NTNU
Norwegian University of
Science and Technology

Stochastic Mixed Integer Problems for Bidding in Sequential Electricity Markets

Scenariowise decomposition approaches

Anniken S Hofgaard
Ingrid Kårstad
Cecilie Skjong

Industrial Economics and Technology Management
Submission date: June 2016
Supervisor: Stein-Erik Fleten, IØT
Co-supervisor: Gro Klæboe, IØT
Asgeir Tomasgard, IØT

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

Problem description

In the competitive Nordic electricity markets, power producers try to develop bidding strategies that maximize their revenues. With lower prices in the day-ahead market, the producers are looking to other markets, seeking to maximize revenues by dividing capacity between different markets. This thesis focuses on bidding in the day-ahead and primary reserve market for a price-taking hydropower producer.

The resulting bidding problem is as a two-stage stochastic optimization problem, with continuous first stage and mixed-integer second stage variables. The stochasticity is represented by discrete price scenarios. To ensure stability the scenario tree quickly becomes large in size and the problem hard to solve. This thesis will investigate scenariowise decomposition methods in order to decrease solution time.

Preface

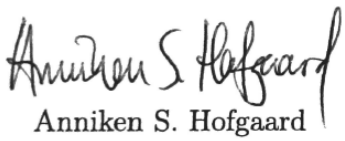
This master thesis is written within the field of Applied Economics and Operations Research at the Department of Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU). The thesis is motivated by the complexity of sequential electricity market modeling.

We would like to thank our supervisors, Professor Stein-Erik Fleten, Professor Asgeir Tomasgard and Postdoc. Gro Klæboe at NTNU for their valuable guidance and helpful discussions. Their interest and insight in the problem has been immensely valuable.

We would also like to express our gratitude to Professor Henrik Anderson and Postdoc. Magnus Stålhane at NTNU, for sharing their experience and insight in the implementation of column generation. In addition we would like to thank Postdoc. Unai Aldasoro, at NTNU, and Postdoc. Adela Pages Bernaus at Universitat de Lleida, for interesting discussions on decomposition of stochastic mixed integer models.

Lastly, we will thank our families for being supportive and understanding throughout the whole process.

Trondheim, June 10th, 2016



Anniken S. Hofgaard

Ingrid Kårstad

Cecilie Skjong

Abstract

With an increasing number of ancillary services and energy markets, the decision making process for a power producer is becoming more complex. This thesis investigates the bidding problem of a Norwegian hydropower producer bidding in the sequential primary reserve market and day-ahead market. The model implemented is a stochastic mixed integer programming (SMIP) problem with continuous first stage variables and mixed integer second stage variables. SMIP problems are known to be hard to solve and by having continuous first stage variables combined with second stage mixed-integer variables the complexity further increases. This thesis investigates the potential benefits of solving the problem by scenariowise decomposition. A comprehensive case study of the problem is conducted with realistic input data. Stochastic problems are dependent upon a good representation of the uncertainty and realistic price scenarios based on historical price data are therefore created.

The results show that scenariowise decomposition of the problem results in better optimistic bounds compared to solving the linear programming relaxation. When the number of scenarios grow, the decomposition methods perform better compared to solving the standard deterministic formulation of the deterministic equivalent. The decomposition methods are however sensitive to the size and complexity of the subproblems, resulting in low scalability with respect to the size of the subproblems. This results in longer solution times than what may be acceptable for a power producer. However, combined with heuristics the methods return bounds which can provide information about the quality of the heuristic solution.

Sammendrag

Med et økende antall reserve- og energimarkeder har budgivningsproblemet for kraftprodusenter blitt mer komplekst. Denne avhandlingen tar for seg budgivningsproblemet til en norsk vannkraftprodusent som sekvensielt byr kapasitet i primærreservemarkedet og deretter energi i døgnmarkedet. Den implementerte modellen er en stokastisk tostegsmodell med kontinuerlige variable i første steg og blandede heltallsvariable i andre steg. Stokastiske modeller med blandede heltall er kjente for å være vanskelige å løse, og med kontinuerlige variabler i første steg og blandede heltall i andre steg øker kompleksiteten ytterligere. Avhandlingen undersøker den potensielle nytten av å dekomponere problemet med hensyn på scenarier. Problemet er undersøkt gjennom et omfattende casestudie med realistisk inputdata. Stokastiske problemer er avhengige av god representasjon av usikkerhet, og realistiske prisscenarier som baserer seg på historisk pridata har derfor blitt generert.

Resultatene viser at dekomponering med hensyn på scenarier gir bedre optimistiske grenser enn den lineære relaksasjonen av problemet. Når antall scenarier øker, viser dekomponeringsmetodene bedre ytelse enn ved å løse den standard formuleringen av den deterministiske ekvivalenten. Dekomponeringsmetodene er derimot sensitive for økt kompleksitet i subproblemene. Når kompleksiteten øker blir løsningstidene lenger enn det som kan tenkes å være akseptabelt for en kraftprodusent. De øvre grensene fra dekomponeringsmetodene kan likevel være nyttige dersom de kombineres med heuristiske løsninger fordi de vil kunne gi nyttig informasjon om kvaliteten på de heuristiske løsningene.

Table of Contents

List of Tables

List of Figures

Nomenclature

Sets

S	Set of scenarios
H	Set of hours
B	Set of reserve blocks
H^b	Set of hours in reserve block b
I	Set of units in the portfolio
F	Set of segments used in the approximation of the production function
J	Set of reservoirs
J^w	Set of reservoirs in watercourse w
P	Set of bid prices
L	Set of end water values cuts
W	Set of watercourses
C	Set of scenario clusters
S^Ω	Set of scenarios in scenario cluster $\Omega \in C$
M	Set of extreme points

Indices

s	Scenario
h	Hour
b	Block
i	Generator
f	Segment
j	Reservoir
p	Bid price
l	End water value cut
w	Watercourse
Ω	Scenario cluster
m	Extreme point

Parameters

Q_i^{max}	Maximum production level of generator $i \in I$
Q_i^{min}	Minimum production level of generator $i \in I$
P_{bp}	Bid price $p \in P$ in block $b \in B$
Pr_s	Probability of scenario $s \in S$
A_{if}	Constant of production function for segment $f \in F$ for unit $i \in I$
B_{if}	Slope of production function for segment $f \in F$ for unit $i \in I$
C_i^{su}	Start-up costs for unit $i \in I$
D_i^{su}	Maximum discharge for unit $i \in I$
I_{hj}	Inflow in reservoir $j \in J$ in hour $h \in H$
F_{lw}	Future income for cut $l \in L$ in watercourse $w \in W$
W_w^0	Initial water value in watercourse $w \in W$
W_{jl}	Marginal water value for cut $l \in L$ for reservoir $j \in J$
V_j^0	Initial water volume in reservoir $j \in J$
V_{jl}	Evaluated reservoir level for cut $l \in L$ for reservoir $j \in J$
U_i^0	1 if a generator is on in its initial state, 0 if it is off for generator $i \in I$
ρ_{sb}^R	Reserve market clearing price for block $b \in B$ for scenario $s \in S$
ρ_{sh}^D	Day ahead market clearing price for hour $h \in H$ for scenario $s \in S$
$\pi_{spp}^{(k)}$	Step length for the Lagrangian multiplier in iteration k for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
$t^{(k)}$	parameter for updating the step length of the Lagrangian multipliers in iteration k
τ	Penalty factor in the objective function of the proximal bundle method
p	Penalty factor in the objective function of progressive hedging
δ_{sbk}^-	Penalty on the surplus variable in the non-anticipativity constraints for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
δ_{sbk}^+	Penalty on the slack variable in the non-anticipativity constraints for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
ε^-	Bound on the surplus variable in the non-anticipativity constraints
ε^+	Bound on the slack variable in the non-anticipativity constraints

Indicator matrices

Γ_{ij}	-1 draws from, 1 discharge into, 0 no connection for reservoir $j \in J$ and unit $i \in I$
$\Lambda_{jj'}$	-1 spills from, 1 spills into, 0 no connection from reservoir $j \in J$ to reservoir $jj \in J$

Variables

$\bar{z}_{bp}$	Common target variable for the capacity bid in the reserve market for bid price $p \in P$ for block $b \in B$
z_{sbp}	Capacity bid in the reserve market for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
x^D_{sh}	Contract in the day-ahead market for hour $h \in H$ in scenario $s \in S$
x^R_{sb}	Contract in the reserve market for block $b \in B$ for scenario $s \in S$
q^D_{shi}	Production by unit $i \in I$ in hour $h \in H$ in scenario $s \in S$
q^R_{shi}	Delivery of reserves by unit $i \in I$ in hour $h \in H$ in scenario $s \in S$
u_{shi}	1 if unit $i \in I$ is committed in hour $h \in H$ in scenario $s \in S$, 0 otherwise
d_{shi}	Discharge by unit $i \in I$ in hour $h \in H$ in scenario $s \in S$
v_{shj}	Reservoir volume in reservoir $j \in J$ in hour $h \in H$ in scenario $s \in S$
s_{shj}	Spill from reservoir $j \in J$ in hour $h \in H$ in scenario $s \in S$
w_{sw}	Approximated end water value in watercourse $w \in W$ in scenario $s \in S$
c_{shi}	Induced start-up cost for unit $i \in I$ in hour $h \in H$ in scenario $s \in S$
μ_{sbp}	Lagrangian multiplier for the relaxed non-anticipativity constraints for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
$\hat{\mu}_{sbp}$	Stability center of the Lagrangian multiplier for the relaxed non-anticipativity constraints for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
$g^{(k)}_{sbp}$	Subgradient of the non-anticipativity constraints in iteration k for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
θ_s	Cutting plane dual value for scenario $s \in S$
$\omega^{(k)}_{sbp}$	Progressive hedging multiplier for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
$\lambda^{(m)}_s$	Weight on extreme point $m \in M$ for scenario $s \in S$
α_{sbk}	Dual variables of the non-anticipativity constraints for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
β_s	Dual variables for the convexity constraints for scenario $s \in S$
y^-_{sbk}	Surplus variable in the non-anticipativity constraints for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$
y^+_{sbk}	Slack variable in the non-anticipativity constraints for bid price $p \in P$ for block $b \in B$ in scenario $s \in S$

Functions

$D(\cdot)$	Lagrangian dual function

Chapter 1

Introduction

In the past years, almost all areas within the European Network of Transmission System Operators for Electricity (ENTSO-E) have experienced an increasing number of frequency deviations and an increase in the amplitude and duration of the deviations (Eurelectric/ENTSO-E, 2011). These patterns are most prominent at the change of hour in the morning and evening due to higher demand and correspondingly higher production during these periods of the day. Frequency deviations are also the result of critical stochastic events such as outages of power plants and loads. The increased amount of reserves needed to counteract the deviations at the change of hour necessarily reduces the available reserves in case of stochastic events. The result is a tighter reserve limit and a risk of insufficient availability of reserves in the case of an outage. These challenges are changing the way power markets are operated.

Over the past 25 years, electricity markets in the Nordic countries have been deregulated in order to improve efficiency and increase security of supply. This means that the operation of power markets have moved from being centralized to competitive, and that the operation of generating units is subject to market rules. To keep a secure supply of electricity and keep the system in balance, several markets have been established. The goal is for markets further away from operating time to settle the majority of demand, while markets closer to operating time take care of smaller deviations (Jiang and Powell, 2015).

Today, by far the largest amount of electricity is traded in the day-ahead market, but the demand for reserves is expected to grow in the years to come. This is due to a closer connected European power system and a larger share of non-flexible renewable, intermittent energy. The intermittent, renewable resources are characterized as being non-dispatchable. Hence, the generating units cannot be regulated in order to match changes in demand. The remainder of the generation capacity consequently has to complement the variability of the non-flexible resources (Lorubio, 2011). This will increase the potential revenue for power producers from bidding in this market. With expected lower prices in the day-ahead market, the producers are also looking to other markets, seeking to maximize revenues by dividing capacity between different markets.

Thus, several markets can provide revenue for power producers, but the amount bid in one market reduces the possible amount that can be bid in other markets. Coordinating

the multi-market bidding decisions can potentially increase the revenue of the power producers, but the bidding decisions are interdependent and the decision problem becomes more complex. There is therefore an increasing need for decision support models that are consistent with the new market conditions (Ventosa et al., 2005).

Much of the research on multi-market bidding has been within the field of optimization. Optimization models consider a single supplier, whose objective is to maximize profit given either exogenous prices, or prices as a function of the producer's decisions. This thesis is a contribution to the optimization literature. We present a stochastic mixed integer programming (SMIP) formulation based on work by Klæboe (2015a) for a short-term hydropower bidding and scheduling problem of a Norwegian hydropower producer who bids in the sequential primary reserve market and day-ahead market and whose objective is to maximize profit. The model includes continuous first stage variables for bidding in the primary reserve market and mixed-integer second stage variables for scheduling generators and bidding in the day-ahead market.

The stochasticity of the model is used to capture the inherent uncertainty in market prices. When incorporating stochasticity in the model, the bidding decisions reflect the possible corrective actions that can be made after the market price has been settled. This flexibility is not captured when uncertainty is ignored. A realistic representation of the uncertainty usually requires a large number of scenarios, resulting in stochastic models being large in size. The problem presented is, in addition, further complicated by the presence of mixed-integer variables in the second stage, making the recourse function lower-semicontinuous with respect to the first stage variables. Such models are particularly difficult to solve within reasonable time.

Due to the size and complexity of the large scale SMIP model considered, it cannot be solved within reasonable time by using off-the-shelf solution methods. Decomposition methods that exploit the structure of the stochastic problem are therefore considered as applicable. Decomposing the problem into smaller subproblems can contribute to mitigate part of the computational difficulty associated with solving larges-scale SMIP problems. The purpose of the thesis is to investigate whether scenariowise decomposition methods can be used to speed up the solution process of solving SMIP models for sequential market bidding that contains continuous first stage variables and mixed-integer second stage variables.

We investigate two different scenariowise decomposition methods that are applicable to a wide range of problems. Essentially, scenariowise decomposition methods exploit the scenario-structure of the stochastic problem by decomposing the problem into one subproblem for each scenario. The methods two methods investigated are Dantzig-Wolfe decomposition and Lagrangian decomposition. They are compared with respect to their convergence properties and the quality of the optimistic bounds they provide compared to each other, and to the linear programming (LP) relaxation of the deterministic equivalent.

We show that the decomposition methods can outperform the LP relaxation with respect to bounds, but they suffer from long solution times due to high complexity of the subproblems associated with the large number of binary unit commitment decisions. Using early stopped solutions from the decomposition methods as a starting point for heuristics,

thereafter combining the heuristic solutions with the valid optimistic bounds provided by the decomposition methods, are therefore considered as relevant practical application of the decomposition methods.

Effort is also put in making a realistic representation of the uncertainty of the stochastic model since high quality results from the stochastic model is dependent on high quality scenarios that manage to capture the characteristics of the separate price processes and their interdependency. A Realistic discrete representations of the joint distribution of day-ahead and primary reserve prices have been achieved by combining time-series analysis and principal component analysis (PCA) with moment matching.

The contribution of this thesis is threefold. To our knowledge, no other work has analysed coordinated multi-market bidding in the Nordic primary reserve market and day-ahead market. Secondly, the problem investigated is a two stage stochastic mixed integer problem (SMIP) with continuous first stage variables and mixed integer second stage variables. Little work has been done investigating decomposition methods for problems with the characteristics of first stage continuous and second stage binary variables. Lastly, the approach of combining principal component analysis with moment matching to create joint discrete scenarios for the panel of prices in the day-ahead and primary reserve market has, to our knowledge, not been used to generate discrete scenarios of electricity prices before.

The implemented model has been tested on data from two watercourses in the NO2 price area operated by Hydro ASA. Hydro is a large integrated global aluminium company with businesses in the whole value chain of aluminum production, and is today the second largest hydropower producer in Norway with an installed capacity of 10 TWh (Hydro, 2016).

The thesis is structured in 11 chapters. Chapter 2 begins with a presentation of the bidding problem for power producers operating in competitive electricity markets. This problem is often modeled as a large-scale stochastic problem and the remainder of the chapter presents the characteristics of a stochastic problem and discusses decomposition methods that exploit the structure of the problem to decrease solution times. Chapter 3 is devoted to —a detailed presentation of the scenariowise decomposition methods that have been implemented, Lagrangian and Dantzig-Wolfe decomposition. Chapter 4 presents the fundamentals of the power markets included in the model and the functioning of a hydropower plant. The SMIP model implemented and analyzed in the thesis is presented in Chapter 5. In Chapter 6 the implementation of the decomposition methods is described. Chapter 7 is devoted to an analysis of the historical primary reserve prices and day-ahead prices and scenarios are thereafter generated. A description of the cases studied is presented in Chapter 8. Results from solving the deterministic equivalent are presented in Chapter 9, in which an analysis is conducted to study the affect different input parameters have on solution times and the quality of the solutions. From this discussion two test instances, that are tested with the decomposition methods presented in Chapter 6, are presented. In Chapter 10, a comprehensive computational study of the decomposition methods is presented. Chapter 11 provides concluding remarks and points at potential areas of future research.

Chapter 2

Optimization theory in electricity market modeling

In this chapter, we present the bidding problem in more detail with references to relevant literature. We then relate the characteristics of the problem to optimization theory and show how it can be represented as a stochastic mixed integer program (SMIP). We give a general introduction and develop a notation for stochastic programs, with main focus on SMIPs. Exact algorithms for solving stochastic programs containing integer decision variables are known to be hard to solve and the solution time grows exponentially with problem size. The final part of this chapter is therefore devoted a review of solution approaches that can be applied to SMIPs to solve this type of problems more efficiently.

2.1 The bidding problem

There exist a variety of approaches to model the behaviour of participants in the power markets. The types of models that exist, differ in time scope, uncertainty, interperiod links, transmission constraints, market representation and game theoretic aspects included (Ventosa et al., 2005). Ventosa et al. (2005) classify the modeling approaches in three categories; equilibrium models, simulation-based models and optimization models. Equilibrium models consider several firms with an overall objective of profit maximization. Models of this type represent the overall market behaviour by using market equilibrium frameworks that include competition between firms and the concept of Nash equilibriums. An example of a paper modeling the electricity markets with an equlibrium model is Baldick et al. (2004). Electricity market models are however often considered to be too complex to solve using equilibrium models and simulation-based models then become an alternative. Simulation-based models are based on simulation instead of exact mathematical modeling. These models are known to be flexible and they facilitate incorporation of complex assumptions. Equilibrium models and simulation models are suitable for long-term planning decisions and analysis of the competitive behaviour of the market participants.

The third class of models, optimization models, considers a single supplier whose objective is to maximize profit, given either exogenous prices or prices as a function of the producer's decisions, subject to operating constraints. Optimization models include more details than the equilibrium and simulation-based models and are therefore better suited for modeling short-term decisions (Ventosa et al., 2005). Optimization models can be divided in two subgroups; single-market and multi-market models.

Single-market models typically consider only the day-ahead market, in which the largest quantities of electricity are traded. The day-ahead market is a short-term market where electricity is traded one day before delivery. The majority of the models of the day-ahead market, focus on constructing optimal bidding curves. A review of different modeling techniques and solution approaches can be found in Kwon and Frances (2012). Construction of the bids depend on the market rules and the physical and economical restrictions on the operation of the power plant. In many models the producer is modeled as a price taker, meaning that the producer has no capability to alter the market prices. Prices are then included as exogenous parameters. Examples of this approach comprise e.g. Conejo et al. (2002) and Fleten and Kristoffersen (2007). In other models, prices are modeled as a function of the firms decisions. This is often done by incorporating the bidding behaviour of competing producers in the optimization model as in Wen and David (2001). Such bidding is also denoted strategic bidding, and a survey covering this topic can be found in David and Wen (2000).

With the increasing number of electricity markets that have been established, market participants are becoming more aware of the opportunities offered by these markets (Fjelldal et al., 2014). In this regard, single market models have their limitations; they do not consider the sequential structure of the electricity markets and cannot capture the opportunity cost associated with bidding in several markets. Thus, multi-market models become relevant. Some references that model optimal bidding in sequential electricity markets are Ugedo et al. (2006) who model a power producer optimizing the bid in a daily, secondary and first intraday market, Boomsma et al. (2014) and Faria and Fleten (2011) who model sequential bidding in a day-ahead market and a balancing market and Attaviriyanupap et al. (2005) who optimal bidding in a day-ahead market and primary reserve market.

When solved by optimization techniques, the bidding problem is usually formulated as a unit commitment (UC) problem (Takriti et al., 2000). The UC problem consists of deciding which units to run in each time step and the amount to dispatch from every generating unit, with the objective of profit maximization (Wallace and Fleten, 2003). The unit commitment problem is an important problem in the operations of power plants, because the problem is aimed at reducing the power production cost by optimal scheduling of the generating units (Zheng et al., 2013). A bibliographic survey covering the unit commitment problem can be found in Padhy (2004). When including the unit commitment decision, the problem is often model as a mixed integer linear programming (MIP) model due to the need to include binary variables to model the on-off decision of the producer. Such models are non-convex and non-differentiable, thus including binary variables results in models of high complexity. If the unit commitment decision is made ex-ante of the bidding decision, other modeling frameworks can be used, such as in Dicorato et al. (2009). Other approaches to reduce the complexity of the unit commitment decision is

by relaxing the binary restrictions, and model the unit commitment decision by partial start-ups (Klæboe, 2015b). There is however a trade-off between the complexity of an optimization model and the realism it provides.

Another important aspect affecting the bidding decision, and that can be incorporated in optimization models, is uncertainty in the input parameters. Stochastic programming is an often used method for optimization problems that includes uncertainty and has become an important modeling framework within the energy sector. A survey of the use of stochastic programming in both regulated and deregulated electricity markets is written by Wallace and Fleten (2003). In deregulated electricity markets, producers have to commit generators without having a certain forecast of the future clearing price. The decision made today should reflect the possible corrective actions that can be made after the market price has been settled. Moreover, the decision made today should be taken with the objective of minimizing the possible cost of recourse since operational decisions often involve irreversibility. A stochastic programming approach is therefore a meaningful and natural framework to use (Kwon and Frances, 2012). This can also be shown to be even more important in sequential electricity markets, in which the decisions made for one market can limit the decisions that can be be made in consecutive markets. Fleten and Kristoffersen (2007) provide an example of using stochastic programming in electricity market modeling. They explore the effect of using a stochastic model compared to a deterministic model for a hydropower producer bidding into the Nordic day-ahead market. The expected profit is shown to be larger in the stochastic model in which the uncertainty in day-ahead market prices is included. This supports modeling the bidding and scheduling problem with stochastic parameters.

When including the unit commitment decision in a stochastic program, there is a question of when the unit commitment decision is made and in what stage it should be included. The timing of the decision is closely related to the flexibility of the power station (Zheng et al., 2013). For flexible power plants, such as hydropower plants, the unit commitment decision can be made close to operating time. Thus, the unit commitment decisions can be included in later stages when additional information from observing realized prices can be taken advantage of. See Faria and Fleten (2011) for an example of this approach. For less flexible plants, such as large coal-fired power plants, the unit commitment decisions have to be taken a longer time in advance since it usually takes longer time for less-flexible generators to be committed (Carøe and Schultz, 1998). The unit commitment decisions must then be included in earlier stages. The complexity of the stochastic problem and the solutions approaches differ depending on the type of variables in each stage. Thus, in which stages the unit commitment decision is included is an important determinant for the complexity of the model.

Different modeling and solution techniques have been used within electricity market modeling for problems that include uncertainty. Historically, the stochastic dynamic modeling approach has been widely applied. Stochastic dynamic programming (SDP) is a solution approach in which the problem is solved by recursively maximizing profit in the current stage and the expected profit in future stages. An example of this approach given by Helseth et al. (2015) who optimize sales of energy and capacity in a multi-market setting. There are however limitations associated with a using SDP, the method cannot handle very complicating constraint and problem definitions. It is furthermore limited

by the stochasticity it can handle (King and Wallace, 2012a). Stochastic dual dynamic programming (SDDP) is a variant of SDP in which all future states do not need to be explored. Stochastic programming is another well-known approach in which the uncertainty is represented by scenarios. This approach is explored in the thesis, and is therefore further described in the next section. Among the benefits of modeling the problem by scenario-based stochastic programming is that the method facilitates using well-studied approaches within the area of stochastic and linear or mixed-integer programing. On the other hand, the size of the problem grows exponentially with the number of scenarios, which complicates common solution-approaches.

2.2 Stochastic mixed integer programming

Stochastic programs are optimization models where some of the data incorporated into the objective function or constraints are uncertain. Models of this type contain several decision stages. A stage can be defined as a point in time where decisions can be made based on the revelation of new information. In this thesis, two-stage stochastic programs are considered. For the two-stage case, decision variables are divided into two groups; first-stage and second-stage variables. First-stage decisions are decided upon before realization of the uncertain parameters, whereas the second-stage decisions, also called recourse decisions, are made after the realization of the uncertainty. Consequently, the second-stage decisions capture the opportunity to adapt to the new information. This flexibility is usually not captured when uncertainty is ignored.

A standard two-stage stochastic maximization problem, has a structure as given below (Lubin et al., 2013),

$$z = \max \{cx + Q(x) : \quad Ax \leq b, \quad x \in X\} \tag{2.1}$$

in which $Q(x)$ is the recourse function given by

$$Q(x) = \mathrm{E}_{\xi} [\min \{q(\xi)y : \quad W(\xi)y \leq h(\xi) - T(\xi)x, \quad y \in Y\}] \tag{2.2}$$

The objective coefficient vector for the first stage variables in the vector x is c, $q(\xi)$ is the objective coefficient vector of the second stage variable vector y, ξ is a random variable with $|S|$ possible realizations and the sets X and Y express the feasible first and second stage decisions. The variables x and y can be defined as continuous or integer, or a combination of the two. When the stochastic problem contains both integer and continuous variables, it is characterized as a SMIP problem. During the solution process of solving a SMIP that contains integer variables, it is of interest to know the distance between the best integer solution found and the upper bound of the problem. This distance can be denoted a duality gap (Beasley, 1996). Note that the same terminology is used to describe the difference between the primal and dual solution when solving a linear optimization problem, but must not be confused with this in the thesis.

Stochastic programs can be categorized based on their type of recourse. If defining the set of first stage feasible set, $K_1 = \{x \in R_+^{n_0} | Ax \leq b\}$ and the second stage feasible set

$K_2 = \{x \in R^{n_0} | Q(x) \leq \infty\}$, a problem is said to have complete recourse if the second-stage is feasible for any value of x. The problem is characterized as having relatively complete recourse if $K_1 \subseteq K_2$, meaning the second-stage is feasible for any x that is also feasible in the first-stage (Zenios and Ziemba, 2006). The recourse of the stochastic problem is also given by the type of variables in the recourse function, for example integer or mixed-integer recourse.

The uncertainty of the above model is approximated by a discrete probability distribution, and the possible outcomes are represented by scenarios. When considering each of the possible scenarios explicitly in the model, we get the deterministic equivalent formulation of the problem as given in (2.3), in which Pr_s is the probability of scenario s, and Z_s represents the feasible solutions of the variables when scenario s is realized.

$$z = \max \left\{ cx + \sum_{s \in S} Pr_s q_s y_s : \quad (x, y_s) \in Z_s, s \in S \right\} \tag{2.3}$$

Alternatively we can introduce a separate first stage variable for each scenario and explicitly define non-anticipativity constraints to ensure feasibility across scenarios, a so-called *split-variable* formulation. The split-variable formulation results in the deterministic equivalent being formulated as in (2.4).

The non-anticipativity constraints comprise a set of common constraints that link the first stage-decisions together. This results in the problem having a dual block angular structure characterized by common variables and the dual having a primal block-angular structure (Birge and Louveaux, 2011). These structures are illustrated in Figure 2.1.

$$z = \max \left\{ \sum_{s \in S} Pr_s (cx_s + q_s y_s) : \quad (x_s, y_s) \in Z_s, s \in S, \hat{x} = x_1 = ... = x_{|S|} \right\} \tag{2.4}$$

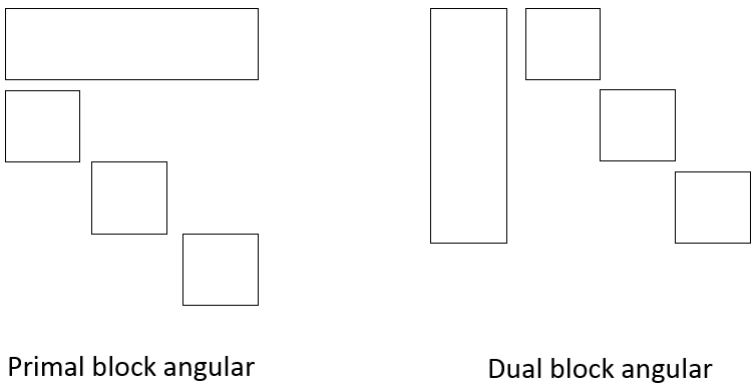


Figure 2.1: Primal and dual block angular structure.

There exist several ways to represent the non-anticipativity constraints. Introducing an additional variable $\hat{x}$, the non-anticipativity constraints can be represented by (2.5). Another representation, that does not include an additional variable, is given by (2.6) for a suitable matrix H that can be defined such that cyclic or *all-versus-all* non-anticipativity

constraints are defined. In theory, these representations are equivalent, but as we will see in Chapter 3, one may be more advantageous than the other depending on the solution algorithm implemented.

$$x_s - \hat{x} = 0 \qquad s \in S \tag{2.5}$$

$$\sum_{s \in S} H_s x_s = 0 \tag{2.6}$$

Evaluating the results from a stochastic optimization model can be done by measuring the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS) (Birge and Louveaux, 2011). The EVPI measures the value of receiving perfect information of the realization of future events. Hence, the EVPI is a benchmark for the value of the stochastic model. For a maximization problem, the EVPI is given by (2.7), in which WS is the value of the wait-and-see solution as given by (2.9), and is RP the value of the recourse problem solution as given by (2.10). The wait-and-see solution is found by removing the non-anticipativity constraints, solving each scenario separately and thereafter finding the expected value. The VSS on the other hand, measures the value of solving the model with stochastic parameters compared to using the expected value of the uncertain parameters. For a maximization problem the VSS is given by (2.8), in which EEV is the expected value of the expected value solution given by (2.11). The optimal decision of the expected value problem, $\bar{x}(\bar{\xi})$, is found by solving the optimization problem when replacing the random variables by their expected value. The relationship between the EEV, RP and WS is given by $EEV \leq RP \leq WS$ for a maximization problem (Birge and Louveaux, 2011).

$$EVPI = WS - RP \tag{2.7}$$

$$VSS = RP - EEV \tag{2.8}$$

$$WS = E_\xi[\max_x z(x, \xi)] \tag{2.9}$$

$$RP = \max_x E_\xi z(x, \xi) \tag{2.10}$$

$$EEV = E_\xi(z(\bar{x}(\bar{\xi}), \xi)) \tag{2.11}$$

.

2.3 Decomposition of stochastic mixed integer programs

Stochastic programming (SP) and mixed integer programming (MIP) are both fields that have solid mathematical foundations and several algorithms and software exist to solve problems of these classes. See for example Birge (1997) and Geoffrion and Marsten (1972) for surveys of solution procedures for SP and MIP problems respectively. The fields of

SP and MIP have however developed independently, and for many years little interest was shown around the combination of the two fields, SMIP problems (Sen, 2005). The applications of SMIP problems however arise in an increasing number of areas (Zhu, 2006), and it is therefore of large interest to find efficient algorithms to solve such problems.

Solving the deterministic equivalent of the SMIP problem is equivalent to solving a large-scale MIP problem. Thus, any solution approach for a general MIP problem is applicable (Zhu, 2006). Some of the many solution methods for MIP problems are guaranteed to give exact solutions. Other methods only provide heuristic solutions that may converge to the optimal solution for certain problem instances but that cannot guarantee convergence in all cases. The explosion in problem size associated with including a large number of scenarios in a SMIP problem complicates the common solution procedures. The dual-angular structure of the stochastic program however enables decomposition of the problem. In many cases it will be computationally advantageous to first separately solve a series of smaller MIP problems, and ultimately combine these solutions into a solution of the large SMIP problem (Sen, 2005). Exploiting the structure of the SMIP problem by decomposition is therefore interesting.

Depending on when (in what stage) the integer decisions are made, different types of SMIPs arise. Stochastic problems with continuous second stage variables are generally easier to solve than stochastic problems containing integer second stage variables. This is because the value function is continuous with respect to the first stage variables when the second stage is continuous. When the second stage variables are integer, the recourse function is however lower semi-continuous with respect to the first stage variables, making the second stage problem non-convex. Given any first stage decisions and the realization of the uncertainty, the second stage decision is an integer programming problem which by itself is a difficult optimization problem. These kinds of problems turn out to be more difficult to solve.

Many solution methods also take advantage of integer variables in the first stage during the solution process. Thus, some methods are constrained by only being applicable when the stochastic program contain integer variables in the first stage. For problems containing continuous first stage variables, and integer second-stage variables the solution methods become more challenging. Few algorithms to solve such problems exist. Based on the importance of variable restrictions in SMIP problems, this will be emphasized in the following presentation of the decomposition methods.

Two of the main groups of decomposition methods for stochastic problems are scenario-wise decompostion (also known as primal decomposition or price directive decomposition) and stagewise decomposition (also known as dual decomposition or resource-directive decomposition). As the name suggests, the problem is decomposed with respect to scenarios in scenariowise decomposition and with respect to stages in stagewise decomposition. In the following, some of the main decomposition methods within the two categories will be presented. The methods will also be classified based on their applicability for only two stage or multi-stage scenariotrees. An overview of the decomposition methods and their characteristics are thereafter summarized in Table 2.1.

2.3.1 Stagewise decomposition

The most widely used decomposition method for solving two-stage SP problems is the L-shaped method developed by Van Slyke and Wets (1969). The L-shaped method is a stagewise decomposition method that can be compared to applying Benders decomposition (Benders, 1962) to a stochastic problem. The original L-shaped method requires convexity of the feasible region, meaning that both the second stage variables need to be continuous. Laporte and Louveaux (1993) developed an integer L-shaped method which extended the L-shaped method to include a branch and cut procedure that can be applied in the presence of binary first stage variables and arbitrary second stage variables. Due to the need to find valid optimality cuts, the method however works best if all first stage variables are binary or all second stage variables are integer (Zhu, 2006). Building on this work, Carøe and Tind (1998) developed a general framework for the L-shaped method in the presence of integer recourse variables. Their work did not address the computational difficulties associated with the method applied to problems with such characteristics, but contributed to close the conceptual gap between MIP and SMIP problems.

Other stagewise decomposition methods for stochastic problems that involve a binary first stage and an integer or mixed integer second stage are the disjunctive decomposition (D^2) algorithm (Sen et al., 2003; Sen and Higle, 2005) and a modified Benders algorithm using the reformulation-linearization technique (RLT) to generate cuts (Sherali and Fraticelli, 2002). For problems with only few scenarios in the second stage, decomposition with branch and cut (D-BAC) has been successfully applied, while for problems with many scenarios in the second stage, the D^2 with branch and cut (D^2-BAC) has shown to be more suitable (Sen and Sherali, 2006). Ntaimo and Sen (2007) present a branch and cut algorithm called D^2-CBAC for SMIPs with continuous first stage variables and finite convergence is established for mixed-binary second stage. The method is based on the D^2 method to derive valid inequalities to the second stage of the SMIP. Branching is done on the first-stage continuous domain, which on its own does not guarantee finiteness. The branch and bound search is therefore guided by disjunctive variables in the second stage, whose number is finite and therefore leads the branch and bound algorithm to terminate in a finite number of steps. Sen et al. (2003) summarize these and other methods for solving two-stage SMIPs using stagewise decompositon.

2.3.2 Scenariowise decomposition

Lagrangian relaxation is a well-known decomposition method for large scale IP and MIP optimization problems that can be applied to both two-stage and multi-stage stochastic problems. The method is applicable to problems with integer restrictions on the variables in the first stage and/or the second stage since the method does not take advantage of specific variable restrictions in the solution procedure. Consequently, the method is applicable to solve quite general SMIP problems (Sen, 2005). Several different updating schemes to update the Lagrangian multipliers exist, resulting in a wide variety of solution approaches proposed within the Lagrangian literature. A survey of different multiplier updating procedures can be found in Bertsekas (2011). The duality gap from using Lagrangian relaxation might be large and Carøe and Schultz (1999) developed the dual

decomposition method for stochastic programs to decrease the duality gap when using Lagrangian relaxation. The method is based on Lagrangian relaxation combined with branch and bound.

Another method to decompose the stochastic problem by scenarios is Dantzig-Wolfe decomposition. The decomposition method was first presented by Dantzig and Wolfe (1960), and is one of the most used tools to solve large and complex optimization problems. The method is applicable to solve general SMIP problems containing both pure and mixed continuous/integer or binary decision variables. When using Dantzig-Wolfe decomposition on problems with integer restrictions, the integrality restrictions can be imposed through a branch and bound procedure, also known as branch and price when combined with column generation. The branch and price procedure can result in large branching trees and heuristics can then be applied to attempt to reduce the size of the trees and to find feasible solutions earlier. Dantzig-Wolfe decomposition is furthermore applicable to multi-stage stochastic programs. An algorithm to solve multi-stage SMIP problems based on column generation combined with branch-and-price is presented in Lulli and Sen (2004). The method they present handles variable restrictions requiring all or some of the variables in all stages to be integer. Another example of decomposing a stochastic program with Dantzig- Wolfe reformulation by scenariowise decomposition is presented in Schulze et al. (2015).

It is also important to note that Lagrangian relaxation and Dantzig-Wolfe reformulation can be applied to decompose the stochastic problem based on units (generating units, ships, cars etc.). In Nowak and Römisch (2000) a dynamic (multi-stage) stochastic programming model is decomposed using Lagrangian relaxation by assigning multipliers to constraints coupling power units. Shiina and Birge (2004) decompose the stochastic unit commitment problem with Dantzig-Wolfe decomposition by imposing separate subproblems for each generating unit. Their method is based on the work of Takriti et al. (2000).

Another scenariowise decomposition method that can be leveraged to solve multi-stage SMIP problems is progressive hedging. The standard progressive hedging algorithm is presented in Rockafellar and Wets (1991). It was originally developed for problems containing only continuous variables, but has later been successfully applied as a heuristic solution procedure for stochastic programs with integer variables in both stages, see for example Løkketangen and Woodruff (1996) and Fan and Liu (2010).

Scenariowise decomposition mitigates part of the computational difficulty associated with solving large problem instances when the number of scenarios increases. The subproblems, which are associated with one scenario each, are much smaller than the original SMIP, and they can be solved in parallel. The methods furthermore exploit the general structure of stochastic programs and scenariowise decomposition methods remain applicable to a wide range of problems, including multi-stage models.A summary of the classifications of the decomposition methods presented is given in Table 2.1.

Table 2.1: Stagewise and scenariowise decomposition methods

	Decomposition method	**Stages**	**First stage variables**	**Second stage variables**
Stage-wise	L-shaped method	2	Binary	Continuous
	Integer L-shaped	2	Binary	Mixed-integer
	Disjunctive decomposition (D^2)	2	Binary	MIP
	Modified Benders' using RLT	2	Binary	MIP
	D-BAC	2	Binary	MIP
	$D^2 - BAC$	2	Binary	MIP
	$D^2 - CBAC$	2	Continuous	Mixed-binary
Scenario-wise	Lagrangian relaxation	Multiple	Arbitrary	Arbitrary
	Dantzig-Wolfe decomposition	Multiple	Arbitrary	Arbitrary
	Progressive hedging	Multiple	Continuous MIP (heuristic)	Continuous MIP (heuristic)

Chapter 3

Scenariowise decomposition by Lagrangian and Dantzig-Wolfe decomposition

This chapter is devoted to a thorough description of the decomposition methods that are applied on the hydropower bidding SMIP problem. We have chosen to explore different variations of Lagragian decomposition, as well as Dantzig-Wolfe decomposition combined with branch and price.

