THE UNIVERSITY *of* EDINBURGH

Stochastic Programming for Hydro-Thermal Unit Commitment

THE UNIVERSITY
of EDINBURGH

Tim Schulze

Supervisor:

Prof. Ken McKinnon

Doctor of Philosophy

The University of Edinburgh

2015

Stochastic Programming for Hydro-Thermal Unit Commitment

Doctoral dissertation

Tim Schulze

School of Mathematics
The University of Edinburgh

Tim Schulze
E-mail: *t.schulze-2@sms.ed.ac.uk*

Supervisor: Prof. Ken McKinnon
E-mail: *k.mckinnon@ed.ac.uk*

Examiners:
Dr. Chris Dent (Durham University)
Prof. Jacek Gondzio (Edinburgh University)

School of Mathematics
The University of Edinburgh
James Clerk Maxwell Building
The King's Buildings
Peter Guthrie Tait Road
Edinburgh, EH9 3FD
United Kingdom

Doctoral dissertation
Initial Submission: July 2015
Final Submission: October 2015
Typeset in LaTeX

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise. Chapters 3 and 4 are based on publications [1] and [2] of which I am the principal author. This work has not been submitted for any other degree or professional qualification.

Karlsruhe, October 13, 2015
Place, Date

Tim Schulze

Lay Abstract

In recent years the liberalisation of energy markets and expansion of renewable energy supplies has increased the uncertainty in operational planning of power systems. Volatile and unpredictable wind power supplies have significant effects on the way conventional power plants are used, because power infeed and demand must be balanced at all times to operate the system safely. Most conventional power plants must be notified three to twelve hours before they become available to generate electricity. To be able to buffer a sudden unpredicted loss of wind power, the system balancing authorities require conventional plants to run part-loaded so that they can respond quickly. Otherwise customers have to be switched off to restore the balance in the system. Running the system with large amounts of part-loaded power plants is costly, but so is switching off customers. To find the cheapest power plant schedule that will permit to operate the power system safely for a given wind forecast while satisfying all demand, system operators use unit commitment models, a type of mathematical optimization model. Traditionally, these models did not include an explicit model of wind uncertainty, but in recent years such models have become more popular. We quantify the added value of using an uncertainty model by performing a two-year evaluation with both, the traditional and new scheduling approaches. For this evaluation we use a model of the British power system in 2020, with a 30% share of wind energy in terms of installed capacity. As planning problems with an explicit representation of uncertainty are much harder to solve than the traditional ones, we derive a dedicated solution methodology for them. It is based on decomposing the problem into multiple smaller problems of the traditional type.

Abstract

In recent years the deregulation of energy markets and expansion of volatile renewable energy supplies has triggered an increased interest in stochastic optimization models for thermal and hydro-thermal scheduling. Several studies have modelled this as stochastic linear or mixed-integer optimization problems. Although a variety of efficient solution techniques have been developed for these models, little is published about the added value of stochastic models over deterministic ones. In the context of day-ahead and intraday unit commitment under wind uncertainty, we compare two-stage and multi-stage stochastic models to deterministic ones and quantify their added value. We show that stochastic optimization models achieve minimal operational cost without having to tune reserve margins in advance, and that their superiority over deterministic models grows with the amount of uncertainty in the relevant wind forecasts. We present a modification of the WILMAR scenario generation technique designed to match the properties of the errors in our wind forcasts, and show that this is needed to make the stochastic approach worthwhile. Our evaluation is done in a rolling horizon fashion over the course of two years, using a 2020 central scheduling model of the British National Grid with transmission constraints and a detailed model of pump storage operation and system-wide reserve and response provision.

Solving stochastic problems directly is computationally intractable for large instances, and alternative approaches are required. In this study we use a Dantzig-Wolfe reformulation to decompose the problem by scenarios. We derive and implement a column generation method with dual stabilisation and novel primal and dual initialisation techniques. A fast, novel schedule combination heuristic is used to construct an optimal primal solution, and numerical results show that knowing this solution from the start also improves the convergence of the lower bound in the column generation method significantly. We test this method on instances of our British model and illustrate that convergence to within 0.1% of optimality can be achieved quickly.

Acknowledgements

First and foremost I would like to thank my PhD supervisor Prof. Ken McKinnon. Ken, you have been an incredible mentor for me throughout the past four years. You have the inquisitive mind of a true researcher and you have taught me the perseverance and perfectionism required to successfully complete a research project. You always made time for long problem discussions and creative brainstorming sessions, and you were involved in my research project with the utmost dedication. You have created a relaxed, yet highly productive work environment and your motivation has kept me afloat at times where mine was faltering. I could not have hoped for a better supervisor and I am very grateful for four interesting, challenging and successful years.

I would also like to thank Dr. Andreas Grothey who contributed ideas, techniqes and solutions in the development phase of the algorithmic methodology. Additionally, I would like to thank Dan Eager and Ian Pope from AF Mercados EMI in Edinburgh for helping to assemble data of the British power system, Peter Kelen from PowerOP for providing feedback on the British model, Samuel Hawkins and Gareth Harrison from the School of Engineering at Edinburgh University for providing the wind data, and Paul Plumptre for explaining National Grid's balancing mechanism and modelling approach. I acknowledge funding obtained through the Principal's Career Development Scholarship scheme of the University of Edinburgh. Prof. Jacek Gondzio served as mentor on this programme and advised me on any academic matters unrelated to my research project. Dr. Chris Dent and Prof. Jacek Gondzio agreed to be my examiners in the final stages of my PhD, and I am very grateful for that. My interest in optimization was fostered by three interesting years spent at the Karlsruhe Institute of Technology, where Prof. Oliver Stein and Dr. Paul Steuermann taught me the basics of optimization and set me on my current career path, and I would like to thank them for that.

Finally but most importantly I would like to thank Annina for her unconditional love and support, particularly in the last months of the write-up phase. Annina, your warm Spanish heart has given me strength at times where I needed it most, while your Finnish honesty and modesty have kept me true to myself and others.

Contents

1

Introduction

1.1 Motivation

For many years, problems in the electric power industry have posed challenges to state of the art solvers and stimulated development of new optimization techniques. Optimization models for efficient planning, scheduling and operation of power systems often involve integer or nonlinear decision variables which make them very challenging. Power systems planning and optimization models are typically classified according to the length of their planning horizon.

- **Long Term Models** are used to devise investment strategies for generation plant or transmission gear, or to analyse the impact of incentive schemes on capacity expansion. Their time horizon covers up to 25 years.
- **Medium Term Models** are concerned with resource management and have a time scale of 1 to 3 years. A typical application is hydro reservoir scheduling, i.e. scheduling of pump storages with yearly storage and inflow cycles.
- **Short Term Models** are used for a variety of tasks at time scales varying from a few minutes up to a week. Unit commitment models are used for week-ahead, day-ahead or intraday power plant scheduling, while economic dispatch models calculate power outputs a few minutes ahead and optimal power flow or load flow models determine physical aspects of power transmission and distribution on a minute-by-minute basis.

The model granularity increases with decreasing time horizon, as models become more and more detailed and include physical aspects of the underlying power system. In this study, we focus on the aspects of hydro-thermal generation unit commitment (UC) at the day-ahead and intraday time scale. The objective of the UC problem is to find a cost-minimal schedule for all available power plants, which will permit to satisfy the demand for electricity while maintaining sufficient reserve and response to operate the system safely in the event of a failure or demand fluctuation. To this end, it aims to find optimal timings for startup and shutdown actions of individual generation units. The nature of the problem is combinatorial, and a plethora of techniques have been proposed to solve it. Most solution methods are based on heuristic search or mathematical programming, or combine both in a hybrid approach. Popular choices of mathematical programming techniques for the UC problem are Lagrangian and augmented Lagrangian relaxation, Dynamic Programming and more recently also Branch & Cut and Progressive Hedging. Heuristic search methods which are often applied in this context are Genetic Algorithms, Simulated Annealing, Particle Swarm Optimization and Artificial Neural Networks.

Traditionally, UC problem formulations did not include a model of the transmission or distribution network, leaving the economic dispatch and optimal power flow stages to deal with this issue. To ensure that the commitments permit network-feasible operation of the power system, the problem was altered i.e. by defining must-run units. However, more recently Branch & Cut solvers have become more powerful, allowing practitioners to include approximate linear models of the power grid in their mixed-integer programming (MIP) formulations of the UC problem. The exact formulation of a UC problem depends on the regulatory regime of the electricity market for which it is used. In regulated market environments it is typically formulated for the whole system, with the aim of minimising the cost of electricity generation to the economy. In a deregulated market, on the other hand, generation companies use UC models to schedule their generation assets so as to achieve maximum profitability. Depending on the market structure, UC models can also be used for market clearing purposes, by deciding on the acceptance or rejection of generation bids. This is common practice in the United States, where independent system operators (ISO) run autonomous electricity markets.

In recent years, there has been an increasing interest in UC formulations which incorporate uncertainty in the model. The traditional source of uncertainty in a regulated market is the electricity demand. However, in many regions customer behaviour is well-predictable, so demand forecasts are quite accurate. Models without explicit representations of demand uncertainty were deemed appropriate if there was a sufficent reserve margin which could be used to buffer small fluctuations. Modern electricity markets, however, are largely deregulated, and modern power systems are being extended continuously to integrate increasing amounts of renewable energy. This introduces new sources of uncertainty to power systems planning and operation in general, and the UC problem in particular:

- **Market price uncertainty** affects generation companies which aim to maximise their profitability in a deregulated market.
- **Renewable supply uncertainty** affects generation, transmission and distribution and companies. Supplies from renewables are weather dependent and diffcult to forecast, and can cause network disruptions if they fluctuate strongly.

These developments have triggered an interest in applying stochastic optimization techniques. In particular, there has been a growing interest in the formulation of stochastic unit commitment (SUC) problems and dedicated solution techniques. Various case studies have modelled this problem as two-stage model with (mixed) integer recourse [3, 4] or as multi-stage model [5, 6, 7]. These models are computationally challenging, especially if the required number of scenarios is large, and dedicated solution techniques are necessary to solve them. This has motivated research into algorithmic strategies for the efficient solution of SUC problems. Besides the necessity for an efficient solver, another important question is to characterise and quantify the added value of stochastic models over previous, deterministic ones. These are the two central questions which we address in this study.

1.2 Previous Research

In the following we briefly review relevant research on solution techniques for SUC models. In the remainder of this study, the focus is on mathematical programming

techniques, in particular decomposition methods. However, a lot of these also incorporate heuristic search methods to find good feasible solutions quickly. In the last paragraph we review articles which address the issue of quantifying the added value of stochastic models over deterministic ones.

Model Formulation. In the past years there have been improvements in the formulation of MIP models for unit commitment. Multiple ways have been proposed to formulate the problem, some of them using one set of binary variables i.e. on-off variables [8], two sets of binary variables where the additional set are startup variables [4], or three sets of binaries where the third set are shutdown variables [9]. Ostrowski et al demonstrated that formulations with less binary variables are not necessarily more efficient in a computational sense, because modern MIP solvers can exploit integrality restrictions to derive cuts which tighten LP relaxations, and profit from branching on the additional variables [9]. Fuel costs are often modelled as linear or piecewise linear (PWL) functions. When PWL models are used they have less detrimental effects on solver performance if they are convex and require no additional binary variables [8, 10]. Additional performance improvements were achieved through the use of tighter valid inequalities. Rajan and Takriti [11] found inequalities for minimum up/down times of individual generation units which tighten the linear programming (LP) relaxation. In the space of on-off and startup variables, their formulation defines facets of the polytope described by the minimum up/down constraints. More recently, Jiang et al [12] showed that the facet-defining quality of these cuts also extends to the stochastic case, and report on their experience with other types of cutting planes devised specifically for multi-stage stochastic formulations.

Decomposition. Despite efficient reformulations and improvements in general MIP software, even deterministic unit commitment is a challenging problem, and this has motivated the application of decomposition techniques. The traditional decomposition approach for UC problems is by generation units. Individual units are bundled by the demand and reserve constraints, and if those are relaxed, e.g. via Lagrangians [5] or augmented Lagrangians [13], the problem becomes separable by generators. Single unit subproblems can be solved efficiently by dynamic programming, or stochastic dynamic

programming if the original model is a stochastic model. Both, Lagrangian relaxation (LR) and Dantzig-Wolfe (DW) decomposition have been proposed to decompose SUC problems by generation units [6, 14]. The number of linking demand and reserve constraints which have to be dualised grows linearly in the number of included scenarios. The Lagrangian dual problem is often solved by a cutting plane or proximal bundle method [3, 5, 6], in which case Lagrangian decomposition becomes the dual equivalent of column generation (ColGen) or bundle column generation [15], respectively.

An alternative to decomposition by generation units is decomposition by scenarios. If the non-anticipativity constraints are relaxed, the problem becomes separable by scenarios. This requires a model where non-anticipativity is formulated explicitly through constraints rather than implicitly by sharing variables among scenarios, which can lead to a substantial amount of redundancy if applied to two-stage day-ahead UC models as in [3]. Scenario decomposition yields deterministic UC problems as subproblems, and any solution technique suitable to them can be used as subproblem solver, e.g. embedded Lagrangian decomposition by generation units as proposed in [7] or standard mixed integer programming techniques. Various approaches have been proposed to achieve separability by scenarios. Carøe and Schultz [3] perform Lagrangian relaxation of non-anticipativity constraints in a two-stage model and solve the dual problem by a proximal bundle method.

Takriti and Birge [7] apply Progressive Hedging [16], which was originally developed for convex stochastic programs and remains a heuristic when applied in the mixed-integer case. Ryan et al [17], Gade et al [18] and Watson and Woodruff [19] describe various modifications of the Progressive Hedging algorithm and report on associated improvements in mixed-integer applications, in particular the UC problem. They devise techniques to identify feasible solutions and improve primal convergence, and ways of bounding the problem from below. These techniques enable fast progress even in large scale UC applications with realistic test systems. However, there is no guarantee that the optimal primal solution will be found or the lower bound will be tight. Progressive Hedging remains a heuristic when applied in the mixed-integer case, albeit a very successful one.

Since LR and DW methods solve the dual problem or, equivalently, a convex relaxation of the primal problem, they do not necessarily find a primal feasible solution

either, and additional work is required to achieve that. Gröwe-Kuska et al [5, 6] use various Lagrangian based heuristics and an additional Economic Dispatch stage to find solutions with small duality gaps. Shiina and Birge [14] use a schedule reduction technique to generate integer solutions from a restricted master problem. Additionally, since UC problems have integer variables, there may be a duality gap, in which case a branching scheme such as Branch & Price is needed to verify that a global solution has been found. Carøe and Schultz [3] embed their dual algorithm in a branching scheme and use a primal rounding heuristic to produce intermediate solutions and accelerate convergence. Beside the quality of the primal solution, the lower bound obtained from the decomposition method plays a central role in achieving fast convergence. Dentcheva and Römisch [20] show that the lower bound obtained through Lagrangian scenario decomposition is at least as good as the lower bound obtained through decomposition by generation units, which provides a strong motivation for the former approach.

An alternative way of decomposing the problem via LR or DW decomposition is by stages or nodes, however this is not popular in SUC problems. An overview of the three ways of decomposing the problem, i.e. by units, scenarios or stages, is given in Römisch and Schultz [21]. Another stage-wise decomposition approach was developed specifically for two-stage problems: Benders decomposition, or the L-Shaped Method was originally devised for problems with continuous recourse but was subsequently generalised for integer recourse models [22]. Zheng et al [23] apply Benders decomposition to decompose two-stage SUC problems. Unlike other two-stage formulations [3, 4], theirs has binary variables on both, the first and second stages.

Finally, Goez et al [4] review two-stage and multi-stage stochastic formulations of the unit commitment problem and test various solution techniques: LR as in [3], Progressive Hedging as in [7] and a heuristic based on successively decreasing the number of relaxed variables in an LP relaxation. A comprehensive review of decomposition techniques for energy optimization problems in general is also given in [24].

Stochastic vs Deterministic Models. Substantial research efforts have gone into developing fast solution methods for MIP based SUC models. Despite that, comparatively little has been published about the added value of stochastic scheduling models over deterministic ones. In the literature, there are two different approaches to evaluate

the expected cost of UC schedules so that they can be compared with one another:

1. Evaluation via Monte-Carlo simulation: for the given schedule, a dispatch solution is calculated on a large number of day-long sample paths generated from a simulator that is thought to represent reality. This is typically done for a set of representative days, e.g. one day per season of the year. The performance of different schedules is measured by their expected dispatch cost.

