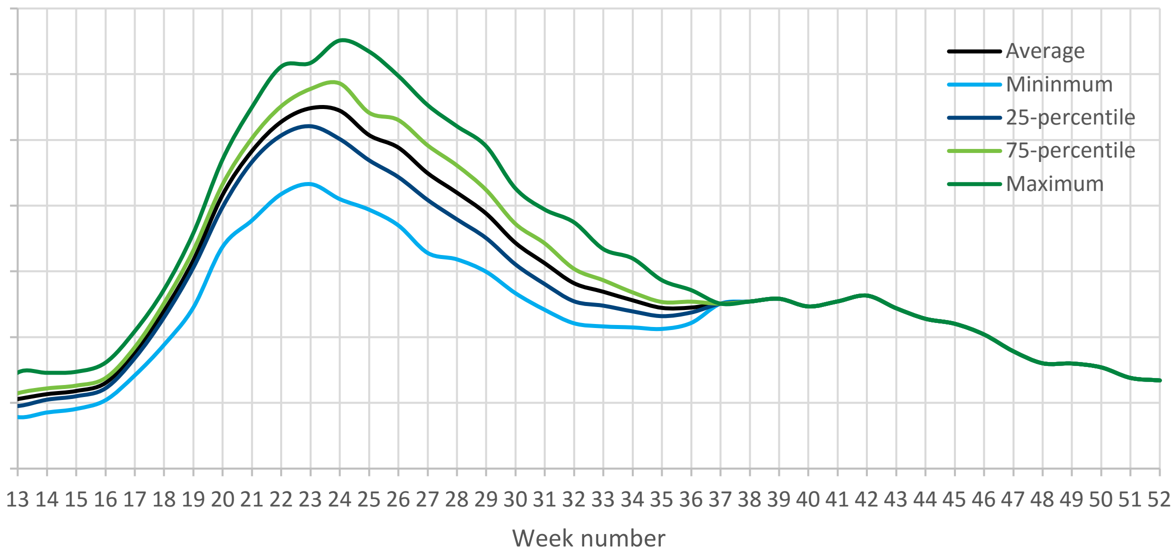


Figure 6 Variation in expected future inflow depending on variation in known snow storage in in week 12.

4.7 Availability of Thermal Capacity

Unavailability of power system components can be classified as either planned outages or forced outages.

Planned outages are revisions scheduled in time, and can be handled in SOVN as is done in the EMPS model. Reduced capacities on hydropower stations are specified in the file REVISJONSPLAN.STAS. Reduced capacities in the transmission grid (transportation model) are specified in MASKENETT.DATA.

Forced outages are probabilistic, i.e., one does not know the timing and severity of such outages.
The EMPS model can treat forced outages using the Expected Incremental Cost (EIC) method. With this method it is possible to compute the expected consequence of unavailability of thermal capacity. The EIC method is computationally efficient and well suited for addressing the impact of generator's reduced availability on price levels in complex market models. References [17, 15] provide relevant theoretical background for the method.

The EIC method has not been implemented in SOVN so far for the following reasons:

- It will increase the size and complexity of the problems considerably.
- Production for individual units is not directly available from the solution. Expected production can be calculated in post processing. Thus, the method cannot easily be combined with start-up costs on the same thermal units or detailed power flow constraints.

5 Scenario Reduction

Uncertainty is represented using a scenario fan in the SOVN model, as described in section 1. One can view the fan as special case of a multi-stage scenario tree, where only the second stage is stochastic, and realizations from the remaining stages of the tree are deterministic given the second stage value.

The SOVN model allows direct use of historical scenarios to represent the second-stage uncertainty, where each scenario has equal probability of occurrence. As an example, if 80 historical inflow scenarios are available and there are 4 price scenarios for each inflow scenario, a total of 320 scenarios can be used to represent the second-stage uncertainty. The direct use of scenarios is convenient in the sense that no refined statistical model is required to represent the stochastic processes. The direct use of observed variables also keeps observed correlations between all uncertain variables in time and space and this is assumed to be a very important property of this implementation.

However, solving each second-stage scenario problem is computationally demanding, and we have therefore implemented a scenario reduction algorithm.

In the context of stochastic optimization, scenario reduction refers to the problem of reducing the number of nodes of a scenario tree such that:

a) The probability distribution represented by the reduced tree is close to the initial distribution.
b) The optimal solution of the stochastic program using the reduced tree is close to the true optimal solution.

In a previous research project [2], a thorough literature review was done covering scenario generation and reduction techniques. Moreover, a set of methods were implemented for this purpose, resulting in an in-house tool for scenario generation and reduction. We have implemented of one of these methods in SOVN, based on the fast-forward selection algorithm documented in [9].

The method is described by the following:

Define:

$$D_{ij} = \sum_{t=1}^{T} \sum_{n=1}^{N} \left(E_{nit} - E_{njt}\right)^2 \qquad \textbf{(32)}$$

Where

D_{ij}	-	Measure for the distance between scenario number i and scenario number j
E_{nit}	-	Value of scenario number i in timestep t unit n
N	-	Number of units (inflow series, wind series, exogenous price series)
T	-	Number of time steps

The reduction algorithm goes through the following steps to exclude one scenario:

1. Calculate probability weighted distance to all other scenarios $p_i * D_{ij}$ where p_i is the probability of scenario i.

2. Remove the scenario with lowest probability weighted distance to another scenario.
3. Update probabilities for scenario p_j (assuming scenario i was removed, being closest to scenario j)
   $p_j = p_j + p_i$

An example of this procedure is shown with Table 2 and Table 3. Table 2 shows the calculated distances between all scenarios and the original probabilities. Because all scenarios have equal probabilities and the distance between scenario 1 and 2 is smallest, in this case scenario 1 is removed. Table 3 shows the updated probabilities and the next scenario to be removed.

Table 2 Calculated distance measure and probabilities (first row) for 10 example scenarios. Green indicate a small distance and red a large distance.

Prob	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0,00	0,73	2,81	2,37	1,42	1,15	1,17	1,50	1,30	0,99
2	0,73	0,00	2,40	2,81	1,06	1,28	1,82	1,25	1,20	1,06
3	2,81	2,40	0,00	3,42	1,93	2,01	3,01	3,52	2,39	2,95
4	2,37	2,81	3,42	0,00	2,06	2,47	2,43	3,06	3,34	2,90
5	1,42	1,06	1,93	2,06	0,00	1,23	2,41	1,64	1,46	1,78
6	1,15	1,28	2,01	2,47	1,23	0,00	2,07	2,38	1,53	1,80
7	1,17	1,82	3,01	2,43	2,41	2,07	0,00	2,21	2,32	2,08
8	1,50	1,25	3,52	3,06	1,64	2,38	2,21	0,00	1,87	2,16
9	1,30	1,20	2,39	3,34	1,46	1,53	2,32	1,87	0,00	1,40
10	0,99	1,06	2,95	2,90	1,78	1,80	2,08	2,16	1,40	0,00

Table 3 Updated probability (left red arrow) and marking of the next scenario (number 10) to be removed and added to scenario number 2.