The two methods considered are both scenariowise decomposition methods. Decomposing the problem by scenarios has several advantages for the SMIP problem considered. Having continuous first stage variables and mixed-integer second stage variables, several of the stagewise decomposition methods are not applicable as seen in the previous chapter. The scenariowise decomposition methods chosen are methods that can handle integer and/or continuous variables in both stages. The methods are furthermore applicable to multistage stochastic problems. This is relevant for the sequential hydropower bidding problem since it might be of interest to include more markets in the model, making it necessary to extend the model to include more stages. The dual block-angular structure of the SMIP problem implemented also makes it suitable for decomposition by scenarios.

Lagrangian duality and column generation are alternative approaches for getting better relaxations of MIP problems than what can be obtained by LP-relaxation. The bounds obtained by standard Lagrangian dualization and column generation are, in fact, the same (Lemaréchal, 2001; Geoffrion, 1974; Brooks and Geoffrion, 1966). The Lagrangian multipliers that belong to the relaxed constraints correspond to the dual variables of the same constraints in the master problem of the Dantzig-Wolfe decomposition. Consequently, a choice between the two methods would largely depend on which of the methods that can find the optimal multipliers or dual values more efficiently (Barnhart et al., 1998). When the optimal multipliers are found, for instance using the cutting plane method, the convex hull of the solution space is expressed in terms of its faces, while the master problem of the Dantzig-Wolfe reformulation expresses the convex hull in terms of its extreme points (Frangioni, 2005). A detailed discussion of the deep link between Lagrangian techniques and column generation is provided by Frangioni (2005).

For a SMIP problem Lubin et al. (2013) show how the cutting plane algorithm, which can be used to update the Lagrangian multiplier in Lagrangian relaxation, is computationally equivalent to column generation used to solve the Dantzig-Wolfe decomposed problem presented in Lulli and Sen (2004).

3.1 Lagrangian decomposition

Lagrangian relaxation is based on relaxing certain complicating constraints and adding a penalty in the objective function for breaking the relaxed constraints. The method works well for models with a block angular structure since coupling constraints may be relaxed and the problem can then be decomposed to smaller subproblems. The only coupling constraints between scenarios in a SMIP problem are the non-anticiaptivity constraints. By relaxing these constraints, the SMIP problem can be decomposed into separate subproblems for each scenario. Lagrangian relaxation provides optimistic bounds (Lagrangian dual bounds) on the solution to the optimization problem containing integer variables, and the bound is guaranteed to be equal to or better than the bound from the LP-relaxation. Sometimes the bound is exact and an optimal solution is found. Even when this is not the case, the method provides a good starting point for recovering primal feasible solutions. Several different Lagrangian decompositions of the same problem are possible, depending on which constraints are relaxed.

If we relax the non-anticipativity constraints of the stochastic problem given by (2.4), we get the Lagrangian dual function of the problem, as given by (3.1) (Lubin et al., 2013), in which non-anticipativitiy constraints are represented with a target variable formulation. Infeasibility with respect to the relaxed non-anticipativity constraints are penalized by the Lagrangian multipliers, denoted by μ_s in (3.1).

$$D(\mu) = \max \left\{ \sum_{s \in S} [Pr_s(cx_s + q_s y_s) - \mu_s(x_s - \hat{x})] : \quad \sum_{s \in S} \mu_s = 0 \quad (x_s, y_s) \in Z_s, s \in S \right\} \tag{3.1}$$

With the restriction on μ_s given by $\sum_{s \in S} \mu_s = 0$, the target variable is eliminated and the Lagrangian dual function is separable with respect to scenarios. Each scenario subproblem is then given by (3.2).

$$D_s(\mu_s) = \max \{ Pr_s(cx_s + q_s y_s) - \mu_s x_s : \quad (x_s, y_s) \in Z_s \} \tag{3.2}$$

To find the best possible bound of the Lagrangian problem, the Lagrangian dual function should be minimized with respect to the Lagrangian multipliers. The resulting problem, given by (3.3), is denoted the Lagrangian dual problem. The Lagrangian dual function is piecewise linear, and the corresponding Lagrangian dual problem is thus a concave, but non-smooth optimization problem.

$$z_{LD} = \min_{\mu} D(\mu) = \min_{\mu} \sum_{s \in S} D_s(\mu_s) \tag{3.3}$$

Lagrangian relaxation is an iterative algorithm. In every iteration, the subproblems are solved and the Lagrangian multipliers are updated based on a predefined procedure. This is repeated until all first stage decisions agree across scenarios or have converged sufficiently. The algorithm for solving the reformulated problem consists of the steps given in Figure 3.1.

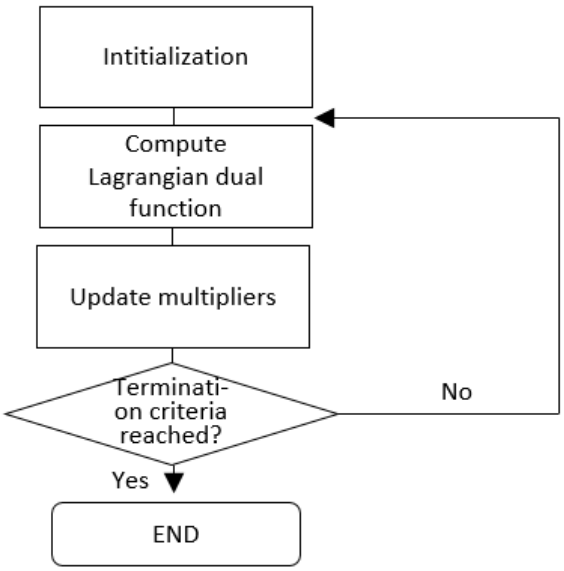


Figure 3.1: Flowchart of the Lagrangian relaxation procedure.

Several variations of Lagrangian relaxation exist, differing with respect to the Lagrangian multiplier updating procedure. The multiplier updating is an essential ingredient of the implementation of the Lagrangian relaxation. Well-known methods from the power systems literature are subgradient, cutting plane and bundle methods. These updating methods are compared with respect to solution time in Escudero et al. (2013). They conclude that the best method is instance dependent. Redondo and Conejo (1999) present a similar comparison using the subgradient method, bundle method and a dynamically constrained cutting plane algorithm. Their tests show that the cutting plane algorithm performs significantly better compared to the others. The multiplier updating procedures explored in this thesis are inspired by the results from these studies. The different methods studied are described in detail below.

3.1.1 Subgradient method

The subgradient method is a simple algorithm for minimizing a non-differentiable convex function, and can be used for solving convex optimization problems. The method is very

similar to the ordinary gradient method for differentiable functions, but in addition it also applies to non-differentiable functions. The subgradient method iteratively updates the Lagrangian multiplier vector according to (3.4).

$$\mu^{(k+1)} = \mu^{(k)} + \pi^{(k)} g^{(k)} \tag{3.4}$$

in which $g^{(k)}$ is the subgradient and the vector $\pi^{(k)}$ is the step length. How the step length is updated may vary across implementations. A commonly used updating function is given in by (3.5) (Escudero et al., 2013),

$$\pi^{(k)} = t^{(k)} \frac{z_{LD} - \bar{z}_{LD}(\mu^{(k)})}{\|g^{(k)}\|} \tag{3.5}$$

in which $\bar{z}_{LD}$ is the best feasible solution found so far and $t^{(k)}$ is a scalar that is gradually decreased.

The full subgradient procedure is given by Algorithm 1.

Algorithm 1: Subgradient method

Choose desired step length updating function π, step length parameter t and its updating scheme, and termination criteria

1. Initialization: $k := 0, \quad \mu^{(0)} \in \mathbb{R}^n$
2. Compute $D(\mu^{(k)})$
3. Choose a subgradient $g^{(k)} = \sum_{s=1}^{r} H x_s^{(k)}$
4. If termination criteria reached: quit
5. $\mu^{(k+1)} = \mu^{(k)} + \pi^{(k)} g^{(k)}$
6. $k := k + 1$, return to step 2

An important factor of the performance of the subgradient is the tuning of the step size parameter. If too short steps are used, the method suffers from slow convergence, while too long steps result in instability and large oscillations. Absolute convergence to the optimal multipliers is guaranteed if, $g^{(k)} \to 0$ and $\sum_{k=0}^{\infty} t^{(k)} \to \infty$ (Boyd et al., 2003).

The subgradient method usually converges slowly and produces oscillating solutions (Feng and Liao, 2006). Still, it is commonly used in practice due to ease of implementation and minimal computational overhead. For practical applications, it is usually combined with warm start procedures or other techniques to speed up the convergence.

3.1.2 Cutting plane method

The cutting-plane method is an outer approximation scheme that solves the Lagrangian dual problem by iteratively adding linear inequalities to narrow down the feasible region. The linear inequalities are also called cuts. In each iteration of the cutting plane algorithm, the Lagrangian multipliers are updated by solving the minimization problem in (3.6)-(3.8).

For the cutting plane algorithm, we adopt the target variable formulation of the non-anticipativity constraints, and therefore also the version of the Lagrangian dual function given by (3.1).

$$\min \quad \sum_{s \in S} \theta_s \tag{3.6}$$

$$\text{s.t.} \quad \sum_{s \in S} \mu_s = 0 \tag{3.7}$$

$$\theta_s \geq D_s(\mu_s^k) - (x_s^k)(\mu_s - \mu_s^k) \qquad s \in S, k = 1, .., K \tag{3.8}$$

The number of the last iteration at a point in time is given by K, and a given iteration up to that point is indexed by k. In order to avoid oscillations, it is possible to put an upper limit on the maximum absolute value of μ_s.

The cutting plane formulation in (3.6)-(3.8) always underestimates the value of the Lagrangian dual, but the two formulations gradually get closer as cuts are added to the problem. A common stopping criteria of the algorithm is therefore to stop when the difference between the Lagrangian dual and the cutting plane dual value is less than a predefined tolerance.

The full algorithm as described in Lubin et al. (2013), but adjusted for a maximization problem, is given in Algorithm 2.

Algorithm 2: Cutting plane method

Choose a relative convergence tolerance ϵ

1. Initialization: $k := 1, \quad \mu_s^k := 0 \ \forall s \in S$.
   Solve (3.2) $\forall s \in S$ and save the optimal value $D_s(\mu_s^k)$ and solution x_s^k.
2. Solve (3.6)-(3.8), obtaining optimal θ_s^* and μ_s^* $\forall s \in S$.
3. k:=k+1, $\mu_s^k := \mu_s^*$ $\forall s \in S$
   Solve (3.2) $\forall s \in S$ and save the optimal value $D_s(\mu_s^k)$ and solution x_s^k.
4. If $\sum_{s \in S}[D_s(\mu_s^k) - \theta_s^*]/[1 + |\sum_{s \in S} D_s(\mu^k)|] < \epsilon$ terminate;
   else add $D_s(\mu_s^k) - (x_s^k)(\mu_s - \mu_s^k)$ to (3.8).
5. Go to step 2.

Redondo and Conejo (1999), present a modified version of the cutting plane algorithm called the Dynamically Constrained Cutting Plane (DCCP) algorithm. This method has been developed to prevent excessive growth in problem size of the master problem due to large numbers of cuts when the number of iterations increases. In this algorithm, only a maximum number of cutting planes, $\hat{n}$, and not the full set, is considered when solving the problem in each iteration. The feasible region of the multipliers is dynamically updated. As long as $k < \hat{n}$ all constraints are considered, but when the number of iterations is larger than the maximum number of cutting plane constraints, the $k - \hat{n}$ most distant cuts are not considered in the model. The distance, d_k, to determine which cuts should be considered, and is computed according to (3.9). The distance is always positive because the cutting plane formulation of the dual function always underestimates the actual dual function.

$$d_k = z_{LD}(\mu^{(K)}) - (z_{LD}(\mu^{(k)}) - \sum_{s \in S} \left(\mu_s^{(K)}) - \mu_s^{(k)}\right) g^{(K)}) \qquad k = 1, .., K-1 \tag{3.9}$$

Redondo and Conejo (1999) report less oscillating behavior than with with the standard cutting plane algorithm. A disadvantage to the method is that the best choice of maximum limit is problem dependent and needs to be carefully tuned.

3.1.3 Proximal bundle method

The proximal bundle method is closely connected to the cutting plane algorithm, and it is the most widely used regularization methods for updating Lagrangian multipliers (Lubin et al., 2013). In order to stabilize cutting plane oscillations, the proximal bundle method updates the multipliers, μ_s, by solving the master problem depending on a given stability center, $\hat{\mu}_s$, instead of just using the multiplier from the previous iteration (as in the standard cutting plane algorithm). It uses a quadratic penalty term, $\sum_{s \in S} \|\mu_s - \hat{\mu_s}\|_2^2$, and penalty factor, τ, in the objective to regulate the step length in each iteration.

The mathematical formulation of the proximal bundle method is achieved by modifying the master problem of the cutting plane algorithm directly. The new master problem containing the stability center is given in (3.10)-(3.12).

$$\min \quad \sum_{s \in S} \theta_s - \frac{1}{2}\tau \sum_{s \in S} \|\mu_s - \hat{\mu_s}\|^2 \tag{3.10}$$

$$\text{s.t.} \quad \sum_{s \in S} (\mu_s - \hat{\mu_s}) = 0 \tag{3.11}$$

$$\theta_s + x_s^k(\mu_s - \hat{\mu_s}) \geq D_s(\mu_s^k) - (x_s^k)(\mu_s - \mu_s^k) \qquad s \in S, k \in K \tag{3.12}$$

The stability center $\hat{\mu}^{(k)}$ is a point one would like $\mu^{(k)}$ to be close to. This is done in order to center the algorithm in the region of interest. Stability centers can be problem specific, or more general, as when the multiplier resulting in the best solution so far is used.

In order to center the search properly, it is necessary to carefully tune-up the penalty and other parameters. The tune-up is problem dependent and can be hard to achieve (Redondo and Conejo, 1999). Kiwiel (1990) present work on how to tune these parameters. For a small enough penalty factor, the algorithm is transformed back to the ordinary cutting plane algorithm and we move back to the stability issues. If, on the other hand, the penalty is large, convergence is slow.

The quadratic program may become very time-consuming to solve. The quadratic term makes the objective function non-linear which potentially results in a more complex master problem compared to standard cutting plane method. In return, the bundle method usually stabilizes faster and uses significantly fewer iterations.

The full procedure is presented in Algorithm 3. It is based on Lubin et al. (2013), who also suggest a termination criteria and an updating scheme for the penalty factor, τ.

Algorithm 3: Proximal bundle method

Choose a termination criteria and updating scheme for τ.

1. Initialization: $k := 1, \quad \hat{\mu}_s^k := 0 \; \forall s \in S$, $\tau := 1$, $m := 0.1$.
   Solve (3.2) with $\hat{\mu}_s \; \forall s \in S$ and save the optimal solution x_s^k.
   $curObj := D_s(\hat{\mu}_s^k)$.
2. Solve (3.10)–(3.12), obtaining optimal θ_s^* and μ_s^*.
3. If the termination criteria is reached, terminate, else continue.
4. $k := k + 1$.
5. Solve $D_s(\mu_s^*) \; \forall s \in S$, saving the optimal value $D_s(\mu_s^k)$ and solution x_s^k.
6. $newObj := \sum_{s \in S} D_s(\mu_s^k)$.
7. Update τ.
8. If $newObj < curObj$, then $\hat{\mu}_s := \mu_s^*$ and $curObj := newObj$
9. Go to step 2.

3.1.4 Progressive hedging

Progressive hedging is a method related to the Lagrangian relaxation in terms that the non-anticipativity constraints are relaxed and penalty terms are added to the objective function for violating the constraints. Progressive hedging was first described in Rockafellar and Wets (1991).

In every iteration, the first-stage variables are updated based on (3.13)–(3.15) in order to coordinate these decisions across scenarios.

$$x_s^{(k+1)} = \text{argmax}_{x,y_s} \left\{ cx + \omega_s^{(k)} x - \frac{\rho}{2} \|x - \hat{x}_s^{(k)}\|^2 : (x, y_s) \in Z_s \right\}, \tag{3.13}$$

in which,

$$\omega_s^{(k)} = \omega_s^{(k-1)} + \rho(x_s^{(k)} - \bar{x}^{(k)}), \tag{3.14}$$

and

$$\bar{x}^{(k)} = \sum_{s \in S} Pr_s x_s^{(k)}. \tag{3.15}$$

The full algorithm is outlined in Algorithm 4 (Watson and Woodruff, 2011).

The distance variable of the progressive hedging algorithm, ω_s, can be seen as somewhat analogous to the Lagrangian multiplier. The quadratic term used to update the first stage decisions constitutes the stabilizing element of the algorithm and is somewhat similar to the quadratic term of the bundle method. However, in contrast to the bundle method, the quadratic term in is part of the subproblems, not the master problem and contains first-stage variable instead of multipliers. Due to the addition of the quadratic term, the subproblems of the progressive hedging algorithm are non-linear. Non-linear subproblems distinguishes the progressive hedging method from the Lagrangian multiplier updating methods.

Convergence of the progressive hedging method for convex problems is proven in Rockafellar and Wets (1991), but no results of convergence of the algorithm are presented in

Algorithm 4: Progressive hedging

Input: penalty factor $\rho > 0$ and termination threshold ϵ.

1. $k := 0$
2. For all $s \in S$, $x_s^{(k)} := \text{argmax}_{x,y_s}(cx + q_s y_s) : (x, y_s) \in Z_s$
3. $\bar{x}^{(k)} := \sum_{s \in S} Pr_s x_s^{(k)}$
4. For all $s \in S$, $\omega_s^{(k)} := p(x_s^{(k)} - \bar{x}^{(k)})$
5. $k := k + 1$
6. For all $s \in S$, solve (3.13)
7. $\bar{x}^{(k)} := \sum_{s \in S} Pr_s x_s^{(k)}$
8. For all $s \in S$, update $\omega_s^{(k)}$ according to (3.14).
9. $g^{(k)} := \sum_{s \in S} Pr_s \| x_s^{(k)} - \bar{x}^{(k)} \|$
10. If $g^{(k)} > \epsilon$, then go to step 5. Otherwise, terminate.

the non-convex case. The first general purpose methods for finding good solutions to multistage SMIPs was presented in Løkketangen and Woodruff (1996). It uses the progressive hedging algorithm described in Rockafellar and Wets (1991) and solutions to the subproblems are obtained using a tabu search algorithm. Watson and Woodruff (2011) discuss issues related to the use of progressive hedging as a heuristic to solve SMIPs and present algorithmic innovations to find effective values for the penalizing factor, termination criteria and to accelerate convergence. In Watson and Woodruff (2011) issues related to the use of progressive hedging as a heuristic to solve SMIPs are discussed and algorithmic innovations are presented to find effective values for the penalizing factor, termination criteria and to accelerate convergence.

The standard progressive hedging method does not give any bounds on the objective value function, making it impossible to draw any conclusions about the quality of the solution obtained. Watson et al. (2013) present a method for computing optimistic bounds by using the estimates of the dual prices in each iteration, and they prove that the best possible optimistic bound obtained using dual prices of the progressive hedging algorithm is as tight as the bounds obtained with Lagrangian decomposition.

The bound equivalence between Lagrangian relaxation and progressive hedging has been utilized to derive a Lagrangian multiplier updating scheme based on progressive hedging as described in Escudero et al. (2013). This variant has no addition of extra terms in multipliers but, similar to progressive hedging, uses the average first stage solution in the multiplier updating formula. It can be seen as a variant of the subgradient method, but instead of using the standard subgradients, the gradient vector is calculated based on the difference between the first stage variables and a common first stage solution based on a weighted average over all first stage decisions, as given by (3.15)

Similar to the subgradient method and the standard progressive hedging, the multiplier updating of is based on a formula and not on an optimization problem. In contrast to the standard progressive hedging algorithm by Rockafellar and Wets (1991), the subproblems are kept as is and are not modified with a quadratic term. We will refer to this new method as the simplified progressive hedging algorithm.

The full procedure is given in Algorithm 5.

Algorithm 5: Simplified progressive hedging

Choose desired step function π, step length factor ρ and termination criteria

1. Initialization: $k := 0, \quad \mu^{(0)} \in \mathbb{R}^n$
2. Compute $D(\mu^{(k)})$
3. Calculate subgradients by $g^{(k)} = x_s^{(k)} - \bar{x}^{(k)}$
4. If termination criteria reached: quit
5. Calculate $\omega^{(k+1)} = \omega^{(k)} + \pi^{(k)} g^{(k)}$
6. $k := k + 1$, return to step 2

The step length π_s is calculated by (3.16)

$$\pi^{(k)} = \rho^{(k)} \frac{z_{LD} - \bar{z}_{LD}(\mu^{(k)})}{\|g^{(k)}\|} \tag{3.16}$$

3.2 Dantzig-Wolfe decomposition

Dantzig-Wolfe decompositon is a method that combines a Dantzig-Wolfe reformulation of an optimization problem with a method to solve the decomposed problem. The idea behind a Dantzig-Wolfe reformulation is to formulate the optimization problem with an inner representation, defined through the extreme points of the feasible region. In the inner representation, also denoted the master problem, the number of constraints is reduced but the number of variables is increased. Finding the full set of solutions that together comprise the solution space of the problem is challenging. Most inner points in the master problem will furthermore not be used in the optimal solution, and are not necessary to include in the formulation. By only including a subset of all the extreme points of the master problem, we get what is called the reduced master problem (RMP). To find the extreme points, or columns, that should be included in the RMP in order to find the optimal solution of the problem, we can use column generation. From optimization theory it is known that the optimal solution to a linear maximization problem is found when the reduced costs of all non-basic variables are less than or equal to zero (Lundgren et al., 2012). In column generation, new columns to be added to the RMP are found by solving a subproblem with the objective of finding the maximum reduced cost (for maximization problems). Columns, or extreme points, with positive reduced cost are added to the master problem as they can improve the objective function of the RMP. The RMP and the subproblems are solved iteratively until an optimal solution is found or the duality gap is of an acceptable size. An overview of the process is given in Figure 3.2.

Applied to a stochastic problem, we are able to decompose the stochastic problem by scenarios. Using the general stochastic problem defined by (2.4), the decomposition method is outlined below (Lulli and Sen, 2004). The master problem is given by (3.17)-(3.21) and is stated through its interior representation in which $x_s = \sum_{m \in M} x_s^m \lambda_s^m$. The weight on each on each extreme point $m \in M$ in scenario s is given by λ_s^m.

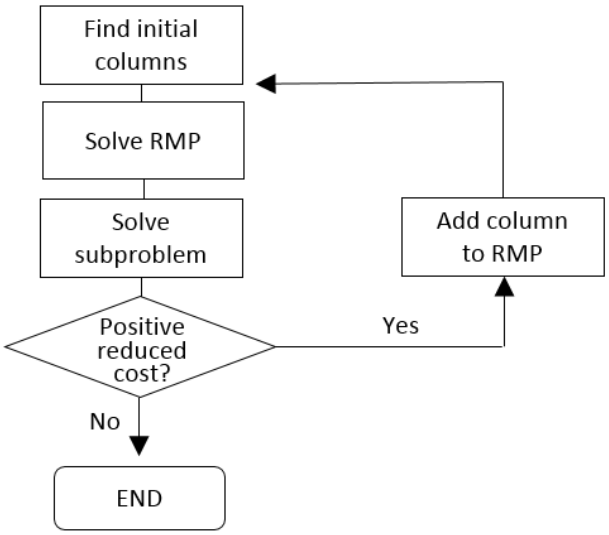


Figure 3.2: Flowchart of the Dantzig-Wolfe decomposition procedure.

$$\max \quad \sum_{s\in S} Pr_s \sum_{m\in M} cx_s^m \lambda_s^m \tag{3.17}$$

$$\text{s.t} \quad \sum_{m\in M} x_s^m \lambda_s^m = \bar{x} \qquad s \in S \tag{3.18}$$

$$\sum_{m\in M} \lambda_s^m = 1 \qquad s \in S \tag{3.19}$$

$$\lambda_s^m \geq 0 \qquad m \in M, s \in S \tag{3.20}$$

$$\bar{x} \geq 0 \tag{3.21}$$

For the RMP, not all the extreme points of the solution space are included in M and the set grows with iterations. The dual values associated with the non-anticipativity constraints (3.18) are given by α_s and the dual values associated with the convexity constraints (3.19) are given by β_s. The reduced cost associated with a column is given by (3.22).

$$\bar{c}_s^m = Pr_s c_s^m - \sum_{s\in S} \alpha_s x_s^m - \beta_s \tag{3.22}$$

The search for columns with positive reduced cost is done by solving the subproblem given by (3.23) for each scenario. Solutions to the subproblems that give $\bar{c}_s \geq 0$ are added to the RMP. Note that not only the columns with the highest positive reduced cost can be added.

$$\bar{c}_s = \max \left\{ Pr_s c_s x_s - \sum_{s\in S} \alpha_s x_s - \beta_s : (x, y_s) \in Z_s \right\} \tag{3.23}$$

The subproblems are independent and similar. This permits the use of parallelization to speed up the solution process.

Before all necessary columns to find an optimal solution to the master problem have been found and added to the RMP, we can find an upper bound on the solution. This optimistic bound, z_{MP}, is the sum of the objective value in the RMP, z^k_{RMP}, in an iteration k and the optimal reduced costs in the subproblems in the next iteration (Lundgren et al., 2012). Thus,

$$z_{MP} \leq z^{(k-1)}_{RMP} + \sum_{s \in S} \bar{c}^{(k)}_s \qquad k = 1, .., K \tag{3.24}$$

in which K is the number of iterations performed so far. The bound is not monotone, and it is the lowest bound found so far that gives the best optimistic bound.

3.2.1 Stabilized column generation

It is well known that column generation can suffer from slow convergence. Some of the reasons for slow convergence can be the size of the master problem, in which a bigger problem converges slower than a small one, primal degeneracy in the optimal solution of the RMP, and instability in the behaviour of the dual variables (Ben Amor et al., 2009; Du Merle et al., 1999). To speed up convergence, several dual stabilization techniques have been developed, both linear and non-linear. Ben Amor et al. (2009) present a study in which the performance of different stabilization techniques are evaluated and the impact of different stabilizing functions and parameters are assessed.

Du Merle et al. (1999) present the first linear stabilization scheme with a focus on Kelley's cutting plane algorithm (Kelley, 1960). Applying column generation to a program P amounts to applying Kelley's cutting plane algorithm to its dual D.

$$\begin{array}{llr} (P) & \min & cx \\ & \text{s.t.} & Ax = b \\ & & x \geq 0 \end{array} \qquad \begin{array}{llr} (D) & \max & b\lambda \\ & \text{s.t.} & A^T\lambda \leq c \end{array}$$

Kelley's algorithm can show undesirable behaviour with slow convergence and when one tries to prove optimality of a degenerate solution of P. To overcome degeneracy we can add bounded surplus and slack variables. We then get the primal problem $\widetilde{P}$ and its dual $\widetilde{D}$.

$$\begin{array}{llr} (\widetilde{P}) & \min & c\tilde{x} - \delta^- y^- + \delta^+ y^+ \\ & \text{s.t.} & A\tilde{x} - y^- + y^+ = b \\ & & y^- \leq \varepsilon^- \\ & & y^+ \leq \varepsilon^+ \\ & & \tilde{x}, y^-, y^+ \geq 0 \end{array} \qquad \begin{array}{llr} (\widetilde{D}) & \max & b\tilde{\lambda} - \varepsilon^- w^- - \varepsilon^+ w^+ \\ & \text{s.t.} & A^T\tilde{\lambda} \leq c \\ & & -\tilde{\lambda} - w^- \leq -\delta^- \\ & & \tilde{\lambda} - w^+ \leq \delta^+ \\ & & w^-, w^+ \geq 0 \end{array}$$

In $\widetilde{P}$, y^- and y^+ are surplus and slack variables which are bounded by ε^- and ε^+ respectively. In the objective function, the surplus and slack variables are used in a stabilizing function and are penalized by δ^- and δ^+ respectively. In the dual program, $\widetilde{D}$, the penalizing parameters work as bounds for the dual variables $\tilde{\lambda}$ such that dual variables outside the stability center, the interval $[\delta^-, \delta^+]$, are penalized.

It is of importance to develop good strategies for adjusting the parameters δ^-, δ^+, ε^- and ε^+ to achieve convergence. The strategies for updating the parameters must together ensure finite convergence. Du Merle et al. (1999) suggest different updating strategies for adjusting the parameters. Some of these suggestions are to use the dual values from the RMP as the penalizing parameters, δ^- and δ^+. For the updating of the bounds, ε^- and ε^+, on the surplus and slack variables, they suggest to decrease the bounds after a given number of iterations such that they vanish in a finite number of iterations.

Ben Amor et al. (2009) suggest to improve the method of Du Merle et al. (1999) by adding additional stabilizing terms to the procedure. They refer to the method of Du Merle et al. (1999) as a method with a 3-piecewise linear stabilizing function, and argue that it is difficult to find penalizing parameters that will provide good convergence. This is because the parameters must be high enough to ensure boundedness, such that only small steps in the non-penalized region can be taken, which, in turn, slows down convergence. They therefore suggest to use a 5-piecewise linear stabilization function with two sets of penalties; one with large values to ensure stability, and one with small values to allow for significant changes in the dual variables.

3.2.2 Branch and price

When the problem contains integer variables, the optimal solution from the column generation procedure does not necessarily provide feasible integer solutions. The branch and price framework can then be applied to find the optimal feasible solution. Branch and price is a version of branch and bound in which column generation is used to solve the problem in each node of the branch and bound tree. The general methodology of branch and price is presented in (Barnhart et al., 1998) and an illustration of the general procedure is provided in Figure 3.3. Branch and price is an exact solution method, guaranteed to find the optimal solution.

An example of branch and price applied to a stochastic problem that is decomposed by scenarios can be found in Lulli and Sen (2004). The branch and price procedure they present is stated for a maximization problem in Algorithm 6. The set of generated columns is denoted G_s, and the lower bound from the procedure is denoted LB.

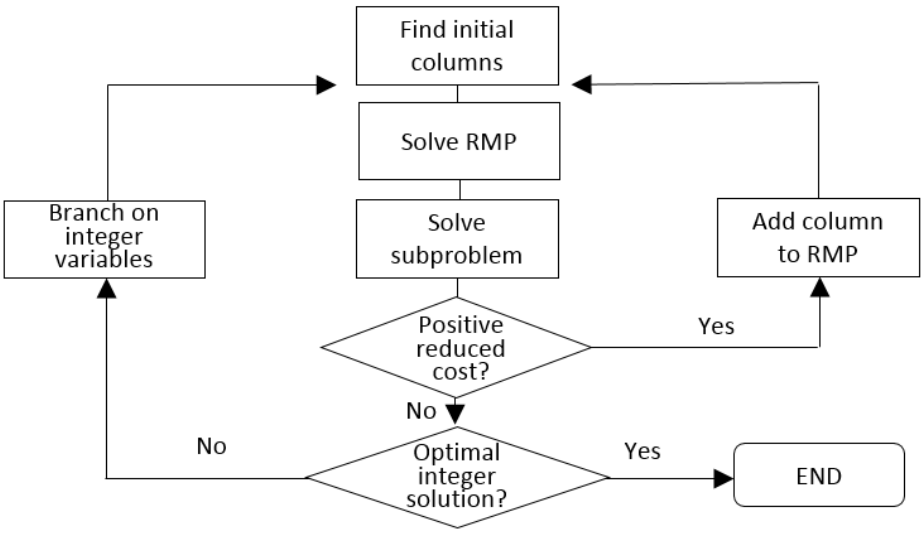


Figure 3.3: Flowchart of branch and price procedure.

Algorithm 6: Column generation with branch and price

1. Initialization: $G_s = \{\}$, $\alpha_s = 0$ and $\beta_s = \infty \ \forall s \in S$, $LB = -\infty$ or the objective function of some computed feasible solution.
2. Solve the pricing subproblem given by (3.23) $\forall s \in S$. If $\bar{c}_s \geq 0$ add the column to the current RMP and $m \rightarrow G_s$ with total cost g_s^m computed using the original cost vector. Otherwise, go to step 6.
3. Solve the linear relaxation of the RMP given by (3.17)–(3.21), and update the dual variables α_s and β_s.
4. If all the integrality conditions are satisfied, update the incumbent solution if necessary, and go to Step 6, otherwise go to Step 5.
5. Start a branch phase using the following rule: find a fractional component of X, say i, required to be integer. Define two subproblems adding the constraints $X_i \leq \lfloor \tilde{X}_i \rfloor$ or $X_i \geq \lceil \tilde{X}_i \rceil$, respectively. In the successor nodes, eliminate all those columns whose components do not satisfy the new branching constraints.
6. If there are no more active nodes in the branching tree, then we declare the incumbent solution as the optimal one and STOP. Else, we select an active node and go to Step 2.

3.3 Accelerating convergence

There exist several general techniques that can be combined with different decomposition algorithms in order to achieve tighter bounds or speed up convergence. Two such methods are outlined below.

Scenario clustering

Scenario clustering is a method where a scenario set, S, is partitioned into disjoint subsets called scenario clusters. The scenarios within the same cluster are regarded as observationally indistinguishable and the non-anticipativity is defined implicitly within the cluster. This can be done in order to achieve tighter bounds, but at the cost of more complex subproblems.

Escudero et al. (2013) propose a scenario clustered version of Lagrangian decomposition for two-stage stochastic mixed 0–1 programs. The mathematical formulation of the Lagrangian dual function is given in (3.25).

$$\max \Bigg\{ \sum_{\Omega \in C} \sum_{s' \in S^{\Omega}} Pr_{s'} \left(c x_{\Omega} + q_{s'} y_{s'} \right) - \sum_{\Omega \in C} \mu_{\Omega} (x_{\Omega} - \hat{x}) : \\ (x_{\Omega}, y_s) \in Z_s, s \in S, \hat{x} = x_1 = ... = x_{|C|} \Bigg\} \tag{3.25}$$

Each cluster is given by the index $\Omega \in C$, and the scenarios $s' \in S^{\Omega}$ are the scenarios that belong to cluster Ω. For clustering applied to Dantzig-Wolfe decomposition, the resulting master problem would be as follows:

$$\max \quad \sum_{\Omega \in C} \sum_{s' \in S^{\Omega}} Pr_s \sum_{m \in M} c x_s^m \lambda_s^m \tag{3.26}$$

$$\text{s.t} \quad \sum_{m \in M} x_s^m \lambda_s^m = \bar{x} \qquad \Omega \in S^{\Omega} \tag{3.27}$$

$$\sum_{m \in M} \lambda_s^m = 1 \qquad \Omega \in S^{\Omega} \tag{3.28}$$

$$\lambda_s^m \geq 0 \qquad m \in M, \Omega \in S^{\Omega} \tag{3.29}$$

$$\bar{x} \geq 0 \tag{3.30}$$

In the subproblems, non-anticipativity constraints are added for the scenarios that belong to each cluster.

Warm-start
Warm start procedures aim to seed the decomposition methods with good initial values. Instead of guessing at random or using zero values, another choice would be to use dual values from the relaxed problem as the initial guess. Borghetti et al. (2003) propose using warm-start procedures for the Lagrangian relaxation based on the solution to a relaxed version of the original problem.

For column generation, the initial set of columns can affect the number of iterations needed. By starting out with a good set of columns, better dual values can be obtained in the beginning of the procedure, which can help reduce the number of iterations needed. Vanderbeck (2005) provides details on initializations of the column generation procedure.

3.4 Heuristics for recovering primal feasible solutions

Heuristics are solution methods that can be used to find feasible solutions within a limited amount of time. As opposed to exact methods, heuristic methods sacrifice optimality,

accuracy, precision, or completeness for speed. Even though heuristics do not provide a guarantee of the quality of the solution, they are much appreciated solution methods for problems that are too complex to be solved efficiently by an off-the-shelf optimization solver.

There exist many types of heuristics, with two of the main groups of heuristics being constructive heuristics and local search heuristics (Lundgren et al., 2012). While constructive methods aim at building a feasible solution, local search methods are used to find better solutions by searching through the landscape of feasible solutions. Heuristics can also be combined with decomposition methods in order to find good feasible solutions within reasonable time. In the case of the decomposition methods described above, we can take either the master problem or the subproblem solution from the latest iteration of the decomposition algorithm and construct a primal feasible solution by making adjustments to the solution chosen. However, there is no guarantee that the subproblem solution is close to the optimal solution of the primal problem. Thus, great adjustments might be required and the quality of the heuristic solution is usually hard to control. Most heuristics are very specific and problem-dependent and require great knowledge about the problem to be solved.

Chapter 4

Nordic electricity markets and hydropower

The power markets in the Nordic countries were deregulated in the early 1990s and the individual day-ahead markets were then brought together to form a common Nordic market run by Nord Pool. The implemented model in this thesis considers a hydropower producer bidding power in the day-ahead market, also known as Elspot, and capacity in the primary reserve market. These two markets are only two of the Nordic power markets, as illustrated in Table 4.1. Due to the majority of power production in Norway being hydropower, the model includes only hydropower production.

This chapter begins with a presentation of the two power markets included in the model. Thereafter the functioning of a hydropower plant will be described. This will set the stage for the next chapter where the optimization model implemented is described in detail.

Table 4.1: Electricity markets in Norway (Eriksrud and Braathen, 2014).

Market place	Physical trade	Financial trade
Nord Pool Spot (NPS)	Elspot Elbas	
Transmission system operator (TSO)	Primary reserves (FNR/FDR) Secondary reserves (FRR) Tertiary reserves (Balancing market)	
Nasdaq OMX commodities		Futures Forwards Options Contracts for differences
Bilateral	Full delivery Load factor contracts Cap and floor contracts	

4.1 Day-ahead and primary reserve market

The day-ahead market, called Elspot, is an auction-based exchange where power is traded for delivery the day after gate closure (Nord Pool, 2016b). Nord Pool is the world's largest market for trading power, resulting in high liquidity and little market power. Trading in the day-ahead market can be done using different types of orders. Single hourly orders, in which the auction participant specifies the purchase and/or sales volume for each hour, accounts for the largest share (Nord Pool, 2016b). Bids must be placed before 12.00 the day before delivery and when when all bids are received, the system price is calculated as the equilibrium between the aggregated supply and demand curves. This type of price formation is called marginal price setting. The capacities between bidding areas are not taken into account when finding the system price and is thus an unconstrained market clearing price. Due to grid congestion, the Nordic area is however divided into several price areas. Prices in each area are calculated in a similar manner to the system price, including only the members of that area (Nord Pool, 2016b).

Most of the balancing between supply and demand in the power market is carried out in the day-ahead market. However, there is also a need for a final balancing process and it is the transmission system operator (TSO) who is responsible for the stability of electricity in an area and for security of supply. In Norway the responsible TSO is Statnett. To maintain the balance between production and consumption, reserves that can be activated when the frequency in the power grid deviates from 50 Hz are needed. The reserves are divided into primary, secondary and tertiary reserves. Momentary imbalance is first regulated by primary reserves and if further reserves are needed for balancing, secondary and tertiary reserves are activated (Statnett, 2016).

The primary reserve market is split into a weekly and a daily market. The weekly market is cleared before the spot market while the daily market is cleared after the spot market to cover remaining demand and possible exchange. This is illustrated in Figure 4.1. Actors in the primary reserve market can choose to participate in one or both of the markets (Statnett, 2016). The weekly market is divided between weekdays (Monday to Friday) and weekends (Saturday and Sunday), and both are again divided into three blocks; night, day and evening. The night block covers the hours from 00:00 to 08:00, the day block runs from 08:00 to 20:00 and the remaining hours from 20:00 to 24:00 are covered by the evening block. All bids for the auction for weekend blocks must be submitted by 12:00 Thursday before delivery, and bids for the weekday blocks must be submitted before 12:00 Friday before delivery (Statnett, 2015). Statnett decides which bids to accept, and the primary criteria for acceptance is the price. Normally, all accepted offers in the same price area receive the marginal price.

The primary reserve capacity is based on a suppliers portfolio of power plants and there is no link between a specific generator and the primary reserve bid. Hence, a supplier can freely distribute the capacity over the portfolio of its generators, allowing the reserve capacity available at a single generator to vary within the delivery week. The TSO must be informed of which generators that are available, so that it is clear that the contracted reserves are available within the portfolio of the supplier. In the Norwegian primary reserve market, there is however a symmetry requirement on unit-level, meaning that the upward and downward reserve delivery has to be equal for a generator delivering

reserves.