2. Rolling horizon evaluation: a rolling scheduling and dispatch procedure is defined in which the system is scheduled for a few hours and evaluated against a historic trajectory by a dispatch model. Following the evaluation, the next few hours are scheduled and the process is repeated. Performance is measured by the dispatch cost on the historic trajectory. This is sometimes referred to as time domain scheduling simulation.

A major disadvantage of the Monte-Carlo simulation approach is that it is not possible to be certain whether the simulator is a correct representation of reality. Also, inter-temporal constraints such as minimum up- and downtimes cannot be considered beyond the end of the simulated day. These shortcomings are avoided in the rolling horizon approach.

The following studies use Monte-Carlo simulation to evaluate UC schedules. Ruiz et al [25] report on an evaluation of deterministic and two-stage stochastic UC under load and generator failure uncertainty, using the IEEE reliability test system [26]. Papavasiliou and Oren [27] apply Lagrangian relaxation and Benders decomposition to solve two-stage stochastic problems with uncertain wind production and security constrained problems with contingency scenarios. They compare different formulations with respect to fuel cost and security of supply by evaluating a typical spring day in the California ISO test system. Constantinescu et al [28] include wind scenarios obtained from a numerical weather prediction model in a two-stage stochastic model. They evaluate this against a deterministic model, using three days of wind data from Illinois and a ten generator test system.

The following studies perform a rolling horizon evaluation of UC schedules. Tuohy et al [29] apply the WILMAR model [30] to data of the Irish electricity system and perform a one year rolling evaluation of deterministic and multi-stage SUC. They re-

port savings between 0.25% and 0.9% when using a stochastic approach instead of a deterministic one, depending on the length of the first stage. However, the authors use perfect information on the first stage, which biases the solutions to become better if the length of the first stage is extended. Additionally, the problems are only solved to an optimality tolerance of 1%. Sturt and Strbac [31] report on the difference between deterministic and stochastic rolling planning in a thermal power system with high wind penetration and a given level of storage capacity, which represents the British (GB) power sytem in 2030. However, mainly continuous relaxations of integer models are used, and transmission network issues arising from the geographical disparity of wind, storage and conventional generation are not addressed.

1.3 Objective

In this study, we address two central issues that arise in relation with the application of stochastic optimization models to the UC problem:

- **Efficient solution techniques** are required to solve these problems. Such methods have to scale well in the number of included scenarios. We derive an efficient scenario decomposition algorithm with dual initialisation and stabilisation procedures and primal heuristics for accelerated convergence. The algorithm is tested on a realistic model based on the GB power system. We vary the number of scenarios and show that the method scales well in the problem size.

- **The added value of stochastic optimization** models over deterministic ones is characterised by performing a long-term rolling horizon evaluation of the two approaches on our model of the GB power system with a significant amount of uncertain wind power.

The solution technique and our evaluation approach are briefly outlined below. To ensure that our findings with respect to the performance of the decomposition algorithm and the comparison of stochastic and deterministic UC are viable for models of realistic size and detail, we assembled data for a model of the GB power system with an aggregated representation of the grid and a detailed representation of pump storage plants. Our model is a central scheduling model with transmission restrictions between

network areas. Demand and generation are balanced locally by areas, but reserve and response are treated as system-wide quantities. The data for conventional generation and wind farms correspond to National Grid's figures for 2020 under the Gone Green Scenario, with an overall wind penetration of 30% in terms of installed capacity.

Our solution approach is based on DW scenario decomposition and can be applied in the two-stage and multi-stage case. A generic framework for this type of decomposition is described by Lulli and Sen [32], and a stochastic version of this algorithm for convex continuous multi-stage stochastic programs has recently been proposed in Higle and Sen [33]. Non-anticipativity constraints are separated from the remaining constraints by dealing with them in the master problem. They are formulated in terms of additional variables which represent the bundled scenarios' common decisions. Applying this formulation allows dual information to be spread flexibly among a scenario bundle. We use a proximal bundle method [34] to stabilise the master problem and accelerate convergence. Additionally, we derive a dual initialisation procedure and construct primal feasible solutions using a fast, novel schedule combination heuristic which uses the schedules generated by the ColGen procedure. Although theoretically, in order to guarantee optimality, the ColGen approach needs to be embedded in a Branch & Price framework, we demonstrate that for the problems considered, sufficiently small duality gaps can be achieved without branching. We apply our algorithm to both, two-stage and multi-stage SUC problems and report on its computational performance on our model of the GB power system.

The performance of stochastic and deterministic UC approaches is compared in the context of both, day-ahead and intraday scheduling. We use a two-stage stochastic model for day-ahead scheduling and a multi-stage stochastic model for intraday scheduling. In the intraday context we consider two cases: one where commitments of large conventional plant can be revised every three hours, and another one where they can be revised every six hours. The study is performed in a rolling horizon fashion over an evaluation period of two years. We characterise stochastic and deterministic schedules, show how they differ from one another, and quantify the savings achieved with stochastic scheduling. While these models are computationally challenging, the savings achieved with them are typically a small percentage of the overall cost, implying the necessity of small optimality tolerances. These are achieved by applying our scenario

decomposition method.

1.4 Structure

The remainder of this study is structured as follows. Chapter 2 provides a brief explanation of stochastic optimization methodology in general and stochastic UC models in particular. Additionally, it provides a brief review of the techniques that can be used to decompose SUC models by scenarios. We start with an introduction to stochastic programming, recourse models, the non-anticipativity property, and how it can be written as a constraint to permit the application of decomposition methods. This is followed by a brief review of LR methodology: we outline how LR is used for lower bounding and decomposition, and what solution techniques exist for the dual problem. Then we proceed with an overview of DW decomposition: we recapitulate Dantzig and Wolfe's decomposition principle for linear problems, and how it can be generalised for the mixed-integer case. DW decomposition has close ties with LR, and we show how they are used to cross-apply techniques developed for either one of the methods. Finally, we discuss Progressive Hedging (PH) as an alternative scenario decomposition method. We close the chapter with a simple example and brief discussion of MIP UC models.

Chapter 3 provides a more extensive discussion of scenario decomposition. First, we state a simple ColGen algorithm for SUC problems. Then we explain how dual regularisation and initialisation can be used to stabilise and hot-start this method, and derive a primal heuristic to find good solutions quickly and accelerate convergence. The heuristic is based on generator schedule exchange between scenarios. Finally, we describe our implementation of this method and discuss results of a performance comparison of decomposition and out-of-the-box Branch & Bound when applied to two-stage and multi-stage versions of our model of the GB power system. Numerical results demonstrate how time savings through decomposition increase rapidly with the number of scenarios included in the SUC model.

Chapter 4 is concerned with the long-term evaluation of deterministic and stochastic scheduling approaches. We provide an algebraic statement of our GB power system model and a description of the various data sources. After discussing error statistics

of the wind forecasts used for this evaluation, we give a brief outline of the scenario generation, reduction and scenario tree construction techniques that were used to generate the wind input data. The rolling horizon evaluation process is then described in more detail, before we proceed with an overview of its results. The chapter closes with a discussion of characteristic traits of stochastic and deterministic schedules and the added value of stochastic scheduling.

Finally, in Chapter 5 we summarise the objective, methodology and results of this study and draw our conclusions.

2

Methods & Models

This chapter reviews some of the theory that we rely upon in the remainder of our study. It gives a brief introduction to stochastic optimization models, UC models, and the techniques that can be used to decompose them by scenarios. We introduce recourse models, the non-anticipativity property, and how it can be written as a constraint to permit scenario decomposition. Then we continue with a brief review of LR methodology, how it is used for lower bounding and decomposition, and what solution techniques exist for the dual problem. This is followed by an overview of DW decomposition: we explain Dantzig and Wolfe's decomposition principle and outline generalisations for the mixed-integer case. LR and DW decomposition are closely related and we show how they can both be used in the same context. Finally, we explain the PH algorithm for scenario decomposition of convex continuous problems, and then close the chapter with a brief discussion of MIP UC models.

2.1 Stochastic Programming

Stochastic programming is a technique that is widely used in applications where one or more of the parameters of an optimization model underlie uncertainty. It has its origin in a publication of George Dantzig [35], which first introduced the recourse model. A stochastic problem arises when at least one of its parameters is described as a random variable, resulting in a blend of an optimization model and a stochastic model. In the following sections we provide a brief introduction to two-stage and multi-stage recourse models and how they are used in modern applications. The contents of these sections are loosely based on [36, 37].

2.1.1 Two-Stage Recourse Problems

The type of stochastic optimization model we use for our UC problems is known as recourse model. The term recourse refers to the possibility to perform a corrective action after an initial decision has been taken and the outcome of the uncertain parameter has been observed. In a two-stage recourse model the sequence of events is *act-observe-react*, so there are two stages on which a decision is made:

1. The first stage decision is a central decision made under uncertainty.
2. The second stage or recourse decision is an individual response to every outcome of the uncertain parameter that is accounted for in the model.

Let S be the uncertain parameter of the underlying decision process. Then S is a random variable defined on a probability space $(\mathcal{S}, \mathcal{A}, P)$. To obtain a recourse problem that can be solved with linear or mixed-integer programming techniques, we require S to be a discrete random variable with a finite number of realizations $|\mathcal{S}|$. In many applications this means that the true continuous stochastic process underlying the uncertain parameter has to be approximated by a discrete one. The quality of the decision depends on the quality of this approximation, so a lot of research is concerned with finding appropriate approximations [38, 39]. Assuming that S is discrete, we can write $p_s := P(S = s)$ for all outcomes $s \in \mathcal{S}$. The outcomes are called scenarios. The general form of a two-stage recourse problem is

$$\begin{aligned} \min_{x} \quad & c^T x + \mathcal{Q}(x) \\ \text{s.t.} \quad & Ax = b,\ x \geq 0 \end{aligned} \tag{2.1}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$ are the first stage data and

$$\mathcal{Q}(x) := \mathbb{E}\left[q\left(x, S\right)\right] = \sum_{s \in \mathcal{S}} p_s q\left(x, s\right) \tag{2.2}$$

is the expected recourse function with

$$\begin{aligned} q\left(x, s\right) := \quad \min_{y_s} \quad & d_s^T y_s \\ \text{s.t.} \quad & W_s y_s = r_s - T_s x,\ y_s \geq 0, \end{aligned} \tag{2.3}$$

where $d_s \in \mathbb{R}^p$, $r_s \in \mathbb{R}^l$, $W_s \in \mathbb{R}^{l \times p}$ and $T_s \in \mathbb{R}^{l \times n}$ are the second stage data for every scenario $s \in \mathcal{S}$. The first stage decision is represented by the variable x and is identical for all scenarios s, while the recourse action is denoted as y_s and can be different under different scenarios. Despite the implicit assumption that decisions are made sequentially in time, the two-stage recourse problem is a specially structured linear program in which optimal values for x and y_s are found simultaneously. In a practical context, the important decision taken by this problem is the here-and-now decision x, which is informed by the what-if decisions y_s under all considered scenarios. An optimal decision x will have the property that it balances profit from good scenarios and vulnerability to bad scenarios. Modelling many bad scenarios in $\mathcal{S}$ acts like an insurance in that it helps to hedge against future risk, but it also increases the computational burden. Finding the right scenario set $\mathcal{S}$ is to strike the balance between modelling risk and keeping computational effort manageable.

In many applications not all second stage data will vary with the scenario. When the second stage constraint matrix is constant in s, $W_s = W$, the problem is said to have fixed recourse. If, beyond that, W is an identity matrix, the problem is said to have simple recourse. Other special types of recourse appear when stochastic programming is applied to integer or mixed-integer problems. In that case x and y_s may both contain integer decisions. Depending on whether y_s is continuous or (mixed-) integer, the problem is said to have continuous or (mixed-) integer recourse. For some dedicated stochastic solution techniques it matters what type of recourse problem (2.1) has. For instance, L-Shaped or Benders decomposition was initially developed to solve continuous recourse models and only later generalised to integer recourse models [36, 22].

When applying solution techniques which decompose the problem into a first stage and a second stage problem, an important question is whether there are values of x for which at least one of the second stage problems (2.3) is not feasible. Generally, if the recourse problem is feasible for any right-hand side, then the problem is said to have complete recourse:

$$\mathcal{Y}(s, r) := \{y_s : W_s y_s = r, y_s \geq 0\} \neq \emptyset \quad \forall r \in \mathbb{R},\ s \in \mathcal{S}. \tag{2.4}$$

Achieving this may not be practical in every model, and a weaker but similarly useful criterion can be used instead. The problem is said to have relatively complete recourse if the second stage problem is feasible for all right-hand sides which can be attained subject to feasibility of the first stage:

$$\mathcal{Y}(s,r) \neq \emptyset \quad \forall r \in \{r_s - T_s x : Ax = b, x \geq 0\}\,,\ s \in \mathcal{S}. \tag{2.5}$$

Both, complete and relatively complete recourse are sufficient to guarantee that the expected recourse function is finite, $\mathcal{Q}(x) < \infty$ [37]. The SUC models which we analyse in this study are examples of mixed-integer recourse models. We choose a formulation for them which ensures that they have relatively complete recourse.

2.1.2 Non-Anticipativity Constraints

The decision structure in a two-stage recourse problem is such that the first stage decision x has to be made before the true outcome of the uncertain parameter S can be observed. Our decision is then said to be non-anticipative because we cannot anticipate the future when making it. The non-anticipativity property is inherent in the model structure, as x cannot change based on the scenario s. An equivalent way of formulating the same problem is by imposing the non-anticipativity property explicitly through constraints. We first introduce additional variables x_s which depend on the scenario s, and then add non-anticipativity constraints which force them to be identical under all scenarios. To avoid confusion with the scenario specific variable x_s, we rename $\bar{x} := x$ in the new formulation. The resulting problem has the following structure:

$$\begin{aligned} \min_{x,y} \quad & \sum_{s\in\mathcal{S}} p_s \left(c^T x_s + d_s^T y_s\right) \\ \text{s.t.} \quad & \left.\begin{aligned} & Ax_s = b \\ & T_s x_s + W_s y_s = r_s \\ & x_s - \bar{x} = 0 \\ & x_s, y_s \geq 0 \end{aligned}\right\} \forall s \in \mathcal{S} \end{aligned} \tag{2.6}$$

The second stage copies x_s of first stage variables $\bar{x}$ and the non-anticipativity constraints $x_s = \bar{x}$ are obviously redundant, and if the problem is to be solved directly by

linear programming techniques it is likely to be less efficient than the original formulation. However, this formulation has other advantages. Firstly, in some applications, it can be an easier, more intuitive way to formulate the problem. This is often the case in multi-stage recourse problems which we introduce in the next section. Secondly, it permits the application of a different type of decomposition technique, because the structure of the constraint matrix has changed. First, consider the structure of the original problem: for a fixed first stage decision x, the recourse problem (2.3) can be solved individually for each scenario s. This is exploited e.g. by the L-Shaped method [22]. The structure of the corresponding constraint matrix is block-angular with binding columns (variables). Let the scenario set be $\mathcal{S} = \{1, \dots, \hat{s}\}$. Then the constraint matrix of the original formulation 2.1 is given by:

$$\begin{bmatrix} A & & & \\ T_1 & W_1 & & \\ \vdots & & \ddots & \\ T_{\hat{s}} & & & W_{\hat{s}} \end{bmatrix} \cdot \begin{bmatrix} x \\ y_1 \\ \vdots \\ y_{\hat{s}} \end{bmatrix} = \begin{bmatrix} b \\ r_1 \\ \vdots \\ r_{\hat{s}} \end{bmatrix}. \tag{2.7}$$

By reformulating the problem with non-anticipativity constraints, we have moved to a block-angular constraint matrix with binding rows (constraints). If we let

$$B_s := \begin{bmatrix} A & 0 \\ T_s & W_s \end{bmatrix}, \quad z_s := \begin{bmatrix} x_s \\ y_s \end{bmatrix}, \quad g_s := \begin{bmatrix} b \\ r_s \end{bmatrix} \quad \forall s \in \mathcal{S} \tag{2.8}$$

then, apart from the non-anticipativity constraints, the constraints of problem (2.6) can be written as $B_s z_s = g_s$, $\forall s \in \mathcal{S}$. The constraint matrix of the new formulation is given by:

$$\begin{bmatrix} B_1 & & & \\ & \ddots & & \\ & & B_{\hat{s}} & \\ N_1 & \dots & N_{\hat{s}} & N_0 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ \vdots \\ z_{\hat{s}} \\ \bar{x} \end{bmatrix} = \begin{bmatrix} g_1 \\ \vdots \\ g_{\hat{s}} \\ 0 \end{bmatrix}. \tag{2.9}$$

Here we assume that the matrices $N_1, \dots, N_{\hat{s}}$ and N_0 have appropriate structure so that the corresponding rows express the non-anticipativity constraints. Again, we can obtain separability by scenarios. However, this time it is achieved by relaxing the non-

anticipativity constraints, i.e. removing the binding rows from (2.9) rather than fixing the first stage variables x. Instead of solving the original problem we can then solve the following subproblems for each $s \in \mathcal{S}$:

$$\begin{aligned} \min_{x_s, y_s} \quad & p_s \left(c^T x_s + d_s^T y_s \right) \\ \text{s.t.} \quad & Ax_s = b \\ & T_s x_s + W_s y_s = r_s \\ & x_s, y_s \geq 0. \end{aligned} \tag{2.10}$$

In contrast to the recourse problem (2.3) which can be solved individually for each scenario if x is fixed, these subproblems have their own copies of first stage variables x_s. Lagrangian relaxation [40], Dantzig-Wolfe decomposition [41] or Progressive Hedging [16] can be applied to remove the non-anticipativity constraints and achieve separability in this way. The different approaches are briefly explained in Section 2.2. We have not specified $N_1, \ldots, N_{\hat{s}}$ and N_0 explicitly, because there are multiple ways of expressing non-anticipativity constraints. The formulation we used so far is applied e.g. in [32]. We refer to it as common target formulation because it uses redundant variables $\bar{x}$ that can be interpreted as target values which are common to all scenarios. Alternative formulations are available, which do not require redundant target variables: in PH the target variables $\bar{x}$ are replaced with the weighted average $\sum_{s \in \mathcal{S}} p_s x_s$. Alternatively, it is also possible to eliminate $\bar{x}$ by using the following chain formulation:

$$x_s = x_{s+1} \quad \forall s = 1, \ldots, \hat{s} - 1, \tag{2.11}$$

or a common target formulation without redundant variables:

$$x_s = x_{\hat{s}} \quad \forall s = 1, \ldots, \hat{s} - 1. \tag{2.12}$$

In the latter, we have chosen the last scenario to be the target for all other scenarios, but the choice is arbitrary: any scenario can be chosen as common target. LR or DW decomposition can be applied to the chain formulation or to either one of the common target formulations. The choice of non-anticipativity constraints determines

the number and meaning of their multipliers, but does not affect the dimension of the dual solution space. This is discussed in more detail in Chapter 3.