Prob	0,00	0,20	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0,00	0,73	2,81	2,37	1,42	1,15	1,17	1,50	1,30	0,99
2	0,73	0,00	2,40	2,81	1,06	1,28	1,82	1,25	1,20	1,06
3	2,81	2,40	0,00	3,42	1,93	2,01	3,01	3,52	2,39	2,95
4	2,37	2,81	3,42	0,00	2,06	2,47	2,43	3,06	3,34	2,90
5	1,42	1,06	1,93	2,06	0,00	1,23	2,41	1,64	1,46	1,78
6	1,15	1,28	2,01	2,47	1,23	0,00	2,07	2,38	1,53	1,80
7	1,17	1,82	3,01	2,43	2,41	2,07	0,00	2,21	2,32	2,08
8	1,50	1,25	3,52	3,06	1,64	2,38	2,21	0,00	1,87	2,16
9	1,30	1,20	2,39	3,34	1,46	1,53	2,32	1,87	0,00	1,40
10	0,99	1,06	2,95	2,90	1,78	1,80	2,08	2,16	1,40	0,00

The user specifies the desired number of scenarios to be used in the fan. Before the scenario reduction starts, all scenarios are assumed to have equal probability. The scenario value E_i (GWh/time step) represent in SOVN the sum energy of all inflow series plus all wind and solar energy production in the system for a given time step. The energy equivalent to sea (kWh/m^3) for each reservoir where a given inflow series is used together with the amount of inflow to find the energy (GWh/time step) for each inflow series.

Note that if there exists more than one price scenario for each inflow scenario current scenario reduction implementation will remove all price scenarios first because prices are not part of the evaluation criteria.

5.1 Example

In this section we show an example of the whole scenario generation process for one specific inflow series for a given first-stage week in the future. The process consists of 4 steps:

1. Scenarios given by observed values (Figure 7)
2. Correction for snow storage information (Figure 8)
3. Scenario reduction of snow-corrected inflows (Figure 9)
4. Smoothing of scenarios based on known inflow in the first-stage week (Figure 10)

The first-stage week is assumed to be week number 12 and the specific inflow series is from south east of Norway.

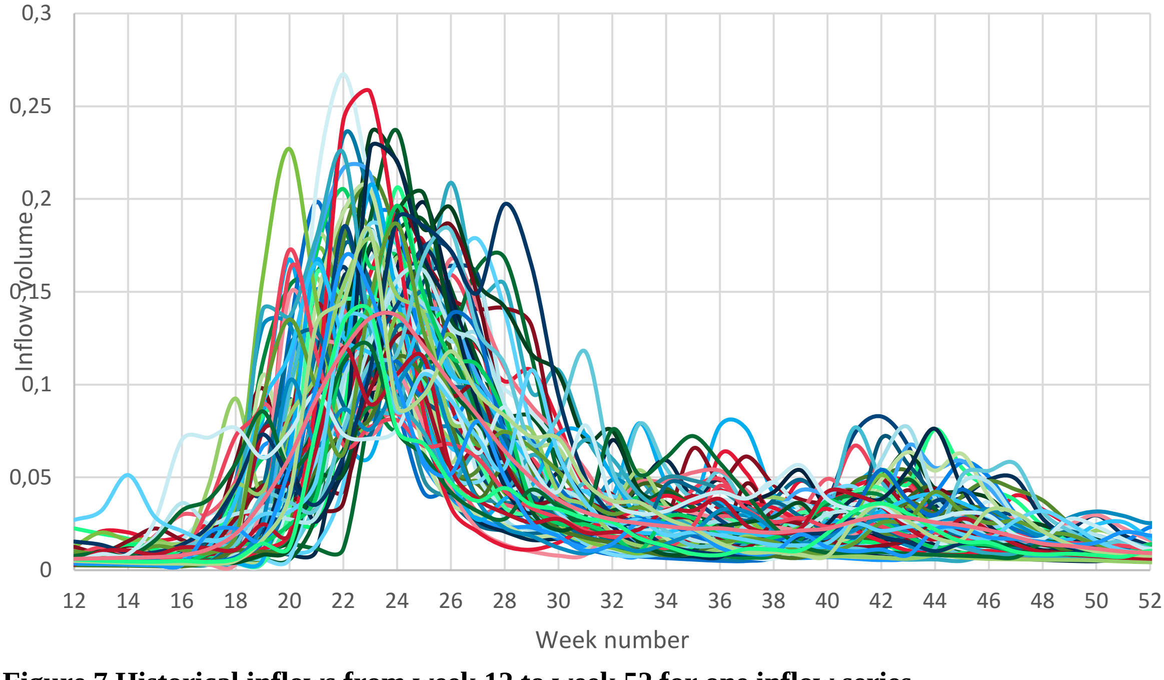


Figure 7 Historical inflows from week 12 to week 52 for one inflow series.

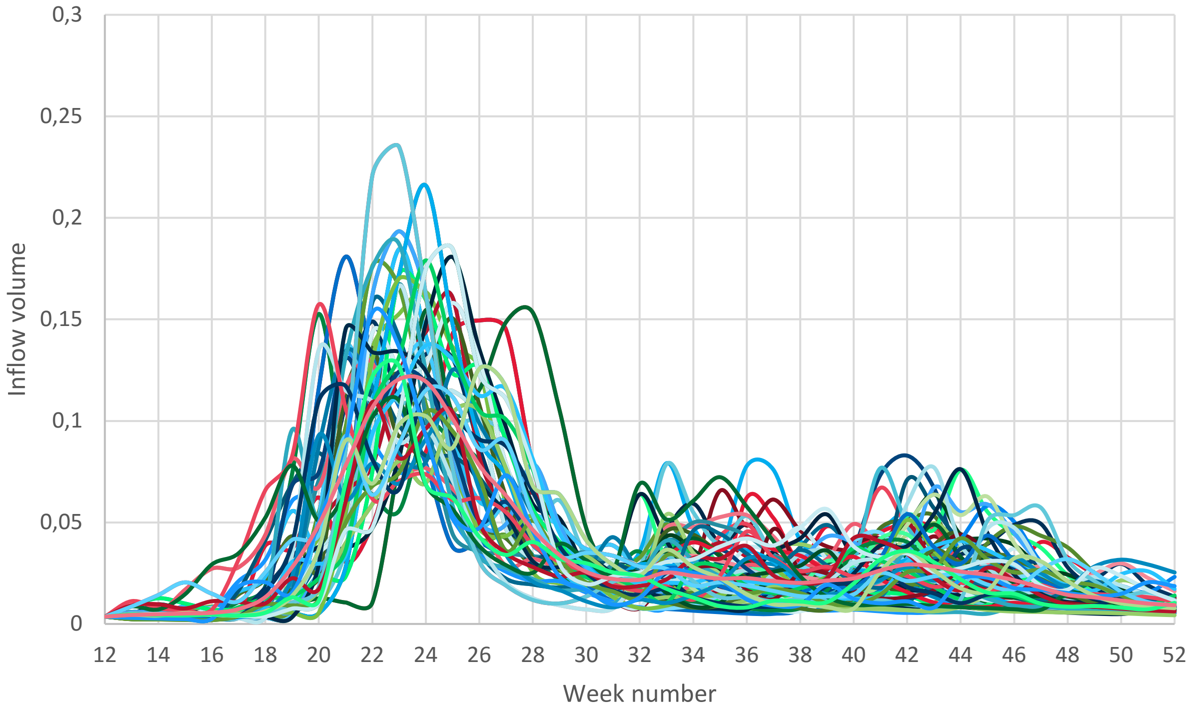


Figure 8 Scenarios for given inflow and snow storage information (snow storage lower than normal).