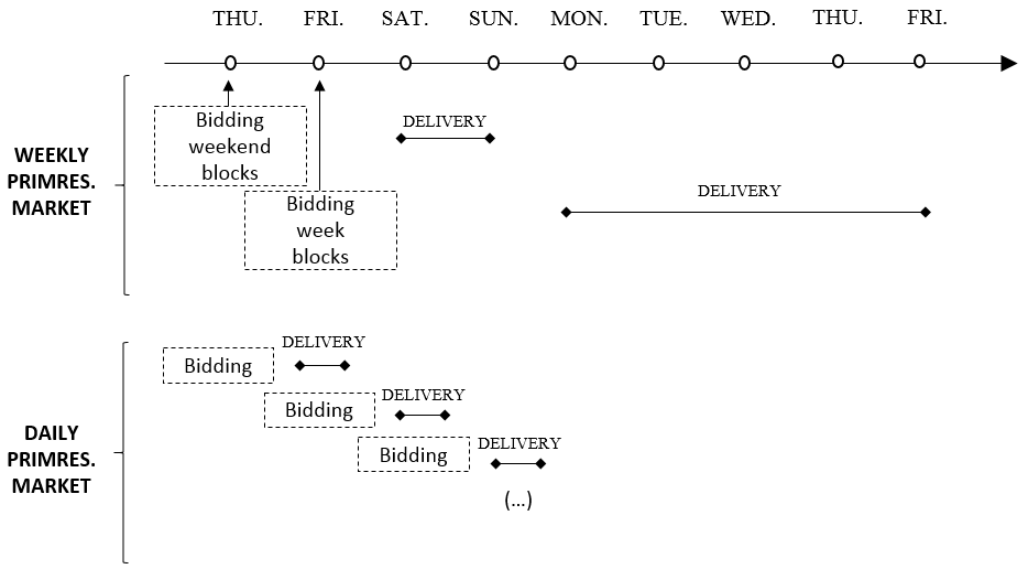


Figure 4.1: Bidding in the weekly and daily primary reserve market.

4.2 Hydropower

Hydropower is a well-established and cost-competitive technology. The technology is furthermore flexible and the response time is fast. Hydopower stations can be classified in three categories; reservoir (or storage) hydropower plants, pumped-storage plants and run-of-river systems (IEA, 2012). The hydropower stations analysed in this thesis are reservoir hydropower plants, and this will consequently be the focus of this section. Hydropower plants have various operational constraints governing their hydraulic operation. In addition, there are regulations that restrict the way a hydropower plant can be operated. These constraints should be included in an optimization model to the extent seen as necessary in order to achieve the technical accuracy wanted. This section provides a brief introduction to the functioning of a hydropower plant and the value of water. For more information regarding their operation see Kjølle (1980).

4.2.1 Operation of hydropower plants

In a hydropower plant, potential energy stored in the water is converted to mechanical energy in turbines before being converted to electricity in generators. The power output from discharging water is given by (4.1) in which ρ is the density of the water given in kg/m^3, g is the gravitational constant given in m/s^2, Q is the discharge given in m^3/s, H is the head level given in m, and η_a is the overall efficiency of the power plant (Kjølle, 1980).

$$P = \rho g Q H \eta_a \tag{4.1}$$

The power output varies with the difference between the water level in the up- and downstream reservoirs of the hydropower plant, also denoted the head. The relationship between power output, discharge and head level is non-linear and an optimization model that includes head level dependencies will therefore be non-linear, regardless of the type of turbine (Klæboe, 2015b).

However, for hydropower plants with large storage capacity and high head levels, the head level has negligible influence on operating efficiency (Catalão et al., 2010). In such a case, the discharge dominates and the output of the power plant can be modeled as a function of discharge alone. This relationship is usually concave, with the point of maximum efficiency denoted the best point (Kjølle, 1980). Efficiency varies due to the need to compensate for the change in rotational speed of the turbine when the electric load changes. This is done by changing the settings of the turbine, making the the turbine less efficient. Operating at the best point is optimal if only energy is delivered. But, if e.g spinning reserves are to be provided or if the producer wants to avoid start-up costs, operating at lower efficiency might be desirable (Klæboe, 2015b). Running at lower efficiency might however contribute to lower the life-time of the turbine and increase maintenance costs.

Hydropower plants are designed and constructed to function optimally with continuous production and few starts and stops. Production patterns are however changing, resulting in higher costs associated with start-stops and with general load changes. There is always deterioration on equipment associated with starting and stopping the plant. Costs associated with repairs and replacement of equipment will thus increase with more start-ups and stops. These are costs that together are in the magnitude of millions of NOK (Bakken et al., 2001). Some of the costs can be directly linked to a specific start-up or stop, while other costs are accumulated over time and only indirectly linked to a specific start-up. In addition, some water is always lost when a unit is started, leading to a cost of lost water. Bakken et al. (2001) propose two approaches to model the start-stop costs; one in which costs are dependent and proportional with the number of start-stops during a period, and one in which costs are a function of accumulated operating time.

4.2.2 Water value

The fuel of hydropower is free, but there is an opportunity cost associated with using the water since it is a scarce resource that can be stored and used in later periods. This means that the hydropower producer has to weigh the profit today against possible future profit, making the hydropower planning problem a challenging one (Klæboe, 2015b).

Ideally, short term planning decisions should take all long-term effects into account with a high level of detail in both the short- and long-term. However, it is not possible to run one optimization model that takes all time periods into account due to computational difficulty. The planning problem is consequently divided into several phases. Fosso and Belsnes (2004) divide the process in four phases; long-term scheduling (1-5 years), medium-term scheduling (3-18 months), short term scheduling (1-2 weeks) and detailed simulation (1-12 weeks). In long-term scheduling optimal storage and dispatch strategies are made by aggregating the system and leaving out the details. In seasonal scheduling, schedules from long-term models are disaggregated to storage and dispatch plans for indi-

vidual reservoirs and power plants. This phase is seen as an intermediate step necessary to give correct boundary conditions to short-term models. Short-term scheduling thereafter concerns the actual operations of the hydropower plants and results in operational plans. If needed, simulations can be performed to account for other aspects not included in the short-term models, and are also used to verify the short term plans.

Since the different scheduling phases cover different time-periods, there are boundary conditions that need to be taken into account when coupling them together. In the end, the short-term scheduling decision should reflect the strategies in the long-term schedules. There are two main methods in use; volume and price coupling (Fosso and Belsnes, 2004). In volume coupling the optimal use of water is based on set end-period reservoir levels calculated in longer-term models. In price coupling the expected future income of the stored water is used. The expected future income of a marginal unit of water is also denoted the water value. Volume coupling is known as a less flexible method than price coupling because changing conditions, e.g prices, will have a larger impact on the optimality since end-period reservoir levels are most probably not optimal when conditions change. Volume coupling is hence not consistent with the fact that optimal decisions in a market based system should be price dependent (Fosso and Belsnes, 2004). The coupling in the model in this thesis is based on price coupling, and this method is therefore further explained.

Water values for each reservoir are calculated in the seasonal model with initial and end period reservoir levels as input parameters. The water value is found as the shadow price of the reservoir balance constraint in the optimization problem. Since the risk of spilling increases with increasing reservoir level, the water value as a function of reservoir level is a concave function. Hence the marginal water value is positive but decreasing. The price coupling should be based on multidimensional water value functions, meaning that the water value of one reservoir depends on the reservoir levels in all other reservoirs in the system. This is because a hydropower system often is cascaded and therefore has an inter-reservoir dependency. Using one-dimensional water value functions should therefore be seen as only an approximation (Fosso and Belsnes, 2004). The multidimensional water value function is often presented as cutting planes and an illustration is given in Figure 4.2.

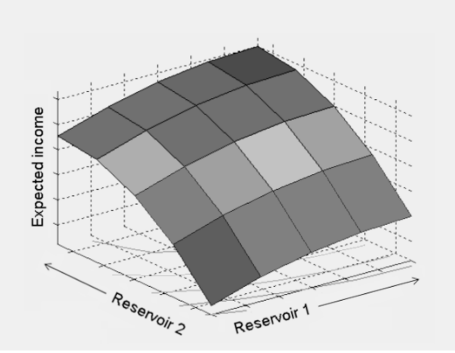


Figure 4.2: Illustration of multidimensional water value function for two watercourses and 16 cuts (Eriksrud and Braathen, 2014).

Chapter 5

Problem formulation

This chapter presents the mathematical formulation of a two-stage SMIP model with relatively complete recourse for coordinated sequential bidding in the primary reserve market and the day-ahead market. The model presented is based on work by Klæboe (2015a). Trading in the day-ahead market is for many power producers the main source of revenue and is seen as a natural choice to include in a multi-market model. Demand for ancillary services, including primary reserves, is expected to grow in the years to come and it is therefore relevant to develop models that take these markets into account. The primary reserve market has been chosen because the market is bid into once a week, and therefore serves well as a start point for developing multi-market models and investigating possible decomposition strategies. The model can later be extended to include other reserve markets instead of, or in addition to, the primary reserve market. Otherwise the model can be used as part of a a set of two-stage rolling-horizon problems representing multi-market bidding.

The model implemented includes only hydropower bidding, but other technologies could be included by adding different technological restrictions. It is noteworthy that a hydropower bidding model must include sufficient details in order to reflect the realistic income and cost of production. Otherwise, bids from the model may lead to production obligations that are not executable (Klæboe, 2015b). It is also preferable that the model is solved fast, but these two objectives are however often conflicting (Nygreen et al., 1998). Considering this, the model implemented and the input data used in the model are carefully chosen.

5.1 Assumptions

The model focuses on the Norwegian electricity market, and the markets are therefore modeled according to the market rules described in Section 4.1. As described, the primary reserve market consists of both a weekly and a daily market. In the model, only the weekly market is taken into account and the modeled primary reserve market is cleared before the day-ahead market.

According to the market rules for primary reserves, producers are only remunerated for

the reservation, not activation, of primary reserves. Producers are remunerated according to the marginal pricing principle, meaning that all producers are paid the market clearing price, regardless of their bid. This principle is also applicable in the day-ahead market.

The producer operates a portfolio of hydropower plants. The number of plants and generators in each plant can be varied based on the modeler's objectives. The plants can be different regarding e.g size and location, but all plants are operated under the same type of constraints. The risk of not being able to deliver contracted capacity is not taken into account. It is assumed that the producer is risk neutral and seeks to maximize expected profit.

For the day-ahead market, only single hourly orders are considered, a simplification that removes complexity at minimal cost (Fleten and Kristoffersen, 2007). Market prices are modeled as stochastic since the benefits of solving the model with stochastic prices are considered high due to the uncertainty associated with electricity price.

5.2 Market modeling

The objective of the producer is to maximize profit when bidding in the primary reserve market and the day-ahead market. Bidding in these two markets takes place sequentially; the market for primary reserves is settled before the day-ahead market, hence the acceptance of a bid in the primary reserve market limits the possible bids in the day-ahead market.

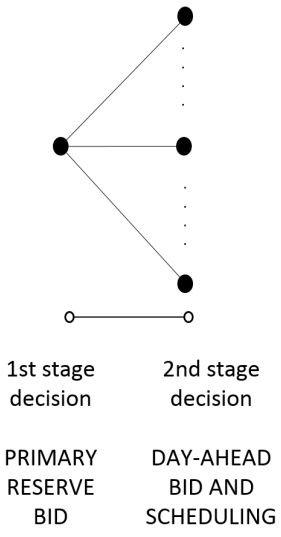


Figure 5.1: Two-stage scenario tree for bidding in the primary reserve market and the day-ahead market.

The here and now decision is the amount to bid in the primary reserve market and in the second stage, the producer decides the day-ahead bid and the scheduling of the

generators. This is illustrated in Figure 5.1. Generating units have to be running in order to deliver primary reserves and the capacity delivered may not be used in the day-ahead market. There is consequently an opportunity cost associated with delivering primary reserves.

The objective function of the SMIP problem is given by (5.1). Income in the primary reserve market is given by the price of primary reserves, ρ^R_{sb} and the contracted capacity in the primary reserve market, x^R_{sb}. In the day-ahead market, the income is similarly given by the price of day-ahead energy, ρ^D_{sh} and the contracted capacity in the day-ahead market, x^D_{sh}. The cost of production is given by the change in the water value, $w_{s,w}$, and the induced start-up cost c_{shi}.

$$\max \sum_{s \in S} Pr_s \left[\sum_{b \in B} |H^b| \rho^R_{sb} x^R_{sb} + \sum_{h \in H} \rho^D_{sh} x^D_{sh} - \sum_{w \in W} (W^0 - w_{ws}) - \sum_{h \in H} \sum_{i \in I} c_{shi} \right] \tag{5.1}$$

The producer bids continuous volumes in both the primary reserve market and day-ahead market. Bidding is only modeled explicitly in the first market clearing process in the primary reserve market. Bids are given as volume-price points (z_{sbp}, P_{bp}) that together constitute a bidding-curve. For each block $b \in B$ the market is cleared by linear interpolation between these volume-price points. This is given by (5.2). The cleared, contracted capacity in the primary reserve market is given by x^R_{sb}. There are also market rules imposing a non-decreasing bid curve which is imposed by (5.3). The maximum amount of capacity that the producer can bid is limited by the capacity in his portfolio. Because the delivery of primary reserves is symmetric on units, the delivery of primary reserves is restricted by half of the difference between the maximum and minimum capacity of the generator. Thus, the maximum amount of reserve capacity a unit can deliver lies at the midpoint between the maximum and minimum production of the unit. This is given by (5.4), in which Q^{min}_i and Q^{max}_i denote the minimum and maximum production of a unit $i \in I$, respectively. The non-anticipativity constraints for the first-stage variables are given in (5.5), in which $\bar{z}_{bp}$ are common target variables for the capacity volume bid in the primary reserve market.

$$x^R_{sb} = z_{sbp} + (z_{sb(p+1)} - z_{sbp}) \frac{\rho^R_{sb} - P_{bp}}{P_{b(p+1)} - P_{bp}} \quad \text{if } P_{bp} \leq \rho^R_{sb} \leq P_{b(p+1)} \qquad b \in B, p \in P \setminus \{|P|\}, s \in S \tag{5.2}$$

$$z_{sbp} \leq z_{sb(p+1)} \qquad s \in S, b \in B, p \in P \setminus \{|P|\} \tag{5.3}$$

$$z_{sbp} \leq \frac{1}{2} \sum_{i \in I} (Q^{max}_i - Q^{min}_i) \qquad s \in S, b \in B, p \in P \tag{5.4}$$

$$z_{sbp} = \bar{z}_{bp} \qquad s \in S, b \in B, p \in P \tag{5.5}$$

The producer must cover its delivery obligations from the market clearing process. Constraint (5.6) and (5.7) handle this for the day-ahead market and reserve market respectively. The resulting production and capacity reservation for each unit $i \in I$ in the

producers portfolio, are given by q^D_{shi} and q^R_{shi}, for each market respectively. Note that the producer freely can distribute the capacity over the portfolio of its generators. Thus, different units within the portfolio can deliver primary reserves during the different hours in a block. The capacity available at a single generator is hence allowed to vary, but the producer is always obligated to cover total delivery obligations.

$$\sum_{i \in I} q^D_{shi} = x^D_{sh} \qquad s \in S, h \in H \tag{5.6}$$

$$\sum_{i \in I} q^R_{shi} = x^R_{sb} \qquad s \in S, b \in B, h \in H^b \tag{5.7}$$

In the Nordic market, reserves are symmetric on unit level. This means that upward and downward regulation reserves have to be the same for each unit. Constraints on production, q^D_{shi}, are thus restricted by contracted delivery in the reserve market and the minimum and maximum production limits as given by (5.8) and (5.9), in which u_{shi} is 1 if a unit is on, and 0 when it is not producing.

$$Q^{max} u_{shi} - q^D_{shi} \geq q^R_{shi} \qquad s \in S, h \in H^b, i \in I \tag{5.8}$$

$$q^D_{shi} - Q^{min} u_{shi} \geq q^R_{shi} \qquad s \in S, h \in H^b, i \in I \tag{5.9}$$

5.3 Hydropower modeling

Change in the volume of water, v_{shj}, in a reservoir is determined by the inflow to the reservoir, I_{hj}, and the discharge, d_{shi}, and spill, s_{shj} from the reservoir. The reservoir balancing constraints are given in the general case by (5.10), and by (5.11) for the first hour. Indicator matrices, Γ_{ij} and $\Lambda_{jj'}$, are used to represent connections between different generators and reservoirs.

$$v_{shj} - v_{s(h-1)j} = I_{hj} + \sum_{i \in I} \Gamma_{ij} d_{shi} + \sum_{j' \in J} \Lambda_{jj'} s_{hj'} \qquad s \in S, h \in H, j \in J \tag{5.10}$$

$$v_{s,1,j} - V^0_j = I_{1,j} + \sum_{i \in I} \Gamma_{ij} d_{s,1,i} + \sum_{j' \in J} \Lambda_{jj'} s_{1,j'} \qquad s \in S, j \in J \tag{5.11}$$

There are known head loss effects of running multiple units in the same plant simultaneously. Production is modeled per unit and this effect is not taken into account. The model, furthermore, does not take the production's dependency on head water level into account. The head water dependency is considered negligible for the problem at hand, and not modeling head water dependency facilitates using linear programming. The effect of the discharge rate on the efficiency of the plant is however important to include. This is included in (5.12), which models the production discharge-curve by a piece-wise linear approximation, in which A_{if} is the constant and B_{if} is the slope of the production

function for segment $f \in F$ for unit $i \in I$. Maximum discharge is modeled in (5.13), in which D_i is the maximum discharge of unit $i \in I$.

$$q^D_{shi} \leq A_{if} u_{shi} + B_{if} d_{shi} \qquad s \in S, h \in H, i \in I, f \in F \tag{5.12}$$
$$d_{shi} \leq D_i u_{shi} \qquad s \in S, h \in H, i \in I \tag{5.13}$$

Costs arise when units are started and stopped. Start-up costs, C^{su}_i, are larger than the shut-down costs and are the only costs included. Shut-down costs can also be incorporated in the start-up costs such that a shut-down and start-up is modeled as one single event. This is handled in the general case by (5.14), and by (5.15) for the first hour.

$$c_{shi} \geq C^{su}_i (u_{shi} - u_{s(h-1)i}) \qquad s \in S, h \in H, i \in I \tag{5.14}$$
$$c_{s,1,i} \geq C^{su}_i (u_{s,1,i} - U^0_i) \qquad s \in S, i \in I \tag{5.15}$$

The water value represents the expected future income of a marginal amount of stored water in the reservoir. For each watercourse $w \in W$ the future income F_{lw} and the marginal water value W_{jl} for each reservoir $j \in J^w$ are estimated for $l \in L$ different reference reservoir levels V_{jl}. The reference level $l \in L$ is also called a cut. The end water value in the short-term model, w_{sw}, for each watercourse is approximated as given by (5.16), in which F_{lw} and V_{jl} are parameters from a seasonal model and v_{shj} is the water level in the reservoirs at the end of the planning period.

$$w_{sw} \leq F_{lw} - \sum_{j \in J^w} W_{jl}(V_{jl} - v_{s|H|j}) \qquad s \in S, l \in L, w \in W \tag{5.16}$$

Constraints (5.17)-(5.28) are the domain restrictions of the variables.

$$\bar{z}_{bp} \geq 0 \qquad b \in B, p \in P \tag{5.17}$$
$$z_{sbp} \geq 0 \qquad s \in S, b \in B, p \in P \tag{5.18}$$
$$x^D_{sh} \geq 0 \qquad s \in S, h \in H \tag{5.19}$$
$$x^R_{sb} \geq 0 \qquad s \in S, b \in B \tag{5.20}$$
$$w_{sw} \geq 0 \qquad s \in S, w \in W \tag{5.21}$$
$$q^D_{shi} \geq 0 \qquad s \in S, h \in H, i \in I \tag{5.22}$$
$$q^R_{shi} \geq 0 \qquad s \in S, h \in H, i \in I \tag{5.23}$$
$$d_{shi} \geq 0 \qquad s \in S, h \in H, i \in I \tag{5.24}$$
$$v_{shj} \geq 0 \qquad s \in S, h \in H, j \in J \tag{5.25}$$
$$s_{shj} \geq 0 \qquad s \in S, h \in H, j \in J \tag{5.26}$$
$$c_{shi} \geq 0 \qquad s \in S, h \in H, i \in I \tag{5.27}$$
$$u_{shi} \in \{0, 1\} \qquad s \in S, h \in H, i \in I \tag{5.28}$$

Chapter 6

Implementation of the decomposition methods

The decomposition methods investigated in this thesis were presented in general terms in Chapter 3. This chapter is more problem-specific and presents the mathematical formulation of the decomposition methods when applied to the hydro bidding SMIP problem.

6.1 Implementation of Lagrangian decomposition

When applying Lagrangian decomposition to the bidding problem we relax the non-anticipativity constraints. The Lagrangian decomposition formulation of the hydro bidding problem is given in (6.1) or (6.2), depending on how the non-anticipativity constraints are formulated in the original; either cyclic or using a target variable respectively. Both of these approached have been implemented, since different versions are used depending on the multiplier updating scheme.

$$
\begin{aligned}
D(\mu_{sbp}) = \max \sum_{s \in S/|S|} Pr_s \Bigg[& \sum_{b \in B} \rho^R_{sb} x^R_{sb} + \sum_{h \in H} \rho^D_{sh} x^D_{sh} - \sum_{w \in W} (W^0_w - w_{ws}) - \sum_{h \in H} \sum_{i \in I} c_{shi} - \\
& \sum_{b \in B} \sum_{p \in P} \mu_{sbp} (z_{(s+1)bp} - z_{sbp}) \Bigg] + Pr_{|S|} \Bigg[\sum_{b \in B} \rho^R_{|S|b} x^R_{|S|b} + \\
& \sum_{h \in H} \rho^D_{|S|h} x^D_{|S|h} - \sum_{w \in W} (W^0_w - w_{w|s|}) - \sum_{h \in H} \sum_{i \in I} c_{|S|hi} - \sum_{b \in B} \sum_{p \in P} \mu_{|S|bp} (z_{1bp} - z_{|S|bp}) \Bigg]
\end{aligned}
\tag{6.1}
$$

$$D(\mu_{sbp}) = \max \sum_{s\in S} Pr_s \left[\sum_{b\in B} \rho^R_{sb} x^R_{sb} + \sum_{h\in H} \rho^D_{sh} x^D_{sh} - \sum_{w\in W} (W^0_w - w_{ws}) - \sum_{h\in H} \sum_{i\in I} c_{shi} - \sum_{b\in B} \sum_{k\in P} \mu_{sbp}(z_{sbp} - \hat{z}_{bp}) \right] \tag{6.2}$$

The constraints in the subproblems are similar to those in the deterministic equivalent. The difference is that each constraint is only defined for the scenario in the given subproblem, and that the non-anticipativity constraints have been relaxed. The constraints are given by (6.3)–(6.27).

$$x^R_{sb} = z_{sbp} + (z_{sb(p+1)} - z_{sbp}) \frac{\rho^R_{sb} - P_{bp}}{P_{b(p+1)} - P_{bp}} \quad \text{if } P_{bp} \le \rho^R_{sb} \le P_{b(p+1)} \quad b \in B, p \in P \setminus \{|p|\} \tag{6.3}$$

$$z_{sbp} \le z_{sb(p+1)} \qquad b \in B, p \in P \setminus \{|P|\} \tag{6.4}$$

$$z_{sbp} \le \frac{1}{2} \sum_{i\in I} (Q^{max}_i - Q^{min}_i) \qquad b \in B, p \in P \tag{6.5}$$

$$\sum_{i\in I} q^D_{shi} = x^D_{sh} \qquad h \in H \tag{6.6}$$

$$\sum_{i\in I} q^R_{shi} = x^R_{sb} \qquad b \in B, h \in H^b \tag{6.7}$$

$$Q^{max} u_{shi} - q^D_{shi} \ge q^R_{shi} \qquad h \in H^b, i \in I \tag{6.8}$$

$$q^D_{shi} - Q^{min} u_{shi} \ge q^R_{shi} \qquad h \in H^b, i \in I \tag{6.9}$$

$$v_{shj} - v_{s(h-1)j} = I_{hj} + \sum_{i\in I} \Gamma_{ij} d_{shi} - \sum_{j'\in J} \Lambda_{jj'} s_{hj'} \qquad h \in H, j \in J \tag{6.10}$$

$$v_{s,1,j} - V^0_j = I_{1,j} + \sum_{i\in I} \Gamma_{ij} d_{s,1,i} + \sum_{j'\in J} \Lambda_{jj'} s_{1,j'} \qquad j \in J \tag{6.11}$$

$$q^D_{shi} \le A_{if} + B_{if} d_{shi} \qquad h \in H, i \in I, f \in F \tag{6.12}$$

$$c_{shi} \ge C^{su}_i (u_{shi} - u_{s(h-1)i}) \qquad h \in H, i \in I \tag{6.13}$$

$$c_{shi} \ge C^{su}_i (u_{shi} - u_{s(h-1)i}) \qquad h \in H, i \in I \tag{6.14}$$

$$c_{s,1,i} \ge C^{su}_i (u_{s,1,i} - U^0_i) \qquad i \in I \tag{6.15}$$

$$w_{sw} \le F_{lw} - \sum_{j\in J^w} W_{jl} (V_{jl} - v_{s|H|j}) \qquad l \in L, w \in W \tag{6.16}$$

$$z_{sbp} \geq 0 \qquad b \in B, p \in P \tag{6.17}$$
$$x^D_{sh} \geq 0 \qquad h \in H \tag{6.18}$$
$$x^R_{sb} \geq 0 \qquad b \in B \tag{6.19}$$
$$w_{sw} \geq 0 \qquad w \in W \tag{6.20}$$
$$q^D_{shi} \geq 0 \qquad h \in H, i \in I \tag{6.21}$$
$$q^R_{shi} \geq 0 \qquad h \in H, i \in I \tag{6.22}$$
$$d_{shi} \geq 0 \qquad h \in H, i \in I \tag{6.23}$$
$$v_{shj} \geq 0 \qquad h \in H, j \in J \tag{6.24}$$
$$s_{shj} \geq 0 \qquad h \in H, j \in J \tag{6.25}$$
$$c_{shi} \geq 0 \qquad h \in H, i \in I \tag{6.26}$$
$$u_{shi} \in \{0, 1\} \qquad h \in H, i \in I \tag{6.27}$$

The upper bound is minimized through the Lagrangian dual problem given by (6.28). We will consider this as the master problem of our implementation.

$$\min_{\mu_{sbp}} D(\mu_{sbp}) = \min_{\mu_{sbp}} \sum_{s \in S} D_s(\mu_{sbp}) \tag{6.28}$$

s. t.

$$\mu_{sbp} \quad free \qquad s \in S, b \in B, p \in P \tag{6.29}$$

Because all constraints that have been dualized are equality constraints, the corresponding Lagrangian multipliers are unrestricted in sign.

The multiplier updating schemes applied are mathematically described below. Two of the methods, the subgradient and the simplified progressive hedging algorithm, update the multipliers according to a formula and makes use of the formulation in (6.1). The other three solve an optimization problem in order to update the multiplier vector, thus allowing for adding the constraint $\sum_{s \in S} \mu_s = 0$ and the target variable formulation of (6.2) can therefore be used. The target formulation of the non-anticipativity constraint is also the one adopted in the Dantzig-Wolfe decomposition.

The different multiplier updating procedure are described in detail in the following sections.

6.1.1 Subgradient method

When implementing the subgradient method we adopt the objective function formulation given in (6.2). Due to the cyclic nature of this formulation, we define a separate constraint for the border case $s = |S|$, denoted by the (x.b) equations.

The subgradient does not perform any optimization as part of the master problem. Instead, the master problem consists of a loop where in each iteration, the multipliers are updated according to (6.30a)-(6.30b).

$$\mu_{sbp}^{(k+1)} = \mu_{sbp}^{(k)} + \pi_{sbp}^{(k)}(z_{(s+1)bp} - z_{sbp}) \qquad s \in S \setminus |S|, b \in B, p \in P, k \in 1...K \tag{6.30a}$$

$$\mu_{|S|bp}^{(k+1)} = \mu_{|S|bp}^{(k)} + \pi_{|S|bp}^{(k)}(z_{1bp} - z_{|S|bp}) \qquad b \in B, p \in P, k \in 1...K \tag{6.30b}$$

The subgradient $g_{sbp}^{(k)}$ is a measure for the violation of the corresponding non-anticipativity constraint and is given by (6.31a)-(6.31b).

$$g_{sbp}^{(k)} = z_{(s+1)bp} - z_{sbp} \qquad s \in S \setminus |S|, b \in B, p \in P, k \in 1...K \tag{6.31a}$$

$$g_{|S|bp}^{(k)} = z_{1bp} - z_{|S|bp} \qquad b \in B, p \in P, k \in 1...K \tag{6.31b}$$

The vector $\pi_{sbp}^{(k)}$ is the step length. The step length formula for calculating π may vary across implementations. We adopt the step size formula of (Escudero et al., 2013) given in (6.32a)-(6.32b).

$$\pi_{sbp}^{(k)} = t^{(k)} \cdot \frac{z_{LD} - \bar{z}_{LD}(\mu_{sbp}^{(k)})}{\|z_{(s+1)bp} - z_{sbp}\|} \qquad s \in S \setminus |S|, b \in B, p \in P, k \in 1...K \tag{6.32a}$$

$$\pi_{|S|bp}^{(k)} = t^{(k)} \cdot \frac{z_{LD} - \bar{z}_{LD}(\mu_{|S|bp}^{(k)})}{\|z_{1bp} - z_{|S|bp}\|} \qquad b \in B, p \in P, k \in 1...K \tag{6.32b}$$

in which $\bar{z}_{LD}$ is the best feasible solution found so far and $t^{(k)}$ is an initial step size parameter, a scalar that is gradually decreased to ensure convergence.

6.1.2 Cutting plane method

In contrast to the subgradient method, the cutting plane master problem is solved as an optimization problem. In each iteration (6.33)–(6.35) are solved.

$$\min \quad \sum_{s \in S} \theta_s \tag{6.33}$$

$$\text{s.t.} \quad \sum_{s \in S} \mu_{sbp} = 0 \qquad b \in B, p \in P, k \in 1...K \tag{6.34}$$

$$\theta_s \geq D_s(\mu_{sbp}^{(k)}) - (z_{sbp}^{(k)})(\mu_{sbp} - \mu_{sbp}^{(k)}) \qquad s \in S, b \in B, p \in P, k \in 1...K \tag{6.35}$$

6.1.3 Proximal bundle method

The master problem of the proximal bundle method is similar to that of the cutting plane method, but with an additional quadratic term in the master problem objective. It is

given by (6.36)–(6.38)

$$\min \quad \sum_{s\in S}\theta_s - \frac{1}{2}\tau\sum_{s\in S}\|\mu_{sbp}-\hat{\mu}_{bp}\|^2 \tag{6.36}$$

$$\text{s.t.} \quad \sum_{s\in S}(\mu_s-\hat{\mu}_{sbp})=0 \qquad b\in B, p\in P, k\in 1...K \tag{6.37}$$

$$\theta_s + z^{(k)}_{sbp}(\mu_{sbp}-\hat{\mu}_{bp}) \geq D_s(\mu^k_{sbp}) - (z^{(k)}_{sbp})(\mu_{sbp}-\mu^k_{sbp}) \quad s\in S, b\in B, p\in P, k\in 1...K \tag{6.38}$$

6.1.4 Progressive hedging

In every iteration k, the variables included in the non-anticipativity constraints are calculated based on (6.39)–(6.41).

$$z^{(k+1)}_{sbp} = \underset{z,x,w,c}{\text{argmax}} \left\{ \sum_{s\in S/|S|} Pr_s \left[\sum_{b\in B}\rho^R_{sb}x^R_{sb} + \sum_{h\in H}\rho^D_{sh}x^D_{sh} - \sum_{w\in W}(W^0_w - w_{ws}) - \sum_{h\in H}\sum_{i\in I}c_{shi} \right.\right.$$
$$\left.\left. - \sum_{b\in B}\sum_{p\in P}\left(\omega^{(k)}_{sbk}x - \frac{\rho}{2}\|z_{sbp}-\bar{z}^{(k)}_{bp}\|^2\right)\right]\right\} \quad s\in S, b\in B, p\in P \tag{6.39}$$

s. t.

$$(6.3)-(6.27)$$

in which,

$$\omega^{(k)}_{sbp} := \omega^{(k-1)}_{sbp} + \rho(z^{(k)}_{sbp} - \bar{z}^{(k)}_{bp}) \qquad s\in S, b\in B, p\in P \tag{6.40}$$

where

$$\bar{z}^{(k)}_{bp} := \sum_{s\in S} Pr_s z^{(k)}_{sbp} \qquad b\in B, p\in P \tag{6.41}$$

6.1.5 Extensions to the Lagrangian relaxation

Warm-start
The choice of initial multipliers can sometimes significantly impact the overall performance of the Lagrangian decomposition method. In order to try to speed up convergence, the Lagrangian decomposition algorithm has been seeded with non-zero initial multipliers. We apply a warm-start procedure where the dual variables from the non-anticipativity constraints from the solution to the LP-relaxation have been used as initial Lagrangian multipliers.

Scenario clustering
When clustering scenarios, the non-anticipativty constraints between the scenarios within

the same cluster are merged and considered implicitly. This can result in tighter bounds and also decrease the number of iterations required, but the size and complexity of these aggregated subproblems do also increase.

A mathematical formulation of the clustered Lagrangian dual function is given in (6.42). The clusters are indexed by Ω, the set of scenarios and reserve bid of cluster Ω are denoted by S^{Ω} and $z_{pb\Omega}$ respectively.

$$D(\mu) = \max \sum_{\Omega \in C} \sum_{s \in S^{\Omega}} Pr_s \left[\sum_{b \in B} \rho^R_{sb} x^R_{sb} + \sum_{h \in H} \rho^D_{sh} x^D_{sh} - \sum_{w \in W} (W^0_w - w_{ws}) - \sum_{h \in H} \sum_{i \in I} c_{shi} \right] - \sum_{p \in P} \sum_{b \in B} \sum_{\Omega \in C} \mu_{pb\Omega} (z_{pb\Omega} - \hat{z}_{pb}) \tag{6.42}$$

6.2 Implementation of Dantzig-Wolfe decomposition

In the Dantzig-Wolfe decomposition of the SMIP problem we let the master problem consist of the objective function (5.1) and the non-anticipativity constraints (5.5) of the deterministic equivalent. Its interior representation is used and the resulting master problem is given by (6.43)–(6.47), in which (6.43) is the objective function and (6.44) represents the non-anticipativity constraints. The convexity constraints of the weights, $\lambda^{(m)}_s$, on each extreme point $m \in M$ in scenario $s \in S$ are given by (6.45). Constraints (6.46) and (6.47) are the domain restrictions on the variables. For the RMP the set of extreme points, M, does not include all extreme points of the feasible solution space of the problem, but are added iteratively.

$$\max \quad \sum_{s \in S} Pr_s \left[\sum_{m \in M} \sum_{b \in B} |H^b| \rho^R_{sb} x^{R(m)}_{sb} \lambda^{(m)}_s + \sum_{h \in H} \rho^D_{sh} x^{D(m)}_{sh} \lambda^{(m)}_s - \sum_{w \in W} (W^0_w - w^{(m)}_{ws} \lambda^{(m)}_s) - \sum_{h \in H} \sum_{i \in I} c^{(m)}_{shi} \lambda^{(m)}_s \right] \tag{6.43}$$

$$\text{s.t.} \quad \sum_{m \in M} z^{(m)}_{sbp} \lambda^{(m)}_s = \bar{z}_{bp} \qquad s \in S, b \in B, p \in P \tag{6.44}$$

$$\sum_{m \in M} \lambda^{(m)}_s = 1 \qquad s \in S \tag{6.45}$$

$$\lambda^{(m)}_s \geq 0 \qquad m \in M, s \in S \tag{6.46}$$

$$\bar{z}_{bp} \geq 0 \qquad b \in B, p \in P \tag{6.47}$$

Each subproblem is defined for a specific scenario s and consists of the remaining constraints of the deterministic equivalent. The constraints in the subproblem are thus the same as those in the subproblems of the Lagrangian decomposition, and are given by

(6.3)–(6.27). The objective function of the subproblem is given by (6.48). It includes the dual values α_{sbp} and β_s of (6.44) and (6.45), respectively, and forms the reduced cost of the columns.

$$\max \quad Pr_s \left[\sum_{b \in B} |H^b| \rho^R_{sb} x^R_{sb} + \sum_{h \in H} \rho^D_{sh} x^D_{sh} - \sum_{w \in W} (W^0_w - w_{ws}) - \sum_{h \in H} \sum_{i \in I} c_{shi} - \sum_{b \in B} \sum_{p \in P} \alpha_{sbp} z_{sbp} - \beta_s) \right] \tag{6.48}$$

6.2.1 Improving convergence by stabilization

To accelerate convergence, the stabilization method of Du Merle et al. (1999) can be implemented. The stabilization procedure is implemented by including slack variables, y^-_{sbp}, and surplus variables, y^+_{sbp}, which are restricted by ε^- and ε^+ respectively. The slack and surplus variables are added to the objective function with parameters δ^-_{sbp} and δ^+_{sbp} deciding the penalization of surplus and slack.

$$\max \quad \sum_{s \in S} Pr_s \left[\sum_{m \in M^t} \sum_{b \in B} |H^b| \rho^R_{sb} x^{R(m)}_{sb} \lambda^{(m)}_s + \sum_{h \in H} \rho^D_{sh} x^{D(m)}_{sh} \lambda^{(m)}_s - \sum_{w \in W} (W^0_w - w^{(m)}_{ws} \lambda^{(m)}_s) - \sum_{h \in H} \sum_{i \in I} c^{(m)}_{shi} \lambda^{(m)}_s \right] - \delta^-_{sbp} y^-_{sbp} + \delta^+_{sbp} y^+_{sbp} \tag{6.49}$$

$$\sum_{m \in M^t} z^{(m)}_{sbp} \lambda^{(m)}_s - y^-_{sbp} + y^+_{sbp} = \bar{z}_{bp} \qquad s \in S, b \in B, p \in P \tag{6.50}$$

$$\sum_{m \in M^t} \lambda^{(m)}_s = 1 \qquad s \in S \tag{6.51}$$

$$y^-_{sbp} \leq \varepsilon^- \qquad s \in S, b \in B, p \in P \tag{6.52}$$

$$y^+_{sbp} \leq \varepsilon^+ \qquad s \in S, b \in B, p \in P \tag{6.53}$$

$$\lambda^{(m)}_s \geq 0 \qquad m \in M^t, s \in S \tag{6.54}$$

$$\bar{z}_{bp} \geq 0 \qquad b \in B, p \in P \tag{6.55}$$

$$y^-_{sbp} \geq 0 \qquad s \in S, b \in B, p \in P \tag{6.56}$$

$$y^+_{sbp} \geq 0 \qquad s \in S, b \in B, p \in P \tag{6.57}$$

6.2.2 The implemented solution process

The solution process is divided in three separate phases that are linked together as illustrated in Figure 6.1 for the branch and price algorithm, in which the problem is solved

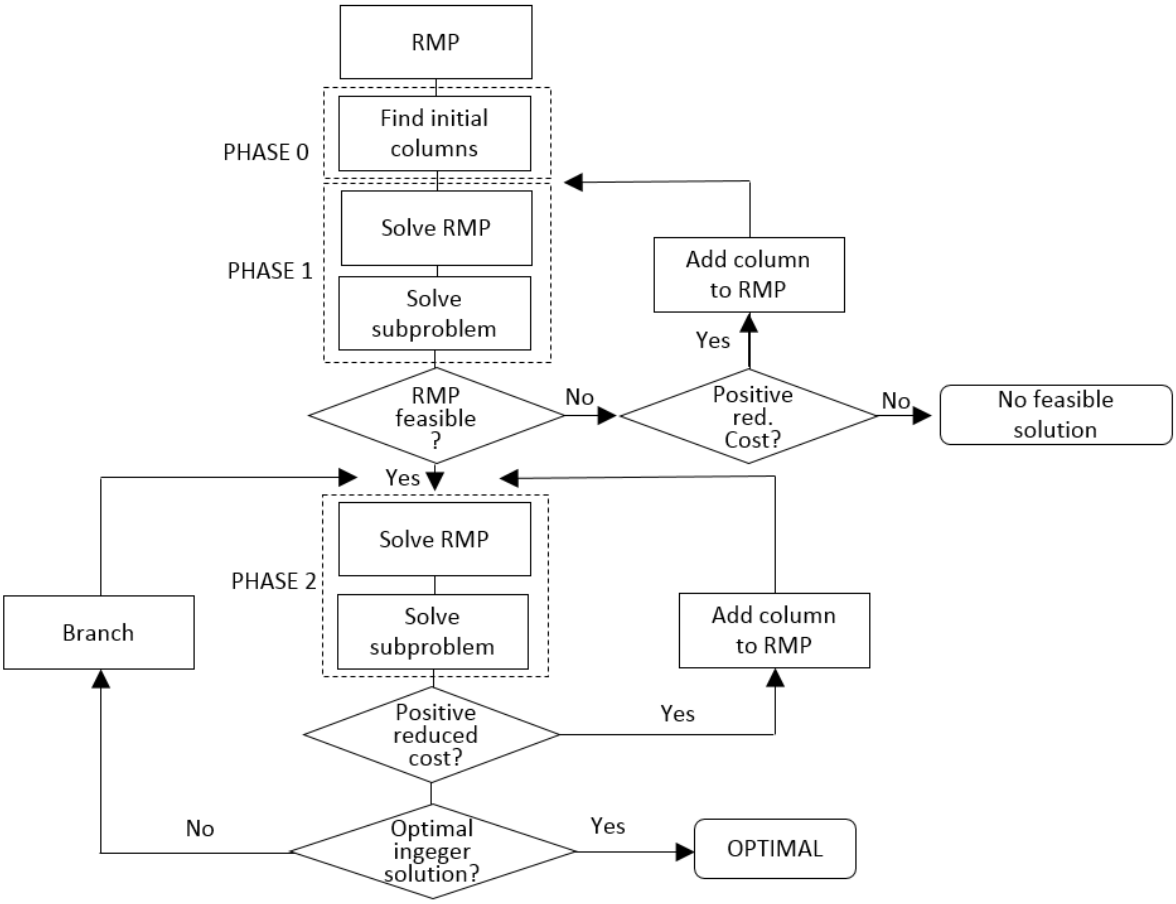


Figure 6.1: Solution procedure of the column generation procedure implemented.

by Dantzig-Wolfe decomposition in each node. The column generation process needs an initial set of columns in the RMP to start the solution process. These solutions are found in an initialization process, that we call Phase 0, in which each subproblem is solved once with all dual values set to zero. Phase 0 is only executed once in the first node.