2.1.3 Multi-Stage Recourse problems

The two-stage recourse problems discussed so far followed the decision structure *act-observe-react*, where we can observe the outcome of the uncertain parameter once and then adapt our reaction. For cases where the uncertain parameter is monitored in regular intervals and decisions can be adapted regularly, the model can be extended to accommodate repeated observations and reactions. This is the structure underlying multi-stage recourse problems. The decision process is typically captured in a decision tree as shown in Figure 2.1, where a new stage is introduced every time new information becomes available. This assumes that the model follows discrete time steps $t = 1, \dots, T$, where stages can span more than one time step, i.e. the model's time grid can be finer than the intervals at which new information becomes available and decisions are adapted. The example tree in Figure 2.1 shows a single realisation of the uncertain parameter on the first stage, three on the second stage and five thereafter. Scenario s_3 only splits off once, indicating that the information becoming available at time t_3 does not affect it.

A multi-stage stochastic problem can be expressed easily using a scenario formulation. We assume that there is a decision vector x_{st} for every time step t and scenario s. Let c_{st} be the corresponding cost vector. Then the multi-stage stochastic program can be written as follows.

$$\begin{aligned} \min_{x} \quad & \sum_{s \in \mathcal{S}} p_s \sum_{t=1}^{T} c_{st}^T x_{st} \\ \text{s.t.} \quad & x_s \in \mathcal{X}_s \quad \forall s \in \mathcal{S} \\ & x_{st} - \bar{x}_{bt} = 0 \quad \forall b \in \mathcal{B},\ t \in \mathcal{T}_b,\ s \in \mathcal{S}_b \end{aligned} \tag{2.13}$$

Constraints on $x_s := (x_{st})\ \forall t = 1 \dots, T$ are expressed individually for each scenario through membership in the feasible sets $\mathcal{X}_s$. The non-anticipativity constraints $x_{st} = \bar{x}_{bt}$ take a similar form as before, but rely on the notion of bundles. Bundles can be defined in various ways, depending on the preferred data structure. Examples can be

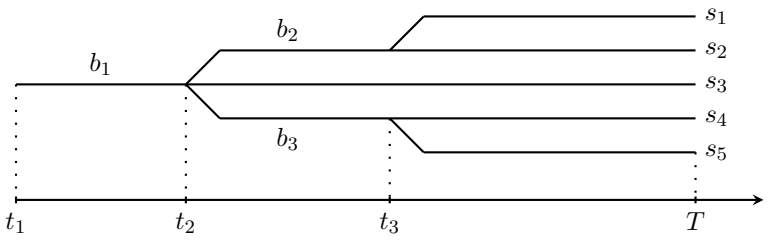


Figure 2.1: Scenario tree example with five leaves. The first stage covers periods $\mathcal{T}_{b_1} = \{t_1, \ldots, t_2\}$, with bundle b_1 that contains scenarios $\mathcal{S}_{b_1} = \{s_1, s_2, s_3, s_4, s_5\}$. On stage two with time steps $\mathcal{T}_{b_2} = \mathcal{T}_{b_3} = \{t_2, \ldots, t_3\}$ we have bundles b_2 and b_3 with $\mathcal{S}_{b_2} = \{s_1, s_2\}$ and $\mathcal{S}_{b_3} = \{s_4, s_5\}$. After that, all scenarios are independent.

found in [4, 37]. In our notation a bundle $b \in \mathcal{B}$ is specified by a tuple of sets $(\mathcal{T}_b, \mathcal{S}_b)$ which indicate the time periods and scenarios contained in the bundle, respectively. A vector of target variables, $\bar{x}_{bt}$, is introduced for every time step $t \in \mathcal{T}_b$ of bundle b, and non-anticipativity constraints require all variables of scenarios contained in the bundled scenario subset $\mathcal{S}_b$ to be equal to the target variables in this period. Figure 2.1 demonstrates how these data structures can be used to model a scenario tree for problem (2.13).

For both, two-stage and multi-stage recourse problems, a formulation which combines all scenarios in one large optimization problem is referred to as *extensive* formulation or deterministic equivalent (of the stochastic formulation). This can be a scenario formulation or a formulation with intrinsic non-anticipativity property. The term extensive formulation is used in the decomposition literature to distinguish between the original stochastic problem and other problems used in the decomposition procedure, i.e. single scenario subproblems and master problems which we introduce later.

Advanced Non-Anticipativity Models. The example tree in Figure 2.1 implies that a single scenario is used during the first time steps $\{t_1, \ldots, t_2\}$, and a unique solution is obtained for *all* variables by imposing corresponding non-anticipativity constraints at these time steps. This is a simplification which makes it easier to apply the underlying concept of iterated observe-and-react sequences to UC problems, and is common in the SUC literature [4]. However, in practical UC applications the scheduling decision for the first time steps $\{t_1, \ldots, t_2\}$ is made a few hours before t_1, and the sample space of the uncertain parameter cannot be approximated well by using a single

scenario between t_1 and t_2. At the same time not all decisions between t_1 and t_2 have to be decided in advance, and some of them can be modelled as recourse decisions. In the UC models discussed in Chapter 4, the first stage decision is a power plant schedule, but the model also includes recourse decisions such as operating levels and pump storage actions during the first time steps. The important decision to be made by this model is only the schedule. All recourse actions are re-decided by subsequent scheduling and dispatch models.

Therefore we will later model *multiple* realisations of the uncertain parameter during time steps $\{t_1, \ldots, t_2\}$. This results in as many solutions as there are distinct scenario bundles during the first time steps, and we have to introduce additional non-anticipativity constraints to make the first stage decision unique across all scenarios. This is explained in more detail in Chapter 4 where we introduce the formulation of our GB SUC model.

2.2 Decomposition

The idea of decomposition is nearly as old as linear programming [42]. Most optimization algorithms scale well up to a certain problem size, but beyond that it becomes very hard to solve the problem. Decomposition algorithms aim to alleviate the adverse effects of problem size by dividing it into smaller subproblems which are solved repeatedly until a sufficient approximation to an optimal solution of the original problem has been found. This is particularly interesting for (mixed-) integer optimization problems, which are NP-hard [43]. In this section we briefly outline Lagrangian relaxation and Dantzig-Wolfe decomposition and explain how they are related. We also briefly discuss Progressive Hedging, a scenario decomposition method for stochastic problems. The contents of this section are loosely based on [40, 41, 44, 16]. Another decomposition method which is frequently applied to two-stage recourse problems is Benders decomposition [45]. However, it results in a decomposition by stages rather than by scenarios, and the underlying idea is different from the methods presented here. Hence we omit it from our discussion.

2.2.1 Lagrangian Relaxation

Lagrangian relaxation is a versatile technique to find lower bounds for minimisation problems by solving a dual problem. It is often applied in mixed-integer optimization where good lower bounds are much sought-after. Consider the following general minimisation problem, called the primal problem.

$$\min_x f(x) \text{ s.t. } x \in \mathcal{X},\ h_j(x) = 0\ \forall j = 1, \dots, m \tag{2.14}$$

If $\mathcal{X}$ is a convex set, f is convex and h_j are affine functions, then the problem is said to be convex. Note that this can easily be extended to cover the case with additional inequalities $g_k(x) \leq 0$ which, in order for the problem to be convex, do not need to be affine but only convex [40, 46]. Lagrangian relaxation is typically applied when the feasible set $\mathcal{X}$ contains constraints subject to which this primal problem is easy to solve, while $h_j(x) = 0$ are complicating constraints which make it harder to solve.

Example 2.1 (Scenario Decomposition) *We have seen in (2.8) and (2.9) that the stochastic problem (2.6) has a block-diagonal constraint matrix with additional non-anticipativity constraints. So if we take*

$$\begin{aligned} \mathcal{X} &:= \{z = (x_1, y_1, \dots, x_{\hat{s}}, y_{\hat{s}})^T : B_s z_s = g_s\ \forall s \in \mathcal{S}\} \text{ and} \\ h_s(x) &:= x_s - \bar{x} \quad \forall s \in \mathcal{S} \end{aligned} \tag{2.15}$$

then h_s are the non-anticipativity constraints and without them the problem is separable by scenarios and easier to solve than it was originally.

Lagrangian relaxation removes the complicating constraints from the problem and introduces a linear price on violating them. The function

$$L(x, \lambda) := f(x) - \sum_{j=1}^{m} \lambda_j h_j(x) = f(x) - \lambda^T h(x) \tag{2.16}$$

with a price vector $\lambda \in \mathbb{R}^m$ is called the Lagrangian. The prices are also called multipliers or dual variables. The dual function is given by

$$\theta(\lambda) := \min_{x \in \mathcal{X}} L(x, \lambda). \tag{2.17}$$

To evaluate the dual function at a given point λ, we have to solve an optimization problem which is similar to the primal problem, but its complicating constraints have been removed and their violation is penalized for in the objective, using the given prices λ. The dual problem is then defined as

$$\max_{\lambda \in \mathbb{R}^m} \theta(\lambda), \tag{2.18}$$

and it follows immediately that for any primal feasible point $\bar{x} \in \mathcal{X}$ with $h(\bar{x}) = 0$ and any dual feasible point $\bar{\lambda} \in \mathbb{R}^m$ we have

$$\theta(\bar{\lambda}) = \min_x \left[f(x) - \bar{\lambda}^T h(x) \right] \leq f(\bar{x}) - \underbrace{\bar{\lambda}^T h(\bar{x})}_{=0} = f(\bar{x}). \tag{2.19}$$

This means that solving the dual problem provides a lower bound on the optimal solution of the primal problem, that is,

$$\max_{\lambda \in \mathbb{R}^m} \theta(\lambda) \leq \min_{x \in \mathcal{X}: h(x)=0} f(x). \tag{2.20}$$

This property is known as weak duality. The difference

$$\Delta := \min_{x \in \mathcal{X}: h(x)=0} f(x) - \max_{\lambda \in \mathbb{R}^m} \theta(\lambda) \geq 0 \tag{2.21}$$

is called duality gap. The gap is guaranteed to vanish, $\Delta = 0$, if the primal problem is convex, f is continuously differentiable and an appropriate constraint qualification holds [46]. This property is referred to as strong duality. However, in the case with integer variables, non-zero duality gaps can occur and there is no guarantee that solving the dual problem provides a feasible solution for the primal problem. It is then common practice to apply a heuristic to recover a primal solution after solving the dual problem. The heuristic may find an optimal primal solution, but even then a non-zero duality gap may remain. If a positive gap remains and no better primal solutions can be found it is necessary to perform branching. An example for an application with branching can be found in Carøe and Schultz [3]. The derivation of branching schemes is also discussed in Section 2.2.2.

Solution Methods. A variety of different techniques can be applied to solve the dual problem (2.18). The simplest among them is the subgradient method, while more sophisticated approaches include cutting plane and bundle methods. In the following we provide a brief description of these methods. More information, including proofs of the described properties can be found in [47, 40]. As our dual problem is a maximisation problem, the methods we describe are based on supergradients rather than subgradients, and the corresponding tangential hyperplanes support the Lagrangian from above, rather than below.

Gradient Methods. The dual function θ is concave regardless of the properties of f, h and $\mathcal{X}$. However, θ is typically not differentiable everywhere, so the solution methods for the dual problem must be able to deal with its non-smooth nature. At every λ_k where $\theta(\lambda_k)$ exists, that is, (2.17) has an optimal solution $x(\lambda_k)$, the vector $s(\lambda_k) := -h(x(\lambda_k))$ is a supergradient of θ:

$$\theta(\lambda) \leq \theta(\lambda_k) + s(\lambda_k)^T(\lambda - \lambda_k) \quad \forall \lambda \in \mathbb{R}^m, \tag{2.22}$$

which we write as $s(\lambda_k) \in \partial\theta(\lambda_k)$. By solving (2.17) to evaluate the dual function at a query point λ_k, a supergradient is obtained for free, and this can be used to construct a method to solve the dual problem. For λ^* strictly better than λ_k we have

$$\theta(\lambda_k) < \theta(\lambda^*) \leq \theta(\lambda_k) + s(\lambda_k)^T(\lambda^* - \lambda_k), \tag{2.23}$$

which implies that $s(\lambda_k)^T(\lambda^* - \lambda_k) > 0$, so if we choose a sufficiently small steplength $\alpha > 0$, then

$$\begin{aligned} \|\lambda_k - \alpha s(\lambda_k) - \lambda^*\|_2^2 &= \|\lambda_k - \lambda^*\|_2^2 - 2\alpha s(\lambda_k)^T(\lambda^* - \lambda_k) + \alpha^2\|s(\lambda_k)\|_2^2 \\ &< \|\lambda_k - \lambda^*\|_2^2, \end{aligned} \tag{2.24}$$

which means that $\lambda_k - \alpha s(\lambda_k)$ is strictly closer to any improving λ^* than λ_k. This motivates the following iterative scheme for solving the dual problem (2.18).

Algorithm 2.1 (Supergradient Method) *Input: $k = 0$, λ_0, and tolerance $\epsilon > 0$*

1. *Find $x(\lambda_k)$ and $\theta(\lambda_k)$ by solving (2.17)*

2. *If* $\|h(x(\lambda_k))\| < \epsilon$ *then terminate*

3. *Otherwise set* $\lambda_{k+1} = \lambda_k - \alpha_k s(\lambda_k)$, $k = k + 1$ *and go to 1.*

Here $\alpha_k \in \mathbb{R}_+$ is a sequence of positive steplengths. It can be shown that if the steplengths satisfy the following conditions

$$\alpha_k = \frac{t_k}{\|s(\lambda_k)\|}, \quad t_k \downarrow 0, \quad \sum_{k=1}^{\infty} t_k = \infty \tag{2.25}$$

the supergradient method will converge to a maximum of θ. However, convergence is often too slow for practical applications. This drawback can be partially alleviated through the right choice of steplength α_k and a change of metric, e.g. $\lambda = Bu$ with an invertible matrix B [40].