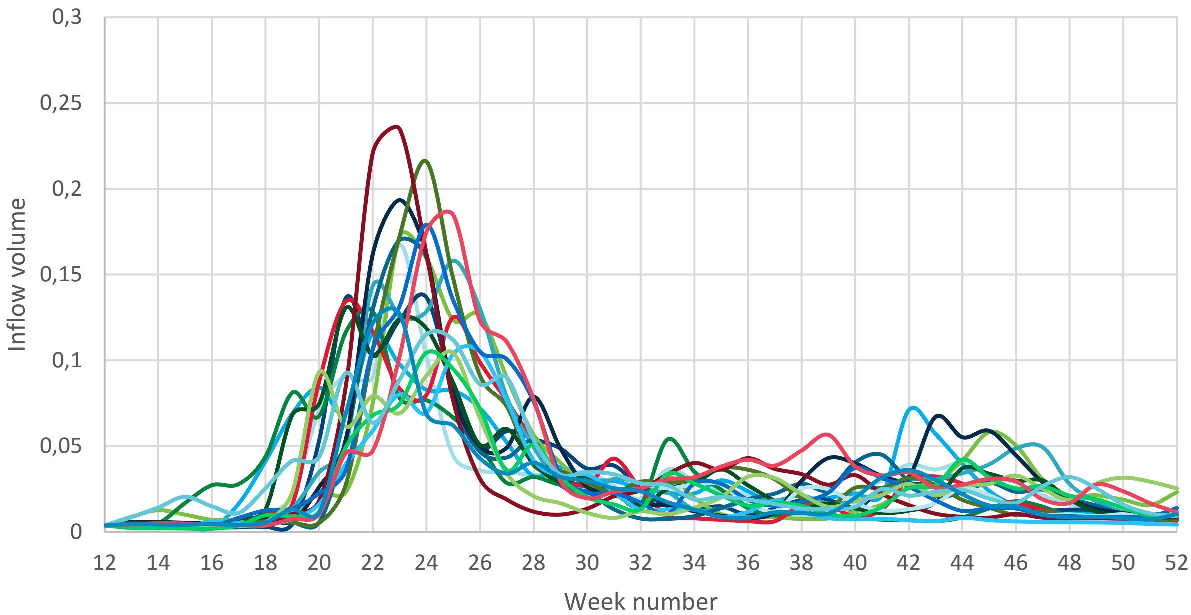


Figure 9 Reduced number of scenarios (20) for snow-corrected inflows.

Finally, we apply the techniques discussed in Section 4.1 to find the smoothed series based on the known inflow in the first-stage week

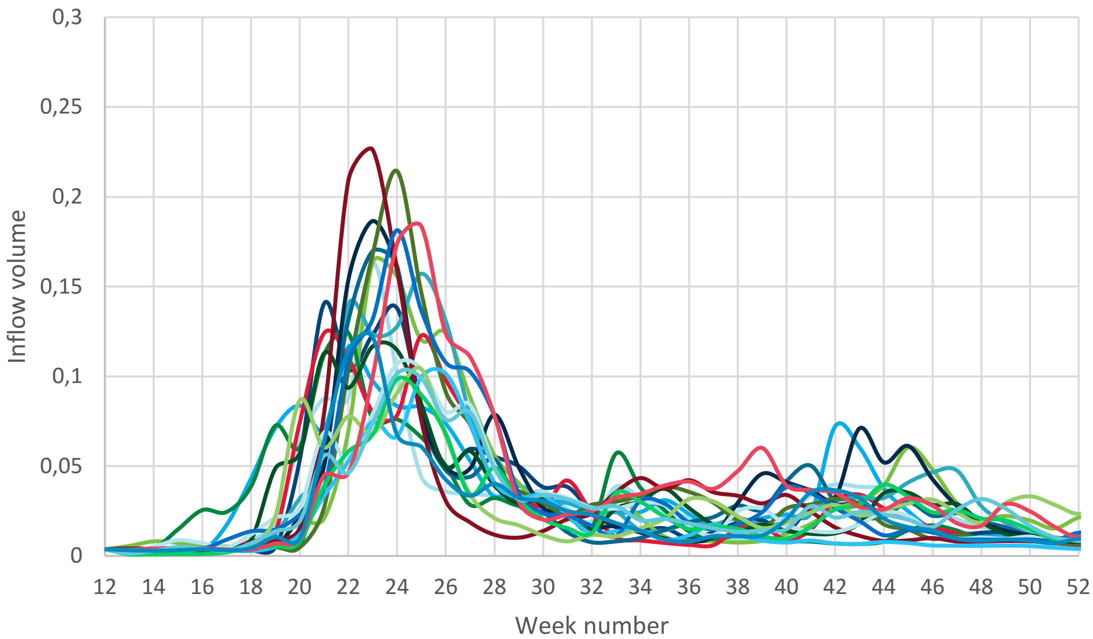


Figure 10 Smoothing: correcting for known inflow in the first-stage week.

6 Power Flow Constraints

The SOVN model allows modelling of the transmission grid by using:

a) A transportation model (similar to the EMPS model)
b) Linearized power flow equations (similar to Samnett)

The following sections describe transmission grid modelling. The standard approach using a transportation model is briefly reviewed in Section 6.1. Two different approaches for modelling linearized power flow equations are elaborated in Sections 6.3 and 6.4. The use of static linearization presented in Section 6.3 does not rely on the presence of detailed grid data. It simply uses a pre-defined set of linear sensitivities (PTDFs) defining how much the flow on an interconnection will change with a change in net power injection in a certain price area. When using dynamic linearization, these PTDF factors are computed internally based on available information about the detailed grid and on some pre-defined weighting scheme.

6.1 Transportation model

In the simulation phase in EMPS model, the flow on a connection *l* between price areas is represented by a non-negative variable f_l^{dir} in both directions (dir):

$$0 \le f_l^{dir} \le F_l^{max,dir} \qquad \textbf{(33)}$$

The capacities $F_l^{max,dir}$ in both directions are defined in the file MASKENETT.DATA (and TRANSCAP_HOUR.DATA), and these can be time-dependent.

The capacity can be load or wind dependent. Where the capacity change linear based firm load in given area or specified windfarms as well as extra input on MASKENETT.DATA (more in EMPS user manual).

6.2 Linearized power flow equations

The linearized (or DC) power flow equations used in the SOVN model can be derived from the full AC power flow equations, as described in [14]. We will briefly restate the equations and describe how the elements are computed in the following.

The linear equations takes the following form:

$$\mathbf{B}\boldsymbol{\delta} = \mathbf{P} \qquad \textbf{(34)}$$

Where, for all busbars (except the swing bus):

B Matrix whose elements are described below;
δ Vector of voltage angles;
P Vector of power injections.