Phase 0 does not assure that the non-anticipativity constraints are adhered to. Thus, Phase 1 is started after Phase 0 to find a feasible solution to the RMP. This is done by penalizing surplus and slack in the non-anticipativity constraints. This results in an RMP which is similar to the stabilized RMP, but without the bounds on the surplus and slack variables. The RMP in Phase 1 is thus given by (6.49)–(6.51) and (6.54)–(6.57).

In phase 2 the main column generation procedure is executed using the stabilized version given by (6.49)–(6.57). Phase 2 ends when no subproblems return columns with strictly positive reduced cost.

To find integer-feasible solutions from the column generation procedure, branch and price is implemented. When branch and price is applied, the current optimal solution is checked for fractional values in the unit commitment decisions. Pruning is conducted if the solution is integer-feasible or if the the objective function value is lower than a feasible integer solution already found. The best-first strategy is used to select the node in which branching should be performed. That is, the node with the highest objective value is chosen to

be the parent node of the two next nodes that are to be solved. In the first new node, the unit commitment decision with the most fractional value is fixed to 1, and only columns in the parent node that have a value of 1 for this variable are transferred to the new node. In the second new node, the same variable is branched to 0, and the remaining columns from the parent node are transferred to this node. Note that branching is done on the original variables from the outer representation of the problem. When each of the new nodes are solved, the variables that have been branched on are fixed in the subproblems. The branch and price procedure originally continues until the optimal integer solution is found.

The iterative nature of the column generation procedure, might result in a large number of iterations to find the optimal solution. To test if the number of iterations could be decreased, a version of the Dantzig-Wolfe decomposition in which scenarios are clustered in the subproblems was implemented. This is inspired by the scenario clustering in Lagrangian relaxation. The method is implemented by clustering two or more scenarios together, and including non-anticipativity constraints in the scenarios that belong to the same scenario cluster in the subproblem. The non-anticipativity constraints in the RMP are thereafter redefined such that they are defined for each scenario cluster instead of each scenario.

6.2.3 Early stopping in the subproblems

It is possible to determine criteria for when to stop the subproblems if they are considered to be too hard to solve until optimality. Two such criteria are used during testing of the decomposition methods. These are maximum time and relative duality gap. When putting a limit on the relative duality gap, the search for better solution stops when that particular gap is reached. When a maximum time is set, the subproblem returns the best feasible solution it has found when it has been solved for the given number of seconds. If no feasible solution is found at that time, the solver will continue to solve the problem until the first feasible solution is found and return this solution.

It is not straightforward to choose which cutoff is the best one. If a maximum time is set, there is no guarantee of the size of the gap in the subproblem when time has run out and columns are sent to the RMP. If the column has negative reduced cost it will not be added to the RMP, and the RMP will register that the subproblem could not provide any new columns. If this happens in the iteration in which Phase 2 terminates, it cannot be determined if the current solution in the RMP is the optimal one.

If the the duality gap is used to terminate the search for better solutions in the subproblems, more can be said about true upper bound of the RMP when Phase 2 terminates. If, for instance, we set the relative duality gap to 5%, it is known that the optimal column from a subproblem can be at most 5% higher than the reduced cost of the column that will be returned by the subproblem. It can therefore be guaranteed that when Phase 2 is over, the upper bound of the problem is at most 5% higher than the objective value in the RMP. However, nothing can be said about the time the subproblems spend to find a solution that gives a gap smaller than the gap limit that has been set.

Interrupting the subproblems based on the the criteria mentioned above will not neces-

sarily return an optimal solution. Still, the solutions can be used to recover an upper bound for the overall problem when Phase 2 is finished. This upper bound is found by (3.24), which states that an upper bound to the solution in the RMP is the current objective value plus the reduced cost for the optimal columns from the subproblems. In the case when the optimal column is not found in the subproblem, the duality gap from the subproblem is also added to this bound to find a valid upper bound. The two stopping criteria can also be combined. In this case, the one that is first reached will be used.

6.3 Heuristics for finding feasible solutions

Because Lagrangian relaxation may not find feasible solutions and the solution time of the branch and price procedure can potentially last for a long time, two simple heuristics are implemented for each of the decomposition methods in order to find feasible solutions. The heuristics implemented for each decomposition method are similar and are based on either fixating the primary reserve bid made in the first stage or fixating the unit commitment decisions made in the second stage stage. Because the problem has relative complete recourse, the heuristics that fixate the primary reserve bid should always return feasible solutions. The same holds for when the unit commitment decisions are fixated because always possible to find a set of reserve bids that corresponds to scenario with the minimum reserve commitments, and excess production from the other scenarios can be delivered to the spot market.

When the reserve bids are fixated based on the solution from Lagrangian relaxation, the bids for each scenario are aggregated into one common bid across scenarios in order for the solution to admissible. We experiment with different aggregation techniques; minimum, maximum, mean and mode. The first stage decisions are then potentially adjusted such that they comply with the restriction of having an increasing bid curve and fixated to the values of these aggregated bids. The standard deterministic equivalent is then solved with these reserve bid fixated. When we apply the heuristic that fixates the reserve bid to the best solution found in the RMP of the Dantzig-Wolfe decomposition we can use the last reserve bid that is found before the column generation is interrupted or has stopped. The deterministic equivalent is then solved with the reserve bid fixated. With the first stage decisions fixated the deterministic equivalent should be easier to solve, but because the number of integer variables is large, the solution time of this heuristic may still be long if it is to be solved to optimality. One may therefore have to use integer solutions that are found before optimality is proved.

The heuristic based on fixating the unit commitment decisions is for Lagrangian relaxation done by using the the unit commitment decisions that belong to the solution that gives the best bound found so far. This can be done without any adjustments. The deterministic equivalent is then solved with these variables fixed. For the Dantzig-Wolfe decomposition, the unit commitment decisions that belong to the last solution found in the RMP are not necessarily binary and need to be rounded to the nearest integer value before they are fixated. Phase 1 and Phase 2 of the column generation algorithm are then run again to reach optimality given the fixed unit commitment decision. When the unit commitment decisions are fixated, the resulting problems are LPs, and this should reduce the solution

times of both the deterministic equivalent and the subproblems in the Dantzig-Wolfe decomposition significantly.

Chapter 7

Price modeling and scenario generation

In order to obtain valid and interesting results from the stochastic optimization model, a good representation of the uncertainty is important. Careful generation of scenarios can contribute to reduce the computational time of a stochastic optimization problem (Boomsma et al., 2014), and the quality of the scenario tree is also an important determinant of the quality of the solutions obtained from the model (Kaut, 2014). Forecasting and representation of the uncertainty of electricity prices is furthermore becoming increasingly important in today's electricity markets where decisions on when and to which market to deliver electricity is becoming a more complex task.

Similar to King and Wallace (2012b), we view the representation and the discretization of the random variables as part of the modeling, and this part is therefore emphasized in the thesis. This chapter begins with a presentation of historical day-ahead prices and primary reserve prices before we present the details of the scenario generation process. The hydropower station modeled in this thesis is located in the NO2 price area. Hence area prices from NO2 are used as input data for the scenario generation. Technical and hydrological data received from Hydro that are used as input in the model are valid for the period between 2016-04-23 until 2016-04-29, hence price forecasts for this period are generated.

7.1 Historical prices

As described in Chapter 4, day-ahead energy and primary reserves are bid in hourly and weekly block intervals respectively. Since the day-ahead prices for all hours are quoted at the same time, one day in advance, hourly prices cannot be seen as a time-series, but should rather be considered as a panel of 24 cross-sectional hours that vary from day to day (Huisman et al., 2007). The same holds for the block-prices in the primary reserve market.

7.1.1 Day-ahead prices

Figure 7.1 illustrates the historical day-ahead prices in the NO2 area for all 24 hours from 2013-01-01 to 2015-04-22. Several features that stem from the physical conditions governing electricity production, transmission and consumption can be spotted in the dynamics of these electricity prices.

Table 7.1: Day-ahead price characteristics, 2013-01-01 to 2016-04-22 [NOK/MWh].

Hour	Mean	Median	SD	Skewness	Kurtosis	Min	Max
00:00-01:00	217.44	226.96	68.64	-0.61	3.16	14.61	398.36
01:00-02:00	211.01	220.40	70.13	-0.61	3.11	5.93	390.70
02:00-03:00	207.12	215.90	71.08	-0.59	3.08	4.93	387.44
03:00-04:00	205.30	215.00	71.82	-0.59	3.04	7.10	388.00
04:00-05:00	207.27	217.43	72.00	-0.62	3.07	10.28	389.72
05:00-06:00	214.11	223.47	71.70	-0.64	3.11	9.29	391.05
06:00-07:00	223.93	231.44	70.42	-0.57	3.16	10.37	438.97
07:00-08:00	237.96	243.21	79.97	0.49	6.08	14.97	713.01
08:00-09:00	247.31	248.55	86.63	1.04	7.66	19.11	824.22
09:00-10:00	245.59	248.14	79.07	0.53	5.35	19.65	809.59
10:00-11:00	243.03	247.55	73.94	0.20	4.53	24.34	726.24
11:00-12:00	239.72	246.03	70.28	-0.12	3.87	24.79	679.53
12:00-13:00	236.65	245.11	68.75	-0.33	3.38	24.79	580.59
13:00-14:00	234.57	243.72	68.10	-0.45	3.16	23.98	564.76
14:00-15:00	233.23	241.81	68.87	-0.44	3.18	23.71	561.98
15:00-16:00	232.90	242.94	69.57	-0.45	3.16	25.61	591.64
16:00-17:00	236.25	244.66	73.22	-0.08	4.09	31.29	668.86
17:00-18:00	242.19	247.69	78.47	0.40	5.12	38.86	912.21
18:00-19:00	242.64	248.47	77.28	0.54	6.50	51.88	833.52
19:00-20:00	239.00	246.51	72.13	0.15	5.52	52.44	720.12
20:00-21:00	234.57	243.25	66.81	-0.46	3.13	50.45	480.77
21:00-22:00	231.45	240.57	65.86	-0.52	3.07	47.27	416.78
22:00-23:00	227.47	237.18	65.46	-0.57	2.99	45.28	405.47
23:00-24:00	220.53	231.28	66.86	-0.58	3.06	35.41	400.30

There is a clear annual seasonal pattern during the year; prices increase during early winter and decrease again moving towards spring and summer. Due to the high share of hydro power in the Norwegian electricity system, electricity prices are highly dependent on the hydrological conditions. Prices generally increase when there is less water available and decrease when there is more water available, for instance due to snow fall or melting respectively. In addition, changes in the consumption pattern during the year affect the seasonal pattern.

The historical price data fluctuates around a mean value. The mean value, however, varies from hour to hour, as can be seen in Table 7.1. The prices are lowest during the night and early afternoon, with the lowest mean price being 205.30 NOK/MWh between 03:00-04:00. Prices are highest during the morning and late afternoon, with prices being above 240 NOK/MWh between 08:00-11:00 and 17:00-19:00. The highest mean price is 247.31 NOK/MWh between 08:00-09:00. Since prices are dependent on demand, the

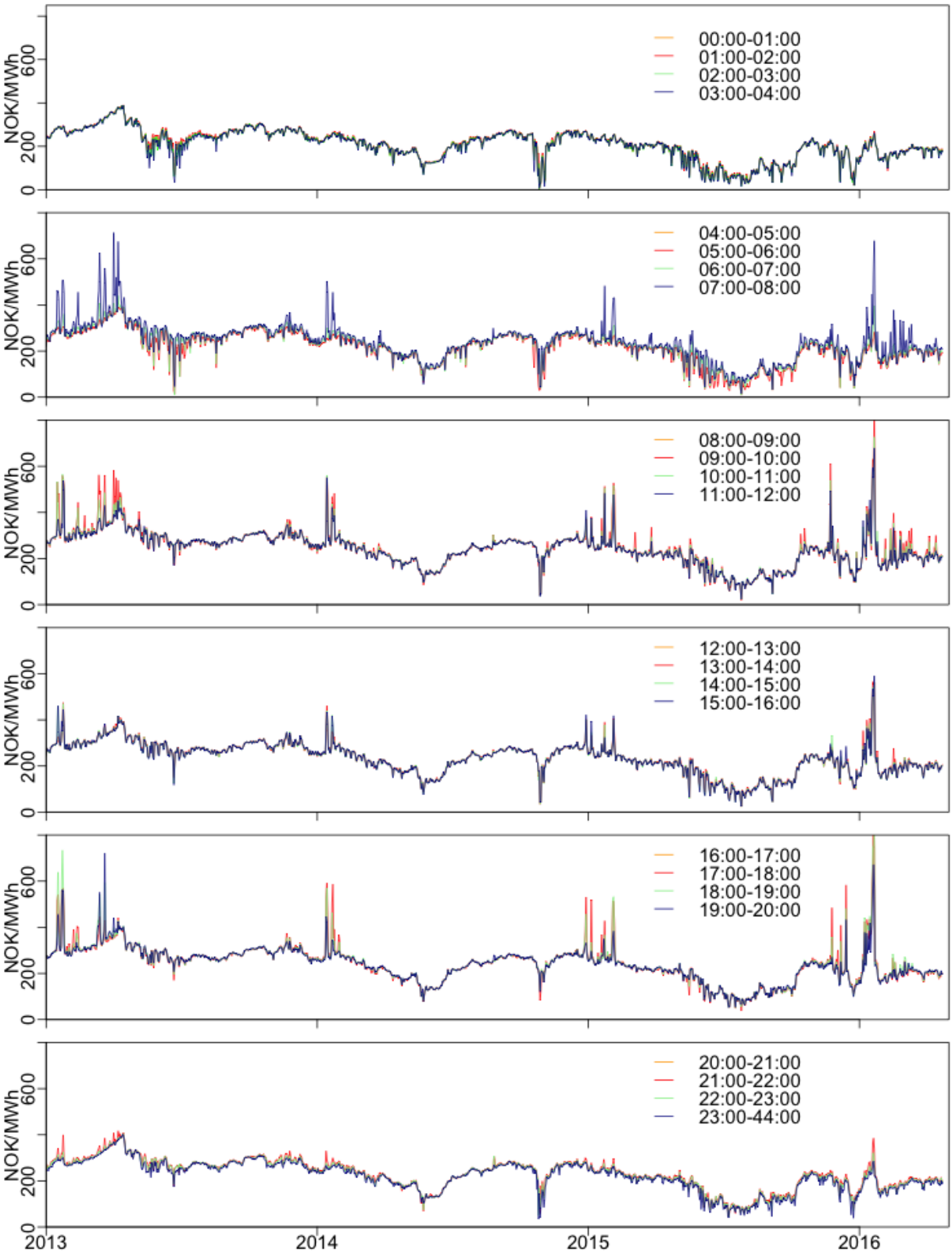


Figure 7.1: Hourly DA prices in NO2, 2013-01-01 to 2016-04-22.

price pattern during the day is a clear indication of the daily consumption pattern, and the common daily load profile is characterized by a pattern of higher demand during the morning and late afternoon.

Electricity prices furthermore show a mean reverting behaviour after periods of positive or negative spikes, meaning that they move towards the mean value of the price process. The direction and frequency of spikes vary throughout the year. During times of high prices, positive spikes occur more frequently and during times of low prices negative spikes are more frequent. There are several reasons for the occurrence of spikes, including unexpected outage of power plants, problems with the transmission grid, extreme temperature, high loads and bidding behaviour of market participants to mention some (Janczura et al., 2013). One common explanation for price spikes in the Norwegian system is the difference in supply and demand level throughout the year. During winter, when there is less water available and higher demand, positive spikes can be the result of sudden scarcity. Negative spikes are often present during spring when the snow is melting. During this period, hydro producers might be obligated to release water to avoid flooding, which again will affect the prices. Table 7.1 also shows that the higher frequency of spikes affects the distribution of the prices. During hours of more spikes, the variance, skewness and kurtosis are higher. The skewness is higher because the spikes are either positive or negative, leading to a less symmetric distribution around the mean. The kurtosis is furthermore higher because the higher spike frequency results in a distribution with larger tails.

The historical data shows that the hourly prices are highly positively correlated. The correlation is on average 0.918, with the smallest correlation being 0.787 and the highest correlation being 0.997. Correlation is largest for consecutive hours. For example, the correlation between consecutive hours is close to 1 from 00:00 and 06:00, while it is closer to 0.8 for hours further apart. The large positive correlation can be explained by the common information set for the panel of 24 hourly prices since they are quoted at the same time. This observation consequently supports modeling electricity prices as a panel of hourly prices. The full correlation matrix can be found in Appendix A.

7.1.2 Primary reserve prices

Figure 7.2 illustrates the historical weekly prices in the NO2 area for the different primary reserve blocks from 2013-01-01 to 2015-04-22. The prices are plotted together with weekly average day-ahead prices for each block. These prices are found as the average weekly price for the hours within the corresponding block in the primary reserve market.

It can be seen that the characteristics and patterns of primary reserves differ considerably from what is seen in the day-ahead market. The most prominent difference is the lower price levels of primary reserves compared to the price levels in the day ahead market. As previously mentioned, producers receive only payment for the reserved capacity, not the possible generated electricity. Most often, the reserves are not generated, and Allen and Ilić (2000) argue that primary reserves generally have an expected low cost and that the price of primary reserves, consequently, is expected to be lower than the price of electricity in the day-ahead market. The volatility of primary reserve prices and the

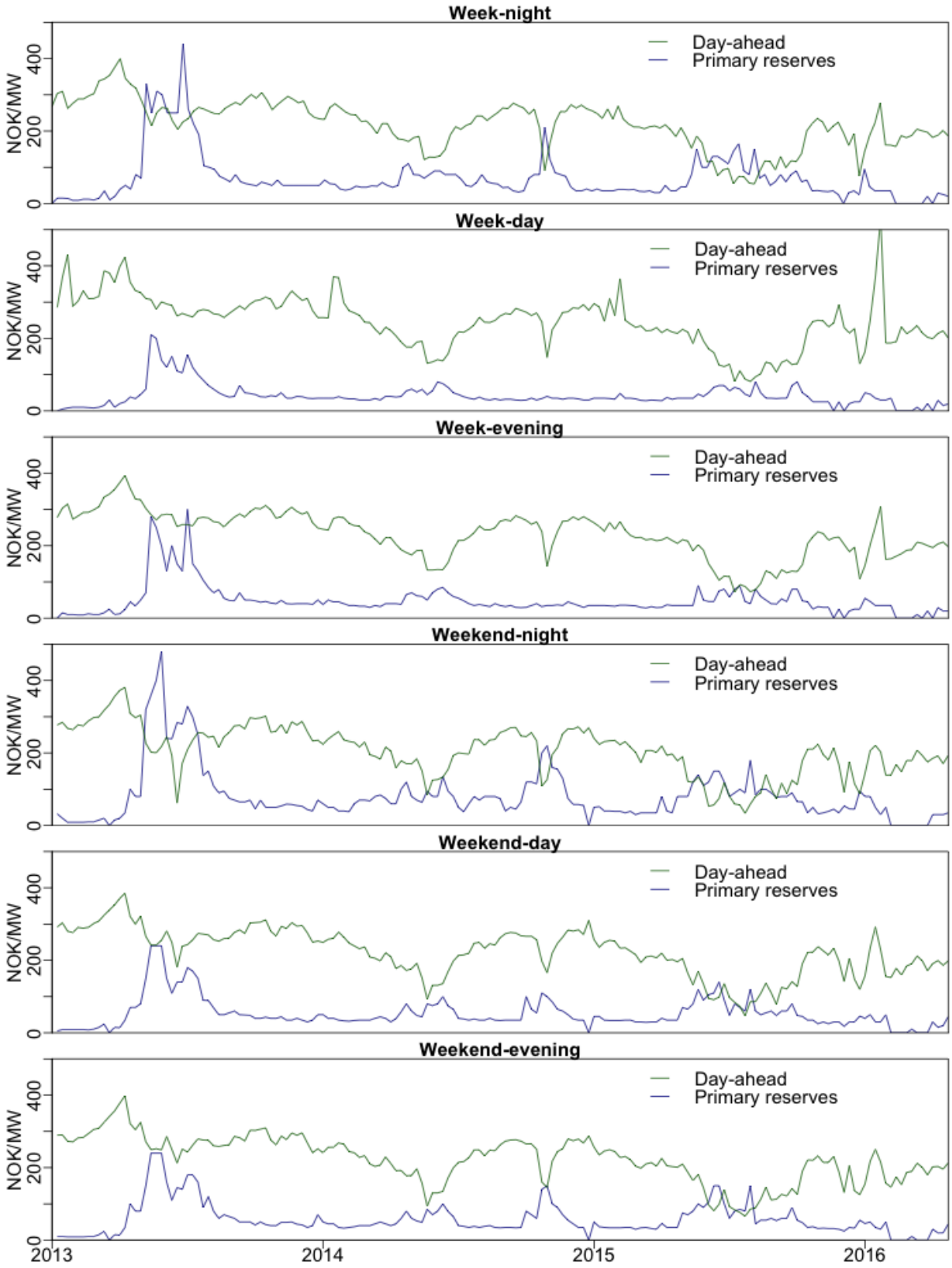


Figure 7.2: Weekly primary reserve prices and average weekly block-wise day-ahead prices, 2013-01-01 to 2016-04-22.

Table 7.2: Primary reserve price characteristics, 2013-01-01 to 2016-04-22 [NOK/MW].

Block	Mean	Median	SD	Skewness	Kurtosis	Min	Max
Week - night	80.65	65.00	66.02	2.66	11.55	0.00	480.00
Week - day	52.73	40.00	31.14	2.60	12.54	0.00	240.00
Week - evening	56.74	45.00	43.50	3.29	16.58	0.00	240.00
Weekend - night	68.81	50.00	75.34	2.51	10.66	0.00	440.00
Weekend - day	41.76	35.00	42.25	2.12	8.89	0.00	210.00
Weekend - evening	48.15	38.00	44.40	1.88	7.31	0.00	300.00

presence of spikes also differ from what seen in the day-ahead market. A further analysis of the primary reserve prices reveals that the price level is higher during the night blocks with a corresponding higher standard deviation. These results are presented in Table 7.2.

Another difference between the two markets is that the seasonal pattern of primary reserves is both less prominent and different from the seasonality in the day-ahead market. Reserve prices are seen to increase during spring and to be on a generally lower level at other times during the year. These variations are negatively correlated with the seasonal variations in the day-ahead market. The negative correlation varies from -0.224 between the weekend-night block and the corresponding average price in the day-ahead market for this block to -0.013 for the week-evening block. The full correlation matrix can be found in Appendix A.

Based on the above discussion, it can be argued that the primary reserve prices are negatively correlated with the expected day-ahead price. The day-ahead market is settled after the primary reserve market, but considering the volumes traded in the primary reserve market compared to the day-ahead market, we argue that the price in the day-ahead market should be modeled as independent of the price of primary reserves. As the day-ahead market prices are not known at the time of bidding in the primary reserve market, the primary reserve market is modeled as dependent on the expected price in the day-ahead market.

As for the day-ahead market, the correlation between the different primary reserve blocks is positive, see Appendix A. The information set is updated at the same time for the weekend block and week blocks separately, thus the correlation is largest between blocks at the same time of the week. The overall positive correlation can be explained by the conditions governing the price formation in electricity markets, such as temperature and rainfall, which will affect the overall price level in the period. The historical prices also show that the correlation is somewhat larger for the weekend blocks compared to the week blocks.

7.2 Scenario generation

Solving a stochastic program with the original continuous distribution can only be done for small instances, otherwise the problem will become too large to solve in a reasonable amount of time with available computational power. Hence, the random variables in

a stochastic optimization problem are often represented in a scenario tree in which the continuous distribution of the random variables is discretized into a set of scenarios.

There are several methods developed to generate scenario trees. Kaut and Wallace (2007) provide a survey of some of the main scenario generation methods. The paper also describes how to evaluate the scenario tree by stability requirements and testing for bias. Another short presentation of relevant scenario generation methods can be found in Conejo et al. (2010). Conejo et al. (2010) divide the methods in four different categories; path based methods, moment matching methods, internal sampling and scenario reduction. Path based methods are methods in which complete paths are built from econometric and time series models. In moment matching methods, a discrete distribution satisfying the statistical properties of the underlying distribution is generated. Internal sampling is a scenario generation method based on sampling from the original distribution. Scenario reduction can be combined with other methods in order to reduce the number of scenarios to a prescribed cardinality. This is often needed because the possible number of scenarios is limited by the computational power when solving the stochastic optimization problem.

The distribution of prices in the day-ahead and primary reserve market is multivariate. Their interdependency is modeled such that both price processes are dependent on the expected price in the day-ahead market, and they are assumed to be independent conditional on the expected day-ahead price. The conditional independence between them is due to the small volumes traded in the primary reserve market, which is the market that is settled first. Because the volumes in the reserve market are much smaller than the volumes in the day-ahead market, totally ~ 1 GW traded primary reserves compared to a 374 TWh in the Nordic-Baltic day-ahead market in 2015 (Statnett, 2016; Nord Pool, 2016a), the revelation of the reserve price is assumed to not affect the realization of the prices in the day-ahead market. An illustration is provided in Figure 7.3.

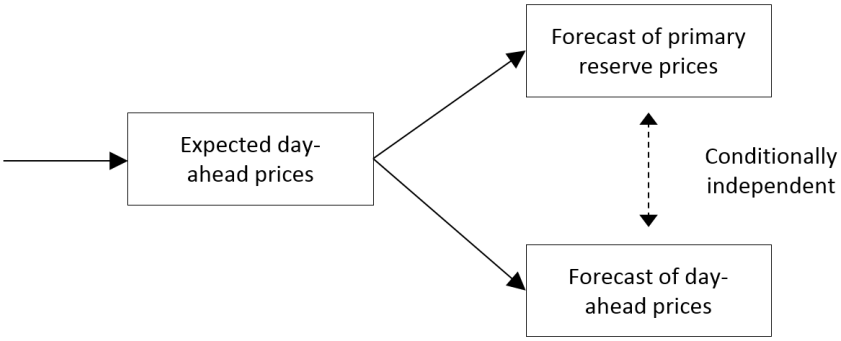


Figure 7.3: Illustration of the relationship between prices in the primary reserve market and day-ahead market.

Many of the methods for scenario generation are only suitable when the probability distribution is univariate. Having a multivariate distribution, only some of the methods are seen as applicable. Amongst these are sampling, path based methods combined with scenario reduction (Conejo et al., 2010), moment matching (Høyland et al., 2003) and copula based methods (Kaut, 2014).

The method chosen in this thesis is based on moment matching, and scenarios are generated in a C, C++ and Fortran IDE with an algorithm for moment-matching provided by Kaut (2003). Moment matching is conducted as one of several steps in a method that combines forecasting and scenario generation as presented in Schütz et al. (2009). By first fitting time-series models to historical prices and thereafter generating scenarios based on the residuals, the correlation between the variables is captured in two ways. The time-series will capture the characteristics of the price processes which implicitly results in capturing the correlation between them. The correlation that is not captured, is included in the correlation between the residual that is input to the moment matching algorithm. The method presented in the articles is adjusted to the specific problem analyzed. An overview of the scenario generation method is given below, and is also presented in the flowchart in Figure 7.4. The flowchart also indicates which programs are used in the different steps of the procedure.

1. Parametrize autoregressive models on historical data.
2. Find the distribution of the historical prediction errors. Error terms are assumed to be identically distributed. The error terms together form a multivariate discrete distribution for all the error terms.
3. To reduce the dimension of the multivariate distribution, perform a principal component analysis (PCA).
4. Denote all principal components P and the principal components explaining the highest part of the variance K. K principal components are chosen as stochastic in the scenario generation process, while $P - K$ components are explained by their mean value. Calculate the first four moments of the empirical distribution of the K stochastic principal components.
5. Generate S scenarios for the K stochastic principal components (PCs) by using a moment matching procedure as presented in e.g Høyland et al. (2003).
6. Transform the scenarios for the PCs back to scenarios of prediction error.
7. Combine the forecasting method with the prediction errors in the scenario tree by using the parametrized autoregressive model with the realization of the error term.

7.2.1 Preprosessing of data

Electricity prices can be viewed as consisting of two separate components; one deterministic and one stochastic. The deterministic component origins from deterministic hourly, daily and seasonal variations, while the stochastic component represents random errors

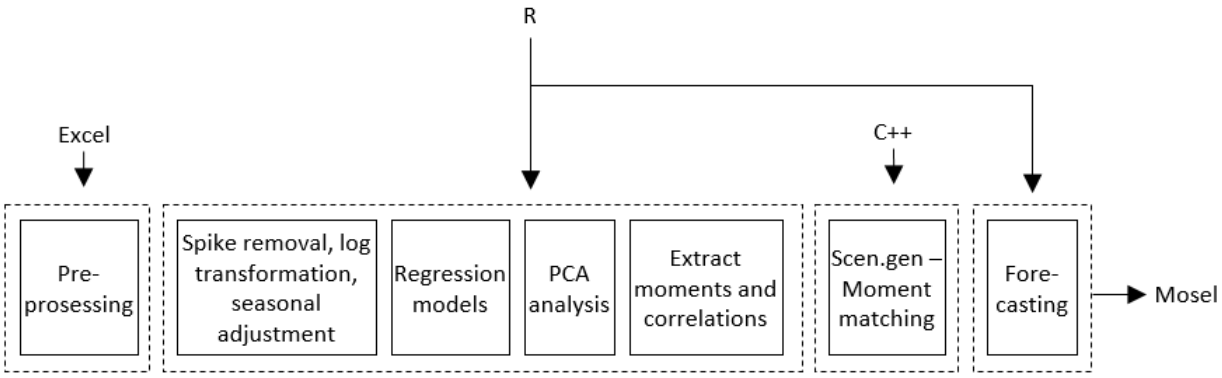


Figure 7.4: Flowchart of scenario generation procedure.

in the data. As in Janczura et al. (2013), we follow the "industry standard" where the day-ahead price Z_t is modeled as the sum of an independent deterministic component, P_t, and a stochastic component, S_t, as given by (7.1). The price models to be fitted to historical data in Step (1) in the scenario generation algorithm are used to model the stochastic component. Hence, the data is adjusted for the deterministic trend and seasonal component prior to model fitting.

$$Z_t = P_t + S_t \tag{7.1}$$

As seen in Section 7.1.1, a characteristic of the prices in the day-ahead market is the occurrence of spikes. Statistical models perform poorly under the presence of spikes and significant evidence has been found for better estimation of time-series components when the data has been treated carefully for outliers and spikes (Weron, 2014). Estimation routines for seasonal patterns are usually quite sensitive to extreme observations (Janczura et al., 2013). Based on their results, Janczura et al. (2013) recommend to filter and replace extreme observations prior to estimation of seasonal components and model fitting. This will provide more appropriate estimates for both the seasonal pattern and better parameters of the stochastic process. Hence, prices are seasonally adjusted before price spikes are filtered and replaced. Thereafter the seasonality that was removed is added such that the original data without spikes can be used as a starting-point for further analysis. Following Geman and Roncoroni (2006), seasonality can be represented by (7.2). For the time series analysed, the function consists of a constant, a linear trend and Fourier sinusoids with yearly period of 365 days for the day ahead prices. The function is estimated with the least squares method in R, and the parameters of the function are given in B. Inclusion of shorter seasonal periods has been tested, but it has shown to not improve the results of the seasonal adjustment. This matches the observation of the price throughout the year having one high-price period and one low-price period. The functional fit illustrated in figure 7.5 also shows this pattern.

$$f^s(t) = \alpha + \beta t + \sum_{i \in I} (\gamma_i sin(\frac{2\pi}{p}t) + \upsilon_i cos(\frac{2\pi}{p}t)) \tag{7.2}$$

There are several methods proposed in the literature to both detect and replace spikes. Weron (2014) mentions recursive filters, variable price thresholds, fixed price change thresholds and regime-switching classification as reasonable methods to detect spikes. Determining which price variations are caused by jumps and which are only variations in the underlying base signal is challenging. It can generally be said that price spikes are prices that surpass a certain threshold for a short period of time, but there is no commonly accepted definition and no common consensus of what the threshold or time interval should be (Weron, 2014; Janczura et al., 2013). In this thesis we define a price spike to be an increase in the absolute value greater than three standard deviations of the mean price. Spikes could be removed recursively, but this has by inspection shown to remove high prices not seen as spikes in the data, we therefore remove all spikes simultaneously. The threshold of three standard deviations from the mean value is a common threshold (Trück et al., 2007), and it works well on our data, removing all obvious spikes.

Spikes need to be replaced in order to avoid the missing value problem when identifying time-series models. This can be done by for example replacing the spike with a chosen threshold, with the mean value of the price before and after the spike or with one of the prices before or after the price spike or with similar day values (Weron, 2014). In this thesis, both positive and negative spikes are replaced with the price for the hour before the spike occurred. Figure 7.5 shows the spike-removal process for hour 8, from 07:00-08:00.

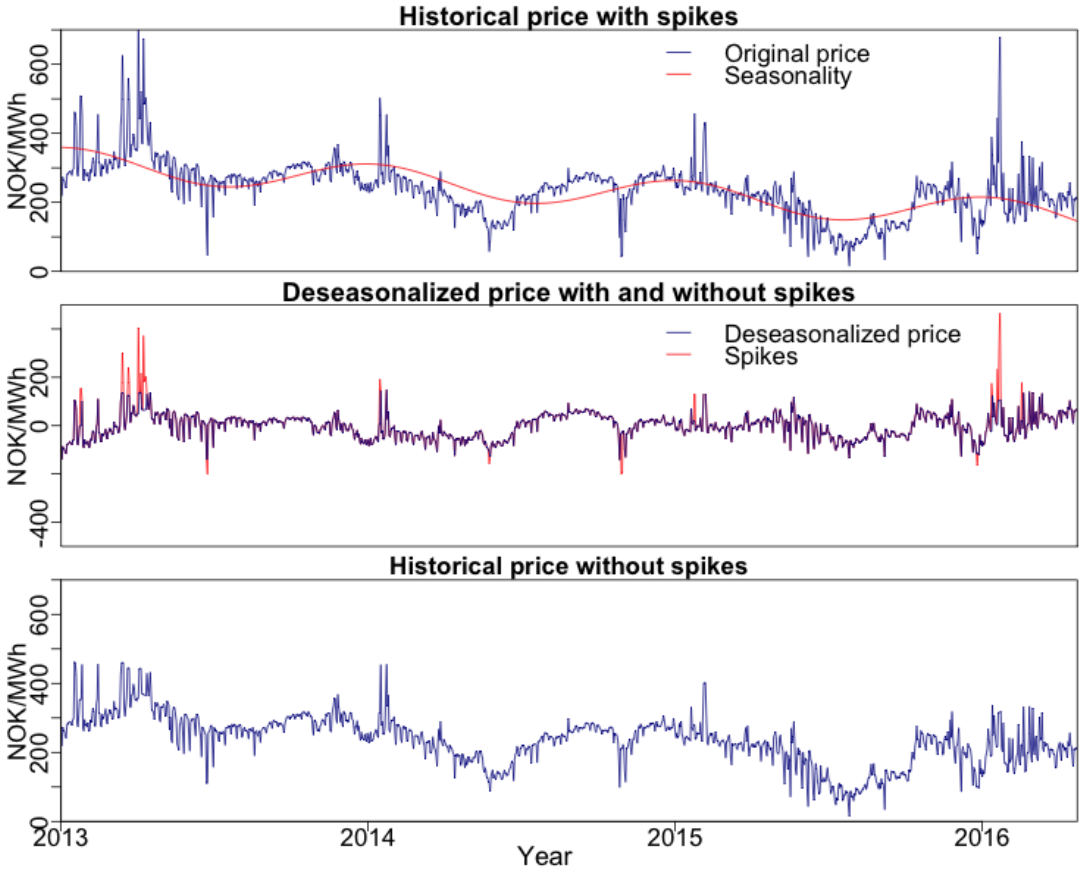


Figure 7.5: Example of spike removal for the hour between 07:00-08:00.

As part of the preprocessing, we need to perform a variance-stabilizing transformation. Such a transformation requires non-negative values, but the seasonal adjustment may create some negative values. Hence, variance stabilizing transformations are conducted first. There exist several variance stabilizing transformation methods, but two of the most used ones are the Box-Cox transformation and logarithmic transformation, in which the logarithmic transformation is a special case of the Box-Cox transformation (Wei, 2006). The logarithmic transformation contributes to normalize the data and stabilize the variance to the extent necessary to achieve a stable variance, and is therefore applied.

After the variance stability transformation, the transformed day-ahead data is seasonally adjusted with the same approach as given by Equation (7.2). The seasonal pattern for the primary reserve prices are not prominent, hence primary reserves are not seasonally adjusted. The stationarity of the tests was tested with the Augmented Dickey Fuller test, and the results show that the hypothesis of the time series being stationary after the transformations cannot be rejected. Thus the tests indicates stationarity.

7.2.2 Identification of time-series models

Following the variance stabilizing transformations and removal of deterministic price components, price models can be fitted to the historical prices. In order to find models that represent the historical data, previous work within the field of price modeling of electricity prices can be used to establish preliminary hypothesis of possible models.