After solving the dual problem, an important question is how to recover a solution of the primal problem. If the primal problem is convex and additional assumptions hold with respect to the steplengths α_k and the Lagrangian, the sequence of weighted averages

$$\hat{x}_k := \frac{\sum_{j=1}^{k} \alpha_j x(\lambda_j)}{\sum_{j=1}^{k} \alpha_j} \tag{2.26}$$

converges to an optimal primal solution [48]. In the mixed-integer case, however, it converges to an optimal solution of a convex relaxation of the primal problem, which is sometimes referred to as a pseudo-schedule in UC applications [40]. The pseudo-schedule can then be used as a starting point for primal heuristics to recover an integer solution.

Cutting Plane Methods. A more sophisticated way of solving the dual problem is via a cutting plane method. Every time we evaluate the dual function θ at a query point λ_k, we obtain a cutting plane (2.22) that is tangential to its graph and lies entirely above it. After k iterations this provides us with a concave piecewise linear model:

$$\theta(\lambda) \leq \hat{\theta}(\lambda) := \min_{i=1,\ldots,k} \left\{\theta(\lambda_i) + s(\lambda_i)^T(\lambda - \lambda_i)\right\} \tag{2.27}$$

which is called cutting plane model. It is characterised by the k-tuple of elements $(\theta(\lambda_i), s(\lambda_i))_{i=1,\ldots,k}$, the so-called bundle of information. In iteration k we find a new

query point λ_{k+1} by maximising the model $\hat{\theta}$:

$$\begin{aligned} \max_{\lambda,r} \quad & r \\ \text{s.t.} \quad & r \leq \theta(\lambda_i) + s(\lambda_i)^T(\lambda - \lambda_i) \quad \forall i = 1, \ldots, k \\ & (\lambda, r) \in \mathbb{R}^{m+1}. \end{aligned} \tag{2.28}$$

The piecewise linear model $\hat{\theta}$ is improved in every iteration by adding a new cutting plane that is tangential at the current query point. Problem (2.28) operates on the area under the graph of the piecewise linear model, the hypograph of $\hat{\theta}$:

$$\text{hypo}(\hat{\theta}) := \left\{ (\lambda, r) \in \mathbb{R}^{m+1} : r \leq \hat{\theta}(\lambda) \right\}. \tag{2.29}$$

By construction we have that $r_{k+1} = \hat{\theta}(\lambda_{k+1}) \geq \hat{\theta}(\lambda) \geq \theta(\lambda)$ and the positive quantity

$$\delta := r_{k+1} - \theta(\lambda_k) = \hat{\theta}(\lambda_{k+1}) - \theta(\lambda_k) \geq 0 \tag{2.30}$$

is the nominal increase in iteration k if $\hat{\theta}$ was an accurate model of θ. We have that $\delta + \theta(\lambda_k) = r_{k+1} \geq \theta(\lambda)$ for all λ, which means that

$$\delta + \theta(\lambda_k) \geq \max_{\lambda} \theta(\lambda) \geq \theta(\lambda_k) \tag{2.31}$$

so if δ is small then $\theta(\lambda_k)$ is a good approximation of the dual optimum and we can stop. The sequence (r_k) is non-increasing and it can be shown that if (λ_k) is bounded then it converges to the dual optimum, $r_k \downarrow \max_\lambda \theta(\lambda)$ [40]. The Cutting Plane Algorithm is stated below, and an exemplary application with dimension $m = 1$ is visualized in Figure 2.2.

Algorithm 2.2 (Cutting Plane Method) *Input: $k = 0$, λ_0, and tolerance $\epsilon > 0$*

1. *Find $x(\lambda_k)$, $s(\lambda_k) = -h(x(\lambda_k))$ and $\theta(\lambda_k)$ by solving (2.17)*
2. *Add a cut to (2.28). Set $k = k + 1$ and solve (2.28) for r_{k+1} and λ_{k+1}*
3. *If $\delta = r_{k+1} - \theta(\lambda_k) < \epsilon$ then terminate, else go to 1*

An inherent weakness of the cutting plane method is its instability. Problem (2.28) is obviously feasible, but not necessarily bounded. For instance, after the first iteration

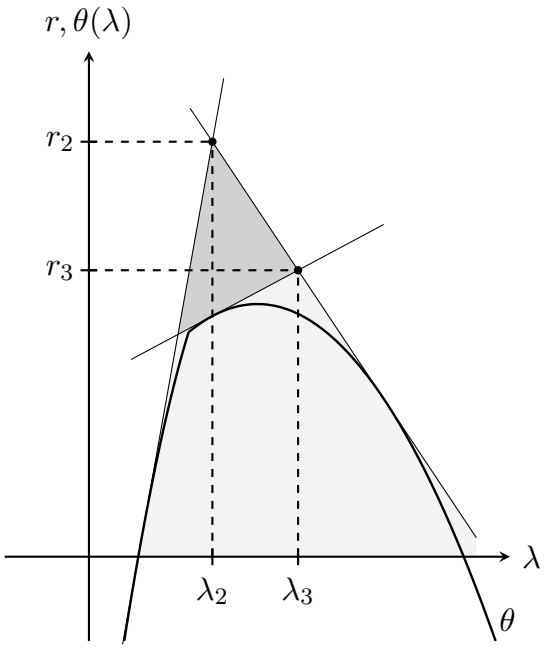


Figure 2.2: Exemplary application of the cutting plane method to maximise the non-smooth function $\theta(\lambda)$. In iterations $k = 0, 1$ two supporting hyperplanes were found and addded to the piecewise linear model $\hat{\theta}$. The hypograph of $\hat{\theta}$ is colored in gray. Solving (2.28) gave (r_2, λ_2) and another hyperplane was generated from query point λ_2. It cuts off the dark gray area of the hypograph. Solving (2.28) again gives (r_3, λ_3).

the model $\hat{\theta}$ consists of a single plane, and without imposing bounds on r or λ we get $r_2 = \infty$. Stability is an issue of practical importance. An example in Chapter XV in [47] shows that to obtain an accuracy of $\delta < \epsilon^2$ for $\theta(\lambda) := \min\{0, 1 - \|\lambda\|\}$ one requires $\mathcal{O}((\frac{1}{\epsilon})^m)$ iterations. The model $\hat{\theta}$ is too optimistic and encourages too large steps in λ. To overcome this issue and stabilise the cutting plane method is the objective of proximal bundle methods.

Proximal Bundle Methods. Bundle methods are an adaptation of the cutting plane method: a stability center $\hat{\lambda}$ is chosen, and the next iterate is required to be close to that center. There are different ways of implementing the proximity restriction: level, trust-region and penalty stabilization [47]. Here we focus on the popular penalty method, in which the cutting plane problem (2.28) is replaced by the following stabilised problem.

$$\max_{\lambda, r} \quad r - \tfrac{1}{2}(\lambda - \hat{\lambda})^T M (\lambda - \hat{\lambda}) \tag{2.32}$$

$$\begin{aligned} \text{s.t.} \quad & r \leq \theta(\lambda_i) + s(\lambda_i)^T(\lambda - \lambda_i) \quad \forall i = 1, \ldots, k \\ & (\lambda, r) \in \mathbb{R}^{m+1}. \end{aligned}$$

Here M is an appropriate scaling factor. A popular choice is $M = tI$, where I is the m-dimensional identity matrix and $t > 0$ is a steplength parameter which controls how close the next iterate stays to the stability center. If t varies from one iteration to another, we refer to this as a variable steplength bundle method. For more general choices of M the approach is called variable metric bundle method. The expected increase in iteration k is now defined in terms of the stability center $\hat{\lambda}$ as $\delta := r_{k+1} - \theta(\hat{\lambda})$, and if the actual increase of the dual function is larger than a given proportion $\kappa \in [0, 1)$ of that,

$$\theta(\lambda_{k+1}) - \theta(\hat{\lambda}) \geq \kappa\delta, \tag{2.33}$$

then the step is deemed successful and the stability center is updated as $\hat{\lambda} = \lambda_{k+1}$. Such steps are called *serious steps*. Otherwise the stability center remains the same, but the cutting plane model is still updated. Such steps are known as *null steps*. The bundle algorithm is stated below.

Algorithm 2.3 (Bundle Method) *Input:* $k = 0$, M, $\kappa \in [0, 1)$ $\lambda_0 = \hat{\lambda}$, *and* $\epsilon > 0$

1. *Find* $x(\lambda_k)$, $s(\lambda_k) = -h(x(\lambda_k))$ *and* $\theta(\lambda_k)$ *by solving (2.17)*
2. *Add a cut to (2.32). Set* $k = k + 1$ *and solve (2.32) for* r_{k+1} *and* λ_{k+1}
3. *If* $\theta(\lambda_{k+1}) - \theta(\hat{\lambda}) \geq \kappa\delta$ *then set* $\hat{\lambda} = \lambda_{k+1}$
4. *If* $\delta + \|M(\lambda_{k+1} - \hat{\lambda})\| < \epsilon$ *terminate, else go to 1*

Besides testing whether the expected increase is small, the termination criterion of this method also tests if the change in the dual iterate is small. The need for a stricter termination criterion is motivated by the fact that (2.31) does not hold any more if we obtain δ from solving the stabilised cutting plane model (2.32). Bundle methods can be shown to converge to a maximum of the dual problem in a finite number of iterations. In practical applications their efficiency depends a lot on the choice of the scale matrix M or the step size parameter t. More information on bundle methods and proofs of these results can be found in [47, 40, 49].

2.2.2 Dantzig-Wolfe Decomposition

DW decomposition was introduced in 1960 as a technique for linear programming [42]. Its application results in an algorithm which is known as column generation. Like LR it offers a tool to separate complicating or binding constraints from the constraint matrix and solve a simpler subproblem iteratively to arrive at the solution of the original problem. However, it is restricted to the case of linear and mixed integer linear programming. DW decomposition relies on the observation that a polyhedron can be expressed as a convex combination of its extreme points. The presentation below is based on [41, 50]. Consider the following primal problem

$$\min_x c^T x \text{ s.t. } Ax = b,\ Dx = d,\ x \geq 0, \tag{2.34}$$

which is a special linear case of (2.14) with $f(x) = c^T x$, $\mathcal{X} := \{x : Dx = d,\ x \geq 0\}$ and complicating constraints $h_j(x) := A^j x - b_j = 0$ for $j = 1, \ldots, m$. For simplicity we assume that the feasible set $\mathcal{X}$ is non-empty and bounded, which holds for the UC applications considered in Chapters 3 and 4. For the unbounded case we refer to [41]. In the bounded case, $\mathcal{X}$ can equivalently be represented as a convex combination of its finitely many extreme points $\{\hat{x}_i\}_{i \in I}$, i.e. for every $x \in \mathcal{X}$ there is an alternative representation with

$$x = \sum_{i \in I} \hat{x}_i w_i, \quad \sum_{i \in I} w_i = 1, \quad w \geq 0, \tag{2.35}$$

where w are convex weight variables and I is a finite index set of extreme points of $\mathcal{X}$. If we define $\hat{a}_i := A\hat{x}_i$ and $\hat{c}_i := c^T \hat{x}_i$ for $i \in I$ and substitute for x in the original problem (2.34), we obtain an equivalent master problem (MP):

$$\begin{aligned} z^* := \quad \min_w \quad & \sum_{i \in I} \hat{c}_i w_i \\ \text{s.t.} \quad & \sum_{i \in I} \hat{a}_i w_i = b \\ & \sum_{i \in I} w_i = 1, \quad w \geq 0. \end{aligned} \tag{2.36}$$

The original problem and the MP are equivalent in that their optimal objective values z^* are identical. However, their feasible polytopes differ combinatorially. A given set

of weights w implies a unique solution x, but not vice versa. In most applications, $\mathcal{X}$ has a large, often exponential number of extreme points, so that it is not efficient to enumerate them. Instead, we work with a small subset of columns in I, and the corresponding version of (2.36) is called restricted master problem (RMP). Additional extreme points are identified in the optimization process and added into the RMP as columns, hence the name column generation. The optimality criterion is the same as in the Simplex algorithm: in every iteration, the pricing step is to find a non-basic variable to enter the basis. Assuming that we have a subset of columns such that the RMP is feasible, let $\bar{\lambda},\ \bar{\sigma}$ be its dual optimal solution. If there is a feasible point $\hat{x}_i \in \mathcal{X}$ with negative reduced cost, that is,

$$\bar{c}_i = c^T \hat{x}_i - \bar{\lambda}^T A \hat{x}_i - \bar{\sigma} < 0 \tag{2.37}$$

then the RMP objective value can be improved if it enters the basis. Testing if such a point exists amounts to solving another linear problem:

$$\bar{c}^* := \min_x \left\{ \left(c^T - \bar{\lambda}^T A\right) x - \bar{\sigma} :\ Dx = d,\ x \geq 0 \right\}. \tag{2.38}$$

This is called pricing problem, subproblem, oracle or column generator. If $\bar{c}^* \geq 0$ then the previous optimal solution of the RMP is also optimal for the MP. Otherwise, if $-\infty < \bar{c}^* < 0$ the optimal solution of (2.38) is an extreme point $\hat{x}_i$ of $\mathcal{X}$ and we add the column $\left[c^T \hat{x}_i, (A\hat{x}_i)^T, 1\right]^T$ to the RMP. After adding the column, the RMP is solved again and the new column can be assigned a positive weight. The RMP is solved over a smaller feasible region than the original problem, so its intermediate solutions $\bar{z}$ provide upper bounds on the optimal objective value z^*. Additionally, it can be shown that the MP's objective value can decrease at most by the reduced cost, which results in the following bounds [51]:

$$\bar{z} + \bar{c}^* \leq z^* \leq \bar{z}. \tag{2.39}$$

The Dantzig-Wolfe decomposition or ColGen method is summarised in the statement below.

Algorithm 2.4 (Column Generation) *Input:* $k = 0$, *columns* $\left[\hat{c}_i^T, \hat{a}_i^T, 1\right]^T, i \in I$

1. *Find* z_k, λ_k, σ_k *by solving the RMP (2.36)*

2. *Find* $\bar{c}^k$ *and* x_k *by solving the pricing problem (2.38)*

3. *If* $\bar{c}^k \geq 0$ *terminate, else set* $k = k + 1$*, add* $\left[c^T x_k, (Ax_k)^T, 1\right]^T$ *and go to 1.*

Mixed-Integer Column Generation. The termination criterion in this statement is based on non-negativity of reduced costs, which implies optimality. However, in practical applications it can take many iterations to achieve this. If an ϵ-optimal solution is sufficient, one can terminate when the relative optimality gap is sufficiently small: $\frac{-\bar{c}^k}{z^k} < \epsilon$. This is of particular interest if ColGen is applied in the mixed-integer case. In the following we assume that some of the variables x must take binary values, as will be the case for the UC applications discussed in Chapters 3 and 4. For a more generic view of ColGen in the case with general integer variables, see Vanderbeck [50]. Let

$$\tilde{\mathcal{X}} := \{x : Dx = d,\ x \geq 0,\ x_j \in \{0,1\},\ j \in J\}\,. \tag{2.40}$$

To apply the Dantzig-Wolfe algorithm we can use a convexification approach. Instead of working with $\tilde{\mathcal{X}}$, we use its convex hull $\text{conv}(\tilde{\mathcal{X}})$. This is a natural extension of the ColGen algorithm, as the MP already operates on the convex hull of its columns. To ensure that the columns are extreme points of $\tilde{\mathcal{X}}$, we only need to adapt the pricing problem:

$$\bar{c}^* := \min_x \left\{ \left(c^T - \bar{\lambda}^T A\right) x - \bar{\sigma} :\ Dx = d,\ x \geq 0,\ x_j \in \{0,1\},\ j \in J \right\}. \tag{2.41}$$

Algorithm 2.4 can be used with this mixed-integer pricing problem. However, on termination of the ColGen method, z^* is now an optimal solution of a convex relaxation of the original problem. It provides a lower bound on the optimal objective value of the original mixed-integer problem because $\tilde{\mathcal{X}} \subset \text{conv}(\tilde{\mathcal{X}})$. Heuristics are typically used to find integer feasible solutions $\tilde{x}$, and any such solution provides an upper bound $c^T\tilde{x}$ on the optimal objective function. The gap between upper and lower bounds can be closed by branching on the RMP, that is, the ColGen procedure can be embedded in a Branch & Price framework by repeating it at every node of a partial enumeration tree [52]. A variety of branching rules have been proposed for Branch & Price algorithms [53]. When branching is performed, it is often preferable to do this on x rather than the weight variables w [41]. To branch on x, the following operations need to be performed

on the master- and subproblems.