The elements of the B-matrix are expressed as:

$$B_{ii} = \sum_{j \in \Omega_i} \frac{1}{X_{ij}} \text{ and } B_{ij} = -\frac{1}{X_{ij}} \tag{35}$$

The set of linear equations in (34) is solved every time the SOVN model needs to check for overloads in the detailed grid. We factorize the B matrix using the NAG routine `F01BRF` and solve for voltage angles using routine `F04AXF`. In the current version of SOVN we assume a static grid, i.e., we do not consider changes in the grid topology and physical parameters during the period of analysis. Thus, the factorization is only done once.

Once the voltage angles are known, power flowing on a given line *l* between busses *i* and *j* can be found as:

$$f_l = \frac{\delta_i - \delta_j}{X_l} \tag{36}$$

Once overloads are detected, linear constraints are added to the related time intervals in the SOVN model. When expressing these linear constraints, we omit representation of the voltage angles, by using power transfer distribution factors (PTDF). A PTDF describe the ratio between change of flow on a given power line and change of power injection at a given busbar, when the corresponding injection change is on the swing bus.

If we conceptually invert the B matrix in (34) we get:

$$\mathbf{\Delta\delta} = \mathbf{B}^{-1}\mathbf{\Delta P} \tag{37}$$

Assume that the net power injection in a busbar *k* is $\Delta P_k = 1.0$ (the corresponding reduction will be put on the swing bus). The change in flow on the line from *i* to *j* will be the difference between elements *ik* and *jk* divided by the reactance of the line.

Knowing the elements of the row *i* and *j* of the inverse B-matrix, we can calculate the impact of any change (combination) in net active power injection. A row/column (note symmetry) of the inverse B matrix can be computed by putting a 1.0 in the right hand side of (37) and solve it. Thus, we can find the relation between any combination of injections at the node by a proper initialisation of (37) and solving it directly. This is simply a linear transformation. The factors required for the transmission line *i-j* can be found by setting the right-hand side to

$$b_i = \frac{1}{X_{ij}} \text{ and } b_j = -\frac{1}{X_{ij}} \tag{38}$$

and solve the equation:

$$B\Theta^l = \mathrm{b} \tag{39}$$

Where:
B matrix defined in (34);
b the right hand side vector;
Θ^l the PTDF vector for transmission line *l*.

These PTDF coefficients are calculated for any line in interconnections (or 'snitt') that are monitored. For an interconnection containing several lines, the b_i's and b_j's for each line are aggregated. It is important to take the defined direction of the line into account when aggregating.

The coefficients Θ^l express the impact on the flow on line *l* for a change in net power injection at any node when the corresponding change is on the swing bus. However, the changes can be positive or negative and it is only the mismatch (the deviation from 0.0) that is put on the swing bus.

6.3 Static linearization

In this Section we describe the use of static (or pre-defined) PTDFs in SOVN. The static PTDFs are read as a matrix from a defined file at start-up, see example in Table 4. This is the only information given by the physical properties of the transmission grid, there is e.g. no information about the detailed lines and buses.

Table 4 Example, static PTDF matrix.

Snitt \ Area	**1**	**2**	**...**
1-2	0.54	-0.67	..
2-3	0.68	0.65	
...			

Simulated power flows (f_l^0) are computed using elements from the PTDF matrix (Θ_i^l) and the simulated net power injection in each model area (p_i^{inj}).

$$f_l^0 = \sum_{i=1}^{NB} \Theta_i^l p_i^{inj} \tag{40}$$

We treat power flow constraints in both the first- and second-stage problems by relaxation. That is, these equations are added when violated. This is done to reduce the LP problem size and therefore save computation time.

If a power flow value $\mathrm{abs}(f_l^0) \geq F_l^{max}$ a PTDF-constraint on the format below is added to the optimization problem:

$$-F_l^{max} \leq \sum_{i=1}^{NB} \Theta_i^l p_i^{inj} \leq F_l^{max} \tag{41}$$

The PTDFs (Θ_i^l) are static parameters, whereas the net power injection p_i^{inj} is expressed by variables. In principle the net power injection can be directly expressed by summing variables representing all production technologies and demand-side flexibility within an area. However, a more computationally efficient technique is to use the flow variables defined when setting up the basic transportation model in SOVN, see Section 6.1. The linear expression then become sparse, in the sense that relatively few variables are used in the expressions. Note that the boundaries on the flow variables in (36) should be set to a large number in order avoid over-constraining the solution space.

The iterative addition of power flow constraints fits nicely into the Benders decomposition scheme used in SOVN. Iterations on adding power flow constraints are integrated with the basic Benders iterations. The following pseudo-code illustrates where the power flow constraints are added in the decomposition loop.

```
While not converged
   solve first-stage problem
   add power flow constraints for first-stage
   solve second-stage (scenario) problems
   add power flow constraints for scenarios
   check convergence
```

The advantage of the static-PTDF approach lies in its simplicity and the modest need for grid-related information. On the other hand, the model will not be able to verify that the DC power flows on bus level are within their boundaries. Moreover, it is difficult to generate a high-quality PTDF matrix without access to detailed grid information.

6.4 Dynamic linearization

SOVN also allows dynamic linearization of power flow constraints, facilitated through dynamically updated PTDFs. This functionality requires detailed grid information, both in terms of a detailed grid data file and a coupling file locating the different market transactions to buses in the detailed grid.

Detailed grid data should be available on the PSS/E saved case format. Currently, PSS/E only provides 32 bit Fortran library USRCAS for extracting data from the saved case. In order to include this library in the SOVN application a two-step procedure was adopted, as shown in Figure 11. First, the PSS/E-file is read by an auxiliary program ReadPSSESavedCase and relevant data are written to a temporary file GridData.SOVN. When started, SOVN reads data from GridData.SOVN and deletes the file. If a 64-bit version of the USRCAS-library is presented in the future, the reading routines can be directly implemented in SOVN.

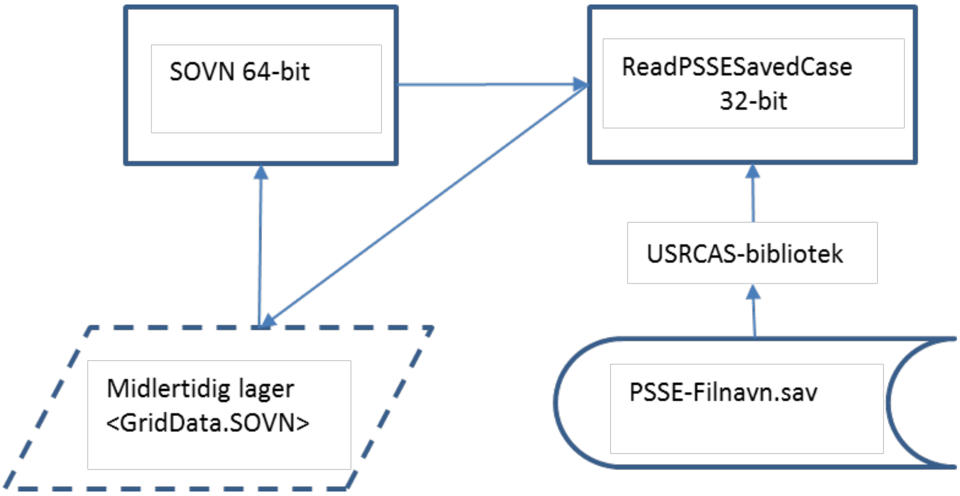


Figure 11 Reading PSS/E saved case in SOVN.