The majority of traded electricity is traded in the day-ahead market, and there has been extensive research on modeling prices within this market. An overview of models can be found in Weron (2014). The paper classifies the modeling approaches in five main groups; multi-agent, fundamental, reduced-form, statistical and computational intelligence techniques. In this thesis, statistical methods based on a mathematical combination of the previous prices and/or previous or current values of exogenous factors are seen as most applicable. Well known statistical models are ARMA and GARCH. ARMA models are used to model time series with constant variance, while GARCH models are used in the presence of heteroskedasticity. There are however examples where GARCH models have performed poorly when applied to electricity price forecasting (Weron, 2014). The quality of the methods generally depend on their ability to incorporate factors characterizing the data. Contrary to day-ahead price modeling, there has been little research on modeling primary reserve prices.

Analysis of the historical prices for the current case is also a necessity in the process of identifying good candidate models. Looking at the acf and pacf plots of the prices facilitates the analysis. Figure 7.6 illustrates the acf and the pacf of the day-ahead price for the first hour and the primary reserve prices for the first block respectively.

Taking this into account, together with the historical use of ARMA models for modeling electricity prices, ARMA models have been chosen to model all hours of the day-ahead market. The ARMA model is given by (7.3). The residual, ε_t, in the model is i.i.d white noise with 0 mean, $E(\varepsilon_t) = 0$, and has finite variance, $E(\varepsilon_t^2) = \sigma^2$. The model consists of both autoregressive and moving average terms.

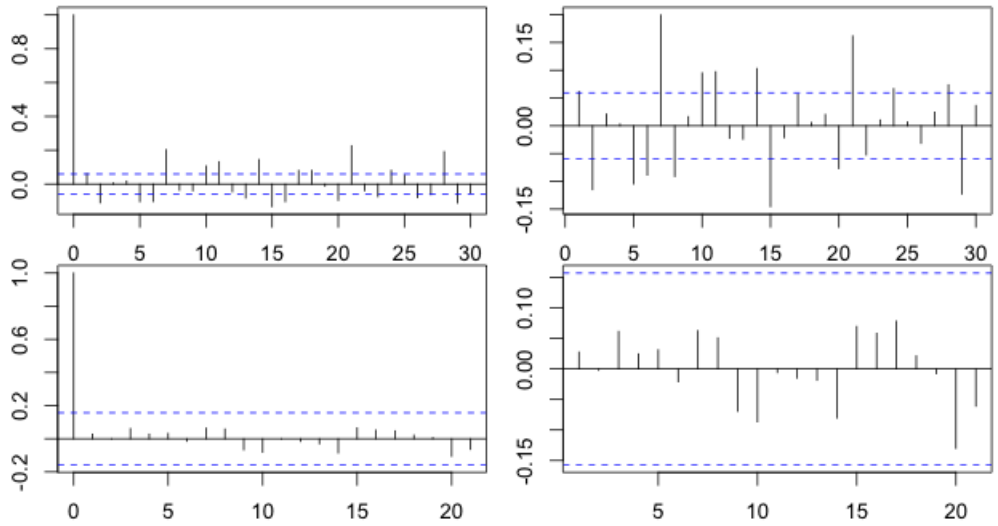


Figure 7.6: Acf (left) and pacf (right) of day ahead prices from 07:00-08:00 (upper) and primary reserve prices for the week-night block (lower) after transformation and removal of trend and seasonality.

$$y_t = c + \sum_{i \in I} \phi i y_{t-i} + \sum_{i \in I} \Theta_i \varepsilon_{t-i} + \varepsilon_t \tag{7.3}$$

Blockwise primary reserve prices are modeled with ARMAX models, with the expected day-ahead price as an exogenous parameter. This is inspired by the method presented in Boomsma et al. (2014) and it contributes to preserve the cross correlations between the primary reserve market and the expectation of prices in day ahead market. The ARMAX model is given by (7.4). The formula includes all of the same terms as the ARMA model, but is expressed with different terms than the ARMA model above to distinguish the models. The ARMAX models also includes an extra term for the exogenous parameter, z_t.

$$x_t = d + \sum_{i \in I} \gamma_i x_{t-i} + \sum_{i \in I} \Gamma_i u_{t-i} + \sum_{i \in I} \alpha_i z_{t-i} + u_t \tag{7.4}$$

All models have been fitted using auto.arima in R which uses unit root tests, minimization of the AICc and MLE to obtain the best-fit model (Hyndman and Athanasopoulos, 2014). Autofit is seen to give many terms for complicated processes, and the maximum number of AR and MA terms are therefore set to 3 for the residuals in both the primary reserve market and the day-ahead market since this gives small errors without overfitting the model. The parameters and test-statistics of the model fit are given in Appendix A.1. The quality of the model is estimated by the log-likelihood values and t-values. Estimation of t-values and p-values indicate that the significance of the parameters are high, rejecting the null-hypothesis that the coefficient has no effect.

7.2.3 Principal component analysis

Principal components analysis (PCA) is a technique used to reduce the dimensions of data sets by transforming correlated variables into a smaller set of uncorrelated variables, while retaining a large share of the variance (Alexander, 2008). The assumption behind the method is that the dimension can be reduced when the variables depend on a smaller set of common factors, denoted principal components (PCs). It is beneficial to conduct a PCA before performing moment matching because the moment matching algorithm by Kaut (2003) is slower when the set of random variables increases (Høyland et al., 2003).

The matrix of uncertain variables, $\xi \in \mathbb{R}^{T \times n}$ (n time series of length T), can be expressed by the vector of PCs, $\tilde{\xi} \in \mathbb{R}^{T \times n}$, of the covariance matrix, Σ, multiplied with the inverse of the orthogonal matrix, denoted $X \in \mathbb{R}^{n \times n}$. The eigenvectors of Σ are columns in the orthogonal matrix (Alexander, 2008). This relationship is expressed in (7.5).

$$\xi = \tilde{\xi} X \tag{7.5}$$

The variance of the PCs is given by the eigenvalues v_i of Σ. By including the PCs that describe most of the variance, the essential information is kept while the dimension of the data is reduced. Keeping only the PCs that explain the highest share of variance $\tilde{\xi}^* \in \mathbb{R}^{T \times \tilde{n}}$, the original uncertain variables ξ can be approximated by (7.6), in which $X^* \in \mathbb{R}^{\tilde{n} \times n}$. Note that X is ordered after the size of the eigenvalues of Σ, with the largest eigenvalue first.

$$\xi \approx \tilde{\xi}^* X^* \tag{7.6}$$

The principal component analysis is conducted separately on the historical residuals in the panel of 24 hourly prices in the day-ahead market and the panel of 6 blocks in the primary reserve market. As can be seen from the results in Figure 7.7, the PCA reduces the dimension from 24 to 10 dimensions for the day-ahead prices and from 6 to 3 dimensions for the primary reserve prices while preserving above 95 % of the variance. Such a high cut-off has been chosen because the time for the moment matching algorithm to converge when including 13 stochastic variables is not considered as a barrier. Note that the Pcs explaining most of the variance, $\tilde{p}^*$ are used as input in the moment matching algorithm. Thereafter the resulting scenarios for the PCs are converted to original variables using (7.6).

7.2.4 Moment matching

Moment matching is a scenario generation procedure, whose output is a discrete joint distribution. It performs mathematical transformations on the moments of the marginal distributions and their correlation matrix in order to generate scenarios. The method can

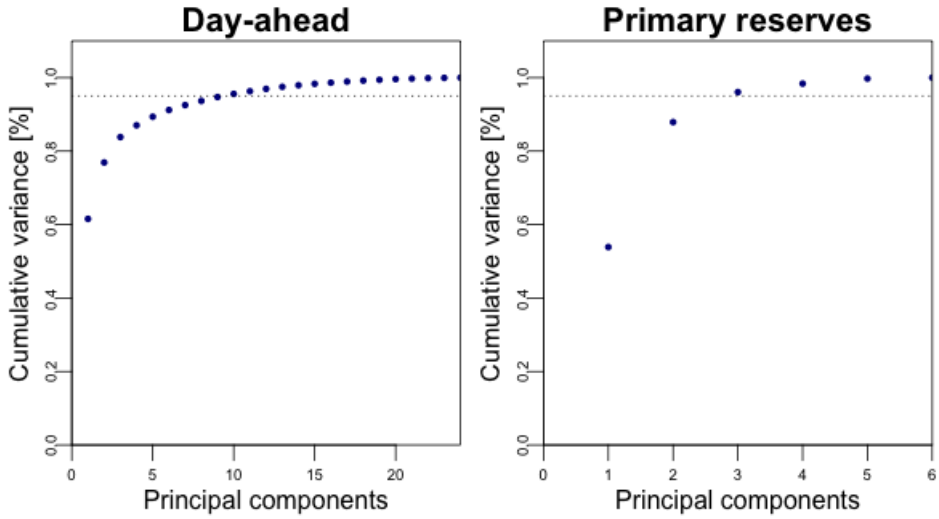


Figure 7.7: Cumulative variance of principal components.

be adjusted to include any statistical property of the marginals, but Høyland and Wallace (2001) refer to own empirical analysis and other studies showing that the important statistical properties of the marginals are the mean, variance, skewness and kurtosis. A great advantage of the method is that there is no need to assume a specific distribution of the marginals to facilitate the scenario generation process.

The moment matching procedure, as described by Høyland et al. (2003), is summarized in Figure 7.8. The procedure is divided into two phases. In the first phase the moments of the original marginal distributions are transformed such that independent univariate outcomes can be generated. In the second phase the whole process is inverted so that the generated outcomes get the desired statistical properties/moments.

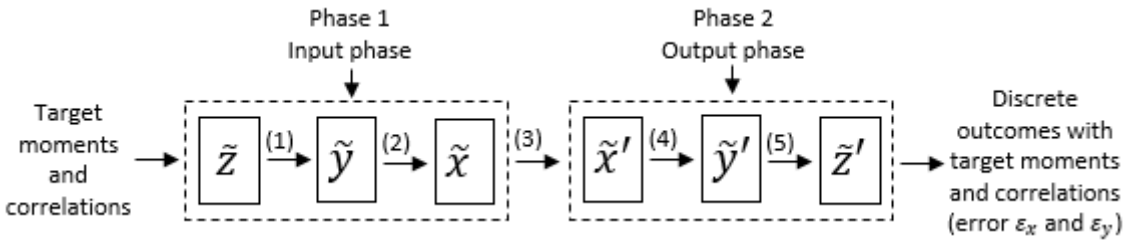


Figure 7.8: Moment matching procedure.

1. Using (7.7), the original random variable $\tilde{\mathcal{Z}} \in \mathbb{R}^n$ is transformed to the random variable $\tilde{\mathcal{Y}} \in \mathbb{R}^n$ such that $\tilde{\mathcal{Y}}$ has $\mu = 0$ and $\sigma^2 = 1$ and the skewness and kurtosis has the same value as $\tilde{\mathcal{Z}}$. In (7.7) α is the square root of the standard deviation of $\tilde{\mathcal{Z}}$ and β is the mean value of $\tilde{\mathcal{Z}}$. This transformation is conducted since the matrix transformation in step 2 needs random variables with $\mu = 0$ and $\sigma^2 = 1$ as input.

2. The matrix transformation given by (7.8) transforms the correlated variables in $\tilde{\mathcal{Y}}$ to independent univariate outcomes $\tilde{\mathcal{X}}$. The matrix transformation uses the lower-triangular matrix $L \in \mathbb{R}^{n \times n}$ from the Cholesky decomposition of the correlation matrix $R \in \mathbb{R}^{n \times n}$, given by $LL^T = R$.

3. Values from $\mathcal{N}(0,1)$ are sampled and used in the cubic transformation given by (7.9) to obtain discrete outcomes $\tilde{\mathcal{X}}'$ with target moments of the random variable $\tilde{\mathcal{X}}$.

4 Using (7.7) the discrete outcome $\tilde{\mathcal{Y}}'$ of $\tilde{\mathcal{Y}}$ are obtained.

5 Using (7.8) the discrete outcome $\tilde{\mathcal{Z}}'$ of $\tilde{\mathcal{Z}}$ with correct correlation are obtained.

$$\tilde{Z} = \alpha\tilde{Y} + \beta \tag{7.7}$$

$$\tilde{X} = L^{-1}\tilde{Y} \tag{7.8}$$

$$\tilde{X}_i = a + b\tilde{U}_i + c\tilde{U}_i^2 + d\tilde{U}_i^3 \tag{7.9}$$

The transformations in the second phase allow us to achieve the target moments and correlations separately. However, the cubic transformation affects the marginal moments and the matrix transformation affects the correlation. Thus, the algorithm only provides approximate results, except from when the random variable $\tilde{X}$ is independent. Note that the original random variables that are input to the moment matching algorithm are uncorrelated, but not independent. The algorithm implemented by Kaut (2003) is therefore iterative in step 3 and 4 with the errors ϵ_x and ϵ_y measuring the distance from the target marginals and target correlation respectively. These errors are calculated as a root-mean-squared-error, and the algorithm is set to iterate until the errors are within a prespecified range.

In addition to the moments and correlation matrix of $\tilde{Z}$, the number of scenarios and their probabilities are needed as input to the moment matching procedure. The probabilities of the scenarios are chosen to be equiprobable such that $p_s = 1/|S| \quad \forall s \in S$ where $|S|$ is the number of scenarios. In addition, the maximum errors ϵ_x and ϵ_y have to be prespecified. We have experienced that the algorithm converges for 55 scenarios and more with a maximum error of ϵ_x and ϵ_y set equal to 0.1. For less than 55 scenarios, the errors have to be increased since the algorithm since it is more challenging to find the correct distribution when fewer scenarios are included. Note however that the convergence of the algorithm given a presoecified maximum correlation and moment errors is not only determined by the number of random variables, but also the structure of their moments (Høyland et al., 2003).

Part of the criticism towards using moment matching is that the method does not capture asymmetric dependence, a situation in which the dependence in down-turns differs from the up-turns. This is often the case in a power system with a large share of hydropower, resulting in electricity prices being higly dependent on rainfall. However, in the short-term model implemented, dependency between electricity prices and rainfall/inflow is not taken into account. Correlation between electricity prices within and between the day-ahead and primary reserve market can furthermore be assumed to not be asymmetrically correlated since they depend on the same factors. Thus, including asymmetric dependence in the scenario generation procedure would not increase the accuracy of the generated scenarios. This is an argument towards using moment matching instead of e.g copula based scenario generation which would require us to make more assumptions about the parameters of the method.

While the moments of the resulting distribution from the scenario generation match the moments of the historical principal components, the maximum value of the resulting distribution does not correspond to the historical maximum value. The kurtosis of the distributions of the residuals are quite high, meaning that the distributions have fat tails. There is only a small probability for the values at the tails to occur, but these scenarios can still be generated. For the primary reserve price this results in scenarios with much higher prices than the prices observed. Scenarios that deviate too much from the historical maximum or minimum value are not desirable and are therefore deleted. After deletion the moments of the resulting distribution is again compared to the moments of the historical principal components since deleting scenarios from the distribution will affect the moments. There is consequently a trade-off between removing outliers and retaining moments of the historical distribution. The resulting prices for the day-ahead market does not exceed the historical prices observed in the market to such an extent that it is seen as beneficial to remove them.

The moment matching algorithm is implemented such that it generates outcomes for one stage. Primary reserve prices for each block are the same for the whole week, and scenarios from the scenario generation can therefore be used directly. However, day-ahead price scenarios for seven consecutive days need to be generated. Høyland and Wallace (2001) state that expanding the scenario generation algorithm from one to several periods complicates the scenario generation procedure because also intertemporal dependencies between stages need to be considered. The intertemporal dependencies between the stages in the scenario generation procedure in this thesis is captured by the time-series models. The residual distribution is the same for all days throughout the week. Based on this, the multistage scenario tree is generated by random sampling scenarios without replacement for all days during the week.

As previously discussed, the presence of spikes is a central characteristic of prices in the day-ahead market. Including spikes in the final scenario tree is important in order to value the opportunities to bid in the day-ahead and primary reserve market correctly. Spikes are added to the scenario tree by random sampling spikes from historical spikes during a period of +/- 15 days around the same time-period that is analysed. Spikes are sampled conditionally on spikes the hour before because spikes often are seen to occur for consecutive hours.

7.2.5 Resulting scenario tree

The resulting day-ahead scenarios and primary reserve scenarios are illustrated in Figure 7.9. The Day-ahead and primary reserve scenarios are seen to follow the true price process closely, capturing changing prices between hours in the day-ahead market and between blocks in the primary reserve market.

For the DA prices, it can be observed that the true price lies within the 1st and 3rd quartile of the generated scenarios the first days before increasing to a generally higher level just above the 3rd quartile. The scenarios capture the price movement during a day well; there are lower prices during the night and higher prices during the morning and late afternoon/evening. It can also be seen that the scenarios manage to capture

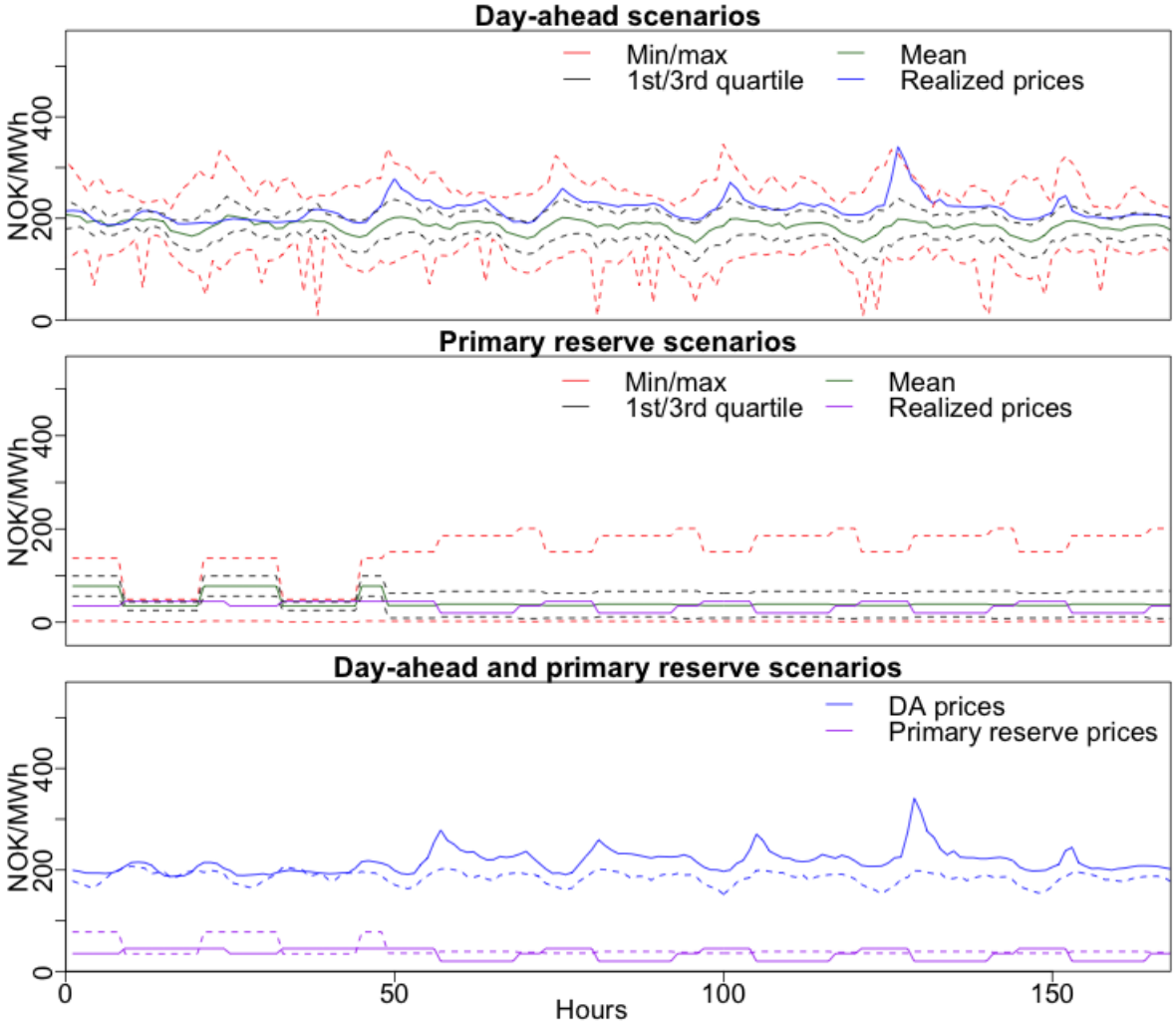


Figure 7.9: Resulting scenarios for day-ahead and primary reserve prices for 2016-04-23 to 2016-04-29. Illustration based on scenariotree consisting of 100 scenarios.

the possibilities of higher prices during the first hours of the day, such as the high price period during the penultimate day. There are historically only negative spikes in the weeks around the time period analysed. Consequently only negative spikes are sampled and added to the scenarios in the scenario tree. The minimum value of the scenario tree therefore vary more than the maximum value.

The realized price of the primary reserves is also seen to lie within the 1st and 3rd quartile. The scenarios capture the higher price level during the night and the lower price level during the day. The last figure represents the relationship between the price scenarios in the day-ahead market and the primary reserve market. Primary reserve prices are, as previously discussed, modelled as dependent on the expected price in the day-ahead market. The negative correlation between these prices are seen to be captured with prices in the primary reserves being high when prices in the day-ahead market are low.

7.2.6 Stability of the scenario tree

Using the same notation as earlier in the thesis, the stochastic random variables are denoted by ξ and the discrete outcome of the stochastic variables are given by $\breve{\xi}$. The error of using the discrete random variables compared to the true distribution, is for a maximization problem expressed by (7.10) (Kaut and Wallace, 2007). The quality of the scenario generation method can be evaluated by investigating the size of this error and the stability of the solutions to the optimization problem (Kaut and Wallace, 2007).

$$e_f(\tilde{\xi}, \breve{\xi}) = F(\text{argmax}F(x, \tilde{\xi}); \tilde{\xi}) - F(\text{argmax}F(x, \breve{\xi}); \breve{\xi}) \tag{7.10}$$

There are two types of stabilities that should be evaluated; out-of sample and in-sample stability. In-sample stability means that the solution to the optimization problem solved with K different scenario trees generated with the same input data, should give the same objective value. This is expressed through (7.11). Out-of sample stability means that the solution of the approximated problem is close to the true optimal solution as expressed by (7.12). Thus, the out-of sample test is the evaluation of the objective function value for a given decision. Having an out-of sample stable solution means that the real performance of the solution does not depend on the scenario tree that is being used.

$$F(x_k^*, \breve{\xi}_k) \approx F(x_l^*, \breve{\xi}_l) \qquad k, l \in K \tag{7.11}$$

$$F(x_k^*, \tilde{\xi}) \approx F(x_l^*, \tilde{\xi}) \qquad k, l \in K \tag{7.12}$$

For a random scenario-generation process, such as moment matching, several trees can be generated with the scenario generation method. In-sample stability is tested for 40-100 scenarios in intervals of 15 scenarios. These tests are run for 10 hours each. The stability of the tree is measured by the coefficient of variance (CV), which indicates the dispersion of the objective function value around the mean value. It is calculated by dividing the standard deviation by the mean. Figure 7.10 illustrates the mean value and the CV with respect to the number of scenarios. The mean value can be seen to be varying around the same value, but with little consistency in the variations given the number of scenarios. The CV however show less variation in the objective function value when solving the problem with 70-100 scenarios. Based on the stabilization of the CV, we conclude that the scenario tree can be considered as stable for 70 scenarios.

In the out of sample tests, the k first stage solutions from the in-sample tests are used as input. Out-of sample stability should be tested with the true distribution, but this is considered very difficult. Approximation tests can be conducted instead by building reference trees (Kaut and Wallace, 2007). The reference tree in this thesis is created by merging 10 smaller trees with 100 scenarios each, resulting in a scenario tree with 1000 scenarios. Note that the problems solved in the out-of sample tests are not stochastic problems since the first stage solutions are fixed. Consequently, the solution of the optimization problem with solved with the reference tree with 1000 scenarios can be solved by solving 10 smaller problems with 100 scenarios each, thereafter finding the expected value

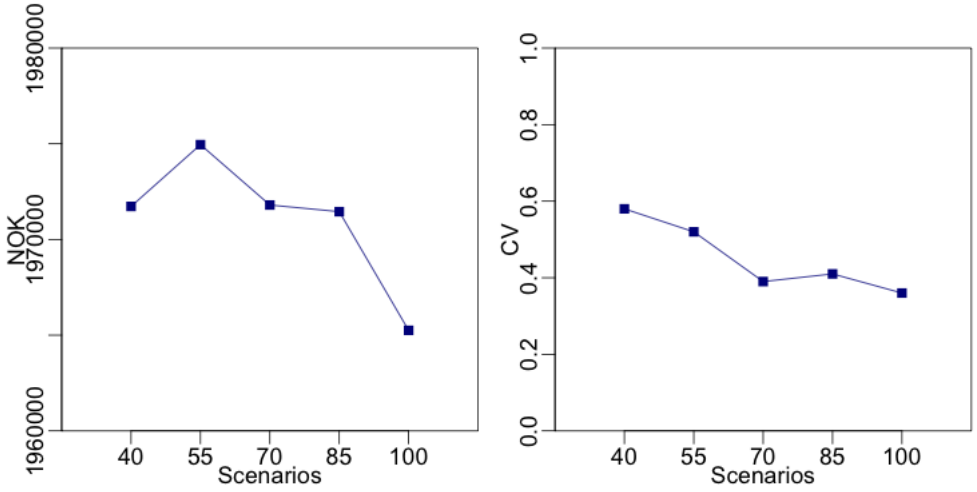


Figure 7.10: In-sample stability.

of the solutions. Similarly as for the in-sample stability testing, the results are given by the CV. The out-of sample stability is only tested for the scenario tree with 70 scenarios that are considered to be stable. The results of the out-of sample tests results in mean value of that the out-of-sample stability of the scenario tree has a CV of 0.1 % with a mean value of 1 970 895. These results show that the scenario tree can be considered as out-of-sample stable.

7.2.7 Implementation considerations

The scenario generation procedure implemented results in realistic price scenarios that manage to capture the price characteristics in both markets. This include the mean-reverting, seasonal and spiky behaviour of day-ahead prices and the correlation between the day-ahead and primary reserve prices. Modeling day-ahead and primary reserve prices as a panel of prices, facilitate capturing the characteristic of the separate hourly day-ahead prices and the blockwise prices. It is clear that the daily price pattern in the day-ahead market is especially well capture with prices being higher in the morning and afternoon. The historical prices for the week modeled is seen to fall within the max-min scenarios in both the day-ahead market and the primary reserve market.

For the day-ahead market it is however seen that the price falls outside the 1st to 3rd quartile for hours during the two last days. Hence, more complex price models that capture the inherent spiky behaviour of the day-ahead price process can be tested to see if they perform better. Including spikes by random sampling as done in thesis, is however considered as sufficient for the purpose of this thesis.

The ARMAX model used for modeling primary reserve prices capture the correlation between the expected day-ahead prices and the primary reserve prices through the exogenous parameter. Including this price response is considered valuable for the scenario generation procedure implemented. Modeling the primary reserve prices with an autoregressive mean reverting process, similarly to the day-ahead prices, can be seen to result

in realistic scenarios. However, considering the small quantities traded and the demand being set by the TSO, other price models can be considered tested in order to capture the large number of zero value prices that can be seen in the historical reserve price data. Integrating other price processes in the scenario generation procedure can successfully be done since the procedure uses the residuals from the price models fitted to historical prices.

Random sampling of the residuals from the moment matching procedure for the day-ahead prices results in a weekly forecast that can be seen to retain the uncertainty in market prices further away from operating time. This is considered as a better approach than only including uncertainty in the forecast for the first day, thereafter treating the residual as zero when forecasting for the consecutive days during the week, which is another possible approach. Due to the mean reverting behaviour of the ARMA model this approach would have resulted in a forecast with less variance in prices further from operating time, which is not seen as a realistic assumption.

The resulting scenario tree shows a highly stable behaviour, both in-sample and out-of-sample. This indicate that the implemented scenario generation procedure function well and that the problem can be solved with different scenario trees. However, as will be discussed in Chapter 9, the optimal decisions of the problem, for which the stability has been tested, strongly depend on the level of prices compared to the marginal water values. Hence, the stability of the tree can be assumed to be lower when solving the problem for other test instances.

Chapter 8

Case description

Two watercourses located in the NO2 price area and operated by Norsk Hydro ASA are modeled in the thesis. Both watercourses are part of larger watercourses with other producers owning and operating power stations upstream and downstream of the modeled part of it. However, in the model, the included power stations are operated independently from the operation of power stations outside the analysis. This chapter begins with a presentation of the two watercourses and the parameters associated with the operations of the associated power stations. All topology and operational data are received from Hydro's operational optimization model and thereafter processed in Excel and R. The data cover the same time-periods as the price forecast generated, 2016-04-23 until 2016-04-29.

8.1 Watercourses

The first watercourse, referred to as watercourse 1, consists of four reservoirs and one small intake reservoir as illustrated to the left in Figure 8.1. Upstream of the first power station there are three reservoirs, which in the model is aggregated to one reservoir. This aggregation will not result in large differences in the way the power station are operated, since the upper reservoirs store an increasing amount of water if there is risk of spillage Eriksrud and Braathen (2014). The three upper reservoirs, both independently and aggregated, are considerably larger than the lower reservoir in the model. This can also be seen from Table 8.1. There is furthermore a small intake reservoir upstream of the lower reservoir, but this reservoir is not included in the model.

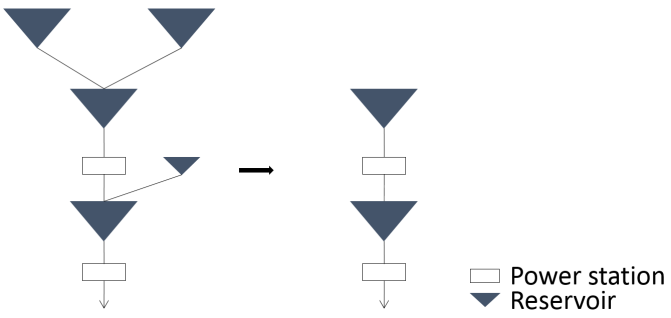


Figure 8.1: Watercourse 1.

Watercourse 2 is illustrated in Figure 8.2. It consists of five reservoirs in cascade, all connected to a power station. The upper reservoir is considerably larger than the other four reservoirs which function only as intake reservoirs that do not store water over a longer time period.

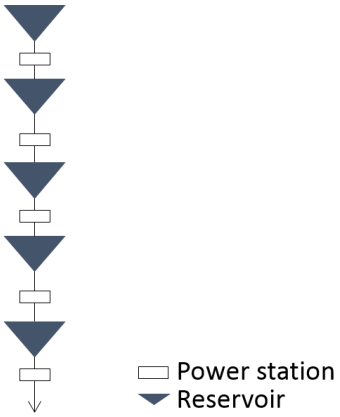


Figure 8.2: Watercourse 2.

For both watercourse 1 and 2, 63 water value cuts are included. The reservoirs within each watercourse are interdependent, but the two watercourses are not. Hence, the watervalue cuts are separate for each watercourse. For watercourse 1, the water value for the aggregated upper reservoir is found as the volume weighted average of the water values for the three aggregated, individual reservoirs. The time-period under analysis is characterized by high inflows due to snow melting and the marginal water values are correspondingly low.

There are seven power stations in the two watercourses altogether. The number and characteristics of the generators in the power stations vary. This can be seen in Table 8.1. Both the minimum and maximum production level vary between the generators, with the smallest production level being as low as 2.3 MW and the maximum production level being as high as 102 MW. These limits and the maximum discharge through the generators are amongst the determinants for the optimal operation of the power stations. Another important determinant is the relationship between the power output and discharge of the power stations.

Table 8.1: Technical overview of watercourse 1 and 2.

Watercourse	Res.	V $[Mm^3]$	Gen.	Pmin $[MW]$	Pmax $[MW]$	Qmax $[m^3/h]$
1	1	178	1	14.1	41.9	57600
	2	1.6	2	29.2	76.0	55 887
			3	29.2	78.0	5 760
2	3	1064	4	2.3	30.4	324 000
	4	1.5	5	4.7	102	1.37961
			6	4.7	102	141 481
	5	0.6	7	14.3	63.8	95 346
			8	15.1	59.4	95 346
			9	13.3	59.4	99 648
	6	0.6	10	4.3	26.5	277 200
	7	0.6	11	10.1	30.3	374 400

The relationship between the power output of a power station and discharge through the turbines are given by production discharge curves (P-Q curves). There are different P-Q curves associated with the possible combinations of producing generators in the power station. We have however chosen to model each generator independently, hence only the combinations where each generator runs alone are included. These curves are normally concave, but the optimization model implemented requires linear curves. The P-Q curves are therefore approximated by piecewise linear curves. It is clear that the best linearization is the one with the highest accuracy and the fewest lines. We have chosen to model the P-Q curves by three lines since this results in production curves with a good enough quality. Including more lines would result in more constraint in the optimization problem and correspondingly higher solution times, which is not desirable given the accuracy needed. The set of curves are chosen such that one line intersects the best point, the point of highest efficiency, and one line intersects the point of maximum production. Two lines have been chosen to approximate this part of the curve. The last line intersect the point of zero discharge and the best point. This results in a break point at the best point of the P-Q curve. This is desirable since the optimal production point is near the best point, and the solution to a linear optimization problem lies at a corner point. An example of the linearization is illustrated in Figure 8.3. Approximations of all the P-Q curves can be found in Appendix C. The linearizations approximate the curves well, except for the part of the curve close to the minimum discharge point. A better approximation of this part of the curve could be included, but the optimization model then needs to be adjusted to account for these changes. These adjustment includes introducing binary variables associated with the unit commitment decision in the P-Q constraint. Initial tests run with these adjustments show that the solution-time increases considerably. The increase in solution time is considered as high compared to the added value of better approximations of the P-Q curves. Hence, the model is implemented with P-Q curves that go through intersect the point of zero.

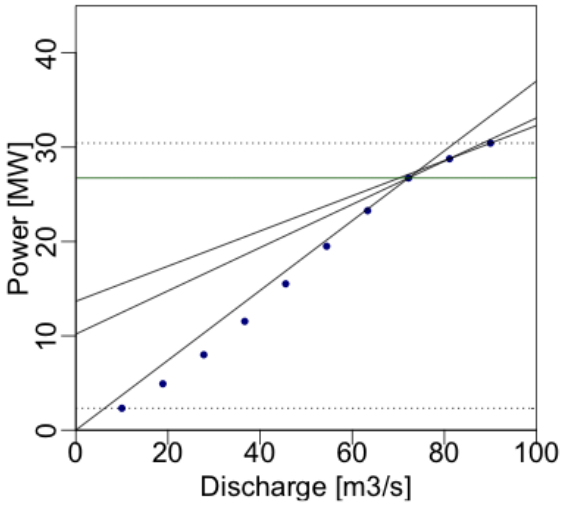


Figure 8.3: Example of piece-wise linear approximation of P-Q curves for generator 4.

Inflow is modeled as deterministic in the model. The expected inflow for the watercourses is illustrated in Figure 8.4. The expected inflow for reservoir 2, 3 and 4 in watercourse 2 is zero and are not included in the figure. In watercourse 1, the expected inflow in the two reservoirs is approximately at the same level and highly fluctuating in the beginning of the week. The inflow in reservoir 3 and 7 in watercourse 2 is very high for the whole period, with the inflow in reservoir 3 being more variable than in reservoir 7. The generally high inflows are due to time of the year as previously mentioned.

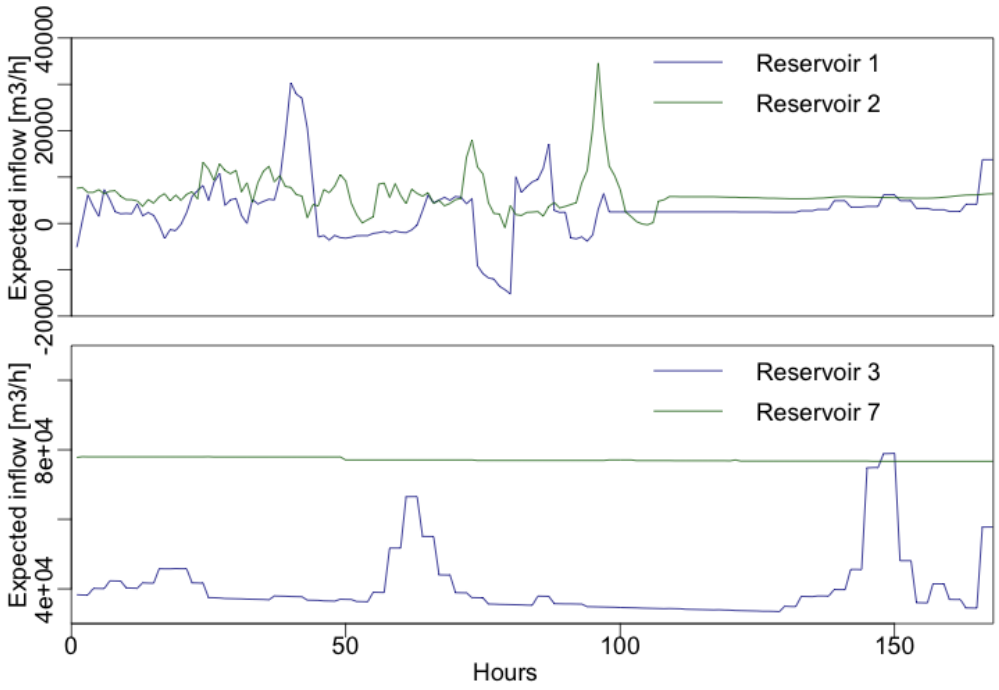


Figure 8.4: Inflow in reservoirs 1, 2, 3 and 7. Inflow in all other reservoirs is zero.

Chapter 9

Computational study of the deterministic equivalent

This chapter presents the main findings from tests performed on the deterministic equivalent of the model. To get an idea of the complexity of the model, the chapter begins with a presentation of the size of the SMIP model. We will then conduct a sensitivity analysis with respect to input parameters in order to investigate how different input parameters of the model affect the complexity and solution time. The sensitivity analysis is then used as a starting point for finding two distinct realistic test instances to be tested with decomposition methods.

The optimization model has been implemented in the Mosel modeling language and solved with Xpress-MP v7.8.0. When solving a problem with binary variables, Xpress first finds the solution of the LP relaxed problem before using root cuts and heuristics and thereafter branch and bound to find the best integer solution. Large-scale SMIP models are difficult to solve to optimality within reasonable time. Feasible solutions found before optimality is reached are therefore stated with the corresponding duality gap when the time limit decided is reached. The duality gaps stated in this and the following chapters are calculated by (9.1) in which LB is the primal feasible lower bound and UB is the upper bound.

$$\text{Duality gap} = \frac{UB - LB}{LB} \cdot 100\% \tag{9.1}$$

All tests have been run on a 64-bit Windows 7 PC with 3.40 GHz Intel Core i7-3770 CPUs (4 cores, 8 threads) and 16 GBRAM.

9.1 Problem size

As can be seen from Table 9.1, the implemented problem is characterized by many constraints and variables, both continuous and binary, and the size of the model grows with the number of scenarios. All restrictions except the bidding curve are given for each

scenario, hence increasing the number of scenarios by a given percentage results in the same percentage increase in the number of variables and approximately same percentage increase in the number of restrictions. The size of the problem is also related to the size of the watercourse. Watercourse 2 is more than twice the size of watercourse 1 with respect to both the number of constraints and variables.

For large problems, there is often some sort of redundancy among the variables and constraints. Thus, as a first step of the solution process, Xpress tries to reduce the problem size by identifying and removing redundant rows and columns by performing a set of presolve procedures on the model. This collection of procedures will often contribute to greatly improve the performance of the optimizer by modifying the problem matrix, resulting in a reduction in both the number of constraints and variables. The results in Table 9.1 show that the presolve procedure reduces the number of variables and constraints, but the percentage decrease is the same regardless of the number of scenarios included. This means that the presolve procedure does not help counteract the increase in problem size associated with including more scenarios in the stochastic optimization problem.

Table 9.1: Problem size of deterministic equivalent before and after presolve for different number of scenarios (S) and watercourses (W).