- When a branching decision is made, columns with contradictory decisions are removed from the RMPs at the child nodes. The set of columns at the parent node is divided in two sets to be used at the first and second child, respectively. We will see below that in some cases the pricing problem is split into multiple subproblems, in which case the column sets at child nodes are non-disjoint subsets of the parent set because not all original variables appear in every column.
- To avoid that the pricing problem generates eliminated columns again at the child node, appropriate constraints must be added to it. If branch decisions are made with respect to original variables x they can be added directly to the pricing problem.

The Branch & Price procedure stops when the gap between upper and lower bounds satisfies a given tolerance.

Ties with Lagrangian Relaxation. In the linear and mixed-integer linear case, LR and DW decomposition are different approaches to achieve the same effect. In the following we briefly review a result which shows that they are dual to one another. Consider the RMP (2.36) and let λ and σ be the multipliers of the complicating constraints and the convexity constraints, respectively. The LP dual of the RMP is then given by

$$\begin{aligned} \max_{\lambda,\sigma} \quad & b^T\lambda + \sigma \\ \text{s.t.} \quad & \hat{a}_i^T\lambda + \sigma \leq \hat{c}_i \quad i \in I \\ & (\lambda, \sigma) \in \mathbb{R}^{|I|+1} \end{aligned} \tag{2.42}$$

We know that strong duality holds for linear problems [51, 46], so for a primal optimal solution $\bar{x} = \sum_{i \in I} \hat{x}_i \bar{w}_i$ of (2.36) and a dual optimal solution $(\bar{\lambda}, \bar{\sigma})$ of (2.42), we have

$$\bar{z} = c^T\bar{x} = \bar{\lambda}^T b + \bar{\sigma}. \tag{2.43}$$

Now consider the Lagrangian dual function of problem (2.34):

$$\theta(\lambda) = \min_{x \in \mathcal{X}} \left[c^T x - \lambda^T (Ax - b) \right] \tag{2.44}$$

For a given vector of multipliers $\bar{\lambda}$, $\bar{\sigma}$, we obtain from (2.38) and (2.43) that the Lagrangian lower bound is given by

$$\theta(\bar{\lambda}) = \left(\bar{\lambda}^T b + \bar{\sigma} \right) + \min_{x \in \mathcal{X}} \left(c^T - \bar{\lambda}^T A \right) x - \bar{\sigma} = \bar{z} + \bar{c}^*. \tag{2.45}$$

By comparing this bound to (2.39) we find that the lower bounds obtained from ColGen and LR are the same. Additionally, we can observe that evaluating the Lagrangian dual function (2.44) is the same as solving the pricing problem (2.38). The same terminology applies in both cases, that is, evaluating (2.44) is also referred to as subproblem, oracle or column generator.

The LP dual of the master problem can be reformulated further. We perform a change of variable $r := \sigma + b^T \lambda$, $r \in \mathbb{R}$ and use this to eliminate $\sigma = r - b^T \lambda$ from (2.42) and obtain the following equivalent problem:

$$\begin{aligned} \max_{\lambda, r} \quad & r \\ \text{s.t.} \quad & r \leq \hat{c}_i - \lambda^T (\hat{a}_i - b) \quad i \in I \\ & (\lambda, r) \in \mathbb{R}^{|I|+1} \end{aligned} \tag{2.46}$$

It is now easy to see that this problem and the cutting plane problem for the Lagrangian (2.28) are identical. Every cutting plane is derived from a dual solution λ_i, $i \in I$. If we let $\hat{x}_i \in \operatorname{argmin}_x L(x, \lambda_i)$ and then substitute $\theta(\lambda_i) = c^T \hat{x}_i - \lambda_i^T (A\hat{x}_i - b)$ and $s(\lambda_i) = -h(\hat{x}_i) = b - A\hat{x}_i$ into (2.28), then we obtain (2.46). This means that applying ColGen to solve the original problem is the primal equivalent of applying the cutting plane method to the Lagrangian dual problem. The duality of the cutting plane and ColGen methods permits us to cross-apply different methods. Techniques developed for one application can be applied in the other and vice versa. For example bundle methods can be applied to stabilise ColGen, and our application described in Chapter 3 will make use of that. The resulting algorithm is sometimes referred to as bundle column

generation [15]. Similarly, branching rules applied to RMPs of mixed-integer problems in the context of Branch & Price can also be applied to the corresponding cutting plane problem to close the duality gap of Lagrangian relaxation [3].

The Decomposition Principle. In the previous paragraph we have seen that evaluating the Lagrangian dual function is the same as solving the subproblem of DW decomposition. In Example 2.1 we demonstrated a case where the constraint matrix of this subproblem is block-diagonal, but we have not shown formally how to exploit this. Exploiting separability of the subproblem is what makes LR and ColGen decomposition techniques attractive, and we now show how to do this. Assume that the feasible set $\mathcal{X} = \{x : Dx = d,\ x \geq 0\}$ is separable, that is,

$$D = \begin{bmatrix} D^1 & & \\ & \ddots & \\ & & D^K \end{bmatrix}, \quad d = \begin{bmatrix} d^1 \\ \vdots \\ d^K \end{bmatrix}. \tag{2.47}$$

Then we can subdivide $x = (x^1, \ldots, x^K)^T$ and write $\mathcal{X}^k := \left\{x^k : D^k x^k = d^k,\ x^k \geq 0\right\}$ for $k = 1, \ldots, K$. Each of the subsets $\mathcal{X}^k$ can be represented independently by the convex hull of its extreme points, indexed by I^k. The objective function is also separable, with $c^T x = \sum_{k=1}^K c^{kT} x^k$. If we write $\hat{a}_i^k = A_k \hat{x}_i^k$ and $\hat{c}_i^k = c^{kT} \hat{x}_i^k$ for the columns, and w_i^k for the weight variables of the k-th subsystem, then the RMP becomes:

$$\begin{aligned} \min_{w} \quad & \sum_{k=1}^{K} \sum_{i \in I^k} \hat{c}_i^k w_i^k \\ \text{s.t.} \quad & \sum_{k=1}^{K} \sum_{i \in I^k} \hat{a}_i^k w_i^k = b \\ & \sum_{i \in I^k} w_i^k = 1,\ w^k \geq 0 \quad \forall k = 1, \ldots, K. \end{aligned} \tag{2.48}$$

If we take σ_k to represent the multiplier of the k-th convexity constraint, then the pricing step amounts to solving the following K independent subproblems for a given set of multipliers $\bar{\lambda}$, $\bar{\sigma}$:

$$\bar{c}^{k*} := \min_{x^k} \left\{ \left(c^{kT} - \bar{\lambda}^T A_k \right) x^k - \bar{\sigma}^k :\ D^k x^k = d^k,\ x^k \geq 0 \right\} \quad \forall k = 1, \ldots, K. \tag{2.49}$$

The ColGen algorithm now terminates when $\bar{c}^{k*} \geq 0$ for all $k = 1, \ldots, K$. Otherwise we add a new column to the RMP for an optimal solution x^{k*} of every pricing subproblem (2.49) with $\bar{c}^{k*} < 0$. The optimal solution of the RMP is now $\bar{z} = \bar{\lambda}^T b + \sum_{k=1}^{K} \bar{\sigma}^k$, and the bounds on the optimal objective value of the original problem are

$$\bar{z} + \sum_{k=1}^{K} \bar{c}^{k*} \leq z^* \leq \bar{z}. \tag{2.50}$$

The dual of the RMP is the same cutting plane problem as before, where we introduce a new cut for every column that is added to the RMP:

$$\begin{aligned} \max_{\lambda, r} \quad & \sum_{k=1}^{K} r^k \\ \text{s.t.} \quad & r^k \leq \hat{c}_i^k - \lambda^T \left(\hat{a}_i^k - b \right) \quad \forall k = 1, \ldots, K,\ i \in I^k \\ & (\lambda, r) \in \mathbb{R}^{|I|+K}. \end{aligned} \tag{2.51}$$

With these alterations the pricing problem consists of K independent, smaller subproblems which can be solved in parallel if the computational environment permits it.

2.2.3 Progressive Hedging

PH is a decomposition technique that was developed by Rockafellar and Wets [16] to decompose stochastic optimization problems by scenarios. It is proven to converge to an optimal solution if it is applied to continuous, convex stochastic problems. Due to its easy implementation, PH is also a popular choice for large-scale mixed-integer problems, including UC problems [19]. However, it remains a heuristic when applied in those cases: although valid lower bounds can be obtained in any iteration [18], there is no guarantee that an integer optimal primal solution is found. PH requires the non-anticipativity condition to be written as a constraint, as demonstrated in Section 2.1.2. It uses an augmented Lagrangian function to relax the non-anticipativity constraints and make the problem separable by scenarios. Consider the following stochastic problem:

$$\min_{x,y} \quad \sum_{s \in \mathcal{S}} p_s f_s(x_s, y_s) \tag{2.52}$$

$$\begin{aligned} \text{s.t.} \quad & (x_s, y_s) \in \mathcal{X}_s \quad \forall s \in \mathcal{S} \\ & x_s - \bar{x} = 0 \quad \forall s \in \mathcal{S} \end{aligned}$$

which we obtain by re-writing (2.6) with $f_s := c^T x_s + d_s^T y_s$ and $\mathcal{X}_s := \{(x_s, y_s) \geq 0 : Ax_s = b, T_s x_s + W_s y_s = r_s\}$. We choose a two-stage formulation to keep the notation simple, however, PH can also be applied to multi-stage problems. The augmented Lagrangian for problem (2.52) is given by

$$L_A(x, y, \lambda, \mu) := \sum_{s \in \mathcal{S}} \left[p_s f_s(x_s, y_s) - \lambda_s^T (x_s - \bar{x}) + \tfrac{1}{2}\mu \|x_s - \bar{x}\|_2^2 \right]. \tag{2.53}$$

In addition to the linear Lagrangian term with multipliers λ, the augmented Lagrangian uses a quadratic penalty term with a scalar penalty parameter $\mu > 0$. A simple algorithmic scheme can be devised for augmented Lagrangians, which is similar to the subgradient method for Lagrangians described in Section 2.2.1. The following multiplier update is derived from first order optimality conditions for the augmented Lagrangian dual problem [46]:

$$\lambda_s^{k+1} = \lambda_s^k - \mu(x_s^k - \bar{x}^k) \quad \forall s \in \mathcal{S}. \tag{2.54}$$

The convergence properties of augmented Lagrangians are superior to those of Lagrangians: finite values of μ are sufficient to achieve convergence to a dual optimal solution λ^* *and* a primal optimal solution (x^*, y^*) [46]. In practical applications it is common to use a positive, finite but increasing sequence of penalties (μ_k). At every iteration k, after updating the multipliers λ^k, the next primal candidate solution (x^k, y^k) is found by solving the subproblem

$$\min_{(x,y) \in \mathcal{X}} L_A(x, y, \lambda^k, \mu^k) \tag{2.55}$$

to evaluate the augmented Lagrangian dual function at λ^k. With the augmented Lagrangian from (2.53) this subproblem is not separable by scenarios due to the quadratic term binding scenario specific variables x_s to the common target variables $\bar{x}$. To achieve separability in PH, the target variables $\bar{x}$ are replaced with a fixed value $\hat{x}$ which is

estimated from the previous iteration's solution:

$$\hat{x} := \sum_{s \in \mathcal{S}} p_s x_s. \tag{2.56}$$

After replacing the common targets in the augmented Lagrangian with a fixed estimate, it becomes separable: instead of (2.55) we can solve the following subproblems for all scenarios $s \in \mathcal{S}$:

$$\min_{(x_s, y_s) \in \mathcal{X}_s} \left[p_s f_s(x_s, y_s) - (\lambda_s^k)^T (x_s - \hat{x}) + \tfrac{1}{2} \mu_k \|x_s - \hat{x}\|_2^2 \right]. \tag{2.57}$$

Although the common targets are set to the expected value of the previous iteration, PH is proven to converge to an optimal solution in linear time if the stochastic problem is convex [16]. The PH algorithm can be summarised as follows:

Algorithm 2.5 (Progressive Hedging) *Input:* $k = 0$, $\lambda^0 = 0$, $\mu_0 = 0$, $\epsilon > 0$

1. *Find* (x_s^k, y_s^k) *for* $s \in \mathcal{S}$ *by solving (2.57)*
2. *Calculate* $\hat{x}^k = \sum_{s \in \mathcal{S}} p_s x_s^k$
3. *If* $\sum_{s \in \mathcal{S}} p_s \|x_s^k - \hat{x}^k\| < \epsilon$ *then terminate*
4. *Else set* $\lambda_s^{k+1} = \lambda_s^k - \mu_k (x_s^k - \hat{x}^k)$ $\forall s \in \mathcal{S}$, $\mu_{k+1} \geq \mu_k$, $k = k + 1$ *and go to 1.*

The termination criterion is based on primal feasibility. In the mixed-integer case, additional measures are required to accelerate convergence and detect non-convergence. These are discussed in [19], along with alternative termination criteria and strategies for choosing a sequence (μ_k).

2.3 Generation Unit Commitment

The UC problem is to find a cost-minimal schedule of startup and shutdown decisions for a given set of generation units. Typically the units considered in this problem are smaller than a whole power plant, i.e. a large coal power plant may consist of multiple steam turbines that can be started individually. For each unit, the decision to have it on or off in any discrete time step is of binary nature, which makes the problem combinatorial. The units have fixed and variable running costs: the fixed cost

is typically referred to as no-load cost while the variable cost is called marginal cost or generation cost and is often modelled as linear, PWL or convex quadratic function of the unit's power output. Usually the units incur additional costs when they are started up, and many models have additional binary variables for these startups, and some also use binaries for shutdowns [9]. The following constraints are present in most UC models:

- **Load balance** constraints ensure that the total power infeed into the system is equal to the demand at all times.
- **Reserve** constraints require the system to keep sufficient backup capacity in part-loaded or quick-start generators or pump storage units. Reserve may be split with respect to the time frame in which it is available, i.e. spinning and non-spinning. Some models also make a distinction between primary and secondary frequency response and other, slower reserve products. These are explained in more detail in Chapter 4.
- **Power output bounds** impose lower and upper output limits on the units when they are on.
- **Polyhedral** constraints establish the connection between on-off, startup and shutdown variables if more than one set of these are included in the model.
- **Minimum up/downtime** requirements force units to stay on (off) for a minimum amount of time after a startup (shutdown), to avoid increased wear and tear of the turbines due to differential expansions.
- **Ramp rate** constraints limit the rate of change in each unit's power output.

Additionally, UC models sometimes contain approximate or aggregated versions of the transmission or distribution grid, interconnectors with other networks, pump storage reservoirs and renewable supplies, the output of which cannot be planned but is weather dependent. The following section contains a basic MIP formulation of a UC problem with the constraints named above. In Chapter 4 we extend this formulation with a network model and a pump storage model to suit the British power system. The formulation shown here uses all three types of binary variables discussed above. An

overview of our notation is also given below. Sets are in calligraphic font, parameters are Latin and Greek capitals, and variables are lower case Latin or Greek letters. Superscripts are used to extend variable names, while subscripts are indices. The planning horizon is $t = 1, \ldots, T$, and where the statement shows or implies variables for $t \leq 0$, they are fixed input data rather than actual variables.

2.3.1 Algebraic Model Statement

Sets

$\mathcal{G}$: set of generation units

Parameters

D^t: length of a single time step [here: 1h]
C_g^{nl}: no-load cost of generator g [\$/h]
C_g^m: marginal (linear) cost coefficient of generator g [\$/MWh]
C_g^{st}: startup cost of generator g [\$]
f_g: cost function of generator g [\$/h]
P_t^{dem}: real power demand in period t [MW]
$P_g^{min,max}$: minimum (maximum) generation limit of generator g [MW]
P_t^{res}: real power reserve requirement in period t [MW]
$P_g^{ru,rd}$: operating ramp up (down) limits of generator g [MW/D^t]
$P_g^{su,sd}$: startup (shutdown) ramp limits of generator g [MW/D^t]
T: last time period of the planning horizon
$T_g^{u,d}$: minimum uptime (downtime) of generator g [D^t]

Variables

$\alpha_{gt} \in \{0, 1\}$: 1 if unit g is on in period t, and 0 if it is off
$\gamma_{gt} \in \{0, 1\}$: 1 if unit g is started up in period t, and 0 otherwise
$\eta_{gt} \in [0, 1]$: 1 if unit g is shut down in period t, and 0 otherwise
$p_{gt} \geq 0$: real power output of generator g in period t [MW]

Using the notation described above, the basic deterministic UC model reads as follows.

$$\min \sum_{t=1}^{T} \sum_{g \in \mathcal{G}} \left(C_g^{st} \gamma_{gt} + D^t C_g^{nl} \alpha_{gt} + D^t C_g^m p_{gt} \right) \tag{2.58}$$

subject to the following constraints.