The interplay between the market model in SOVN and the detailed grid analysis is designed similar to what has been done in Samnett [14]. The interplay is illustrated in Figure 12.

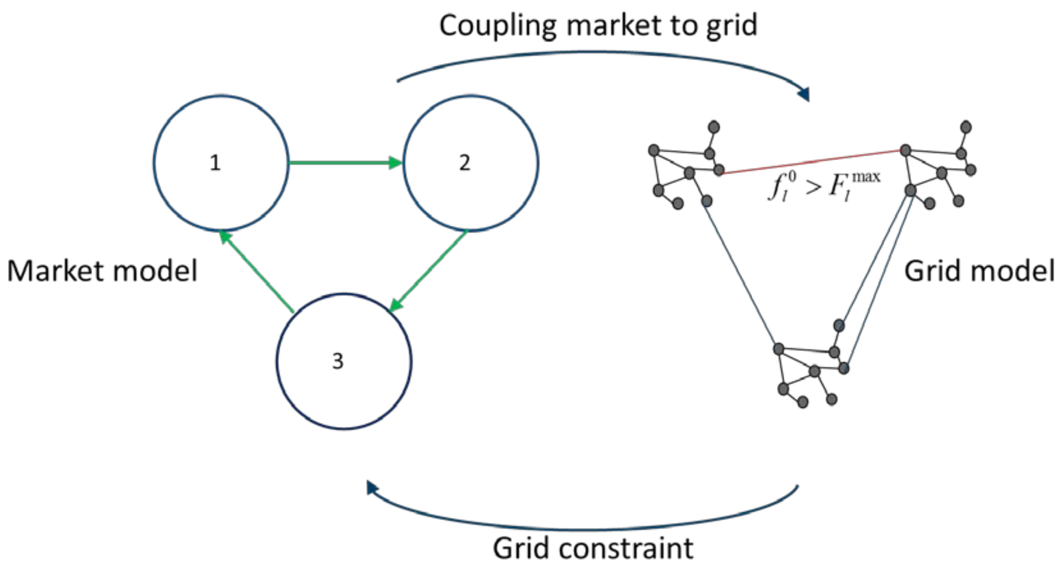


Figure 12 Illustration of the interplay between the market and grid models in SOVN.

The basic steps are described below:

1) The market problem is solved.
2) Market transactions, i.e., production, demand, exchange etc. is mapped to the detailed grid model through coupling keys. The net power injections on each transmission grid bus are defined.
3) A DC power flow (34) is performed, and the model checks for overloads on monitored interconnections.
4) If overloads then:
    a. The PTDF matrix is updated according to the current net power injections
    b. Linear PTDF-constraints are added to the market problem.

c. Re-iterate by repeating steps 1-4.

In step 2, the market solution is coupled to the detailed grid. The market-grid coupling keys are defined in a single XML-file. This file comprises couplings for all hydro modules, all market steps (preference function), all areas with firm-power, all HVDC cables involving areas with detailed grid, and all wind power series. All couplings can be one-one(bus) or one-many (buses). In addition, the electric area definition is provided in this XML-file.

The PTDFs are computed in step 4, according to the following formula:

$$\Theta_i^l = \frac{\sum_{j=1}^{NB(i)} w_j \Theta_j^l}{\sum_{j=1}^{NB(i)} w_j} \quad \textbf{(42)}$$

Where Θ_i^l is the PTDF of area *i* on line *l*, *wj* are weight factors defined for each bus *j*, Θ_j^l is the PTDF for a bus *j* on the line *l*, and NB(i) is the total number of buses in area *i*. Many different weighting schemes are possible (and reasonable) to use in the SOVN model, some of these are discussed in [14]. We have implemented 2 schemes; a) weight based on net power injection, and b) equal weight on all buses.

The PTDF constraints added in step 4 are as in (43), and are slightly different from those presented in (41) in Section 6.3. The constraint in (43) ensures that the overload f_l^0 is controlled within the defined limits F_l^{max} by adjusting the net power injection p_i^{inj} according to the base-case net power injection $p_i^{inj,0}$ which was obtained in the previous market model solution. The delta-formulation in (43) is similar to what is used in Samnett [14].

$$-F_l^{max} \leq f_l^0 + \sum_{i=1}^{NB} \Theta_i^l \left(p_i^{inj} - p_i^{inj,0}\right) \leq F_l^{max} \quad \textbf{(43)}$$

Dynamic linearization is only made available for the first-stage problem. Using dynamic linearization on the second-stage problems would introduce a non-convex relationship (through updated PTDFs) and could violate the convergence properties in the decomposition scheme. A possible solution is use the PTDFs computed in a previously solved SFP as static input to the current SFP, similar to how head is treated.

7 Parallel processing

7.1 Introduction

The SOVN model is a large-scale two-stage stochastic linear programming model that is well suited for massive use of parallel processing, especially for parallel simulation. Firstly, the model is run on sequences of historical records, where each sequence can be treated independently for a parallel simulation. Secondly, the model's basic structure allows using decomposition to solve each second-stage scenario problem separately. Thus, each scenario can be solved in parallel. This can be utilized both for serial and parallel simulation modes.

This section describes how parallel processing is implemented in SOVN. The parallel processing scheme involves parallelization in two layers. On the first (top) level the simulation of weekly decision problems along a historical sequence are distributed to several groups of parallel processes (only used for parallel simulation). This parallel scheme is more or less identical to the parallelization of simulations in other SINTEF models and involves definition of a master – slave concept. The SOVN – implementation differs in the use of groups of processes rather than individual processes.

On the second (lower) level the solution of the weekly decision problem is solved using the available processes within a group of processors. At this level there is not applied a master – slave scheme but rather a predefined allocation of tasks between the processes in the group. The number of processes in a group is either one, or the number of scenarios in the scenario fan plus one. In the first case there is obviously no parallel processing of the sub-problem on level 2.

The level of parallelization is chosen depending on the total number of available processors and type of simulation mode, and may involve only parallelization of the top level, the lower level or on both levels.

A parallel simulation start from a given initial reservoir level where all simulated scenarios are independent. The simulation horizon for each scenario is typically 156 or 260 weeks (similar to a parallel simulation in EMPS) and the 'simulation horizon' for each scenario fan is given by the user. The number for simulated scenarios is typically ranging from 50 to 80 scenarios depending on the inflow statistics.

We use MPI (Message Passing Interface) standard for parallel processing in the SOVN – application. This standard defines the syntax and semantics used. There are many implementations of MPI, we use Microsoft MPI (ms-mpi) which is Microsofts implementation of MPI. In SOVN, only a relatively small set of MPI-routines are applied. To facilitate parallelization in two levels, we utilize the concepts of communicators and communication handles.

The simulation involves the following iteration loops:

7.1.1 First level parallelization – Simulated scenarios

At this level the simulation of each weather scenario is independent of the other scenarios. One process is used as a master process or administrator. The master process identifies new tasks to be solved, i.e. week and scenario for the next weekly decision problem to be solved, selects working group (communicator) and sends initial reservoir levels for given problem to the working group. Information about week, scenario and reservoir levels are broadcasted from the master to the group. The master also receives the results from each weekly decision problem and stores the results.