		Before presolve			**After presolve**			
S	W	Constraints [#]	Variables [#]	Bin. var. [#]	Constraints [#]	Δ Const. [%]	Variables [#]	Δ Var. [%]
40	1	146 798	131 337	20 160	137 865	-6.1	115 852	-11.7
	2	363 014	334 449	53 760	349 565	-3.7	320 922	-4.0
	1&2	496 511	458 976	73 920	480 722	-3.2	480 722	-4.9
70	1	259 673	232 347	35 280	243 869	-6.1	204 951	-11.7
	2	642 209	591 699	94 080	618 410	-3.7	567 762	-4.0
	1&2	878 396	812 016	129 360	850 454	-3.2	772 344	-4.9
100	1	372 575	333 357	50 400	349 892	-6.1	294 042	-11.7
	2	921 431	848 949	134 400	887 267	-3.7	814 587	-4.0
	1&2	1 260 308	1 165 056	184 800	1 220 192	-3.2	1 108 110	-4.9

The solution time is highly related to problem size and complexity. As an illustration of the relationship between solution time and the size of the problem, a comparison of the solution times for watercourse 1, 2 and the two combined are given in the plot to the left in Figure 9.1. It can be seen that the test for the combined problem instance does not find a feasible solution, clearly indicating the large increase in problem complexity and solution time when including more units. The differences between watercourse 1 and 2 are however not prominent even though watercourse 2 increases the size of the model by above 50 % with respect to the number of variables and constraints. The small difference might however be due to other input parameters such as the technicalities of the power station and the marginal water values. In addition, the relationship between the number of scenarios and the solution time is investigated by testing the model with different number of scenarios. Figure 9.1 shows that the solution time increases with the number of scenarios included. This is especially present when the number of scenarios is high.

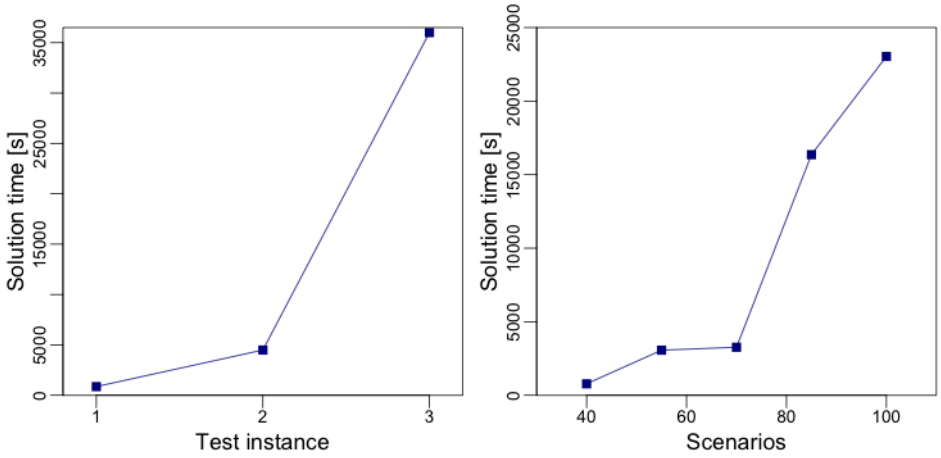


Figure 9.1: Left: Relationship between solution time and problem instance. Tests run with a scenario tree with 70 scenarios and maximum solution time set to 10 hours[1]. Right: Relationship between average solution time given number of scenarios for watercourse 2. Tests run on 5 different scenario trees within each interval.

9.2 Computational results and sensitivity analysis

In addition to the number of scenarios, the input data can greatly impact the complexity of the problem and hence the solution time. In this section we will therefore perform a sensitivity analysis with respect to various input parameters. As a basis for the sensitivity analysis, we use the original data from Hydro, and adjustments are made to these parameters to explore how the difficulty of the problem may be input-dependent. Through this we try to identify test instances that would be particularly interesting to apply decomposition methods on. The goal of the optimization model is to provide decision support to power producers when the bidding decision is characterizes as complex. When the bidding decision is seen as trivial, for example when the producer wishes to commit all generators at all times, the need for such models is however considered to be small. It is furthermore assumed that the decomposition methods to be tested are more useful when it is non-trivial for the producer to find the optimal bid in the coordinated bidding problem since decomposition methods often only become truly useful when the non-decomposed problem is too hard to solve with an optimizer such as Xpress.

Tests are performed on watercourse 1 and watercourse 2 separately. All tests are done using the same scenario tree consisting of 70 scenarios, which is considered a stable number of scenarios, such that the results and the solution times are easily comparable. The maximum solution time is set to 10 hours. If the first MIP solution found is the same as the last, it is indicated by writing the last MIP solution in *italics*.

The sensitivity analysis conducted considers thee factors; marginal water values, minimum production level and variability in inflow. The analysis provides indications on how difficult it is to solve the problem when prices are closer to the marginal water values, for generators that are less efficient and when inflow to the reservoirs differ. Technical dif-

[1]No feasible solution found within 10 hours for the combined test instance

ferences such as minimum production level exist between watercourses and hydro power stations. In addition variations in inflow and marginal water value throughout the year is a well known phenomenon. Looking at iso-price curves of the relationship between the time of the year and the water value in a typical reservoir, for example in Doorman (2015), it is clear that marginal water values differ throughout the year. Furthermore, looking at the relationship between inflow in a watercourse and time of the year, it is clear that the inflow is high during spring and low during winter. All changes to input parameters are adjusted such that the case still can be considered realistic.

Increasing the marginal water values means increasing the cost of water, giving the producer less incentive to use water today. The results from changing the marginal water value by a given percentage relative to the water value in the original water value cuts are provided in Table 9.2. Changes in marginal water value have been tested for every 10 % increase within the interval 10%-100%, but only the results from three tests resulting in the largest duality gap in the first node of the root cutting heuristic are presented.

Table 9.2: Sensitivity with respect to marginal water value. Marginal water value increased by fixed percentage.

		First integer solution				**Last integer solution**			
		Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]	Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]
1	**Orig.**	1 947	1 978	1.55	861	1 975	1 977	0.11	36 000
	80 %	707	780	10.26	12 767	733	779	6.28	36 000
	90 %	609	697	14.47	10 259	622	693	11.40	36 000
	100 %	605	662	9.53	919	639	655	2.49	36 000
2	**Orig.**	6 200	6 202	0.02	4 490	6 201	6 202	0.01	13 444
	70 %	3 053	3 228	5.72	31 371	*3 053*	*3 228*	*5.72*	36 000
	80 %	2 743	3 120	13.75	30 907	2 805	3 120	11.24	36 000
	90 %	2 811	3 148	11.98	23 015	2 818	3 148	11.69	36 000

A higher minimum production level means that the generating units have to produce more electricity when being on. Thus, when prices are low or water values are high, the producer will have a larger incentive to turn the generators off. The maximum production level for the generators in watercourse 1 is already characterized as high and a sensitivity analysis increasing these values would result in an unrealistic case. This is not desirable, hence, only the minimum production level for the generators in watercourse 2 are adjusted. The changes from the original minimum production levels for the tree test instances can be seen in Table 9.3. The results of using these minimum production levels are provided in Table 9.4.

Two different tests have been conducted to investigate the sensitivity with respect to inflow. In the first test, the inflow is set to zero in all reservoirs, and in the second the inflow in reservoir 1 in watercourse 1, which is the most variable inflow, is set as the inflow in the other reservoirs. The results are given in Table 9.5.

Increasing only one parameter at the time indicates that the sensitivity with respect to marginal inflow has the largest effect on the initial MIP gap and the solution time. Tests where the marginal water value is higher is therefore interesting to combine with

Table 9.3: Changes in minimum production level for generators in watercourse 2.

	Gen 4 [MW]	Gen 5 [MW]	Gen 6 [MW]	Gen 7 [MW]	Gen 8 [MW]	Gen 9 [MW]	Gen 10 [MW]	Gen 11 [MW]
Orig.	2.3	4.7	4.7	14.3	15.1	13.3	4.3	10.1
1	10	30	30	15	15	15	10	10
2	10	30	30	20	20	20	10	10
3	10	40	40	20	20	20	10	10

Table 9.4: Sensitivity with respect to increasing the minimum production level in watercourse 2 according to values in Table 9.3.

	First integer solution				**Last integer solution**			
	Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]	Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]
Orig.	6 200	6 202	0.02	4 490	6 201	6 202	0.01	13 444
1	6 155	6 162	0.11	12 691	6 160	6 162	0.02	36 000
2	6 144	6 148	0.07	8 230	6 147	6 148	0.02	36 000
3	6 134	6 137	0.04	23 166	6 135	6 137	0.03	36 000

Table 9.5: Sensitivity with respect to inflow. Test performed setting the inflow equal to zero in all reservoirs and introducing more variable inflow in two reservoirs.

		First integer solution				**Last integer solution**			
		Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]	Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]
	Orig.	1 947	1 978	1.55	861	1 975	977	0.11	36 000
1	0	1 536	1 569	2.12	84	1 565	1 567	0.14	36 000
	Variable	1 804	1 836	1.73	163	1 833	1 835	0.10	36 000
	Orig.	6 200	6 202	0.02	4 490	6 201	6 202	0.01	13 444
2	0	4 711	4 717	0.13	6 684	4 714	4 716	0.05	36 000
	Variable	4 473	4 479	0.15	7 841	4 476	4 479	0.06	36 000

adjustment on inflow and minimum production level. This sensitivity analysis is only conducted for watercourse 2 and the marginal water value has been increased by 80 % in all these tests. The results from an increase in the marginal water value of 80 % is stated as *original* in the following tables with results.

The results from this sensitivity analysis are given in Table 9.6 and 9.7. Note that the maximum solution time is set higher when adjusting both the marginal water values and the minimum production level since a solution time of 10 hours only resulted in non-integer solutions. The solution time is therefore set such that the optimization procedure runs until the first integer solution is found.

Table 9.6: Sensitivity with respect to marginal water value and minimum production level. Marginal water value increased by 80 % in all test instance. Minimum production level as increased according to Table 9.3.

	First integer solution				**Last integer solution**			
	Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]	Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]
Orig.	2 743	3 120	13.75	30 907	2 805	3 120	11.24	36 000
1	2 649	3 002	13.34	21 866	2 662	2 998	12.76	36 000
2	2 677	2 983	11.44	46 695	*2 677*	*2 983*	*11.44*	*46 695*
3	2 679	2 968	10.76	88 739	*2 679*	*2 968*	*10.76*	*88 739*

Table 9.7: Sensitivity with respect to marginal water value and inflow. Marginal water value increased by 80 % in all test instance. Inflow set to zero and more varying inflow introduced in two of the reservoirs.

	First integer solution				**Last integer solution**			
	Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]	Best sol. [1000 NOK]	Best UB [1000 NOK]	Gap [%]	Time [s]
Orig.	2 743	3 120	13.75	30 907	2 805	3 120	11.24	36 000
0	68	35	413.38	58 341	*68*	*35*	*413.38*	*58 341*
Variable	264	546 721	106.94	28 334	316	547	73.02	36 000

9.3 Discussion

The analysis above shows that the solution time of the model is highly sensitive to input-parameters. Observing the solution time and initial duality gap, it can be seen that increasing the marginal water value has the largest individual effect on the initial duality gap and the time until this gap is reached. Individual adjustment of the other parameters without increasing the marginal water value increases the gap only by 0.01 % to 2 %. These observations indicate that the unit commitment decision of the producer when the marginal water values are low can be considered as more or less given by the relationship between the overall price level and marginal water values, almost unaffected by the uncertainty of the prices.

These observations are also correlated with the unit commitment decision. When the initial duality gap in the root cut and heuristic is small, only a small percentage of the resulting binary unit commitment variables are 1. For example for the tests done with the initial data, only 95.4 % and 99.7 % of the unit commitment variables are equal to 1 in watercourse 1 and 2 respectively. Thus, the producer wishes to schedule almost all of the generators at all times. This pattern is especially prominent in watercourse 2 where the minimum production level is very low. When the marginal water value increases by 90 % in watercourse 1 and by 80 % in watercourse 2, the number of resulting binary variables being equal to 1 decrease to 38.7 % and 38.6 % in watercourse 1 and 2 respectively. This support the hypothesis that more variable unit commitment decisions results in an increase in model complexity and an increase in solution times.

Looking more closely at the changes when varying the minimal production level and the inflow, it can be seen that the first duality gap however changes some, indicating that higher minimum production levels results in somewhat increase in the complexity of the problem. Similarly when changing the inflow. Varying these parameters when the marginal water value is increased however results in large differences in solution time. In several of the tests no duality gap is found within 10 hours, and the initial duality gap is above 10 % for all test. The combination of higher marginal water values and changes to inflow has the largest effect. Inflow affect the level of water in the reservoirs which again affect the marginal water values. Hence, it is assumed that this implicit adjustment of marginal water values are one of the major reasons for the large increase in solution time associated with these tests.

Based on these results and the results from the sensitivity analysis conducted, we have identified two problem instances, test instance 1 and 2, that we will apply decomposition methods on. Test instance 1 has been decided to comprise watercourse 1 with an increase in marginal water value of 90 % while test instance 2 comprises watercourse 2 with an increase in marginal water value of 80 %. These adjustments are seen as realistic since the marginal water value changes throughout the year and the price of electricity then lies closer to the marginal water value. Comparing the production patter of these test instances to the production pattern given above, show that the generator turns more on-off as previously discussed. Hence, the second stage decision is considered more complex for the power producer.

It is also of interest to investigate the convergence of the branch and bound procedure when solving the deterministic equivalent for the two test instances. The relationship between the duality gap and the solution time is illustrated in Figure 9.2. It is clear that the convergence of the problem is low. The initial duality gap is high for both test instances, but is largely reduced for test instance 1 when the solution time passes 13 hours before stabilizing at 3.5 %. For test instance 2 the duality gap is reduced during the first iterations in the root cut and heuristic before stabilizing at 11 %. It is well known that the branch and bound procedure can be slow, thus the results are not surprising. Branching on the separate unit commitment decisions in the branch and bound tree only result in a minor restriction due to the large number of unit commitment variables, hence improvement in the best feasible integer solution found improves slowly.

To examine the effects of changes in input data on the value of the stochastic model, we evaluate the two test instances with respect to expected value of perfect information (EVPI) and the value of stochastic solution (VSS) and compare these values with the unmodified input data. The results are given in Table 9.8. Note that the two test instances only find a feasible solution wihin a gap of 3.54 % and 10.76 % for watercourse 1 and 2 respectively during 80 hours.

It can be seen that the values of the EVPI increases for test instance 1 compared to the unmodified data. The VSS however decreases, indicating that the value of solving the model with stochastic parameters compared to the expected value is small. However, due to the duality gap of the feasible solution found, the EVPI given can only be considered an upper bound of the true EVPI, while the VSS can be considered as a lower bound of the true VSS.

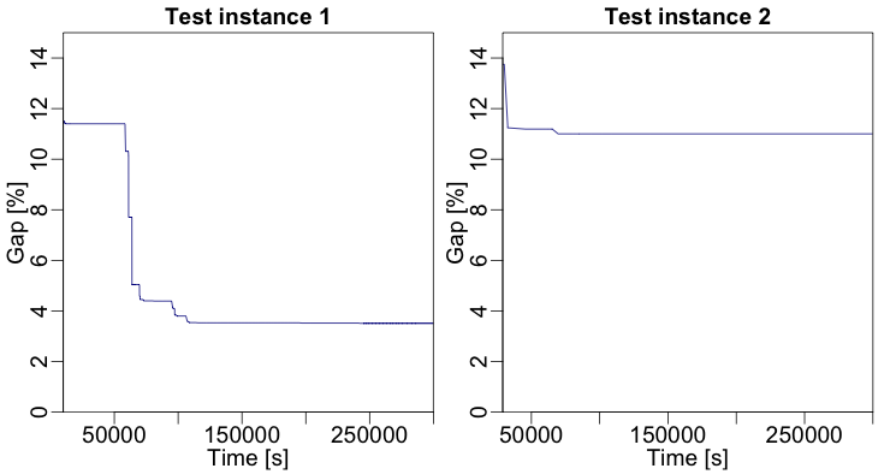


Figure 9.2: Duality gap during the root cut and heuristic, and branch and bound solution procedure for test instance 1 and 2 respectively. Tests run on scenariotree with 70 scenarios for 80 hours.

For test instance 2, no value of the RP within a small enough duality gap was found during a solutiontime of 80 hours. Hence, it is considered too difficult to find a proxy of the EVPI and VSS for this test instance. However, the difference between the EEV and WS show that there is a difference in almost 200 000 NOK between the EEV solution and the WS solution, hence it can be assumed that there will be some value added by solving a stochastic model.

Table 9.8: EVPI and VSS for the two test instances compared to EVPI and VSS with initial data. For value where a duality gap exist, the EVPI and VSS are given as an interval.

		RP (NOK)	WS (NOK)	EEV (NOK)	EVPI (NOK)	EVPI (%)	VSS (NOK)	VSS (%)
Original	1	1 974 508	1 984 145	1 907 934	9 636	0.49	66 575	3.37
	2	6 201 152	6 254 708	6 195 043	53 556	0.86	6 109	0.10
Test instances	1	669 163	717 154	667 805	47 991	7.17[2]	1357.6	0.20[3]
	2	NA	3 063 227	2 864 710	NA	NA	NA	NA

Since the reserved capacity in the primary reserve market can be divided within the portfolio of the producer it assumed that the value of the SMIP model presented is larger when combining the two problem instances compared to solving the two test instances separately. Thus, the increased flexibility associated with having a larger portfolio of generating units have an added value when delivering primary reserves. Tests have been conducted in order to find the RP solution of the two problem instances combined. However, no feasible solution is returned from the solving the problem during a solution-time of 48 hours. Hence, the EVPI and VSS of the combined test instances have been found to be too difficult to obtain within a reasonable amount of time. However, the EEV and

[2]Upper bound on EVPI. RP only solved to 3.5% optimality, thus exact EVPI $\in$ [3.54 %, 7.17%]
[3]Lower bound on VSS. RP only solved to 3.5% optimality, thus exact VSS $\in$ [0.20 %, 3.58 %]

WS solutions have been found and it can be seen that these values are larger than the combined values of the EEV and WS from solving the SMIP problems separately. The EEV for the two combined test instances is 8 189 017 and the WS is 8 260 833 (initial test instances). Thus, the increase in objective function value is 1% for the EEV and 0.2 % for the WS compared to solving the two problem instances separately. These numbers are not large, but for a power producer with an income in the size of millions, including all generators in the portfolio will result in an increased profit.

Chapter 10

Computational study of decomposition methods

This chapter is devoted an analysis of the performance of the Lagrangian and Dantzig-Wolfe decomposition methods when applied to the SMIP problem. We start by analysing each of the methods separately with respect to their input parameters, before the two are compared with respect to the tightness of their bounds and convergence properties. All tests are initially done for test instance 1, which is watercourse 1 with an increase in marginal water value of 90 % compared to the initial data, before being applied to test instance 2, which is watercourse 2 with an increase in marginal water value of 80 % compared to the initial data. Tests are mainly conducted using the same scenario tree with 70 scenarios as the deterministic equivalent. This facilitate easily comparison of the solution methods both with respect to the results and the solution times.

The results from the decomposition methods are evaluated based on the value of their returned upper bound; the lower the better. This bound is furthermore compared to the value of the best upper bound from the branch and bound tree of the deterministic equivalent found within the same time limit. All methods have been run for ten hours, and their performances are only compared within this time limit. Thus, fast convergence is seen as more important than absolute convergence. This is in line with the requirements of a power producer who has limited time to make decisions about what to bid into the different markets.

In most cases neither the Lagrangian nor Dantzig-Wolfe decomposition methods are able to completely close the duality gap to the optimal objective value of the original problem. Consequently, the solutions returned from the methods are infeasible with respect to the non-anticiaptivity constraints or the unit commitment decisions depending on the decomposition method applied. The Lagrangian decomposition returns solutions that are not feasible with respect to the non-anticipativity constraints, while the solutions returned from the Dantzig-Wolfe decomposition are usually infeasible with respect to the integer requirements, unless branching is applied. In order to recover feasible solutions from the decomposition methods, two simple constructive heuristics are implemented and tested.

10.1 Experimental methodology

Similar to the deterministic equivalent, all decomposition methods have been implemented in the Mosel modeling language, solved with Xpress-MP v7.8.0 and tested on 64-bit Windows 7 PC with 3.40 GHz Intel Core i7-3770 CPUs (4 cores, 8 threads) and 16 GB RAM.

Parallelization has been used to utilize the computational advantages of decomposing the problem into separate subproblems. When solved in parallel, the subproblems are started at the same time and solved simultaneously. Parallelization in Xpress is implemented by using the module *mmjobs*, which includes facilities for model management, synchronization of concurrent models based on event queues, and a shared memory IO driver. With 70 subproblems and one master problem, the program could potentially solve 70 subproblems in parallel in separate threads. However, we have been solving all submodels on the same computer without a distributed architecture. Thus, tests on the programs are limited by the number of threads the running computer can process in parallel. As the tests have been run on computers with 8 threads, the full potential of parallelization is therefore not exploited.

10.2 Lagrangian decomposition

This section presents the main results of solving the SMIP problem by Lagrangian relaxation using different Lagrangian multiplier updating schemes. The subgradient, cutting plane, proximal bundle and progressive hedging methods are implemented and compared. In general, these methods differ with respect to aspects such as computational complexity, stability and memory usage, and it is of interest to compare their performances against each other when applied to the SMIP problem.

The optimal parameters are problem dependent and extremely hard to find, and it is outside the scope of this thesis to go into a deep analysis of these parameters. We therefore only seek to find reasonably good parameter values and have investigated three different values for the input parameters of each multiplier updating scheme; high, medium and low.

The results are illustrated in figures showing the bound from the iterations. The number of iterations completed within 10 hours differ across methods and across differently calibrated instances of the same methods. Results are only plotted until the smallest number of iterations, since this is considered the most interesting part to compare. For more information regarding the convergence properties of the methods, we refer to tables.

Oftentimes, the Lagrangian relaxation methods suffer from slow convergence. Thus, some additional measures are implemented and combined with the multiplier updating methods to try improve convergence and reduce the computational time. Methods tested are scenario clustering and warm-start procedures.

10.2.1 Subgradient method

The subgradient updating scheme is highly dependent on the step length. To test the sensitivity of the method with respect to this parameter, three tests have been conducted. In the three tests, the initial value of the step length has been set to 0.2, 2 and 10 respectively. Independent of the initial step length, the values are diminished to ensure convergence. It is decreased by 5% every 10^{th} iteration. The size of the reduction in step length parameter given iteration is based on trial and error on . Figure 10.1 shows the objective value for each iteration given different values for the step length parameter. The best bound reached within 10 hours is given in Table 10.1.

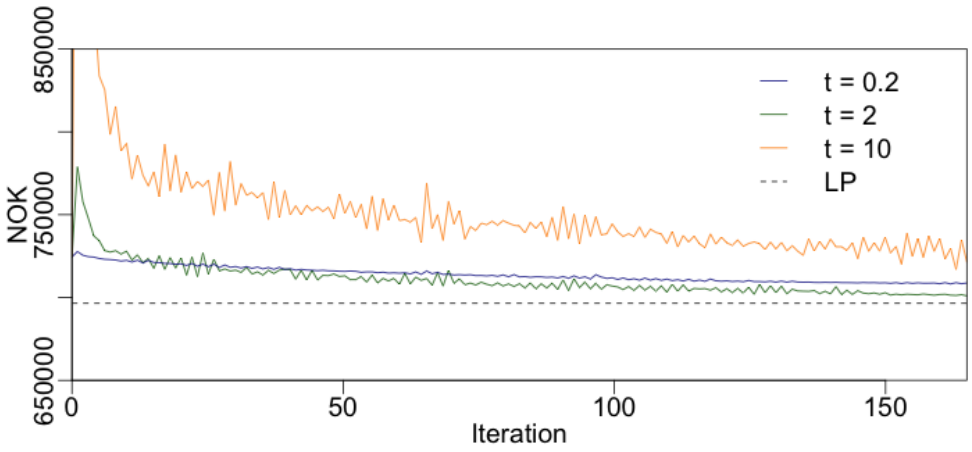


Figure 10.1: Comparison of solutions of subgradient method with varying step length parameter.

Table 10.1: Best bound achieved for different initial values on the step size parameters in the subgradient method.

t	Best UB
0.2	706 625
2	700 741
10	716 965

Applying the subgradient updating scheme for the Lagrangian multiplier is seen to result in slow convergence. Within the time-limit of 10 hours, none of the step length values applied result in an upper bound on the objective function value that is lower than the upper bound from the value of the LP-relaxation. A step length equal to 2 however results in a solution that is closer to the solution of the linear relaxation than what is seen for both larger and smaller step lengths. The results also show a clear relationship between high initial values for the step length and the oscillating behaviour of the objective value. For high initial values for the step length, the subgradient updating scheme is seen to result in highly oscillating behaviour. In some cases this oscillation is necessary to ensure a complete search to reach the optimal multiplier value, but in this case, the oscillations are not stabilized within the time limit of 10 hours. Thus, our requirement for fast convergence cannot allow for oscillations that might be necessary to allow for absolute

convergence if the tests had been run longer. We conclude that $t = 2$ is the best choice for initial step parameter for this SMIP problem, given the time restriction.

10.2.2 Cutting plane method

Figure 10.2 shows how the objective value of the Lagrangian dual develops during 150 iterations of the cutting plane algorithm. There are large oscillations during the first iterations, before the objective value stabilizes after approximately 80 iterations. The extreme oscillations occur in the beginning when few cuts have been added and the feasible region of the objective function is large. The feasible region then gradually shrinks with the number of iterations as more cuts are added to the master problem. The smaller feasible region eventually results in more stable results.

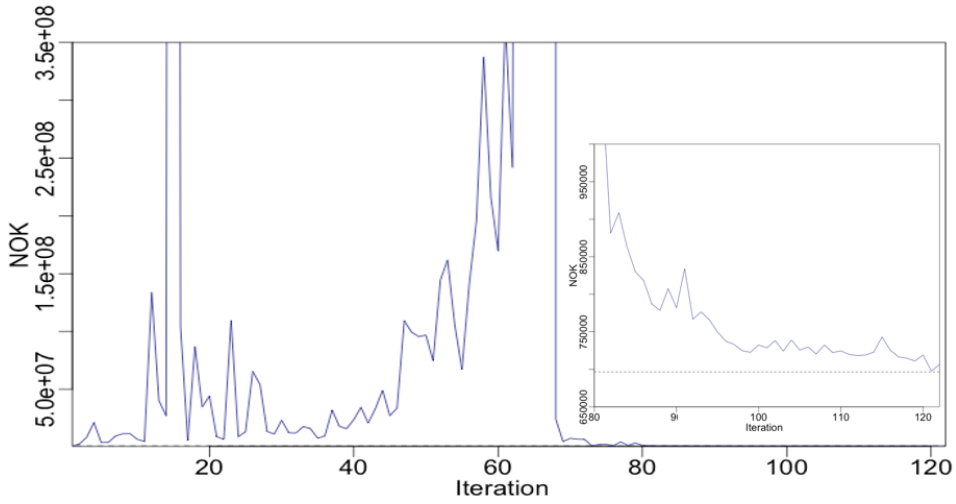


Figure 10.2: Standard cutting plane algorithm.

Putting a limit on the multiplier of the cutting plane algorithm can facilitate faster convergence by preventing unrealistically high and low multiplier values and avoid unnecessary oscillations during the first iterations, as the initial feasible region of the multipliers shrinks proportionally to the limit. However, if the limit is too restrictive, it can prevent optimal values from being found.

Tests are performed with three different limits on the multipliers, and the results are illustrated in Figure 10.3. All tests, except for the test with the limit set to 100, are seen to find a better bound than the LP-relaxation.

When the maximum limit on the multiplier is set to 100, the jump in the second iteration, seen for the two other test runs, is avoided, but the optimal bound cannot be reached due to some of the best multipliers lying outside the feasible region defined by the maximum limit. Hence, this limit can be considered as too restrictive. A limit of 1000 includes the first jump and is also seen to converge faster. However, a limit of 10 000 finds a better upper bound during the solution time, even though it allows for a large degree of variability during the first 30 iterations. The best bound reached within 10 hours for each of the limits are given in Table 10.2.

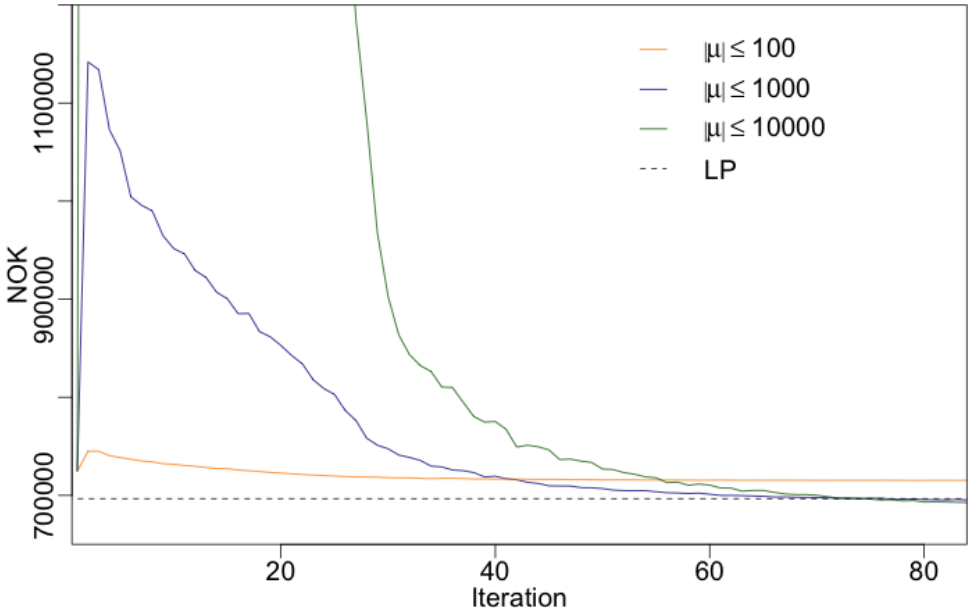


Figure 10.3: Comparison of solution of cutting plane procedure varying max limits on μ.

Based on the above, the recommendation would be to put the limit at 10 000. It oscillates, but stabilizes relatively fast within 10 hours. Moreover, this limit does not seem to be too restrictive to find the optimal multipliers.

Table 10.2: Best bound for different max limits on the multipliers μ in the cutting plane algorithm.

max $\lvert\mu\rvert$	Best UB
100	714 968
1000	695 292
10000	689 980

In order to prevent excessive growth in problem size with increasing number of iterations, the cutting plane algorithm can be modified to a dynamically constrained cutting plane (DCCP) algorithm. Based on the results from the analysis above, a max limit of 10 000 on the multipliers was chosen when implementing the DCCP. The DCCP algorithm has been implemented to see how restricting the problem size, by limiting the number of cuts considered, can have an effect on memory usage and solution time. The objective value for different values of maximum number of cuts considered, $n_{\max}$, is plotted against objective value in Figure 10.4, where $n_{\max} = \infty$ means no upper limit on the number of cuts.

As indicated by Figure 10.4, limiting the number of cuts has a large effect on the stability of the cutting plane method. It starts to oscillate after the maximum limit is reached, and the fewer cuts considered, the more instability. The cutting plane algorithm completes around 150 iterations during the time limit of 10 hours, which is within the interval of what the solver can handle, hence problem size with respect to the number of cuts is not such a big issue. The loss in stability, measured by the oscillations, when limiting

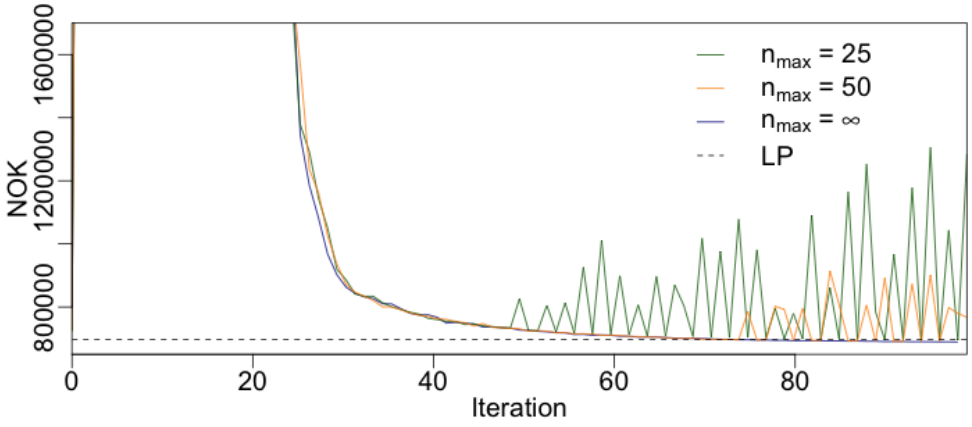


Figure 10.4: Comparison of DCCP varying the maximum number of cutting planes.

the number of cuts does not make it worth limiting the total number of cuts for the problem instance considered. Thus, when the number of iterations is rather limited, it is recommended using the original cutting plane algorithm without any limits on the number of cuts.

10.2.3 Proximal bundle method

The proximal bundle method can contribute to stabilize the cutting plane algorithm and avoid extreme oscillation in the earliest iterations by penalizing the objective function for multiplier values that deviate from the current stability center. This can be seen as soft constraints, in contrast to the hard constraint in the cutting plane when the maximum limits was introduced. Thus, the multipliers are not prevented from reaching their optimal values, and at the same time unnecessary oscillations in objective values in the first iterations are avoided.

The procedure is highly sensitive to the penalty factors τ. As can be seen from Figure 10.5, varying the penalty factor has great impact on convergence. Table 10.3 shows the best bounds achieved after 10 hours for the tested values of τ.

Table 10.3: Best bound achieved for different values of penalty factor τ in the proximal bundle algorithm

τ	Best UB
0.1	691 882
1	689 728
10	724 451

For large τ, the penalization becomes dominating and it is extremely costly to move the multiplier significantly away from the current stability center. Thus, convergence is slow. For low values of τ the bundle method starts to look more like the standard cutting plane method and it oscillates more during the first iterations. Based on the tested values, the method performs best for $\tau = 1$ and the bound after 10 hours is better than the bound from the LP-relaxation.

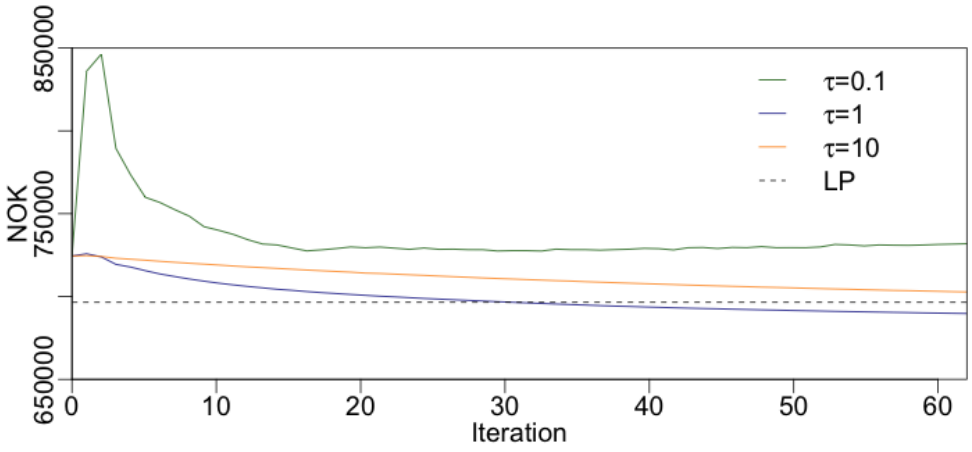


Figure 10.5: Comparison of bundle method for different values of penalty factor τ.

10.2.4 Progressive hedging

The success of the stabilization when applying the proximal bundle method to the SMIP problem, makes progressive hedging particularly interesting. Similar to the bundle method, the standard progressive hedging method does also contain a stabilizing quadratic term, but in the subproblems instead of the master problem.

When performing initial tests with the standard progressive hedging algorithm, not a single iteration is completed before the solver reports out of memory. The quadratic subproblems complicate the solution procedure. Adding a quadratic term to already complex subproblems makes the overall problem extremely hard to solve, and the growth of the branch and bound trees in the sub problems cannot be handled by a single computer. A distributed architecture, running each subproblem on separate computer could potentially be used to handle this issue, but still, it gives us an indication of the complexity and scalability of the subproblems, and that adding more into them is not a recommended approach to speed up the solution process.

As a consequence of the memory problems of the standard progressive algorithm, the simplified version described by Escudero et al. (2013) was implemented, as this method does not include quadratic terms in the subproblems which was seen to complicate the solution procedure due to the complexity of the subproblems. In the following, this simplified method of progressive hedging is referred to as progressive hedging, not the original.

The tests performed on this method are presented in Figure 10.6. The plot indicates that the progressive hedging variant is stable, but converges to a point quite far from optimal. It achieves the best bound when the step size parameter, ρ, is set to 10 000. However, it turns out to be hard to tune the parameters of the model, both the initial step size and the step size regulating procedure, to ensure fast convergence to a good bound. For the parameters tested the method does not manage to return better bounds than the LP-relaxation.

Table 10.4 shows the best bounds achieved after 10 hours for the tested values of ρ.

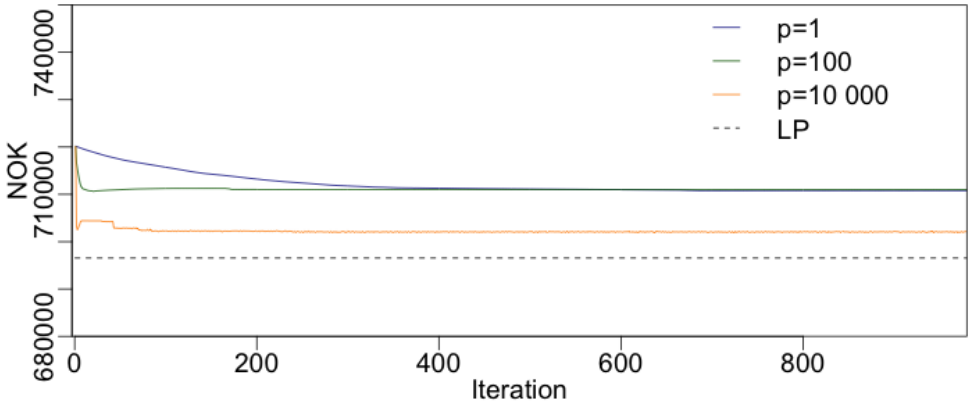


Figure 10.6: Comparison of the simplified progressive hedging algorithm for different values of the initial step size parameters.

Table 10.4: Best bound achieved for different values of step size parameter ρ in the progressive hedging algorithm.

ρ	Best UB
1	716 339
100	716 266
10 000	708 764

10.2.5 Comparison of the multiplier updating schemes

From the analysis above, it should be clear that there are big differences among the tested updating schemes with respect to complexity, stability and solution times. The different characteristics of these updating schemes make it interesting to investigate which multiplier updating procedure performs best on the SMIP problem. Comparing the methods is therefore of great interest.

When comparing the methods, we have been using the most promising input parameters identified through the analysis in the previous section. These were the subgradient with $t = 2$, cutting plane with $|\mu| \leq 10\ 000$, proximal bundle with $\tau = 1$ and the simplified progressive hedging algorithm with $\rho = 10\ 000$. These instances are compared along several dimensions; best bound, complexity of each iteration, and by the total number of iterations completed within the time limit. A summary of the results is presented in Table 10.5.

It can be observed that the proximal bundle method converges significantly faster than the other updating schemes. The method also results in the best upper bound. The cutting plane algorithm provides almost the same bound, but the convergence of the method is slower. The multipliers of the bundle method have the same feasible region as the cutting plane, but the quadratic penalty term in the objective function guides the search in the most promising areas. Thus, resulting in faster convergence. Fewer iterations limit the number of cuts added and prevent the problems from becoming very large. This is advantageous considering possible memory issues associated with the decomposition method.