- Load balance and spinning reserve equations for all $t = 1, \ldots, T$

$$\sum_{g \in \mathcal{G}} p_{gt} = P_t^{dem} \tag{2.59}$$

$$\sum_{g \in \mathcal{G}} \left(\alpha_{gt} P_g^{max} - p_{gt} \right) \geq P_t^{res} \tag{2.60}$$

- Generator bounds for all $g \in \mathcal{G},\ t = 1, \ldots, T$

$$P_g^{min} \alpha_{gt} \leq p_{gt} \leq P_g^{max} \alpha_{gt} \tag{2.61}$$

- Polyhedral/Switching constraints for all $g \in \mathcal{G},\ t = 1, \ldots, T$

$$\alpha_{gt} - \alpha_{g(t-1)} = \gamma_{gt} - \eta_{gt} \tag{2.62}$$

$$1 \geq \gamma_{gt} + \eta_{gt} \tag{2.63}$$

- Minimum up- and downtime constraints for all $g \in \mathcal{G},\ t = 1, \ldots, T$

$$\sum_{i=t-T_g^u+1}^{t} \gamma_{gi} \leq \alpha_{gt} \tag{2.64}$$

$$\sum_{i=t-T_g^d+1}^{t} \eta_{gi} \leq 1 - \alpha_{gt} \tag{2.65}$$

- Ramp rate constraints for all $g \in \mathcal{G},\ t = 2, \ldots, T$

$$p_{gt} - p_{g(t-1)} \leq P_g^{ru} \alpha_{gt-1} + P_g^{su} \gamma_{gt} \tag{2.66}$$

$$p_{g(t-1)} - p_{gt} \leq P_g^{rd} \alpha_{gt} + P_g^{sd} \eta_{gt} \tag{2.67}$$

The objective (2.58) is to minimise the total cost of electricity generation, consisting of startup, no-load and linear cost factors. Load balance constraints (2.59) require supply and demand to be balanced at all times, and spinning reserve constraints (2.60) impose a lower limit on the total headroom provided by part-loaded generation units. This is the simplest way of formulating a requirement for spinning or online reserve.

State	α_{gt}	$\alpha_{g(t-1)}$	γ_{gt}	η_{gt}
Startup	1	0	1	0
Shutdown	0	1	0	1
Cont. Off	0	0	0	0
Cont. On	1	1	0	0

Table 2.1: Integer solutions of constraints (2.62) and (2.63), and the operational state associated with them. Without (2.63), the cases where a generator is continuously off or on are ambiguous with respect to the values of γ_{gt} and η_{gt}, as those can either both be one or both be zero. When generators are off for two consecutive periods, the startup costs make solutions with $\gamma_{gt} = \eta_{gt} = 1$ suboptimal. However, when they are on for successive periods, the right hand side of ramp rate constraints (2.66) and (2.67) are relaxed by setting $\gamma_{gt} = \eta_{gt} = 1$, so there is an incentive for that which may outweigh the startup cost. To eliminate solutions where startup and shutdown variables are both one, constraints (2.63) need to be included.

In our GB model described in Chapter 4 we formulate separate requirements for online reserve and frequency response, which was omitted here as it requires additional variables. Minimum and maximum stable generation limits (2.61) establish the logical connection between power output variables and on-off variables: if a generator is off its power output must be zero, while if it is on it has to operate between its stable limits $0 < P_g^{min} < P_g^{max}$. Polyhedral or switching constraints (2.62) and (2.63) express the logical connection between on-off variables, startup and shutdown variables. Table 2.1 shows the four operational states which satisfy these constraints.

The model with all three variables (α, γ, η) is what Ostrowski et al [9] call the 3-Binary Variable Formulation. It is a popular way of formulating MIP UC models since it has a very tight LP relaxation. The integrality restriction of either startup variables γ or shutdown variables η can be relaxed and it follows from (2.62) that the relaxed variables must take integer values. We relax shutdown variables η, since this gives the best performance with our dataset and solver.

Minimum up- and downtimes for each generator are expressed as (2.64) and (2.65), using Rajan's and Takriti's facet defining cuts [11]. They can be interpreted as follows: For any period $t \in \{1, \ldots, T\}$ and generator $g \in \mathcal{G}$, constraints (2.64) say that the generator can only have been switched on once in the T_g^u preceding periods, and if that is the case then it must be on in period t. On the other hand, if it is off in period t then it cannot have been switched on in any of the T_g^u preceding periods. Constraints

(2.65) work identically for downtimes: the generator can only have been switched off once in the T_g^d periods preceding t, and if that is the case then it must be off in period t. On the other hand, if it is on in period t then it cannot have been switched off in any of the T_g^d preceding periods

Ramp rate constraints (2.66) and (2.67) impose upper limits on the change in power output between adjacent periods. Constraints (2.66) say that if a generator was already on in period $t-1$ ($\alpha_{g(t-1)} = 1$, $\gamma_{gt} = 0$) then it may ramp up by at most P_g^{ru} in period t, while if it is being started up in period t ($\alpha_{g(t-1)} = 0$, $\gamma_{gt} = 1$) it may ramp up from zero to at most P_g^{su}. On the other hand, constraints (2.67) say that if a generator continues to be on in period t ($\alpha_{gt} = 1$, $\eta_{gt} = 0$) then it may ramp down by at most P_g^{rd}, while if it is being shut down in period t ($\alpha_{gt} = 0$, $\eta_{gt} = 1$) it may ramp down from at most P_g^{sd} to zero. The distinction between ramp rates for continuous operation and startup/shutdown procedures is necessary to avoid infeasibilities if the data is such that the generators' minimum stable limits are larger than the ramp rates. For the model above, it is necessary that $P_g^{su}, P_g^{sd} \geq P_g^{min}$ but it is *not* necessary that $P_g^{ru}, P_g^{rd} \geq P_g^{min}$. This workaround is required because in terms of the above model a generator is available and operating above its minimum stable limit as soon as its startup variable takes a value of one. In more accurate models, generators follow a startup trajectory after being switched on, and may operate at output levels below their stable limit while doing so [54]. However, data on startup trajectories is not available to us, so we use the approximate model shown here.

2.3.2 Stochastic Formulations

The deterministic UC formulation given in the previous section can easily be extended to a stochastic one. To do so, we choose a scenario formulation because

- With the non-anticipativity property written as constraints, the necessary structure for scenario decomposition is already there. We merely need to apply LR or DW decomposition.

- To switch between a two-stage and a multi-stage recourse model, it is sufficent to adapt the scenario data and the non-anticipativity constraints. The rest of the model need not be altered.

Let $\mathcal{S}$ be the set of scenarios and let α_{gts}, γ_{gts}, η_{gts} and p_{gts} be the decision variables associated with scenario $s \in \mathcal{S}$. In the case studies in Chapters 3 and 4 the source of uncertainty are renewable infeeds. For now we assume that they enter the model on the right-hand side through uncertain residual demand to be satisfied by conventional generators. Let the demand in scenario $s \in \mathcal{S}$ be given by P_{ts}^{dem}. Then the constraint set for each scenario is given by

$$\mathcal{X}_s := \left\{ (\alpha_s, \gamma_s, \eta_s, p_s) \;\middle|\; \sum_{g \in \mathcal{G}} p_{gts} = P_{ts}^{dem},\ (2.60) \text{ to } (2.67),\ \forall g, t \right\} \tag{2.68}$$

Here $(\alpha_s, \gamma_s, \eta_s, p_s)$ denotes the vector of decision variables for all $g \in \mathcal{G}$ and $t = 1, \ldots, T$ but for a specific scenario s. We assume that $\mathcal{X}_s \subset \{0,1\}^{3|\mathcal{G}| \cdot T} \times \mathbb{R}_+^{|\mathcal{G}| \cdot T}$, i.e. the variable restrictions are implicit in $\mathcal{X}_s$. Note that $\mathcal{X}_s$ still includes the reserve constraints (2.60) because reserve is also kept for reasons other than dealing with wind uncertainty, e.g. to deal with outages. Additionally, let

$$f_g\left(\alpha_{gts}, \gamma_{gts}, \eta_{gts}, p_{gts}\right) := C_g^{st} \gamma_{gts} + D^t C_g^{nl} \alpha_{gts} + D^t C_g^{m} p_{gts}, \tag{2.69}$$

and let π_s be the probability of scenario $s \in \mathcal{S}$. Then the extensive formulation of the SUC model is given by

$$\min_{\alpha,\gamma,\eta,p} \quad \sum_{s \in \mathcal{S}} \pi_s \sum_{t=1}^{T} \sum_{g \in \mathcal{G}} f_g\left(\alpha_{gts}, \gamma_{gts}, \eta_{gts}, p_{gts}\right) \tag{2.70}$$

$$\text{s.t.} \quad (\alpha_s, \gamma_s, \eta_s, p_s) \in \mathcal{X}_s,\ \forall s \in \mathcal{S} \tag{2.71}$$

$$\alpha_{gts} = \bar{\alpha}_{gbt},\ \forall g \in \mathcal{G},\ b \in \mathcal{B},\ s \in \mathcal{S}_b,\ t = t_b^{st}, \ldots, t_b^{end} \tag{2.72}$$

$$p_{gts} = \bar{p}_{gbt},\ \forall g \in \mathcal{G},\ b \in \mathcal{B},\ s \in \mathcal{S}_b, t = t_b^{st}, \ldots, t_b^{end} \tag{2.73}$$

The scenario bundles are $b \in \mathcal{B}$, with start times t_b^{st} and end times t_b^{end} and subsets of bundled scenarios $\mathcal{S}_b$. Non-anticipativity constraints for commitment variables α_{gts} and power output variables p_{gts} are formulated in terms of common target variables $\bar{\alpha}_{gbt}$ and $\bar{p}_{gbt}$. Note that non-anticipativity constraints for γ_{gts} and η_{gts} are not required: in the presence of constraints (2.62) and (2.63), non-anticipativity of on-off variables α_{gts} implies non-anticipativity of γ_{gts} and η_{gts} as well.

In analogy to the example shown in Figure 2.1 (Section 2.1), these data structures can be used to shape a suitable decision tree to obtain a multi-stage SUC problem. In the multi-stage problem, generator commitments and their power outputs can be updated after every scenario split, i.e. they are updated on an *intraday* basis. On the other hand, their commitments *and* power outputs have to be identical for all bundled scenarios $s \in \mathcal{S}_b$ at all times $t_b^{st}, \ldots, t_b^{end}$. For this formulation to work, the realisations of the uncertain parameter must be identical for bundled scenarios:

$$P_{ts_1}^{dem} = P_{ts_2}^{dem} \quad \forall t, s_1, s_2 : \exists b \in \mathcal{B} : t \in \{t_b^{st}, \ldots, t_b^{end}\} \wedge s_1, s_2 \in \mathcal{S}_b \tag{2.74}$$

Scenario sets $\mathcal{S}$ which satisfy this property are said to form a scenario tree. The process that is used to generate the data P_{ts}^{dem} for the scenario tree typically consists of three steps: scenario sampling, reduction and scenario tree construction. The central responsibility of the scenario tree construction step is to ensure property (2.74) for a pre-defined tree structure. Scenario sampling, reduction and tree construction are explained in more detail in Chapter 4.

Two-stage SUC formulations have a different structure: in a two-stage model, the first stage decisions are the commitments α_{gts} of conventional generators, while the recourse decision is the output level p_{gts} at which to operate them. This is true for the whole planning horizon of the problem which, in our case, covers 24 hours. The commitment schedule is typically made well in advance of the beginning of the problem horizon, i.e. a schedule made between noon and 4pm on one day will usually become active at midnight. This scheduling approach is referred to as *day-ahead* unit commitment. To obtain the corresponding model, we drop constraints (2.73) and include (2.72) for a single bundle, that is, $\mathcal{B} = \{b_0\}$ with $t_{b_0}^{st} = 1$ and $t_{b_0}^{end} = T$ and $\mathcal{S}_{b_0} = \mathcal{S}$. Unlike in the multi-stage formulation, scenarios for the uncertain parameter do not have to form a scenario tree, i.e. they do not have to satisfy property (refeq:ScenTreeProperty).

Using a scenario formulation for two-stage SUC models is only sensible if the problem is solved by a scenario decomposition approach. If it is solved e.g. by applying Branch & Bound to the extensive formulation, it is typically more efficient to formulate it with first stage variables α_{gt}, γ_{gt}, η_{gt} which are common to all scenarios, while the only second-stage variables are p_{gts} for scenario-specific recourse actions. In the two-stage

version of our British UC model described in Chapter 4, there are also integer decisions on the second stage, because pump storage and open-cycle gas turbine (OCGT) commitments are recourse actions. Other examples of two-stage and multi-stage SUC models are described in Goez et al [4].

Non-Anticipativity and Degeneracy. Non-anticipativity constraints in UC problems typically have a high degree of redundancy which causes dual degeneracy. In the numerical examples in Chapter 3 we explore the extent of dual degeneracy and its effect on scenario decomposition methods which rely on the dual solution of the non-anticipativity constraints. In the following we briefly demonstrate typical causes of degeneracy.

Consider equations (2.61) with lower and upper stable generation limits $0 < P_g^{min} < P_g^{max}$. With α being binary, these imply the following relation between power output and on-off variables:

$$p_{gts} = 0 \Leftrightarrow \alpha_{gts} = 0 \quad \text{and} \quad p_{gts} \in \left[P_g^{min}, P_g^{max}\right] \Leftrightarrow \alpha_{gts} = 1 \tag{2.75}$$

This means that non-anticipativity of p variables implies non-anticipativity of α variables. Consequently, constraints (2.72) are redundant in the presence of constraints (2.73). However, we find that if an appropriate dual stabilisation technique is applied, scenario decomposition techniques can be more successful when both types of non-anticipativity constraints are present.

An additional cause of redundancy are the minimum up- and downtime constraints: non-anticipativity of α_{gts} for all $s \in \mathcal{S}$ often implies non-anticipativity of $\alpha_{g(t+k)s}$, for some integer k smaller than the minimum up- and downtime. Unlike the former cause, this also affects two-stage models which only contain non-anticipativity constraints on α_{gts}. There is no easy way of finding a non-redundant subset of constraints that guarantees non-anticipativity, so the solution techniques must be able to deal with redundant constraints.

In the GB model described in Chapter 4 there are additional variables which require non-anticipativity constraints, e.g. variables for pump storage operation or reservoir levels. Often these are related to other variables in a way that will make it sufficient

to include non-anticipativity constraints only for a subset of them. We select a subset for which we include the constraints, such that the solution is guaranteed to be non-anticipative. The selection is based on performance of the decomposition method. The subset still contains redundant constraints, and how redundancy and the resulting degeneracy affect our decomposition method is explored in the following chapter.

3

Scenario Decomposition of UC Problems

In this chapter we develop and test a practical DW scenario decomposition algorithm for stochastic UC problems with two or more stages. First, we state a simple ColGen algorithm for this class of problems. Then we explain how dual regularisation and initialisation can be used to stabilise and hot-start our method, and derive a primal heuristic to construct solutions quickly and accelerate convergence. Finally, we describe our implementation and discuss results of a performance comparison of decomposition and out-of-the-box Branch & Bound when applied to two-stage and multi-stage versions of our GB scheduling model. The results demonstrate how time savings through decomposition increase rapidly with the number of scenarios included in the stochastic model.

3.1 Dantzig-Wolfe Scenario Decomposition

DW decomposition was originally proposed for continuous linear programs with block-angular structure of the constraint matrix. As demonstrated in Section 2.2.2, the underlying idea is to exploit separability of the subproblem after removing a set of binding constraints. This approach has been generalised to the mixed-integer linear case, however there may be a non-zero duality gap after achieving convergence of the master problem, in which case branching is required if the gap is to be closed. In the following, we outline how mixed-integer DW decomposition can be applied to the stochastic UC problem (2.70) to (2.73) to separate non-anticipativity constraints from the remaining constraints of the problem and achieve separability by scenarios. Relaxing non-anticipativity to derive scenario decomposition schemes is a common concept in

two-stage and multi-stage stochastic programming: Lagrangian decomposition [33, 55], Progressive Hedging [16] and DW decomposition [32] have all been proposed to achieve separability by scenarios. Progressive Hedging remains a heuristic when applied in the mixed-integer case, but is still a popular algorithm in many such applications [19]. DW decomposition and Lagrangian decomposition extend naturally to the mixed-integer case since the master or cutting plane problems can be branched on. However, in the following sections we will show that for the stochastic UC problems in our examples, sufficiently small optimality gaps can be achieved without branching.