Note also that at this level the number of processes in each group is irrelevant. At this level tasks are only distributed on groups of processes using the communicator handle identifying the relevant group allocated to solve the given task (weekly decision problem).

7.1.2 Second level parallelization – Decomposition

The weekly decision problem, defined in Section 1.2, which is solved in a two-stage decomposition scheme, can be solved using parallel processing, as illustrated in Figure 12. Synchronization between the two stages is needed, but the LP-problems for each second-stage scenario in the SFP are independent.

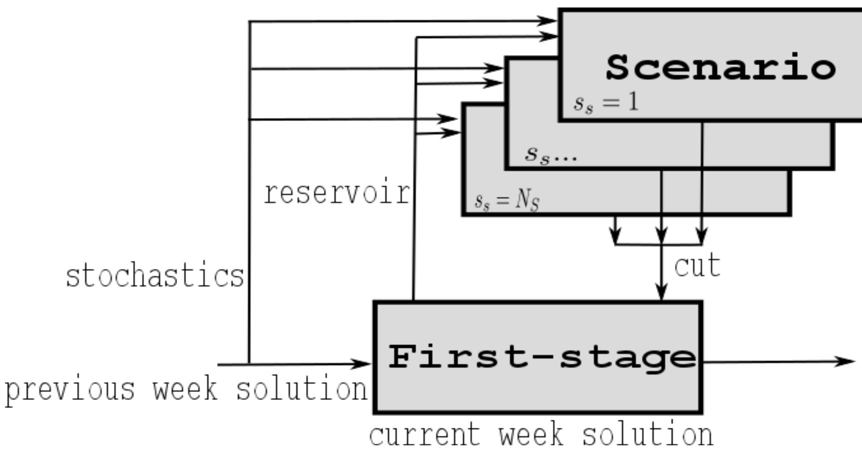


Figure 13 Illustration of parallelization level 2.

The two-stage decomposition results in a number of iterations where the first- and second-stage problems are solved with slightly different input. For the first-stage problem the number of constraints will increase as more cuts from the second stage are added, while the second-stage scenarios are updated only by initial reservoir levels found in the first-stage problem. Thus, there will be a significant gain in computation speed in later iterations by using so called warm start, i.e., using results from previous solutions as a start base. Therefore, the same second-stage scenario is always solved by the same process.

It is assumed that the number of processes available in the work group is either one or equal to the number of scenarios in the SFP plus one. With only one process this has to solve all the problems, i.e., the first-stage problem and the second-stage problems in the decomposed SFP. With a number of processes available equal to the number of scenarios in the scenario fan plus one, one process (named the grp_master) solves the first-stage problem and sends results to the master process/administrator. The other processes solve one scenario each.

Note that the iteration loops for the group master and the other processes are identical. All processes solving the weekly decision problem have to complete the same number of iterations. The group master solves the first-stage problem and sends the initial reservoir levels to the processes that solves the different scenario problems. The other processes solve their scenario problem and send the coupling data (cuts) to the group master. The group master checks for convergence and sends information to the other processes.

The current parallelization of the weekly problem is relatively inflexible, the tasks for each process is fixed. It also requires that the number of processes available for parallel processing equals the number of tasks (first-stage problem + number of scenarios). A lower number of processes than tasks would imply a significantly increase in either calculation time or use of internal memory. This is linked to the possibility of utilizing warm start. The most efficient utilization of warm start is probably applied when the LP-solver uses a single LP-process (allocates a separate set of data) for each scenario problem, but this would require very much memory. The second best option is to store a minimum set of information to provide start basis for a warm start. This is applied when only one process solves the weekly decision problem.

Based on the definition of communicator and group of processes a more flexible parallel processing logic may easily be applied at a later stage.

8 Other Computational Aspects

SOVN is an extremely computational demanding model. Significant (absolute) savings in computation time can be achieved by improving minor details in the computational scheme. Below we discuss some important factors.

8.1 LP solver and problem structure

SOVN currently runs using freely available COIN Clp solver [3] or the commercial solver CPLEX. These solvers are called through separate C/C++ interfaces. Our experience shows that CPLEX is much faster than COIN for large problem. For a typical model of the Nordic system the computation time with COIN can be 20 to 45 times longer than CPLEX depending on the time resolution. Figure 14 shows the solution time ratio COIN/CPLEX as function of time steps in the master problem. For small test problems the factors is only 2-3 and in some cases COIN may even be faster. For real problems of the Nordic systems, a commercial solver therefore is a must. The above-mentioned solvers where also benchmarked on a typical scenario problem in [12].

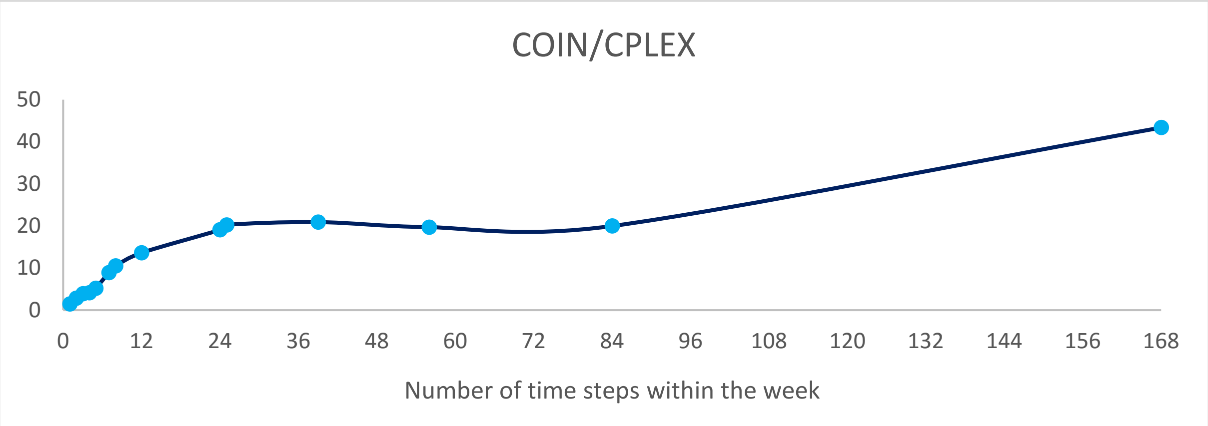


Figure 14 Ratio between COIN and CPLEX for some number of time steps for master week problem.

In a standard set-up, the second-stage (scenario) LP problems are much larger than the first-stage problem, and thus the majority of computation time is spent solving the second-stage problems. We keep the structure of the scenario problems constant (same number of variables and constraints) to facilitate efficient use of starting basis. Since scenarios are only marginally updated between subsequent Benders iterations, the use of starting basis (warm start) is crucial for computation time.

The calculation time for CPLEX also increase fast with increasing number of time steps, as shown in **Figure 15**. The example is taken from the scenario problem and the number of time steps include a mix of weeks and within week time steps. A scenario problem of 56 weeks with weekly time resolution have longer calculation time than a weekly problem with 3 hour resolution.