The performance of the simplified progressive hedging is the least satisfying one. Convergence is extremely slow and the resulting upper bound is higher than the upper bound provided from

Table 10.5: Comparison of average improvement in upper bound per iteration, average time spent per iteration and the total number of iterations for each of the methods when run for 10 hours.

	Best UB [NOK]	Avg. imp. pr. iteration [NOK]	Avg. time pr. iteration [s]	Iterations [#]
Subgradient	700 741	139	210	172
Cutting plane	689 980	374	408	94
Proximal bundle	689 728	500	500	60
Simpl. prog. hedg.	708 764	16	37	980

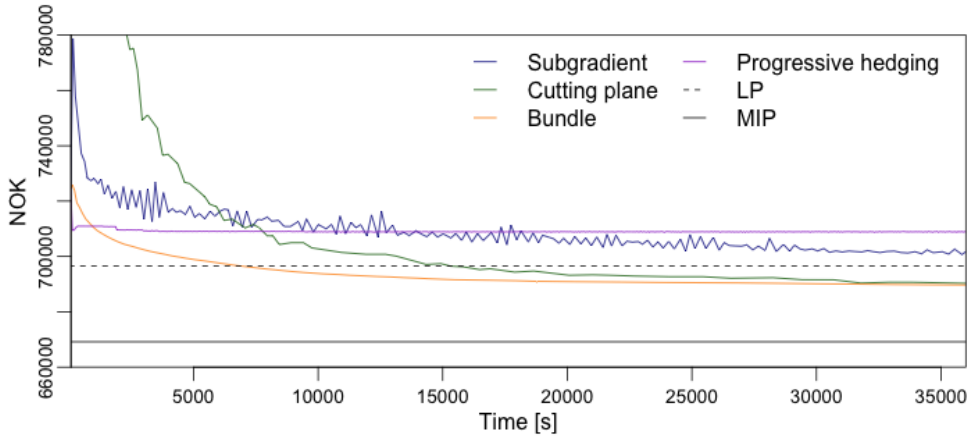


Figure 10.7: Comparison of the different Lagrangian multiplier updating schemes with respect to their objective values.

the branch and bound tree within the same time-frame. This is because tuning the parameters correctly is particularly crucial for the performance, but also very difficult to achieve, especially since the relaxed constraints are equality constraints. Due to this, we are not able generate feasible solutions during the run of a Lagrangian relaxation without the use of additional techniques.

Similar to the progressive hedging algorithm, the subgradient method performs quite poorly due to the same tuning issues mentioned above. The method is seen to provide good upper bounds compared to the cutting plane method during the first iterations, but is surpassed by this in later iterations. The feasible region of the multipliers gradually decreases as more cutting planes are added, and the multipliers are chosen more targeted than the subgradient relying on a simpler, less intelligent approach to adjust the step length as it progresses towards better solutions.

From column three of Table 10.5 it is clear that the solution time per iteration of the decomposition methods varies significantly depending on the chosen updating method. The results also indicate a relationship between the average solution time and average improvement in upper bound per iteration.

There are mainly two factors driving the differences in solution time per per iteration; complexity of the master problem and the particular values of the multipliers.

The complexity of the master problem differs between the methods, resulting in different average solution times. The proximal bundle method results in the longest average time per iteration.

This is partly due to the quadratic term in the objective function of the master problem, resulting in a much more complex non-linear problem compared to the other multiplier updating methods. In return, the average improvement per iteration is relatively large and the proximal bundle method needs significantly fewer iterations to reach the same bound as any of the other methods. For comparison, the relationship between the average solution time and average improvement in bound per iteration for each of the updating schemes is given in Table 10.5.

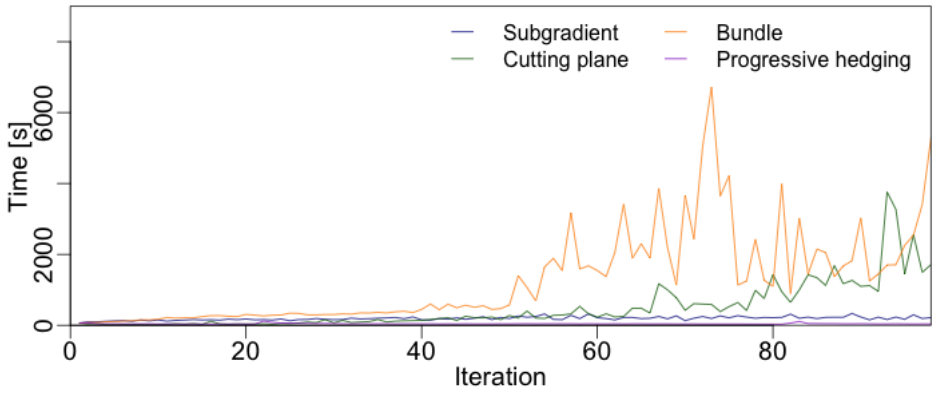


Figure 10.8: Comparison of time per iteration for subgradient, cutting plane and bundle method.

Some multipliers resulting in harder subproblems than other. This is illustrated by the peaks of the graphs in Figure 10.8. Figure 10.8 also shows that the solution time per iteration has a trend towards growing with increasing number of iterations. This is because the subproblems are harder to solve for more "qualified" multipliers. This is another factor driving the long average time per iteration for the bundle method. Fast convergence to good multipliers result in long solution time per iteration also in early iterations.

For the cutting plane and bundle method, the time per iteration will also be affected by the growth in problem size as more cuts are added in each iteration.

In total, the proximal bundle method turns out to be the best among the tested updating methods, but closely followed by the cutting plane algorithm. The stabilizing effect used as part of the proximal bundle algorithm, penalizing the objective function value based on the distance to the proximal stability center turns out to be an important feature in order ensure fast convergence when applying Lagrangian relaxation to the SMIP bidding problem. We will therefore use the proximal bundle method for multiplier updating when comparing Lagrangian decomposition with Dantzig-Wolfe and the standard branch and bound solution procedure.

10.2.6 Extensions to the bundle method

Initial multipliers

Initializing the multipliers with non-random values can compromise better starting point for the multiplier updating algorithm and result in quicker convergence to the optimal multipliers compared to initializing the multipliers to 0. Figure 10.9 provides a comparison of the original bundle method initialized with multiplier values of zero and a version using a warm start procedure based on initializing the multipliers with the dual variables from the non-anticipativity constraint returned from the LP-relaxation.

The graph shows that there is only a small bound between the two versions, and after 10 hours the method without the warm-start is returning the best bound. These results indicate that after many iterations the warm-start procedure will not guide the search in the correct area to such a different extent than initially. After many iterations the original proximal bundle updating scheme without a warm start is returning a slightly better bound than the warm-started updating procedure. In this particular case, applying the warm start procedure is beneficial compared to using zero-valued multipliers when the goal is to find the best upper bound during a shorter-time period than 10 hours.

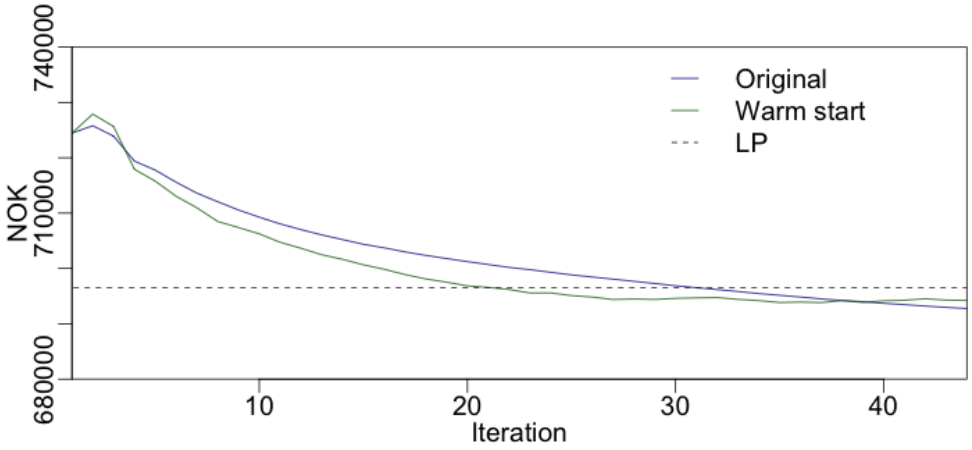


Figure 10.9: Comparisons of the bundle methods with and without a warm-start procedure.

Scenario clustering
Scenario clustering is implemented in combination with the proximal bundle method in order to try to achieve tighter bounds faster. When running scenario clustering for clusters of size two, the subproblems halt in the first iteration and the solver runs out of memory. Due to the large number of integer constraints in the subproblems, the branch and bound tree generated during the solution process quickly becomes large and unmanageable for the solver. The individual subproblems are complex in themselves, thus the resulting problems from aggregating pairs of these cannot be solved to optimality with the available resources.

With more computational resources, this could however have been different. By making use of a fully distributed architecture, each subproblem would get access to more memory that potentially could allow the completion of the full branch and bound searching process, and it is therefore of interest to explore the potential effect form making use of scenario clustering. The clustering has been performed on problem instances with fewer scenarios in order to investigate the potential advantages in terms of tighter bounds. Results after first iteration, using scenario trees of size 10 and 20 with clusters of two and two scenarios are shown in Table 10.6.

Table 10.6 shows that there is a large cost in terms of solution time, and only small gain in terms of tighter bounds when applying scenario clustering. Thus, for such hard subproblems as in this particular case, there seems to be little gain of using scenario clustering.

10.2.7 Heuristics

In order to recover feasible solutions from the solution to the Lagrangian relaxation, the two different Lagrangian based heuristics described in Section 6.3 have been implemented. These

Table 10.6: Improvements in objective value bound by applying scenario clusters of size two to scenario trees of different sizes.

Scenarios [#]	Cluster size [#]	1st bound [NOK]	Δ bound [%]	Time [s]
10	1	669 910	0.9	9
	2	663 793		2 430
20	1	726 698	0.8	15
	2	720 905		4 883

are based on fixating reserve bids and unit commitment decisions respectively.

The unit commitment decisions can be obtained directly from the Lagrangian algorithm because the Lagrangian solutions are integer feasible. For reserve bid heuristic, on the other hand, we need to aggregate the bids from different scenarios into one common bid. We have been experimenting with different aggregation techniques; minimum, maximum, mean and mode. Table 10.7 shows the MIP solutions resulting from each of these procedures and from the heuristic based on fixating the unit commitment decisions for tests performed on a single scenario tree with 70 scenarios. The heuristics are applied to the Lagrangian solution found in the first iteration of the algorithm run.

Table 10.7: MIP solutions from heuristics applied to Lagrangian solution.

Fixated variable	**Aggregation operator**	**MIP solution**
Reserve bid	Max	270 108
Reserve bid	Mean	636 525
Reserve bid	Mode	664 332
Reserve bid	Min	644 741
Unit committed	N/A	613 861

The heuristic based on fixating the unit commitment is outperformed by all the reserve bid based heuristics. This heuristic fixates the unit commitment decisions based on the solutions returned from the first iteration of the Lagrangian relaxation. These unit commitment decisions are the same solutions as we would get by just solving the deterministic equivalent without considering the non-anticipativity constraints. When these decisions are fixated, the quality of the resulting MIP solution is characterized by the fact that all scenario subproblems have been optimized based on only taking the prices of that particular scenario into account. Hence, the uncertainty in the model will have a large impact on the solution and the flexibility of the stochastic model is not taken into account. The heuristics could however potentially perform better if it was applied after a larger number of iterations since the unit commitment decisions would be better adjusted to the present uncertainty.

The best MIP solution was found found by using the reserve bid heuristic based on the mode operator. When using the mode operator, the resulting bidding curve can be characterized as good in many of the subproblems. Aggregating bid by using the most common value can be seen as an "intelligent" adjustment to the Lagrangian solutions in order to fulfill the non-anticipativity requirement. We will therefore use the mode heuristic in the coming analyses.

10.3 Dantzig-Wolfe decomposition

This section presents the main results of solving the SMIP problem by Dantzig-Wolfe reformulation combined with column generation. Similarly as for the Lagrangian relaxation, the initial tests presented in the following have been run on test instance 1. The optimal variant of the Dantzig-Wolfe decomposition method has thereafter been applied to test instance 2.

Initially the convergence of the column generation procedure is presented. Thereafter the results from solving the problem with column generation is presented with a focus on possible ways to reduce the computational time. Dantzig-Wolfe decomposition of problems containing integer variables do usually not result in feasible solutions. Branch and price is therefore implemented to search for feasible solutions when the integer restrictions are violated. A simple heuristic has also been implemented to find feasible solutions more rapidly by using infeasible solutions from the Dantzig-Wolfe decomposition as a starting point.

10.3.1 Convergence of the column generation procedure

Initial tests on the Dantzig-Wolfe decomposition result in a large number of iterations of the RMP and long solution times. For a test run with 10 scenarios, the development of the objective value in the RMP for the first 10 hours is a given in Figure 10.10.

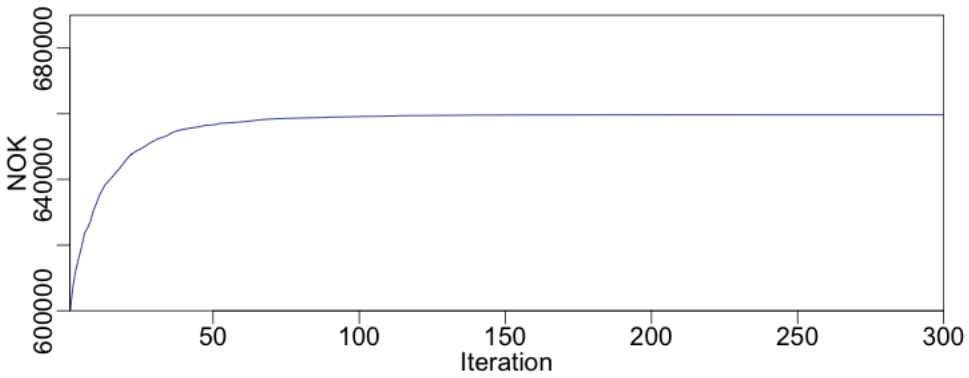


Figure 10.10: Upper bound for the column generation procedure with 10 scenarios for without stabilization for 10 scenarios.

The figure shows that changes in the objective value are small after completion of around 100 iterations. When studying the columns added to the RMP when the change in objective value flattens out, we observe that the profit of these added columns are approximately identical to the profit of columns that have already been added to the problem, and that the second stage solutions are identical while the first stage solutions differ. These small changes in objective value are therefore likely to be a result of tolerance errors in Xpress and that the objective value is the same in all the last iterations ; there exist several equivalent solutions made up of different columns. This is due to the fact that several bidding curves may result in the exact same reserve contract. Consequently, adding these new columns will not improve the quality of the solution and the solution process could potentially have been terminated much earlier without considerable loss in the final objective function value.

Due to the slow convergence described above, a stabilization method has been implemented as described in Section 6.2.1. The penalization on the surplus and slack variables is reset in every

iteration to the dual values of the RMP from the previous iteration. Tests are conducted to find strategies for updating the bounds on the surplus and slack variables. Different initial bounds are chosen, and are decreased by 0.5% after a given number of iterations. The stabilization procedure is evaluated based on the reduction in the number of iterations and the final bound on the surplus and slack variables. Due to the large size and the complexity of the model, tests have been run on a smaller problem instance with a scenario tree consisting of 10 scenarios. Each test is done for one complete run with column generation until it is stopped because no new columns with positive reduced cost are found, and the bounds on slack and surplus are set to be equal. The results are presented in Table 10.8.

Table 10.8: Varying stabilizing parameters in the column generation procedure. Initial bounds on slack and surplus variables and reduction in bound are tested.

Initial bound on slack/ surplus	Decrease bound every n iterations	Number of iterations	Final bound on slack/surplus
No stabilization		576	-
0.5	5	110	0.209
	10	105	0.349
	15	103	0.387
	20	103	0.427
0.75	5	98	0.366
	10	93	0.551
	15	92	0.611
	20	93	0.643
1	5	94	0.488
	10	88	0.735

The results show that stabilizing the algorithm by introducing slack and surplus variables, decreases the number of iterations by more than 80% for all tests. The number of iterations is seen to be lower when the initial bound is high, but in these cases, the final bound on the variables is also higher. The same is seen when the bound is reduced less frequently. There is a trade-off between the number of iterations and the resulting violation of the non-anticiaptivity constraints.

Based on these results, we have chosen to start with a bound of 0.5 and decrease this bound every 10^{th} iteration of Phase 2 in all further tests. This combination has resulted in a reduction of the number of iterations by 82% in the test case, which is a bit less than for most of the other tests. However, the non-anticipativity constraints are violated by only ± 0.349 MW. It is considered important that the violation is small since the model should provide an accurate bidding curve for the power producer that is applicable for all future outcomes of the uncertainty. As will be seen later in this chapter, more iterations are needed when the problem is solved with more scenarios, and the violation of the non-anticipativity constraints will thus be further reduced due to the reduction of the bound of the surplus and slack variables every 10^{th} iteration.

10.3.2 Results from the column generation procedure

Results from running the stabilized column generation procedure for 10 hours with the scenario tree consisting of 70 scenarios is illustrated in the two upper graphs in Figure 10.11. These

graphs show the how the upper bound and the RMP objective function develop given time and iterations respectively. The upper bound is found by adding the objective in the RMP and the reduced cost from the subproblems together, as described in Section 3.2. The relationship between solution time and iterations for the RMP and one of the subproblems is illustrated at the top of Figure 10.12. In the top left corner of the figure, the computation time in the RMP, including waiting for the subproblems to be finished, is given, and in the top right corner the solution times for one subproblem is given. It is seen that the computational time in the same subproblem can vary greatly depending on the dual values sent from the RMP and there can be large variations in the solution time between the different subproblems.

Solving each subproblem to optimality is time consuming due to the presence of binary unit commitment variables. Because columns added to the RMP do not need to be the optimal, we can end the calculation in the subproblems earlier. Some variations for early stopping have been tested by setting the parameters *MAXTIME* and *MIPRELSTOP* in the subproblems. *MAXTIME* sets the maximum allowable time to run the optimization procedure. Thus, when an optimal solution has not been found within the maximum time set by *MAXTIME*, Xpress returns the current best solution. *MIPRELSTOP* sets the relative size of the duality gap. When the duality gap found in the branch and bound procedure in Xpress is below the duality gap set by *MIPRELSTOP*, Xpress returns the current best solution. The tests are run until no subproblem returns columns with positive reduced cost.

The results from one run of each test instance are presented in Table 10.9. In this regard it should be mentioned that the solution process in Xpress is non-deterministic due to several threads being active at the same time. The number of iterations and columns generated for two identical test instances with identical values on the control parameters run on the same machine may therefore vary. However, when all parts of the problem is run to optimality, the resulting objective values will be the same in every run. The reader should keep this in mind when interpreting the results.

Table 10.9: Varying forced stopping parameters in the subproblems in the column generating procedure. Combinations of *MAXTIME* and *MIPRELSTOP* tested.

MAX-TIME [s]	MIPREL-STOP [%]	Objective node 1 [NOK]	Upper bound node 1 [NOK]	Solution time node 1 [s]	Iterations node 1
-	0.1	Not terminated after 48 hours			
-	1.0	Not terminated after 48 hours			
300	-	685 942	688 489	58 038	226
300	0.1	685 948	688 377	64 605	247
600	-	685 972	687 285	89 809	194
600	0.1	685 974	687 357	89 405	195
900	-	685 986	686 754	97 144	161
900	0.1	685 989	686 849	116 727	180
1200	-	685 995	686 430	139 231	175
1200	0.1	685 992	686 312	138 021	172

It can be seen that without any measures to reduce the time spent in each subproblem, total solution times become long. Using *MIPRELSTOP* alone to reduce solution time in the subproblems still results in long solution times. Using *MAXTIME* alone however reduces the solution time. For lower values of *MAXTIME* there is however a need for more iterations before the column generation terminates. This indicates that there is value in letting the subproblem spend

more time on finding better solutions. For higher *MAXTIME*, the upper bound on the objective solution in the RMP is also lower. Using a combination of *MIPRELSTOP* and *MAXTIME* gives varying performance over using *MAXTIME* alone. Based on these results it is decided to set *MAXTIME* to 300 when the decomposition methods are compared.

It is interesting to investigate further how the bound on maximum time affects the solutions during 10 hours compared to the case where the solution time in the subproblems is not bounded. In the lowermost graphs in Figure 10.11 the upper bound and the RMP objective value is illustrated when *MAXTIME* is set to 300 in the subproblems. Solution times per iteration in the RMP and one subproblem are illustrated in the two lowermost graphs in Figure 10.12.

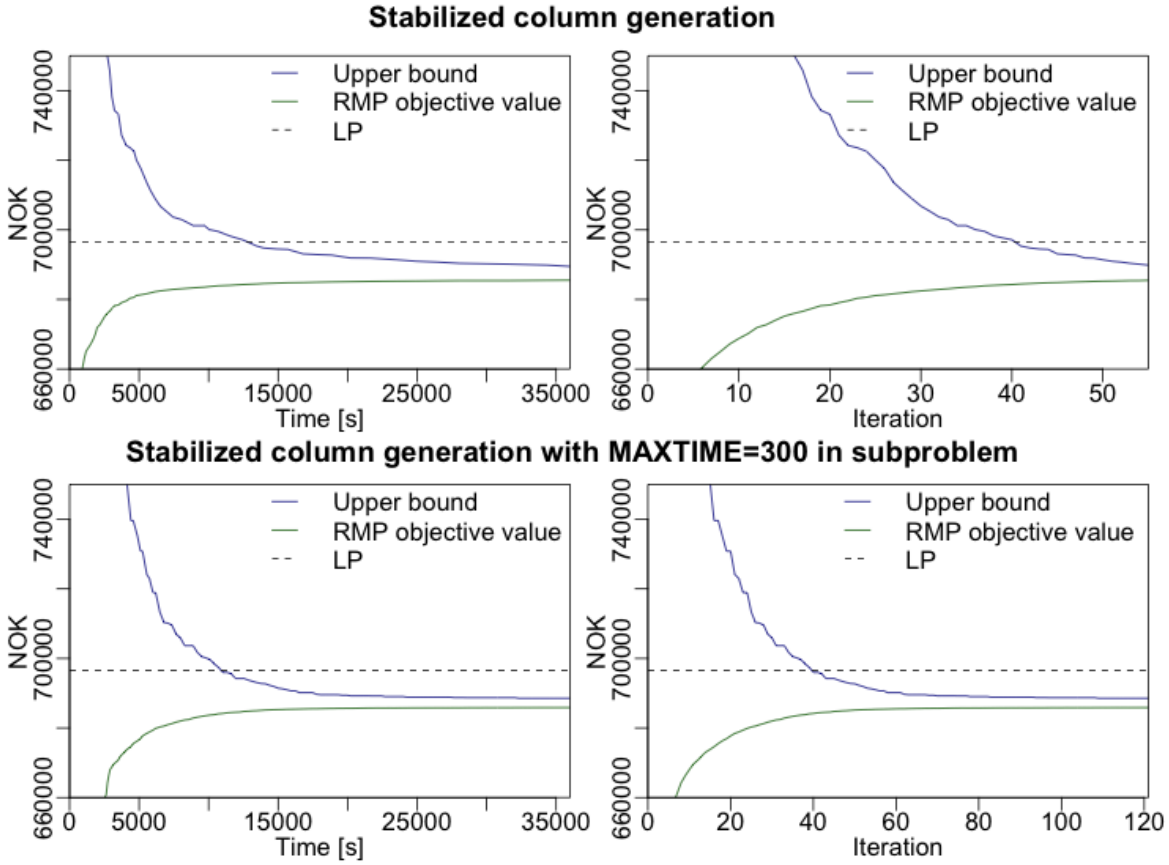


Figure 10.11: Upper and lower bound from stabilized Dantzig-Wolfe with (top) and without (bottom) *MAXTIME* in the subproblems.

When the time in the subproblem is bounded it can be seen that the gap between the upper bound and the RMP objective value after 10 hours is slightly smaller than without the limit. The method also converges to a smaller gap earlier than when there is no bound on the time in the subproblem. The bounded problem completes about twice as many iterations as the unbounded one, and in about half of the last iterations the bound is better in the bounded problem that what is reached after 10 hours in the unbounded problem.

Looking at the time per iteration in the RMP compared to the time in the subproblem for the bounded problem it is clear that the solution time of the subproblems is what drives the total computational time of the algorithm. It is also noted that the subproblems become harder to

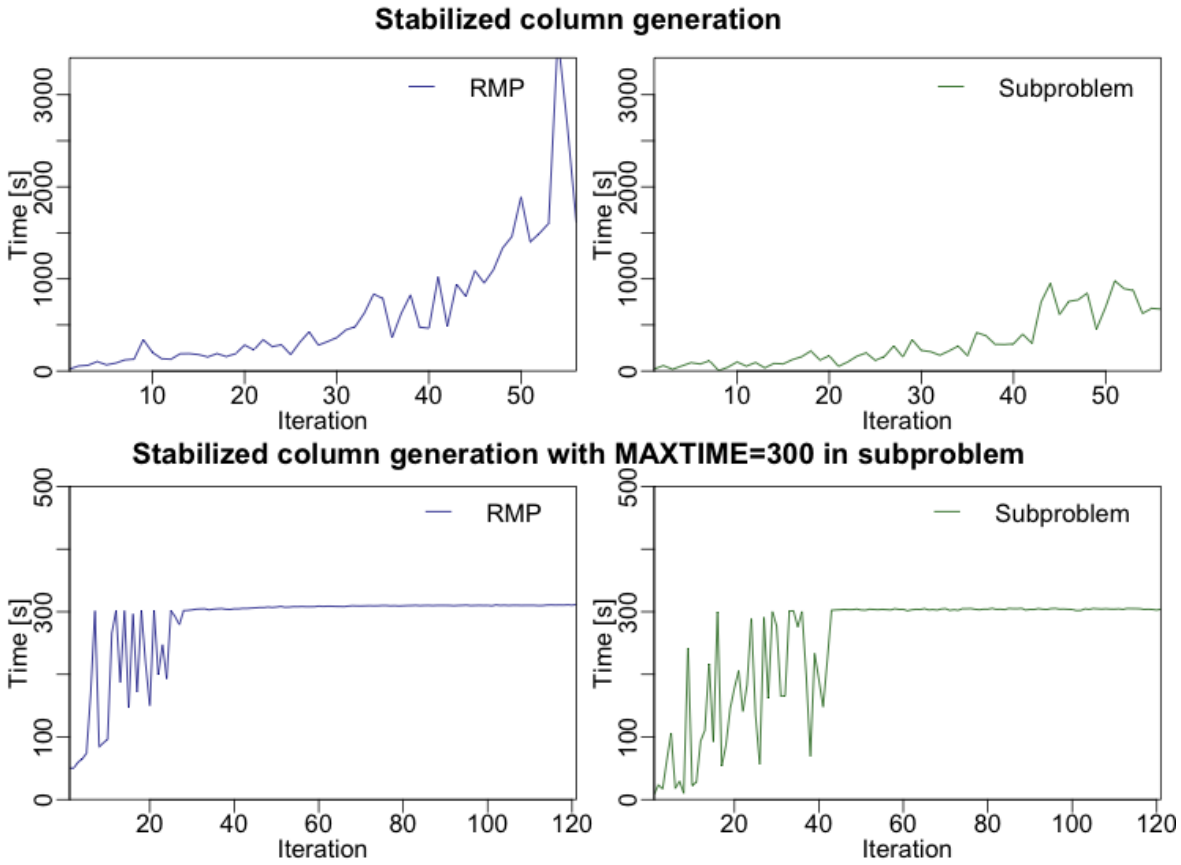


Figure 10.12: Time per iteration in RMP and subproblem with (top) and without (bottom) *MAXTIME* in subproblem.

solve with the iterations. This indicates that when the dual variables sent form the RMP get closer to their optimal values, the subproblems become harder to solve.

It is also seen from the graph for the bounded problem that after around 60 iterations, only small improvements are made in the RMP objective and the upper bound. This shows that there is potential in improving the stabilizing method for the column generation. As mentioned in Section 3.2.1, it can be difficult to find parameters for the 3-piecewise linear stabilization function that both stabilizes the column generation and allows for significant changes in the dual variables. Considering the close connection between column generation and Lagrangian relaxation, and the results from the different updating schemes for the Lagrangian multipliers, where it was seen that the proximal bundle method had the best and most stable performance, it would be interesting to see if a bundle-based stabilization method could improve the stabilization. Given the short solution times seen in the RMP it is stated as a hypothesis that even if the objective function of the RMP is quadratic it may still be solved in a reasonable amount of time. If stabilization is improved, there will in addition be much to gain by reducing the number of iterations. This is not further investigated in this thesis, but is suggested as a starting point for anyone who would be interested in improving the column generation based solution approach presented in this thesis.

10.3.3 Scenario clustering

Tests in which subproblems consist of several scenarios clustered together were performed with two scenarios in each cluster. Solving the subproblems to optimality without using a cutoff with either *MIPRELSTOP* or *MAXTIME* was not successful because Xpress runs out of memory. This indicates that the complexity of the subproblems largely increases when the size of them increases.

The problem was also run with a *MAXTIME* in the subproblems set to 1200 seconds. A high maximum solution time in the subproblems was chosen because the subproblems are likely to be more difficult to solve when being larger. The model was run for 80 hours. When *MAXTIME* was set to 1200, no memory problems are encountered. However, columns sent from the subproblems are far from optimal. In many cases, the duality gap is above 100%, and also as high as 5000% during some iterations. At the point the program was stopped, 237 iterations from the RMP had been solved. Thus, the number of iterations would not be reduced by solving the subproblems as scenario clusters. In fact, comparing the clustered version with the non-clustered version that was let run for a maximum of 1200 seconds in the subproblems, it is seen that combining scenario clustering with column generation does not provide any benefits for the SMIP problem investigated.

10.3.4 Primal feasible solutions

All the tests described above were only conducted in the first node of the branch and price tree. The solutions provided are therefor not feasible with respect to the unit commitment decisions. Thus, the solutions provides only an upper bound to the feasible solutions. The branch and price procedure was therefore implemented in order to obtain feasible primal solutions from the column generation procedure. In addition two different heuristics have been tested.

Because the solution time of the branch and price tree is high in the first node, tests on the branch and price algorithm have only been run with a scenario tree consisting of 10 scenarios. The maximum solution time for the tests was set to 24 hours. During this time, 75 nodes were solved, but none of them provide a feasible solution to the problem. The change in the upper bound from the first to the last node is only 0.008% and the difference between the lowest objective value in a node and the first node is only 0.05%.

The branch and price algorithm alone is seen to not being able to provide feasible solutions within a reasonable amount of time. Two heuristics are therefore implemented that can be used to find integer feasible solutions. The feasible solutions can then be used in the branch and bound tree to enable pruning of nodes. However, looking at the small changes in the upper bound between the nodes in the branch and bound tree, it is reasonable to believe that nodes will not be pruned before many nodes have been processed.

Two heuristics are considered; one based on fixating the bid and one based on fixating the unit commitment decisions as discussed in Section 6.3. Feasible solutions with respect to the first stage variables can be obtained from the solution of the column generating procedure similarly as feasible unit commitment decisions from the Lagrangian decomposition can be obtained directly from the Lagrangian solution. Fixating the unit commitment decision is on the other hand done by rounding the unit commitment decisions to the closest value. To test the performance of these heuristics, five tests were conducted at different times during the column generation procedure. When fixating the unit commitment decisions the heuristic solution is found within a couple of minutes. When fixating the reserve bids the solution time is however considerably

longer due to the presence of binary unit commitment variables. One of the objective of the heuristic is to provide solutions within a short time-frame, hence the maximum solution time of the heuristic is set to 1 000 seconds. The results presented are thus the best feasible solutions found after 1 000 seconds. The results are provided in Table 10.10.

Table 10.10: MIP solutions from heuristics applied to Dantzig-Wolfe solutions

Time [s]	Objetive value; UC fixated [NOK]	Objective value; reserve bid fixated [NOK]
1 000	595 418	544 860
10 000	636 894	615 824
20 000	637 518	559 443
30 000	634 199	630 122
40 000	639 966	563 795

It is seen that the heuristic that rounds and fixates the unit commitment decisions performs better than the one that uses the target reserve bids directly. This is due to the fact that most of the unit commitment decisions returned from Dantzig-Wolfe are rather close to either 0 or 1, rounding these decisions is therefore a suitable approach in order to fulfill the binary requirements. This is somewhat analogous to the results from Lagrangian decomposition, where the aggregation of reserve bids outperformed using the unit commitments decisions directly.

It is also noteworthy that the heuristic solutions are not necessarily better even though they are constructed from solutions with better bounds. For further comparisons, the heuristic which rounds and fixates the unit commitment decisions is used.

10.4 Increasing the size of the subproblems

To investigate how the decomposition methods perform on larger test instances, not only with respect to scenarios, but with respect to reservoirs and generators, the decomposition methods have been tested on test instance 2 which contains above two times as many generators and reservoirs than test instance 1. Based on the discussion and the results in Chapter 9, the initial water value is set 80% higher than in the initial in these tests.

Without any limits on maximum time or relative gap, both decomposition methods run out of memory due to extensive branching in the subproblems when tests are run for test instance 2. For the Dantzig-Wolfe decompostion, the problem was also run with *MAXTIME* set to 300 in the subproblems, and this resulted in solution times of approximately 44 hours.

It has already been seen that when scenarios are bundled, the subproblems become too hard to solve. However, when the scenarios are bundled, non-anticipativity constraints are also present in the subproblems, increasing the complexity of them. When increasing the number of generators and reservoirs, the increase in complexity of the subproblems cannot be directly compared.

Seeing that the decomposition methods perform poorly with the larger test instance 2, and with scenario clusters for test instance 1 it is concluded that the scenariowise decomposition methods investigated in this thesis do not scale well with respect to the size of the subproblems. It is believed that with a distributed architecture, better performance would have been experienced

for these larger test instances but that further increases in the size of the subproblems would still be a problem. Due to these observations, further tests on test instance 2 are not executed.

10.5 Comparison of the decomposition methods

Given the results presented for test instance 1, it is clear that solving the SMIP problem by scenariowise decomposition contributes to finding better upper bounds compared to solving the deterministic equivalent by branch and bound. This is also illustrated in Figure 10.13, which shows the results from solving the SMIP model with the best-performing versions of the Lagrangian relaxation (proximal bundle) and Dantzig-Wolfe decomposition (stabilized version with *MIPRELSTOP*=300). The figure also includes results from solving the model with 40 and 100 scenarios since it is of interest to see how the decomposition methods perform when the number of scenarios is lower or higher. The upper bounds from all methods, the MIP solution from the branch and bound tree of the deterministic equivalent and the different best heuristic solutions are illustrated. The heuristic MIP solutions are derived from applying the best performing heuristic from Lagrangian decomposition and the Dantzig-Wolfe decomposition. The MIP solution from the deterministic equivalent is the best solution found within the same time frame or from the first node in the branch and bound tree if this is found later. MIP solutions found after 10 000 seconds are used for comparison since it is of interest for power producers to early find good feasible solutions during the solution procedure.

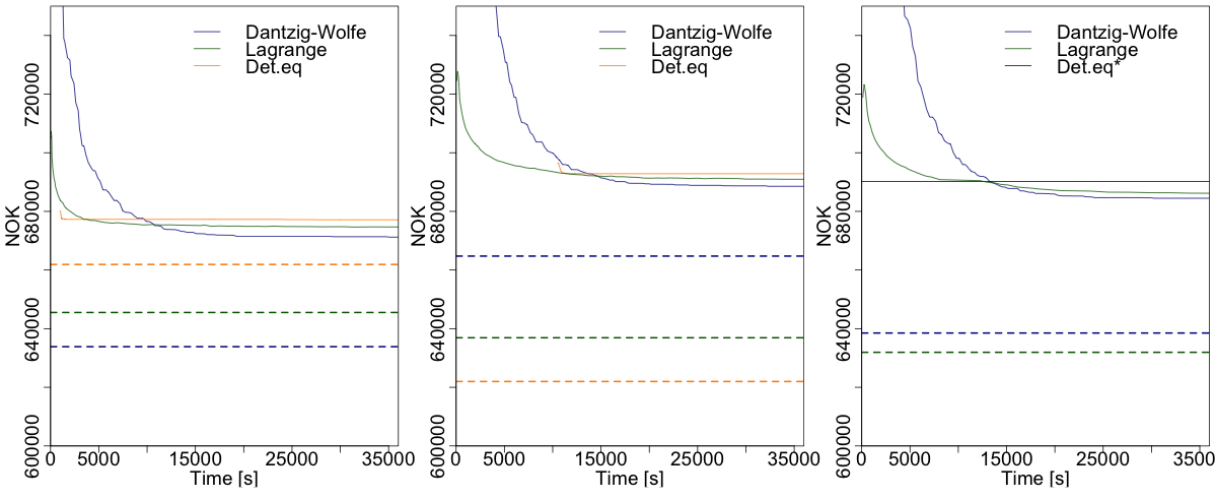


Figure 10.13: Comparison of Dantzig-Wolfe, Lagrange and deterministic equivalent with respect to upper bound given time. Best MIP solution from the respective methods using the best heuristic solution for the decomposition methods and the MIP solution from the branch and bound tree are illustrated with dotted lines For test instance with 100 scenarios, no solution of solving the deterministic equivalent is found within a solution time of 10 000 seconds; black line representing the best upper bound found after 12 hours.

It is clear that the decomposition methods outperform the standard branch and bound procedure by generating tighter optimistic bounds after a solution time of 10 hours for all scenario trees. It is also observed that the Dantzig-Wolfe decomposition converges more slowly than the Lagrangian relaxation, but results in a better upper bound after a given time. Note however that the bound from Lagrangian decomposition and Dantzig-Wolfe decomposition would be the

same if the SMIP problem had been solved solved to optimality with the decomposition methods as discussed in Chapter 3.

For the smallest scenario tree with 40 scenarios, the linear relaxation solution in the first node in the branch and bound tree is seen to be lower than the corresponding bound from both decomposition methods at the time being found. However, as time passes, both decomposition methods provide better upper bounds compared to the upper bound from solving the deterministic equivalent. When the number of scenarios is increased to 100, the deterministic equivalent does not manage to return an optimal linear relaxed solution nor a feasible solution within the time frame of 10 hours. These observations indicate that the decomposition methods are more valuable when including many scenarios, thus the scalability of the methods with respect to scenarios is clear.

It is optimal for a power producer to find an optimal solution of the multi-market bidding problem within a short time frame. Neither of the decomposition methods provide feasible solutions within 10 hours. The implemented heuristics are therefore used to recover feasible solutions such that we can get an idea of the gaps that are possible to achieve within a chosen time frame. This time frame has been chosen to be 10 000 seconds in the comparison presented. The heuristics implemented are shown to provide feasible solutions that are better than the MIP solution found from the branch and bound procedure for the test performed on the scenario tree with 70 and 100 scenarios. For the scenario tree with 100 scenarios the heuristics return feasible solutions when the branch and bound procedure does not return any feasible solutions. Combining decomposition methods with heuristics in such a case is considered interesting because the combination of decomposition methods and heuristics then provide a valid upper bound to feasible solutions at all times during the solution procedure. The heuristics implemented are not advance and show unstable behaviour in that the quality of the solutions they find do not seem to correspond to the quality of the upper bound solutions they are derived from. We consequently need more efficient heuristic that can translate solutions from the decomposition methods into high quality feasible solutions.

10.6 Practical value of scenariowise decomposition

For a hydropower producer the primary reserve bid should be scheduled within a reasonable amount of time, typically within a couple of hours (Klæboe, 2015a). Results from the tests conducted and presented in this chapter show that it is not possible to find exact optimal solutions of the problem with the implemented decomposition methods, within the strict time limit. The results from the simple heuristics applied after 10 000 seconds did provide reasonably good feasible solutions and it is believed that the quality of these solutions could be improved with better heuristics or by combining constructive heuristics with local search heuristics.

Solving the problem with a decomposition method results in a valid upper bound early in the solution process that can be used as a benchmark for evaluating the quality of potential heuristic solutions found. These upper bounds can also facilitate earlier stop of the search for better feasible solutions compared to solving the standard formulation of the problem to find feasible solutions. This is especially evident when including large scenario trees in the model. The deterministic equivalent does not find the optimal solution for the linear relaxation, hence providing no indication of the upper bound of the solution.