The DW Master Problem for Scenario Decomposition. Here we follow the approach explained in Section 2.2.2 to construct a basic scenario decomposition algorithm for the stochastic UC problem. To apply DW decomposition to the SUC problem, we first replace the original problem by an MP. The MP solves a relaxation of the original SUC problem by constructing convex combinations of individual scenario subproblem solutions. These convex combinations satisfy the non-anticipativity constraints. For all scenarios $s \in \mathcal{S}$, let $\text{conv}(\mathcal{X}_s)$ be the convex hull of feasible points of scenario s. Also, let I_s be the index set of extreme points of $\text{conv}(\mathcal{X}_s)$. In typical UC formulations $\mathcal{X}_s$ is bounded and I_s is finite, and throughout the remainder of this chapter we assume that this holds true. The extreme points are used as columns in the MP: at the i-th extreme point of $\text{conv}(\mathcal{X}_s)$, let the on-off, startup, shutdown and power output decisions of generator g at time t be denoted by A_{gtsi}, Γ_{gtsi}, H_{gtsi} and P_{gtsi}, respectively. For all $s \in \mathcal{S}$, $i \in I_s$ we record the operational cost of the corresponding column,

$$c_{si} := \sum_{g \in \mathcal{G}} \sum_{t=1}^{T} f_g \left(A_{gtsi}, \Gamma_{gtsi}, H_{gtsi}, P_{gtsi} \right), \tag{3.1}$$

and introduce a convex weight variable w_{si}. Then the MP for scenario decomposition can be stated as

$$\min_{w, \bar{\alpha}, \bar{p}} \quad \sum_{s \in \mathcal{S}} \pi_s \sum_{i \in I_s} c_{si} w_{si} \tag{3.2}$$

$$\text{s.t.} \quad \sum_{i \in I_s} A_{gtsi} w_{si} = \bar{\alpha}_{gbt}, \ \forall g \in \mathcal{G}, \ b \in \mathcal{B}, \ t = t_b^{st}, \ldots, t_b^{end}, \ s \in \mathcal{S}_b \tag{3.3}$$

$$\sum_{i \in I_s} P_{gtsi} w_{si} = \bar{p}_{gbt}, \ \forall g \in \mathcal{G}, \ b \in \mathcal{B}, \ t = t_b^{st}, \ldots, t_b^{end}, \ s \in \mathcal{S}_b \tag{3.4}$$

$$\sum_{i \in I_s} w_{si} = 1, \ \forall s \in \mathcal{S} \tag{3.5}$$

$$w \geq 0, \ \bar{\alpha}, \ \bar{p} \text{ free.} \tag{3.6}$$

As in the original problem, the objective (3.2) is to minimise the expected total cost. The non-anticipativity constraints for convex combinations of individual columns are (3.3) and (3.4). We denote their dual variables by λ^{α}_{gbts} and λ^{p}_{gbts}, and they can be interpreted as a price for deviation from the bundle's common commitment decision or common power output decision, respectively. The duals of constraints (3.5) are denoted by σ_s.

Choosing a Formulation of Non-Anticipativity Constraints. Artificial target variables $\bar{\alpha}_{gbt}$ and $\bar{p}_{gbt}$ are retained as variables in the MP. Instead of using a non-anticipativity formulation with artificial common target variables, we could use either the chain formulation (2.11) or formulation (2.12). In a chain formulation, the multipliers would be a price for pairwise non-anticipativity violation between scenario couples, while in the third formulation they would be a price for deviation from the decision of a pre-selected central scenario. A number of other formulations are possible, and scenario decomposition algorithms can be derived from all of them. Here we focus on the approach with common target variables, as we find that it requires less iterations to converge to an optimal solution than e.g. a chain formulation. A possible interpretation of this observation is that dual information is spread faster and more flexibly among the bundle if a common target formulation is used, however, an in-depth investigation of this hypothesis is outside the scope of this study. The use of common target variables does not increase the amount of degeneracy: in comparison to the alternative formulations without target variables there is exactly one additional equality constraint and one additional variable, so the dimensions of both, the primal and dual feasible spaces remain unchanged.

Restricting the Column Set. The number of extreme points $|I_s|$ can be expected to be large: for a given scenario s, Figure 3.1 illustrates a projection of the convex hull $\text{conv}(\mathcal{X}_s)$ onto the space of α and p variables. This is done for a very simple case where there is only a single generator and time period and the feasible set is described only

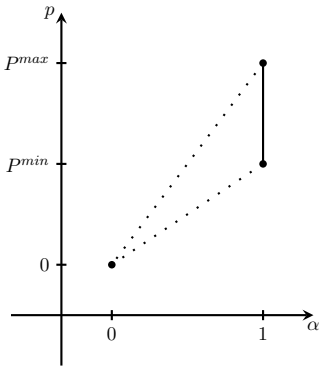


Figure 3.1: Projection of the convex hull $\text{conv}(\check{\mathcal{X}}_s)$ of feasible points of a single scenario s onto the space of α and p variables. This is shown for a case where there is only a single generator and time period, and the feasible set is described by minimum and maximum stable generation limits P^{min} and P^{max} only. There are three extreme points.

by minimum and maximum stable generation limits:

$$\check{\mathcal{X}}_s = \left\{ (\alpha_s, \gamma_s, \eta_s, p_s) \in \{0,1\}^{3|\mathcal{G}|\cdot T} \times \mathbb{R}_+^{|\mathcal{G}|\cdot T} \;\middle|\; P_g^{min}\alpha_{gts} \leq p_{gts} \leq P_g^{max}\alpha_{gts} \right\}. \tag{3.7}$$

In this simple case the convex hull of feasible points has three extreme points per scenario, so with $|\mathcal{G}|$ generators and T time periods there will be $\mathcal{O}\left(3^{|\mathcal{G}|\cdot T}\right)$ extreme points per scenario (assuming that there are no additional constraints that link the generators). The exact number of extreme points depends on the other constraints modelled in $\mathcal{X}_s$, but in general an exponential number of extreme points has to be expected. Instead of enumerating all columns in I_s for each scenario $s \in \mathcal{S}$, we work with a reduced subset of columns in the RMP. To determine whether the restricted solution is optimal for the original MP, we check for every scenario $s \in \mathcal{S}$ if there are any solutions in $\mathcal{X}_s$ which can improve the RMP's objective value. Let $(\lambda_s^\alpha, \lambda_s^p, \sigma_s)$ be the dual RMP solution for scenario s. Then the reduced cost of column $i \in I_s$ of scenario $s \in \mathcal{S}$ is given by

$$\bar{c}_{si} := \pi_s c_{si} - \sum_{g\in\mathcal{G}} \sum_{b\in\mathcal{B}:s\in\mathcal{S}_b} \sum_{t=t_b^{st}}^{t_b^{end}} \left(\lambda_{gbts}^p P_{gtsi} + \lambda_{gbts}^\alpha A_{gtsi} \right) - \sigma_s. \tag{3.8}$$

The RMP objective value can be improved if this is negative for any scenario $s \in \mathcal{S}$. For each scenario, finding the smallest reduced cost amounts to solving the s-th pricing

subproblem, given by:

$$\min_{(\alpha_s,\gamma_s,\eta_s,p_s)\in\mathcal{X}_s} \quad \pi_s \sum_{t=1}^{T} \sum_{g\in\mathcal{G}} f_g\left(\alpha_{gts}, \gamma_{gts}, \eta_{gts}, p_{gts}\right) \tag{3.9}$$
$$- \sum_{g\in\mathcal{G}} \sum_{b\in\mathcal{B}:s\in\mathcal{S}_b} \sum_{t=t_b^{st}}^{t_b^{end}} \left(\lambda^p_{gbts} p_{gts} + \lambda^\alpha_{gbts} \alpha_{gts}\right) - \sigma_s.$$

Assuming that the subproblems can be solved to optimality, their solutions yield the minimum reduced cost for all scenarios. If any of these are negative, the corresponding solutions are added as columns to the RMP. Otherwise we terminate with an optimal solution of the MP.

Overview of the Method. The ColGen procedure avoids solving the MP with all extreme points of $\text{conv}(\mathcal{X}_s)$, $s \in \mathcal{S}$, whose number is prohibitively large. We start with a small number of columns in the RMP. Every time a new set of columns is added, we solve the RMP again, pass its new dual solution to the subproblems and repeat. A flowchart for the basic ColGen method is shown in Figure 3.2. The solution method described in this section is essentially an application of the algorithm described in [32] to the SUC problem. However, to arrive at a practical version of this algorithm, additional analysis is required, including a dual reformulation of the RMP.

Since the MP is a convex relaxation of the SUC problem, solving it provides a *lower bound* on the optimal objective value [41]. Any feasible solution of the SUC gives an *upper bound*, and the optimal solution gives the best possible upper bound. The gap between the best upper bound and the lower bound from the optimal MP solution is the *duality gap*. In practice the gap getween the best known bounds is bigger than the duality gap: the optimal solution of the SUC may not have been found and only a lower bound on the optimal MP solution may be known. If the obtained gap is not sufficiently small it may be necessary to perform branching on the RMP [53]. If the overall gap is too large because there is a positive duality gap then branching will always be necessary. However, if the gap is too large because a suboptimal primal solution has been found, heuristics can be a cheaper alternative than branching to find better solutions and upper bounds. In all examples on which we tested our algorithm, it achieved sufficiently small gaps at the root node of the Branch & Price tree, so no

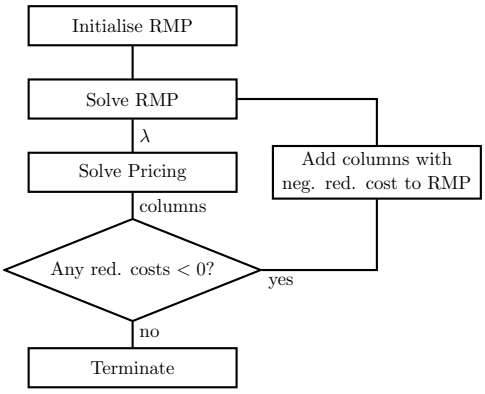


Figure 3.2: Flowchart for the basic ColGen method. On termination, we obtain the solution of a convex relaxation of the original SUC problem. The gap between the best known integer solution and the solution of the relaxation can be closed by branching on the variables of the original problem and repeating this ColGen procedure at each node of the Branch & Price tree.

branching was ever needed.

3.2 Practical Aspects of Scenario Decomposition

A plain ColGen procedure as described above has practical drawbacks which affect its convergence speed. The following problems are common in ColGen applications and need to be accounted for by appropriate algorithmic modifications. The names are due to Vanderbeck [53].

1. Heading-in effect. In the first iterations the RMP lacks a sufficient variation of columns to produce good primal and dual solutions.

2. Plateau effect. While multiple *dual* optimal solutions exist, the *primal* RMP solution is often observed to remain constant over multiple iterations.

3. Bang-bang effect. Dual RMP solutions jump from one extreme value to another and cause fluctuations in the convergence of the lower bound.

4. Tailing-off effect. After some initial progress, many additional columns need to be generated before optimality can be proven.

We address the former two effects by deriving a powerful heuristic to find primal solutions and hot-starting our procedure with dual estimates from an LP relaxation. The

latter two issues are alleviated by using a proximal bundle method to stabilise the dual iterates. Since deterministic UC problems are challenging problems themselves, the subproblems cannot always be solved to optimality, and we explain how this is handled in our approach. For a description of simpler alternative stabilisations, see [41].

3.2.1 Dual Stabilisation of the RMP

In the previous chapter we have seen that LR and DW decomposition are closely related: both can be used to obtain separability in the same way, and both give the same lower bound for a given set of multipliers. If the Lagrangian dual problem is solved by a cutting plane algorithm, the LR and ColGen procedures are identical since the RMP and the cutting plane problem form a primal-dual pair of LPs. We work with the cutting plane problem because it provides a natural motivation for dual stabilisation via proximal bundle methods. Let λ^α be the dual variables of (3.3), λ^p the duals of (3.4) and σ those of constraints (3.5). The stabilised cutting plane problem is given by

$$\text{dRMP}(\rho): \quad \max_{\sigma,\lambda} \sum_{s\in\mathcal{S}} \sigma_s - \frac{\rho}{2}\left(\|\lambda^p - \hat{\lambda}^p\|_2^2 + \|\lambda^\alpha - \hat{\lambda}^\alpha\|_2^2\right) \tag{3.10}$$

$$\text{s.t.} \quad \sigma_s \le \pi_s c_{si} - \sum_{g\in\mathcal{G}} \sum_{b\in\mathcal{B}: s\in\mathcal{S}_b} \sum_{t=t_b^{st}}^{t_b^{end}} \left(\lambda^p_{gbts} P_{gtsi} + \lambda^\alpha_{gbts} A_{gtsi}\right), \ \forall s\in\mathcal{S}, \ i\in I_s \tag{3.11}$$

$$\sum_{s\in\mathcal{S}_b} \lambda^p_{gbts} = 0, \ \forall g\in\mathcal{G}, \ b\in\mathcal{B}, \ t = t_b^{st},\dots,t_b^{end} \tag{3.12}$$

$$\sum_{s\in\mathcal{S}_b} \lambda^\alpha_{gbts} = 0, \ \forall g\in\mathcal{G}, \ b\in\mathcal{B}, \ t = t_b^{st},\dots,t_b^{end} \tag{3.13}$$

$$\lambda^p, \ \lambda^\alpha, \ \sigma \text{ free.} \tag{3.14}$$

When $\boldsymbol{\rho = 0}$ this is the LP dual of the RMP. The artificial variables $\bar{\alpha}_{gbt}$ and $\bar{p}_{gbt}$ of (3.3) and (3.4) translate to dual constraints (3.12) and (3.13), respectively. These constraints force non-anticipativity multipliers to sum up to zero among bundled scenarios. They remove the additional dimension introduced to the dual problem by having one additional non-anticipativity constraint in every bundle (in the non-anticipativity formulation with common target variables). The maximum possible value of $\sum_s \sigma_s$ subject to constraints (3.11), (3.12) and (3.13) gives an upper PWL approximation

to the original SUC problem's Lagrangian dual function, and when the I_s contain *all* extreme points of $\text{conv}(\mathcal{X}_s)$ it gives the exact Lagrangian dual function [40, 41]. Every column of the RMP corresponds to one supporting hyperplane. Without a sufficient variety of cutting planes (3.11), the PWL model of the Lagrangian is too optimistic and encourages too large dual steps. The duals λ^α and λ^p are unrestricted and have zero coefficients in the objective, so arbitrarily large values are possible if they are compensated for by small values which ensure that (3.12) and (3.13) hold. The columns or cuts generated from such large multipliers are usually not meaningful in that they do not form part of a good primal solution.

With $\boldsymbol{\rho > 0}$ the procedure is stabilised through the quadratic bundle term centered on the current dual iterate $(\hat{\lambda}^\alpha, \hat{\lambda}^p)$. Controlled by the steplength parameter ρ, the stabilised problem dRMP(ρ) strikes a balance between maximising the current PWL model of the Lagrangian and not moving too far from the last successful iterate. It produces a candidate solution $(\tilde{\lambda}^\alpha, \tilde{\lambda}^p)$ which we either accept by setting $(\hat{\lambda}^\alpha, \hat{\lambda}^p) := (\tilde{\lambda}^\alpha, \tilde{\lambda}^p)$ (serious step) or reject by keeping the old iterate (null step). The step is accepted if the new iterate improves the current lower bound on the MP solution, which corresponds to a choice of $\kappa = 0$ in Algorithm 2.3. Irrespective of whether the step is accepted or rejected, cutting planes are added to dRMP(ρ) for all subproblem solutions which have negative reduced cost. After adding the cuts, dRMP(ρ) is solved again. For additional convergence improvements, we use a variable steplength logic: we increase ρ if the lower bound has deteriorated, keep it constant if the bound has improved, and decrease it if the bound has not improved over multiple iterations. Additional stability is gained by allowing the subproblems to add multiple cutting planes with negative reduced cost in every iteration.