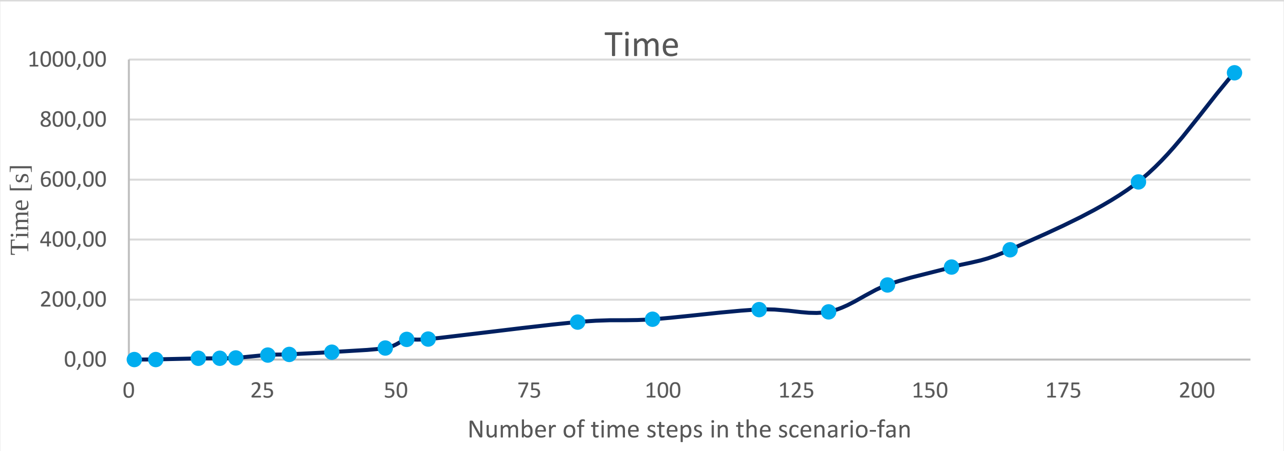


Figure 15. Computation time in seconds to solve one large scenario problem as function of number of time steps.

Total calculation time for one model of the Nordic power system is around 2 weeks (340 hours) with the following computer resources and model setup:

- 101 core/processes and using CPLEX 12.2
- master week with 3 hour resolution
- Planning horizon of 235 weeks (parallel simulation)
- 83 weather years
- Scenario length of 52 weeks using weekly resolution
- Using scenario reduction to 19 scenarios. Optimal parallel processing in two levels, the setup allows for 5 scenarios to be solved in parallel utilizing all available processes

Theoretical minimum calculation time with "unlimited" computational resources (1661 processes) for the same model setup is 20 hours.

Another model of Nordic power system with a different model setup is solved in 51 hours:

- 20 cores/process with CPLEX 12.2
- master week with 5 load periods
- 52 weeks (serial simulation)
- 51 weather years
- Scenario length of 51 week using weekly resolution
- Only parallel processing in the second level because of serial simulation mode

8.2 Flexible time resolution in scenario-fan

It is possible to have varying time resolution along the scenario-fan. This is done using several versions of type PRISAVSNITT.DATA files, see SOVN user manual. All data are input with the finest time resolution and accumulated to the chosen resolution for a given week in the scenario-fan. Master week is always solved with the finest resolution.

8.3 Naming variables and constraints

In the implementation phase we have focused on establishing a flexible model generator for building and adjusting LP problems. All variables and constraints are tagged to easily facilitate:

a) Extraction of results, both when preparing what shall be written to file and when using results internally in the model, e.g. when locating state variables to be used in Benders cuts.
b) Debugging of LP problems. By naming variables and constraints the resulting *.lp files that can be written by SOVN are much easier to read, see example in Figure 16. The three-letter codes can be changed by the user. As examples the codes 'sal' and 'buy' refers to sales and purchase of energy, respectively. This is an important tool for the developers, e.g. when adding new functionality or searching for bugs in the existing code.

```
Minimize
OBJROW: alp0 -25 sal0 -17 sal1 -15 sal2 -14 sal3 -13 sal4 -12 sal5 -10 sal6 -9 sal7 -8.500
 -8 sal9 -7 sal10 -7 sal11 -5 sal12 -0.400000 sal13 -6 sal14 -4 sal15 + buy1 + 9.600000 bu
 + 14 buy4 + 14.300000 buy5 + 15 buy6 + 17 buy7 + 18 buy8 + 19.799999 buy9 + 26 buy10 + 40
 + 56.700001 buy14 + 62.099998 buy15 + 62.099998 buy16 + 68.400002 buy17 + 68.400002 buy18
 + 208.300003 buy24 + 298.200012 buy25 + 298.200012 buy26 + 362 rat0 + 9 buy27 + 11 buy28
 + 68.750000 buy31 + 68.750000 buy32 + 68.750000 buy33 + 76.250000 buy34 + 76.250000 buy35
 + 85 buy41 + 116.033302 buy42 + 116.033302 buy43 + 116.033302 buy44 + 116.033302 buy45 +
 + 252.116699 buy51 + 252.116699 buy52 + 252.116699 buy53 + 366.899994 buy54 + 366.899994
Subject To
POW0:  - sal0 - sal1 - sal2 - sal3 - sal4 - sal5 - sal6 - sal7 - sal8 - sal9
 - sal10 - sal11 - sal12 - sal13 - dmp0 - sal14 - sal15 + buy0 + buy1 + buy2
 + buy3 + buy4 + buy5 + buy6 + buy7 + buy8 + buy9 + buy10 + buy11 + buy12
 + buy13 + buy14 + buy15 + buy16 + buy17 + buy18 + buy19 + buy20 + buy21 + buy22
 + buy23 + buy24 + buy25 + buy26 + rat0 + buy27 + buy28 + buy29 - exc0 + 0.980000 exc1
 + 0.990000 exc4 - exc5 + 0.763889 dis7 + 0.763889 dis8 + 0.729167 dis9 + 0.568182 dis1
 + 1.180556 dis21 + 0.102993 dis26 + 0.102993 dis27 = 14.991304
HYD13:  - spi11 - byp10 - byp11 - dis18 - dis19 + spi13 + byp13 + dis22 = 0
HYD14:  - spi12 - byp12 - dis20 - dis21 + spi14 + byp14 + dis23 + dis24 = 0.056738
```

Figure 16 Example: Naming of variables and constraints in *.lp file.

8.4 Memory use

The memory use in SOVN is potential high and use of several processes increase the memory consumption, both input data and time resolution in optimization are important factors.

For the input data, time resolution and the number of wind parks, price series and inflows are the main contributors. In SOVN, these data are handled by the administrator process which may require up to 6GB memory for a typical Nordic dataset.

Several measures are taken to reduce the memory consumption. In general, it is only the administrator that read and process information about the different input series. All processes know the main information as the number of hydro modules/pumps, first year, number of years, first week, number of weeks etc., but it is only the administrator that know all values (all wind power data, inflow statistics etc.). The other processes only know the values for a specified year and week (or scenario in the scenario-fan).