Through the analysis, it is seen that the scenariowise decomposition methods scale well with respect to scenarios. This facilitates the inclusion of a large number of scenarios that manage to

capture the uncertainty well, having positive impact on the reality representation of the model. On the contrary, the methods do not scale well with respect to the size of the subproblems. The solution times of the decomposition methods are seen to be driven by the complexity of the subproblems. The solution times may vary significantly between scenarios, depending on the realized prices in that particular scenario. One scenario may become the limiting factor, resulting in long solution times for the overall problem.

When testing the decomposition methods with larger problem instance Xpress runs out of memory before a solution can be found. These problems are due to the large number of binary variables in the subproblems associated with the unit commitment decisions. The memory issues can however be different if the decision maker has access to a fully distributed architecture. Still, the issues related to scalability with respect to the complexity of the subproblems remain.

As indicated by the analysis of the EVPI and VSS, the value of solving the SMIP problem presented is assumed to be larger when solving the two test instances together compared to solving them separately. Hence, including a large share of the portfolio (optimally the whole portfolio) is seen as beneficial for a producer who wishes to use the SMIP model presented in practice. This however increases the size of the subproblems resulting in high solution times or no possibility of finding a solution.

Modeling measures could be taken to reduce the complexity of the model with respect to the unit commitment decisions. One possibility is to relax the binary restriction on the unit commitment variables, hence solving the relaxed problem. This however results in the solution obtained in the first stage not being implementable in the second stage. This can however be taken care of by for example optimizing the second stage decisions over again when the uncertainty has been revealed. Another possibility is to include the unit commitment decision in an earlier stage in the stochastic model, hence avoiding the problem of mixed-integer recourse. Note that the reality representation of the model most probably will be affected by the above mentioned changes, but that the possible reduction in solution time might be worth the trade-off. It is outside the scope of this thesis to investigate the possible gain of implementing these measures, but future research within the area is encouraged.

Overall, it can be concluded that solution time is high for all decomposition methods, and in most cases longer than accepted if applied as decision support for power producers. Using the decomposition methods to find valid upper bounds is however considered valuable, since it can contribute to validate possible heuristic solutions found. In addition, early solutions from the decomposition methods may provide good starting points that can be used by heuristics or other techniques to recover feasible solutions.

Chapter 11

Conclusion

In this thesis, a model for optimal bidding in the sequential primary reserve market and day-ahead market has been investigated. The problem investigated is a large-scale stochastic program with continuous first stage and mixed binary second stage variables. Stochastic problems with these characteristics are characterized as complex, and few problem-specific solution procedures exist. A comprehensive modeling framework has been developed and implemented in the thesis to investigate scenariowise decomposition of the problem. This framework consists of price forecasting, scenario generation and implementation of the scenariowise decomposition methods.

For the work done on input data, our main contribution has been the development of a procedure for generating joint price scenarios for hourly day-ahead prices and weekly primary reserve prices. This was accomplished by combining time series forecasting for reserve and day-ahead prices with a scenario generation process based on PCA and moment matching. Day-ahead prices are modeled by an ARMA model, while primary reserve prices have been modeled with an ARMAX model dependent on the expected prices in the day-ahead market. The results of the scenario generation procedure shows that the scenarios capture the price characteristics of electricity prices well, resulting in realistic scenarios.

Two scenariowise decomposition methods have been successfully implemented; Lagrangian and Dantzig-Wolfe decomposition. Scenariowise decomposition makes the problem scalable in the number of scenarios, mitigating part of the computational difficulty associated with solving large problem instances when the number of scenarios increases. Both decomposition methods outperform the linear programming relaxation by providing better upper bounds on the optimal objective value within the maximum time limit of 10 hours. However, the solution time required for the methods to return relatively good bounds are outside the time available in most bidding situations.

The Lagrangian relaxation has been combined with different types of multiplier updating schemes to determine the best fit for the problem at hand. Due to high level of oscillations and instability, the proximal bundle method performed best. Lagrangian relaxation combined with this multiplier updating scheme provides better upper bounds than the linear programming relaxation within reasonable time. These bounds can be used to achieve better gaps when combined with non-optimal integer solutions from solving the standard formulation of the deterministic equivalent or from heuristic solutions.

The solutions from the Lagrangian relaxation fulfill integer requirements, but violate the non-anticipativity constraints. They can be converted into feasible solutions by making use of con-

structive heuristics. Tests performed with two simple constructive heuristics however show that it is hard to control the quality of the solutions. Thus, a remaining issue is still to develop heuristics or find other techniques to recover good feasible solutions from the solutions to the Lagrangian relaxed problem.

The Dantzig-Wolfe decomposition also suffers from long solution times. This is due to both long solution times of the subproblems and instability of the column generation procedure. Different methods have been used to speed up the solution process. These are somewhat successful, but it is believed that there is potential for improving the stabilization methods. If one wishes to pursue the use of use column generation to solve the bidding problem, alternative stabilization techniques should be considered to see if the number of iterations can be reduced. Because the proximal bundle method has shown stability in the Lagrangian decomposition, a stabilization approach based on a bundle method could also be interesting to investigate.

The performance of the decomposition methods is characterized by subproblems that are hard to solve, symmetry, high dimensionality and many unit commitment decisions in terms of binary variables. This results in little scalability with respect to size of the subproblems. The unit commitment part of the subproblems is particularly difficult, resulting in long solution times for the overall problem. When the size of the subproblems grows, not only the solution times are an obstacle, but also memory problems are encountered due to the generation of large branch and bound trees during the solution process. This is seen for the cases where more constraints and integer variables are included in the subproblems (watercourse 2), when subproblems are combined into larger subproblems (scenario clustering) and when the subproblems become non-linear (standard progressive hedging algorithm).

To sum up, the value of the decomposition methods is not to use them to solve the problem to optimality. Instead, early solutions from the decomposition methods may provide good starting points that can be used in combination with heuristics or other techniques to recover feasible solutions. In addition, the decomposition methods can be used to find valid upper bounds, outperforming the bounds from the linear programming relaxation, that can contribute to validate the quality of these solutions. More work should therefore be done on finding techniques for generating good feasible solutions. Another interesting future area of research is to investigate reformulations of the problem that contribute to reduce solution-times while at the same time reatain the realism of the model.

Bibliography

Alexander, C. (2008). *Practical Financial Econometrics*, volume II. Chichester: John Wiley & Sons.

Allen, E. H. and Ilić, M. D. (2000). Reserve markets for power systems reliability. *IEEE Transactions on Power Systems*, 15(1):228–233.

Attaviriyanupap, P., Kita, H., Tanaka, E., and Hasegawa, J. (2005). New bidding strategy formulation for day-ahead energy and reserve markets based on evolutionary programming. *International Journal of Electrical Power & Energy Systems*, 27(3):157–167.

Bakken, B., Bjørkvoll, T., Belsnes, and Skåre, P. (2001). Start/stopp-kostnader for vannkraftverk. Technical Report TR A5351, Trondheim: SINTEF Energiforskning.

Baldick, R., Grant, R., and Kahn, E. (2004). Theory and application of linear supply function equilibrium in electricity markets. *Journal of Regulatory Economics*, 25(2):143–167.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Beasley, J. E. (1996). *Advances in Linear and Integer Programming*. New York: Oxford University Press, Inc.

Ben Amor, H. M., Desrosiers, J., and Frangioni, A. (2009). On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167–1184.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252.

Bertsekas, D. P. (2011). Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010:1–38.

Birge, J. R. (1997). State-of-the-art-survey-stochastic programming: Computation and applications. *INFORMS Journal on Computing*, 9(2):111–133.

Birge, J. R. and Louveaux, F. (2011). *Introduction to Stochastic Programming*. New York: Springer Science & Business Media.

Boomsma, T. K., Juul, N., and Fleten, S.-E. (2014). Bidding in sequential electricity markets: The Nordic case. *European Journal of Operational Research*, 238(3):797–809.

Borghetti, A., Frangioni, A., Lacalandra, F., and Nucci, C. A. (2003). Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems*, 18(1):313–323.

Boyd, S., Xiao, L., and Mutapcic, A. (2003). Subgradient methods. *Lecture Notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005.

Brooks, R. and Geoffrion, A. (1966). Letter to the Editor - Finding Everett's Lagrange Multipliers by Linear Programming. *Operations Research*, 14(6):1149–1153.

Carøe, C. C. and Schultz, R. (1998). A two-stage stochastic program for unit commitment under uncertainty in a hydro-thermal power system. Preprint SC 98–11, Berlin: Konrad-Zuse-Zentrum für Informationstechnik.

Carøe, C. C. and Schultz, R. (1999). Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1):37–45.

Carøe, C. C. and Tind, J. (1998). L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83(1-3):451–464.

Catalão, J., Mariano, S., Mendes, V., and Ferreira, L. (2010). Nonlinear optimization method for short-term hydro scheduling considering head-dependency. *European Transactions on Electrical Power*, 20(2):172–183.

Conejo, A. J., Carrión, M., and Morales, J. M. (2010). *Decision Making Under Uncertainty in Electricity Markets*, volume 1. New York: Springer.

Conejo, A. J., Nogales, F. J., and Arroyo, J. M. (2002). Price-taker bidding strategy under price uncertainty. *IEEE Transactions on Power Systems*, 17(4):1081–1088.

Dantzig, G. B. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, 8(1):101–111.

David, A. K. and Wen, F. (2000). Strategic bidding in competitive electricity markets: a literature survey. 4:2168–2173.

Dicorato, M., Forte, G., Trovato, M., and Caruso, E. (2009). Risk-constrained profit maximization in day-ahead electricity market. *IEEE Transactions on Power Systems*, 24(3):1107–1114.

Doorman, G. L. (2015). *Hydro Power Scheduling*, volume 1. Trondheim: Norwegian University of Science and Technology.

Du Merle, O., Villeneuve, D., Desrosiers, J., and Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics*, 194(1):229–237.

Eriksrud, A. L. and Braathen, J. (2014). Spot market bidding taking the balancing power market into account. Master's thesis, Trondheim: Norwegian University of Science and Technology.

Escudero, L. F., Garín, M. A., Pérez, G., and Unzueta, A. (2013). Scenario cluster decomposition of the Lagrangian dual in two-stage stochastic mixed 0–1 optimization. *Computers & Operations Research*, 40(1):362–377.

Eurelectric/ENTSO-E (2011). Deterministic frequency deviation root causes and proposals for potential solutions. *Report, December 2011*.

Fan, Y. and Liu, C. (2010). Solving stochastic transportation network protection problems using the progressive hedging-based method. *Networks and Spatial Economics*, 10(2):193–208.

Faria, E. and Fleten, S.-E. (2011). Day-ahead market bidding for a nordic hydropower producer: taking the elbas market into account. *Computational Management Science*, 8(1-2):75–101.

Feng, X. and Liao, Y. (2006). A new lagrangian multiplier update approach for lagrangian relaxation based unit commitment. *Electric Power Components and Systems*, 34(8):857–866.

Fjelldal, B., Nafstad, S. M., and Klæboe, G. (2014). Optimal day-ahead electricity market bidding considering different ancillary services. *11th International Conference on the European Energy Market (EEM)*.

Fleten, S.-E. and Kristoffersen, T. K. (2007). Stochastic programming for optimizing bidding strategies of a nordic hydropower producer. *European Journal of Operational Research*, 181(2):916–928.

Fosso, O. B. and Belsnes, M. M. (2004). Short-term hydro scheduling in a liberalized power system. *International Conference on Power System Technology*, 2:1321–1326.

Frangioni, A. (2005). About Lagrangian methods in integer optimization. *Annals of Operations Research*, 139(1):163–193.

Geman, H. and Roncoroni, A. (2006). Understanding the fine structure of electricity prices. *The Journal of Business*, 79(3):1225–1261.

Geoffrion, A. M. (1974). *Lagrangean Relaxation for Integer Programming*. Berlin: Springer.

Geoffrion, A. M. and Marsten, R. E. (1972). Integer programming algorithms: A framework and state-of-the-art survey. *Management Science*, 18(9):465–491.

Helseth, A., Mo, B., Fodstad, M., and Hjelmeland, M. N. (2015). Co-optimizing sales of energy and capacity in a hydropower scheduling model. In *PowerTech Eindhoven 2015*, pages 1–6. Eidenhoven: IEEE Press.

Høyland, K., Kaut, M., and Wallace, S. W. (2003). A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2-3):169–185.

Høyland, K. and Wallace, S. W. (2001). Generating scenario trees for multistage decision problems. *Management Science*, 47(2):295–307.

Huisman, R., Huurman, C., and Mahieu, R. (2007). Hourly electricity prices in day-ahead markets. *Energy Economics*, 29(2):240–248.

Hydro (2016). Om Hydro. `http://www.hydro.com/no/Hydro-i-Norge/Om-Hydro/`. [Accessed 07-June-2016].

Hyndman, R. J. and Athanasopoulos, G. (2014). 8.7 ARIMA modelling in R. `https://www.otexts.org/fpp/8/7`. [Accessed 16-April-2016].

IEA (2012). Technology Roadmap - Hydropower. `https://www.iea.org/publications/freepublications/publication/2012_Hydropower_Roadmap.pdf`. [Accessed 29-May-2016].

Janczura, J., Trück, S., Weron, R., and Wolff, R. C. (2013). Identifying spikes and seasonal components in electricity spot price data: A guide to robust modeling. *Energy Economics*, 38:96–110.

Jiang, D. R. and Powell, W. B. (2015). Optimal hour-ahead bidding in the real-time electricity market with battery storage using approximate dynamic programming. *INFORMS Journal on Computing*, 27(3):525–543.

Kaut, M. (2003). Scenario-generation codes. `http://work.michalkaut.net/downloads.html`. [Accessed 16-April-2016].

Kaut, M. (2014). A copula-based heuristic for scenario generation. *Computational Management Science*, 11(4):503–516.

Kaut, M. and Wallace, S. W. (2007). Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271.

Kelley, J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712.

King, A. J. and Wallace, S. W. (2012a). *Modeling with stochastic programming*. New York: Springer.

King, A. J. and Wallace, S. W. (2012b). Scenario-Tree Generation: With Michal Kaut. In *Modeling with Stochastic Programming*, pages 77–102. New York: Springer.

Kiwiel, K. (1990). Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46:105–122.

Kjølle, A. (1980). *Vannkraftmaskiner*. Oslo: Universitetsforlaget.

Klæboe, G. (2015a). Private communication.

Klæboe, G. (2015b). *Stochastic Short-term Bidding Optimisation for Hydro Power Producers*. PhD thesis, Trondheim: Norwegian University of Science and Technology.

Kwon, R. H. and Frances, D. (2012). Optimization-based bidding in day-ahead electricity auction markets: A review of models for power producers. In *Handbook of Networks in Power Systems I*, pages 41–59. Berlin: Springer.

Laporte, G. and Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142.

Lemaréchal, C. (2001). *Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions*. London: Springer-Verlag.

Løkketangen, A. and Woodruff, D. L. (1996). Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming. *Journal of Heuristics*, 2(2):111–128.

Lorubio, G. (2011). Flexible generation: Backing up renewables. *Union of the Electricity Industry - EURELECTRIC*, page 26.

Lubin, M., Martin, K., Petra, C. G., and Sandıkçı, B. (2013). On parallelizing dual decomposition in stochastic integer programming. *Operations Research Letters*, 41(3):252–258.

Lulli, G. and Sen, S. (2004). A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. *Management Science*, 50(6):786–796.

Lundgren, J., Rönnqvist, M., and Värbrand, P. (2012). *Optimization*, volume 1. Lund: Studentlitteratur.

Nord Pool (2016a). Historical Market Data. `http://www.nordpoolspot.com/historical-market-data/`. [Accessed 19-February-2016].

Nord Pool (2016b). Trading. `http://www.nordpoolspot.com/TAS/`. [Accessed 21-February-2016].

Nowak, M. P. and Römisch, W. (2000). Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. *Annals of Operations Research*, 100(1-4):251–272.

Ntaimo, L. and Sen, S. (2007). A branch-and-cut algorithm for two-stage stochastic mixed-binary programs with continuous first-stage variables. *International Journal of Computational Science and Engineering*, 3(3):232–241.

Nygreen, B., Christiansen, M., Haugen, K., Bjørkvoll, T., and Kristiansen, Ø. (1998). Modeling Norwegian petroleum productionand transportation. *Annals of Operations Research*, 82:251–268.

Padhy, N. P. (2004). Unit commitment-a bibliographical survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205.

Redondo, N. J. and Conejo, A. (1999). Short-term hydro-thermal coordination by Lagrangian relaxation: solution of the dual problem. *IEEE Transactions on Power Systems*, 14(1):89–95.

Rockafellar, R. T. and Wets, R. J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147.

Schulze, T., Grothey, A., and McKinnon, K. (2015). A stabilised scenario decomposition algorithm applied to stochastic unit commitment problems. Technical report, Edinburgh: The University of Edinburgh, School of Mathematics.

Schütz, P., Tomasgard, A., and Ahmed, S. (2009). Supply chain design under uncertainty using sample average approximation and dual decomposition. *European Journal of Operational Research*, 199(2):409–419.

Sen, S. (2005). Algorithms for stochastic mixed-integer programming models. *Handbooks in Operations Research and Management Science*, 12:515–558.

Sen, S. and Higle, J. L. (2005). The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: set convexification. *Mathematical Programming*, 104(1):1–20.

Sen, S., Higle, J. L., and Ntaimo, L. (2003). A summary and illustration of disjunctive decomposition with set convexification. In *Network Interdiction and Stochastic Integer Programming*, pages 105–125. New York: Springer.

Sen, S. and Sherali, H. D. (2006). Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223.

Sherali, H. D. and Fraticelli, B. M. (2002). A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1-4):319–342.

Shiina, T. and Birge, J. R. (2004). Stochastic unit commitment problem. *International Transactions in Operational Research*, 11(1):19–32.

Statnett (2015). Vilkår for tilbud, aksept, rapportering og avregning i marked for primærreserver til Statnett. [Accessed 15-February-2016].

Statnett (2016). Reservemarkeder. `http://www.statnett.no/Drift-og-marked/Markedsinformasjon/`. [Accessed 15-February-2016].

Takriti, S., Krasenbrink, B., and Wu, L. S.-Y. (2000). Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. *Operations Research*, 48(2):268–280.

Trück, S., Weron, R., Wolff, R., et al. (2007). Outlier treatment and robust approaches for modeling electricity spot prices. *Proceedings of the 56th Session of the ISI. Available at MPRA: http://mpra. ub. uni-muenchen. de/4711*.

Ugedo, A., Lobato, E., Franco, A., Rouco, L., Fernandez-Caro, J., and Chofre, J. (2006). Strategic bidding in sequential electricity markets. *IEE Proceedings of Generation Transmission and Distribution*, 153(4):431–442.

Van Slyke, R. M. and Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663.

Vanderbeck, F. (2005). Implementing mixed integer column generation. In *Column generation*, pages 331–358. Boston: Springer.

Ventosa, M., Baıllo, A., Ramos, A., and Rivier, M. (2005). Electricity market modeling trends. *Energy policy*, 33(7):897–913.

Wallace, S. W. and Fleten, S.-E. (2003). Stochastic programming models in energy. *Handbooks in Operations Research and Management Science*, 10:637–677.

Watson, J.-P., Gade, D., Hackebeil, G., Ryan, S. M., Wets, R. J., and Woodruff, D. L. (2013). Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. Technical report, Albuquerque: Sandia National Laboratories (SNL-NM).

Watson, J.-P. and Woodruff, D. L. (2011). Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370.

Wei, W. W.-S. (2006). *Time Series Analysis*, volume 2. Boston: Addison-Wesley.

Wen, F. and David, A. (2001). Strategic bidding for electricity supply in a day-ahead energy market. *Electric Power Systems Research*, 59(3):197–206.

Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4):1030–1081.

Zenios, S. and Ziemba, W. (2006). *Handbook of Asset and Liability Management: Theory and Methodology*. Handbook of Asset and Liability Management. Oxford: Elsevier Science.

Zheng, Q. P., Wang, J., Pardalos, P. M., and Guan, Y. (2013). A decomposition approach to the two-stage stochastic unit commitment problem. *Annals of Operations Research*, 210(1):387–410.

Zhu, X. (2006). *Discrete two-stage stochastic mixed-integer programs with applications to airline fleet assignment and workforce planning problems*. PhD thesis, Virginia: Virginia Polytechnic Institute and State University.

Appendix A

Correlation matrices

Table A.1: Correlation matrix of primary reserve prices

	Week-N	Week-D	Week-E	Weekend-N	Weekend-D	Weekend-E
Week-N	1.000					
Week-D	0.899	1.000				
Week-E	0.924	0.968	1.000			
Weekend-N	0.872	0.832	0.834	1.000		
Weekend-D	0.866	0.863	0.849	0.954	1.000	
Weekend-E	0.854	0.831	0.809	0.950	0.983	1.00

.

Table A.2: Correlation matrix between primary reserve and day-ahead prices.

		Day-ahead					
		Week-N	Week-D	Week-N	Weekend-N	Weekend-D	Weekend-E
P	Week-N	-0.175	-0.131	-0.084	-0.228	-0.133	-0.112
	Week-D	-0.127	-0.093	-0.046	-0.180	-0.098	-0.075
	Week-E	-0.094	-0.060	-0.013	-0.153	-0.068	-0.043
	Weekend-N	-0.141	-0.101	-0.040	-0.224	-0.110	-0.088
	Weekend-D	-0.196	-0.152	-0.101	-0.271	-0.174	-0.150
	Weekend-E	-0.215	-0.171	-0.119	-0.288	-0.187	-0.169

Table A.3: Correlation matrix of day-ahead prices.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000																							
2	0.995	1.000																						
3	0.987	0.997	1.000																					
4	0.980	0.991	0.997	1.000																				
5	0.974	0.985	0.992	0.996	1.000																			
6	0.972	0.978	0.980	0.983	0.991	1.000																		
7	0.952	0.952	0.949	0.950	0.960	0.982	1.000																	
8	0.862	0.859	0.857	0.858	0.867	0.890	0.933	1.000																
9	0.793	0.790	0.787	0.787	0.795	0.817	0.867	0.966	1.000															
10	0.826	0.820	0.815	0.813	0.820	0.839	0.885	0.961	0.984	1.000														
11	0.859	0.851	0.845	0.842	0.847	0.865	0.904	0.953	0.957	0.987	1.000													
12	0.885	0.877	0.869	0.865	0.869	0.886	0.920	0.936	0.927	0.963	0.986	1.000												
13	0.908	0.899	0.891	0.886	0.891	0.908	0.938	0.934	0.910	0.948	0.974	0.993	1.000											
14	0.915	0.907	0.899	0.895	0.900	0.917	0.945	0.932	0.901	0.937	0.965	0.985	0.995	1.000										
15	0.917	0.910	0.902	0.898	0.904	0.922	0.948	0.931	0.897	0.932	0.960	0.979	0.992	0.995	1.000									
16	0.914	0.907	0.900	0.896	0.901	0.919	0.945	0.926	0.894	0.928	0.956	0.976	0.988	0.991	0.996	1.000								
17	0.870	0.863	0.857	0.854	0.860	0.876	0.902	0.901	0.889	0.923	0.947	0.961	0.964	0.962	0.969	0.976	1.000							
18	0.806	0.799	0.794	0.790	0.795	0.810	0.844	0.888	0.883	0.915	0.936	0.935	0.937	0.935	0.940	0.948	0.968	1.000						
19	0.816	0.808	0.803	0.799	0.803	0.817	0.852	0.897	0.891	0.922	0.941	0.935	0.937	0.935	0.939	0.945	0.947	0.979	1.000					
20	0.885	0.876	0.870	0.865	0.868	0.881	0.905	0.913	0.884	0.911	0.934	0.941	0.951	0.954	0.955	0.959	0.945	0.946	0.972	1.000				
21	0.946	0.935	0.927	0.921	0.923	0.936	0.953	0.911	0.862	0.893	0.921	0.944	0.961	0.968	0.969	0.969	0.936	0.899	0.915	0.967	1.000			
22	0.959	0.948	0.940	0.933	0.935	0.947	0.956	0.895	0.837	0.869	0.901	0.926	0.947	0.955	0.958	0.956	0.919	0.868	0.882	0.946	0.993	1.000		
23	0.965	0.954	0.946	0.938	0.940	0.950	0.952	0.874	0.809	0.844	0.878	0.906	0.930	0.940	0.943	0.940	0.899	0.839	0.851	0.922	0.981	0.994	1.000	
24	0.970	0.962	0.955	0.949	0.949	0.955	0.949	0.863	0.795	0.829	0.863	0.892	0.917	0.927	0.931	0.928	0.885	0.821	0.833	0.906	0.969	0.985	0.994	1.000

Appendix B

ARMA and ARMAX model parameters

Table B.1: Parameters for day-ahead ARMA model, 00:00-08:00.

Hour	Param.	Estimate	Std. error	Log-lik	t-value	p-value
00:00-01:00	ϕ_1	0.0192	0.1064	967.51	0.180	0.852
	ϕ_2	0.3396	0.0886	967.51	3.833	0.0001
	ϕ_3	0.5718	0.0725	967.51	7.887	0.0001
	Θ_1	0.6311	0.1019	967.51	6.193	0.0001
	Θ_2	0.1377	0.0706	967.51	1.95	0.054
	Θ_4	-0.3789	0.0332	967.51	11.41	0.001
01:00-02:00	ϕ_1	0.9757	0.0072	717.27	135.51	0.0001
	Θ_1	-0.4000	0.0295	717.27	13.56	0.0001
	Θ_2	-0.1552	0.0282	717.27	5.50	0.0001
02:00-03:00	ϕ_1	0.0847	0.0536	592.1	1.58	0.1144
	ϕ_2	0.8684	0.0518	592.1	16.76	0.0001
	Θ_1	0.5183	0.0589	592.1	8.80	0.0001
	Θ_2	-0.5053	0.0375	592.1	13.47	0.0001
	Θ_3	-0.2047	0.0297	592.1	6.89	0.0001
03:00-04:00	ϕ_1	0.9733	0.0077	545.61	126.40	0.0001
	Θ_1	-0.3673	0.0293	545.61	12.54	0.0001
	Θ_2	-0.1906	0.0281	545.61	6.78	0.0001
04:00-05:00	ϕ_1	0.9766	0.0073	469.71	133.78	0.0001
	Θ_2	-0.4048	0.0293	469.71	13.82	0.0001
	Θ_3	-0.2029	0.0284	469.71	7.14	0.0001
05:00-06:00	ϕ_1	0.9790	0.0069	599.56	141.88	0.0001
	Θ_2	-0.3862	0.0289	599.56	13.36	0.0001
	Θ_3	-0.2291	0.0279	599.56	8.21	0.0001
06:00-07:00	ϕ_1	0.1196	0.0566	736.46	2.11	0.0351
	ϕ_2	0.8457	0.0557	736.46	15.18	0.0001
	Θ_1	0.5161	0.0645	736.46	8.00	0.0001
	Θ_2	-0.6228	0.0307	736.46	20.29	0.0001
	Θ_3	-0.2353	0.0377	736.46	6.24	0.0001
07:00-08:00	ϕ_1	0.4802	0.0287	491.69	16.73	0.0001
	ϕ_2	-0.3912	0.0559	491.69	7.00	0.0001
	ϕ_3	0.8316	0.0381	491.69	21.83	0.0001
	Θ_1	0.0045	0.0424	491.69	0.1	0.992
	Θ_2	0.4323	0.0686	491.69	6.3	0.0001
	Θ_3	-0.5471	0.0389	491.69	14.06	0.0001

Table B.2: Parameters for day-ahead ARMA model, 08:00-16:00.

Hour	Param.	Estimate	Std. error	Log-lik	t-value	p-value
08:00-09:00	ϕ_1	0.1037	0.0503	534.23	2.06	0.0396
	ϕ_2	0.8578	0.0495	534.23	17.33	0.0001
	Θ_1	0.4692	0.0582	534.23	8.06	0.0001
	Θ_2	-0.6830	0.0308	534.23	22.17	0.0001
	Θ_3	-0.2387	0.0360	534.23	6.63	0.0001
09:00-10:00	ϕ_1	0.4888	0.0637	727.76	7.67	0.0001
	ϕ_2	0.7467	0.0817	727.76	9.14	0.0001
	ϕ_3	-0.2594	0.0517	727.76	5.017	0.0001
	Θ_1	0.1027	0.0529	727.76	1.94	0.0526
	Θ_2	-0.7058	0.0493	727.76	14.31	0.0001
10:00-11:00	c	-0.0030	0.0586	862.95	0.05	0.9601
	ϕ_1	0.1280	0.1726	862.95	0.74	0.4594
	ϕ_2	0.8698	0.0528	862.95	16.47	0.0001
	ϕ_3	-0.0365	0.1365	862.95	0.26	0.7949
	Θ_1	0.4405	0.1641	862.95	2.68	0.0075
	Θ_2	-0.6057	0.0669	862.95	0.05	0.9601
	Θ_3	-0.1445	0.0805	862.95	1.79	0.0737
11:00-12:00	ϕ_1	1.3365	0.0530	1075.85	2.52	0.0001
	ϕ_2	-0.3504	0.0505	1075.85	6.94	0.0001
	Θ_1	-0.7163	0.0398	1075.85	18.00	0.0001
12:00-13:00	c	0.0170	0.0478	1142.07	0.05	0.961
	ϕ_1	2.1930	0.0127	1142.07	172.677	0.0001
	ϕ_2	-2.1524	0.0221	1142.07	97.39	0.0001
	ϕ_3	0.9373	0.0132	1142.07	71.01	0.0001
	Θ_1	-1.6161	0.0300	1142.07	53.87	0.0001
	Θ_2	1.4126	0.0403	1142.07	35.05	0.0001
	Θ_4	-0.4078	0.0287	1142.07	14.20	0.0001
14:00-15:00	ϕ_1	1.1906	0.1232	1031.54	9.66	0.0001
	ϕ_2	-0.2076	0.1195	1031.54	1.74	0.0821
	Θ_1	-0.5287	0.1234	1031.54	4.28	0.0001
	Θ_2	-0.1290	0.0632	1031.54	2.04	0.0416
15:00-16:00	c	-0.0025	0.0597	967.48	0.0418	0.9967
	ϕ_1	0.3492	-	967.48	-	-
	ϕ_2	0.7629	-	967.48	-	-
	ϕ_3	-0.1425	0.0884	967.48	1.61	- 0.1007
	Θ_1	0.329	-	967.48	-	-
	Θ_2	-0.5757	-	967.48	-	-
	Θ_3	-0.1516	-	967.48	-	-

Table B.3: Parameters for day-ahead ARMA model, 16:00-24:00.

Hour	Param.	Estimate	Std. error	Log-lik	t-value	p-value
16:00-17:00	c	0.0259	0.0508	1045.07	0.51	0.61601
	ϕ_1	2.1990	0.0109	1045.07	201.74	0.0001
	ϕ_2	-2.1682	0.0180	1045.07	120.46	0.0001
	ϕ_3	0.9466	0.0115	1045.07	82.31	0.0001
	Θ_1	-1.6291	0.0317	1045.07	51.39	0.0001
	Θ_2	1.4176	0.0444	1045.07	31.93	0.0001
	Θ_3	-0.4010	0.0311	1045.07	12.86	0.0001
17:00-18:00	ϕ_1	2.1345	0.0550	1013.61	38.81	0.0001
	ϕ_2	-1.7266	0.0858	1013.61	20.12	0.0001
	ϕ_3	0.5759	0.0425	1013.61	13.55	0.0001
	Θ_1	-1.4613	0.0542	1013.61	26.96	0.0001
	Θ_2	0.718	0.045	1013.61	15.96	0.0001
18:00-19:00	ϕ_1	1.4053	0.0606	1066.89	23.19	0.0001
	ϕ_2	-0.4191	0.0576	1066.89	7.28	0.0001
	ϕ_3	-0.7227	0.0475	1066.89	15.2147	0.0001
19:00-20:00	ϕ_1	0.0117	0.0222	1164.31	0.527	0.5983
	ϕ_2	0.9348	0.0214	1164.31	43.68	0.0001
	ϕ_3	0.7098	0.0362	1164.31	19.61	0.0001
	Θ_1	-0.4147	0.0384	1164.31	10.80	0.0001
	Θ_2	-0.1919	0.0304	1164.31	6.31	0.0001
21:00-22:00	ϕ_1	0.0328	0.0273	1376.6	1.20	0.2304
	ϕ_2	0.9242	0.0264	1376.6	35.00	0.0001
	ϕ_3	0.7415	0.0388	1376.6	19.11	0.0001
	Θ_1	-0.3616	0.0378	1376.6	9.57	0.0001
	Θ_2	-0.2055	0.0286	1376.6	7.19	0.0001
22:00-23:00	ϕ_1	0.9788	0.0066	1432.18	148.30	0.0001
	Θ_1	-0.2082	0.0296	1432.18	7.03	0.0001
	Θ_2	-0.1230	0.0289	1432.18	4.26	0.0001
	Θ_3	-0.0497	0.0276	1432.18	1.80	0.0721
23:00-24:00	ϕ_1	0.2998	0.1719	1466.44	1.74	0.0821
	ϕ_2	0.6660	0.1674	1466.44	3.98	0.0001
	Θ_1	0.4697	0.1700	1466.44	2.76	0.0001
	Θ_2	-0.2509	0.0513	1466.44	4.89	0.0001
	Θ_3	-0.1619	0.0283	1466.44	5.72	0.0001

Table B.4: Parameters for primary reserves ARMAX model

Block	Param.	Estimate	Std. error	Log-lik	t-value	p-value
Week-night	d	1.5954	6.2316	-167.63	0.26	0.7952
	γ_1	1.5954	0.1882	-167.63	8.48	0.0001
	γ_2	-0.9205	0.4482	-167.63	2.05	0.0419
	γ_3	0.2257	0.2656	-167.63	0.85	0.3965
	Γ_1	-1.1396	0.1404	-167.63	4.54	0.0001
	Γ_2	0.7429	0.2509	-167.63	2.96	0.0035
	α_1	-0.4721	0.2513	-167.63	1.88	0.0618
Week-day	d	4.8794	1.4536	-156.19	3.36	0.010
	γ_1	0.0836	0.2815	-156.19	0.30	0.7645
	γ_2	0.4090	0.1519	-156.19	2.69	0.0078
	γ_3	0.2281	0.1828	-156.19	1.25	0.2130
	Γ_1	0.2881	0.2646	-156.19	1.09	0.2772
	Γ_2	0.4441	0.1300	-156.19	3.42	0.0008
	α_1	-0.2828	0.2603	-156.19	1.09	0.2772
Week-evening	d	4.9436	1.8405	-167.31	2.69	0.0078
	γ_1	0.5162	0.2386	-167.31	2.16	0.0322
	γ_2	0.3279	0.1926	-167.31	1.70	0.0909
	Γ_1	-0.1491	0.2374	-167.31	0.63	0.5295
	Γ_2	0.2917	0.1193	-167.31	2.45	0.0153
	Γ_3	-0.1564	0.1325	-167.31	1.18	0.2396
	α_1	-0.2817	0.3363	-167.31	0.84	0.4021
Weekend-night	d	4.9433	1.1572	-174.02	4.27	0.0001
	γ_1	1.7517	0.1476	-174.02	11.87	0.0001
	γ_2	-0.8069	0.1303	-174.02	6.19	0.0001
	Γ_1	-1.1894	0.1463	-174.02	2.86	0.0048
	Γ_2	0.4284	0.1008	-174.02	4.25	0.0001
	α_1	-0.1778	0.2164	-174.02	0.82	0.4134
Weekend-day	d	5.5867	1.7485	-177.54	3.20	0.0016
	γ_1	0.8884	0.0433	-177.54	20.5	0.0001
	Γ_2	-0.3440	0.0844	-177.54	4.08	0.0001)
	α_1	5.5867	1.7485	-177.54	3.20	0.0016
Weekend-evening	d	6.2549	1.8782	-178.18	3.33	0.0011
	γ_1	0.8856	0.0431	-178.18	20.55	0.0001
	Γ_2	-0.3348	0.0836	-178.18	4.00	0.0001
	α_1	6.2549	1.8782	-178.18	3.33	0.0012

Appendix C

Linear approximation of P-Q curves

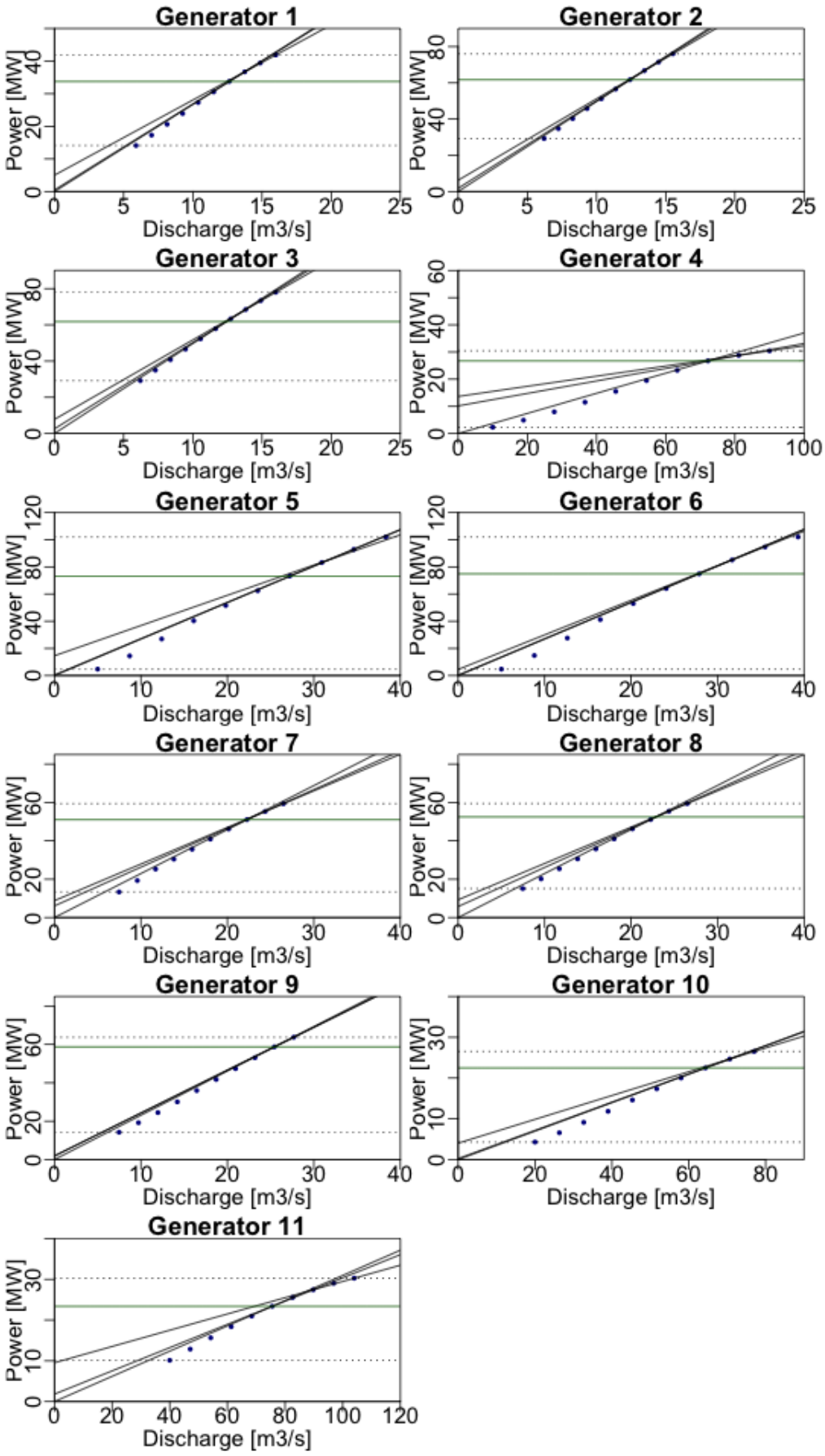


Figure C.1: Linearization of P-Q curves.