The main memory consumption for the group master and the slaves is the time resolution of the optimization problem and how detailed the power system is described (start cost, capacity reservation, ramping constraints ramping). For the master week, the memory consumption has not been a problem for the different power systems that are tested so far, but it can be a problem with large-scale parallelization and hourly time resolution. 56 and 168 load periods need up to 1.5 GB and 3.0 GB memory, respectively on each process. The memory requirements for the processes used to solve the scenario-fan are more or less the same as for the master but the number of time steps in scenarios are possibly higher.

It is possible to reduce the memory requirements of the model further, but as mentioned before current version of SOVN runs without memory problems. Possible memory reductions measures include reducing the number LP-problems in memory from three to two and more direct reading of data from file.

9 Risk aversion/calibration

The objective function in SOVN is minimization of expected operation costs which is equivalent to maximization of expected socioeconomic surplus. This is a reasonable criteria that gives model results independent of users preferences. This is one of the model strengths, calibration is not needed to give good and consistent results.

The EMPS model used to today for this type of analysis include a calibration possibility that the users use to influence model results. The calibration criteria is usually based on a combination of the following:

- Maximization of expected socioeconomic surplus
- Poor simulated operation of reservoirs
- Risk aversion shown by too risky simulated reservoir operation
- The need for some adaption to observed market behaviour based on model comparison with observed prices historically and in the forward market.

The weighting of the points above may differ between users and applications. The reason behind the two first points are mainly because of EMPS model deficiencies and simplifications; aggregation, disaggregation heuristics and optimization for aggregated models. The model does not necessary find the best solution without some help. These deficiencies should in principle not be present in SOVN because formal optimization is used. However the need for and reason behind the two last points may also be present for the SOVN model.

Therefore, we have implemented a rather simple version of a calibration/risk aversion type functionality. The details of this are described more thoroughly in the user manual. The methodology is based on use of two additional artificially made inflow scenarios that represent the most extreme (maximum and minimum of sum energy inflow) for each week, individually. These two scenarios are given weights by the user. Simplified; higher weights more risk aversion. We have tested that the implemented method works as expected but have not put much effort into more refinement of the method.

We are aware that more formal methods for modelling risk aversion like the CVaR (Conditional Value at Risk) criteria has been developed and applied to this kind of problems in recent years. We have not looked into these types of method in this project. In the end this criteria will also end up with user-defined parameters that affect operation and results.

10 Concluding Remarks

This report describes the model that has been developed in the project "Stokastisk optimaliseringsmodell for Norden med individuelle vannverdier og nettrestriksjoner". The model/software program is called SOVN and are the main result from the project.

The purpose of the project was to develop a hydro-thermal market model able to analyse the future electricity market with more renewables and stronger coupling to Europe. It was assumed from the beginning that such a model needed to be based on formal optimization and include a detailed hydro representation. The SOVN model fulfil these requirements and the results from the model, not shown in this report, are very promising especially with regard to consistency. With consistency, we mean that e.g. a more flexible system for example due to transmission investment should always have lower operation costs. Such a property is not guaranteed with models based on heuristics.

The main drawback with the new model is the computation time, which is very long, but this was expected from the beginning. The long computation time will, at least for now, limit somewhat the immediate applicability of the model.

SOVN is based on formal optimization which makes it easier to understand for new trainees and easier to adapt to new types of markets (e.g. balancing markets) or changes in system properties. SOVN is implemented as a simulator that solves two stage stochastic optimization problems in a sequence. Several parts of the model/program code will have other applications outside SOVN. The master problem that solves the weekly market clearing problem will be used and further developed in the MAD project [18].

Solving only one two stage problem, e.g. for the first week with given input reservoirs, give marginal costs (water values) for each individual hydro plant in the whole system. This is a very useful result that can be calculated in minutes depending on the time resolution. It can e.g. be used as input to separate short-term price forecasting models or the master problem itself can be the short-term price forecasting model if provided with good updated input data.

We believe that the new SOVN model and program code will be the basis for much of the future development related to hydro-thermal market modelling at SINTEF. Improved computing speed will increase the applicability of the model every year.

REFERENCES

[1] EMPS user manual. Technical report, SINTEF Energy Research, 2016.
[2] M. M. Belsnes, T. Follestad, O. Wolfgang, and O. B. Fosso. Aspects of stochastic models for short-term hydropower scheduling and bidding. Technical Report TR A7223, SINTEF Energy Research, 2012.
[3] Coin-CLP.
[4] A. Diniz and T. M. Souza. Short-term hydrothermal dispatch with river-level and routing constraints. *IEEE Transactions on Power Systems*, 29(5), 2014.
[5] M. Fodstad, A. L. Henden, and A. Helseth. Hydropower scheduling in day-ahead and balancing markets. In *Proc. of European Energy Markets (EEM)*, Lisbon, Portugal, 2015.
[6] A. Gjelsvik. Hydropower scheduling with variable water head. Technical Report TR F7617, SINTEF Energy Research, 2016.
[7] A. Gjelsvik and A. Haugstad. Considering head variations in a linear model for hydro scheduling. In International Centre for Hydropower, editor, *The backbone of Sustainable Energy Supply*, Trondheim, Norway, 2005.
[8] A. Gjelsvik, B. Mo, and A. Haugstad. *Handbook of Power Systems I*, chapter Long- and medium-term operations planning and stochastic modelling in hydro-dominated power systems based on stochastic dual dynamic programming, pages 33–55. Springer, 2010.
[9] N. Growe-Kuska, H. Heitsch, and W. Romisch. Scenario reduction and scenario tree construction for power management problems. In *IEEE Bologna PowerTech Conference*, 2003.
[10] A. Helseth. Prodrisk user guide. Technical Report AN 16.14.93, SINTEF Energy Research, 2016.
[11] A. Helseth. Stochastic network constrained hydro-thermal scheduling using a linearized progressive hedging algorithm. *Energy Systems*, 7(4):585–600, 2016.
[12] A. Helseth, B. Mo, K S Gjerden, and O Wolfgang. Stochastic optimization model with individual water values and power flow constraints. Technical Report TR A7375, SINTEF Energy Research, 2014.
[13] A. Helseth, B. Mo, and G.Warland. Long-term scheduling of hydro-thermal power systems using scenario fans. *Energy Systems*, 1(4):377–91, 2010.
[14] A. Helseth, G. Warland, B. Mo, and O. B. Fosso. Samnett – the emps model with power flow constraints. Technical report, SINTEF Energy Research, 2011.
[15] M. E. P. Maceira and M. V. F. Pereira. Analytical modeling of chronological reservoir operation in probabilistic production costing. *IEEE Transactions on Power Systems*, 11(1), 1996.
[16] E. Olaussen and A. Haugstad. Vannverdiberegning med informasjon om snømagasinet. Technical Report EFI TR A4104, EFI, 1993.
[17] K. M. Sidenblad and S. T.Y. Lee. A probabilistic production costing methodology for systems with storage. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(6), 1981.
[18] G. Warland, A. Haugstad, and E. S. Huse. Including thermal unit start-up costs in a long-term hydro-thermal scheduling model. In *Proc. 16th Power System Computation Conference*, Glasgow, Scotland, 2008.
[19] G. Warland and B. Mo. Stochastic optimization model for detailed long-term hydro thermal scheduling using scenario-tree simulation. *Energy Procedia*, 87:165–172, 2016.

Technology for a better society

www.sintef.no