3.2.2 Lower Bounds for the MP

The Lagrangian dual function for our SUC problem is

$$\begin{aligned}\theta(\lambda^\alpha, \lambda^p) \quad = \quad & \min_{(\alpha,\gamma,\eta,p)\in\mathcal{X}} \Bigg(\sum_{s\in\mathcal{S}} \pi_s \sum_{t=1}^{T} \sum_{g\in\mathcal{G}} f_g\left(\alpha_{gts}, \gamma_{gts}, \eta_{gts}, p_{gts}\right) \\ & - \sum_{g\in\mathcal{G}} \sum_{b\in\mathcal{B}} \sum_{s\in\mathcal{S}_b} \sum_{t=t_b^{st}}^{t_b^{end}} \left(\lambda^\alpha_{gbts}(\alpha_{gts} - \bar{\alpha}_{gbt}) + \lambda^p_{gbts}(p_{gts} - \bar{p}_{gbt}) \right) \Bigg)\end{aligned}$$

$$\begin{aligned} \overset{(3.12),(3.13)}{=} \quad & \sum_{s\in\mathcal{S}} \min_{(\alpha_s,\gamma_s,\eta_s,p_s)\in\mathcal{X}_s} \Bigg(\pi_s \sum_{t=1}^{T} \sum_{g\in\mathcal{G}} f_g\left(\alpha_{gts},\gamma_{gts},\eta_{gts},p_{gts}\right) \\ & - \sum_{g\in\mathcal{G}} \sum_{b\in\mathcal{B}:s\in\mathcal{S}_b} \sum_{t=t_b^{st}}^{t_b^{end}} \left(\lambda^{\alpha}_{gbts}\alpha_{gts} + \lambda^{p}_{gbts}p_{gts}\right) \Bigg) \\ = \quad & \sum_{s\in\mathcal{S}} \theta_s(\lambda^{\alpha}_s,\lambda^{p}_s), \end{aligned} \tag{3.15}$$

and for feasible multipliers λ^α and λ^p this provides a lower bound on the optimal MP value. The terms in $\bar{\alpha}$ and $\bar{p}$ vanish due to dual constraints (3.12) and (3.13), respectively. For given multipliers λ^α, λ^p and σ, we have that $\theta_s(\lambda^\alpha_s,\lambda^p_s) = \bar{c}^*_s + \sigma_s$, so the lower and upper bounds $\underline{z}$ and $\bar{z}$ on the optimal MP objective value z^* are given by

$$\underline{z} := \theta(\lambda^\alpha,\lambda^p) = \sum_{s\in\mathcal{S}}(\sigma_s + \bar{c}^*_s) \le z^* \le \sum_{s\in\mathcal{S}} \sigma_s =: \bar{z}. \tag{3.16}$$

The lower bound is obtained by solving the pricing subproblems (3.9) and adding their reduced costs to the current RMP objective value $\bar{z}$. For the special choice $\lambda^\alpha = \lambda^p = 0$, the scenario subproblems are solved independently under perfect information, and the resulting bound $\underline{z}$ is called the expected value *under* perfect information. The gap between the optimal SUC objective value and this bound is known as the expected value *of* perfect information (EVPI).

We solve the subproblems (3.9) with a Branch & Cut solver and since they are large MIPs, optimality cannot always be guaranteed. The solver terminates with a set of (sub)optimal solutions and a lower bound on the objective, i.e. the reduced cost. Let this bound be $\bar{c}^{lb}_s \le \bar{c}^*_s$. Then, instead of (3.16) we use

$$\underline{z} = \bar{z} + \sum_{s\in\mathcal{S}} \bar{c}^{lb}_s \tag{3.17}$$

as a valid lower bound for the MP. To decide if the (sub)optimal solutions should be added as columns to the RMP, we evaluate their reduced costs individually and test them for negativity. The termination criterion for the ColGen procedure is adapted to allow for subproblem non-optimality: we stop when the following overestimate of the relative MP gap

$$\delta^{MP} := \frac{-\sum_{s\in\mathcal{S}} \bar{c}^{lb}_s}{\bar{z}} \tag{3.18}$$

satisfies a pre-defined optimality tolerance. The smallest achievable gap δ^{MP} depends on the MIP gaps of the pricing subproblems.

3.2.3 Dual Initialisation of the RMP

In presence of the quadratic stabilisation term, dRMP(ρ) can theoretically be solved without any cutting planes at all, if additional box constraints on σ_s are used to prevent unboundedness of the objective. In that case the resulting dual solution is the provided initial point $(\hat{\lambda}^\alpha, \hat{\lambda}^p)$, and the first pass of ColGen is performed with these multipliers. A simple dual initial guess which requires no additional effort is $\hat{\lambda}^\alpha = \hat{\lambda}^p = 0$, and this can be used to cold-start the ColGen procedure. The first lower bound is then given by the objective value under perfect information and the upper bound is determined by the box constraints on σ_s, so can be expected to be poor. The method would proceed by producing a series of poor upper bounds until there is a primal feasible (non-anticipative) solution among the generated columns. However, this can take many iterations. Additionally, the initial lower bound will also be poor unless the EVPI is small. The convergence process can be sped up significantly if better initial upper and lower bounds are obtained by

1. Providing a good initial estimate of the multipliers $(\hat{\lambda}^\alpha, \hat{\lambda}^p)$.

2. Providing initial columns among which there is a non-anticipative solution.

In this section we describe how we obtain the initial dual iterate, while the next section is dedicated to finding a primal feasible solution.

We obtain an initial dual point by solving the LP relaxation of the original SUC problem and extracting its dual solution. This can be done in different ways: we can solve the extensive formulation of the SUC problem's LP relaxation or apply the same scenario decomposition as in the integer case. In Section 3.3 we report on test results with different strategies. Due to the redundancy of non-anticipativity constraints, the dual solutions of the LP relaxation of the extensive form are similarly degenerate as those of the RMP. We explore the effect of using different dual optimal solutions of the relaxation as starting points for the ColGen algorithm. This requires control over the obtained dual solution. In the following we outline briefly how a quadratic penalty term can be included in the *primal* relaxed problem in order to obtain *dual* optimal solutions

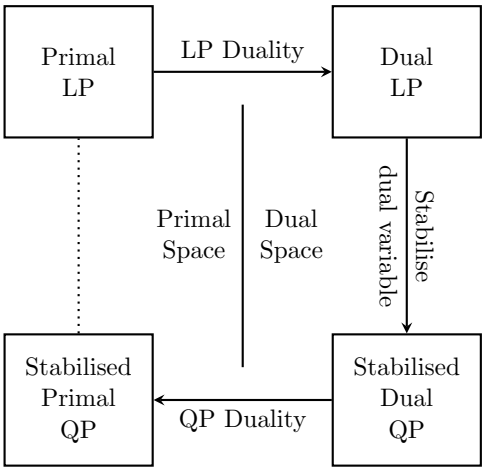


Figure 3.3: Dual stabilisation of the initial LP relaxation. We dualise the initial LP to include a quadratic penalty term on some of the dual variables. The obtained QP is then dualised again to obtain its primal equivalent and thus reveal how to dually stabilise a primal problem without explicitly formulating its dual.

of various sizes. This is more convenient to implement than a dual reformulation of the SUC problem's LP relaxation. The approach is visualized in Figure 3.3.

The penalty term is centered on zero and favours dual solutions of least magnitude. This idea is similar to centering the proximal term in dRMP(ρ) on zero by simply initialising $\hat{\lambda}^{\alpha} = \hat{\lambda}^{p} = 0$. However, in the absence of a sufficient number of cutting planes this can shrink the dual solution, and if zero is a bad multiplier estimate this will hinder progress by suppressing the generation of useful solutions. The SUC relaxation, on the other hand, can be thought of as a ColGen problem in which all possible columns have been generated. This is equivalent to saying that all cutting planes have been included in the corresponding dual, in which case the criterion of choosing the smallest optimal multipliers is more sensible.

Consider the following primal-dual pair of linear programs where the primal has two sets of equality constraints.

$$\begin{aligned} \min\ & c^T x && (3.19) \\ \text{s.t. } & Ax = b \\ & Bx = d \\ & x \geq 0 \end{aligned} \qquad \begin{aligned} \max\ & b^T\gamma + d^T\lambda && (3.20) \\ \text{s.t. } & A^T\gamma + B^T\lambda \leq c \\ & \gamma, \lambda \text{ free} \end{aligned}$$

Here γ are the dual variables associated with $Ax = b$, and λ are the dual variables associated with $Bx = d$. We assume that these are degenerate in that there is a continuum of dual optimal solutions λ. To favour dual solutions of smaller magnitude, we include a quadratic penalty term with a small $\mu > 0$ in the dual problem, giving (3.22). Then we obtain the equivalent primal formulation of the stabilised dual by dualising it again. The primal quadratic problem corresponding to (3.22) is given by (3.21).

$$\begin{aligned} \min\ & c^T x + \tfrac{1}{2}\mu\lambda^T\lambda \qquad (3.21) \\ \text{s.t. } & Ax = b \\ & Bx + \mu\lambda = d \\ & x \geq 0,\ \lambda \text{ free} \end{aligned} \qquad \begin{aligned} \max\ & b^T\gamma + d^T\lambda - \tfrac{1}{2}\mu\lambda^T\lambda \qquad (3.22) \\ \text{s.t. } & A^T\gamma + B^T\lambda \leq c \\ & \gamma, \lambda \text{ free} \end{aligned}$$

In problem (3.21) the variables λ are equal to the dual variables of constraints $Bx + \mu\lambda = d$. To make it easier to interpret this stabilised primal problem, we can write it equivalently as (3.23), with $\tilde{\lambda} = \mu\lambda$.

$$\begin{aligned} \min\ & c^T x + \tfrac{1}{2\mu}\tilde{\lambda}^T\tilde{\lambda} \qquad (3.23) \\ \text{s.t. } & Ax = b \\ & Bx + \tilde{\lambda} = d \\ & x \geq 0,\ \tilde{\lambda} \text{ free} \end{aligned} \qquad \begin{aligned} \min\ & c^T x \qquad (3.24) \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$

In problem (3.23), $\tilde{\lambda} = d - Bx$ is the violation of constraints $Bx = d$. This violation must be paid for in the objective, but its price decreases if μ is increased. In the SUC problem, $\tilde{\lambda}$ is bounded, so we have $\frac{1}{2\mu}\tilde{\lambda}^T\tilde{\lambda} \to 0$ for $\mu \to \infty$. This means that – for large μ – the constraints $Bx = d$ are relaxed, because violating them is free. Thus the solution of (3.23) converges to the solution of (3.24), and the dual variables of the relaxed constraints approach zero: $\lambda = \frac{1}{\mu}\tilde{\lambda} \to 0$. It is the value of these dual variables that we are after, so μ must be chosen with care. We use the magnitude of the violation, $\|\tilde{\lambda}\|_2$, to measure if the level of μ is appropriate. If it is non-negligible, then μ is too large.

We use this approach to regularise the duals of the non-anticipativity constraints in the LP relaxation of the SUC problem: We choose a perturbation level μ and include copies of the dual variables λ^α and λ^p in the objective and the non-anticipativity constraints of the primal problem. The resulting stabilised SUC LP relaxation is shown below.

$$\text{LPR}(\mu): \quad \min_{\alpha,\gamma,\eta,p} \sum_{s\in\mathcal{S}} \pi_s \sum_{t=1}^{T} \sum_{g\in\mathcal{G}} f_g\left(\alpha_{gts}, \gamma_{gts}, \eta_{gts}, p_{gts}\right) \tag{3.25}$$

$$+\frac{\mu}{2}\left(\|\lambda^p\|_2^2 + \|\lambda^\alpha\|_2^2\right)$$

$$\text{s.t.} \quad (\alpha_s, \gamma_s, \eta_s, p_s) \in \mathcal{X}_s, \ \forall s \in \mathcal{S} \tag{3.26}$$

$$\alpha_{gts} + \mu\lambda^\alpha_{gbts} = \bar{\alpha}_{gbt}, \ \forall g \in \mathcal{G}, \ b \in \mathcal{B}, \ s \in \mathcal{S}_b, \ t = t_b^{st}, \ldots, t_b^{end} \tag{3.27}$$

$$p_{gts} + \mu\lambda^p_{gbts} = \bar{p}_{gbt}, \ \forall g \in \mathcal{G}, \ b \in \mathcal{B}, \ s \in \mathcal{S}_b, \ t = t_b^{st}, \ldots, t_b^{end} \tag{3.28}$$

3.2.4 MIP Heuristics

A central issue of LR and DW decomposition for mixed-integer problems is that integer primal solutions may not be found unless the master problem is branched on [41]. Primal solutions are required to find upper bounds for the problem and eventually solve it, and we use MIP-based heuristics to construct them. This is done in the beginning of the ColGen process: the results discussed in Section 3.3 confirm that the decomposition is sped up significantly if a good solution is known early on.

A Schedule Combination Heuristic. Our core heuristic is a MIP-based schedule combination (SC) heuristic which constructs a non-anticipative solution for the SUC problem. It is inspired by Takriti and Birge's refinement heuristic for deterministic problems [56]. The idea is to solve the SUC problem as a MIP, but with the solution space reduced to a set of pre-defined schedules. Here a schedule is a plan of binary on-off decisions for a *single generator* g for the *whole planning horizon* $t = 1, \ldots, T$. Let n_g be the number of known schedules for generator g and $\tilde{A}_{gjt}$ the on-off decision at time t under its j-th schedule. Note that $\tilde{A}_{gjt}$ has no scenario index s, since the same universe of schedules is used for *all scenarios*. We formulate a stochastic mixed-integer schedule selection problem, which – under every scenario – picks one of the schedules for each of the generators, subject to satisfying non-anticipativity. It uses weight variables

w_{gjs} which take a value of one if generator g chooses schedule j under scenario s, and zero otherwise. The schedule selection problem is obtained by *adding* variables w_{gjs} and the following constraints to the original SUC problem:

$$\sum_{j=1}^{n_g} w_{gjs}\tilde{A}_{gjt} = \alpha_{gts},\ \forall g \in \mathcal{G},\ t = 1,\dots,T,\ s \in \mathcal{S} \tag{3.29}$$

$$\sum_{j=1}^{n_g} w_{gjs} = 1,\ \forall g \in \mathcal{G},\ s \in \mathcal{S} \tag{3.30}$$

$$w_{gjs} \in \{0,1\},\ \forall g \in \mathcal{G},\ s \in \mathcal{S},\ j = 1,\dots,n_g. \tag{3.31}$$

Constraints (3.29) ensure that if schedule j is chosen for generator g under scenario s, its binary decisions α_{gts} are equal to $\tilde{A}_{gjt}$ at *all* times $t = 1,\dots,T$. For a given generator g and scenario s, constraints (3.30) say that exactly one of the binary weights w_{gjs} can be one, while all others are zero. The weights are then said to form a specially ordered set of order 1 (SOS1). As the generators must follow their chosen schedules for the whole planning period, some of the constraints which are typically formulated in $\mathcal{X}_s$ can be dropped. For instance minimum up/down times will be satisfied by every schedule, so the constraints are no longer required.

The schedules $\tilde{A}_{gjt}$ are generated through ColGen iterations: after solving the subproblems, we add all distinct schedules that appear in their solutions to $\tilde{A}_{gjt}$. This is done individually for each generator $g \in \mathcal{G}$. Schedules from different scenario subproblems are pooled in $\tilde{A}_{gjt}$ and made available to all other scenarios. Thus scenarios can exchange their own schedule for one proposed by another scenario on a generator-by-generator basis. The tests confirm that this produces good solutions even if $\tilde{A}_{gjt}$ is populated from a pass of ColGen with multipliers $\lambda^\alpha = \lambda^p = 0$, i.e. a perfect foresight solution. However, better solutions are obtained if the multipliers are such that they encourage the subproblems to generate schedules which are a compromise between bundled scenarios. The SC heuristic can be used to:

1. Initialise dRMP(ρ) with a non-anticipative solution: we populate $\tilde{A}_{gjt}$ from subproblem solutions of the first pass of ColGen, performed with multipliers equal to zero or estimated from LPR(μ).

2. Find solutions during the ColGen process: in every iteration, $\tilde{A}_{gjt}$ is extended by

new schedules from the subproblems. The heuristic is repeated and improving solutions are added to dRMP(ρ).

The SC problem is solved with a general purpose MIP solver. We apply integrality restrictions to the SOS1 weights w_{gjs} and relax the on-off variables, since this improved solver performance. Additionally, the problem is pre-processed: we iterate over all on-off variables, and whenever the schedule set allows this generator only to be on or only to be off at a certain time, the corresponding variable is fixed and removed from the problem. For generators with only one schedule, the weights and on-off variables are all fixed and eliminated. In the test runs described in Section 3.3, roughly 70% of the weights and on-off variables are eliminated in this way.[1]

An Over-Commitment Heuristic. MIP solvers typically use local search heuristics to improve upon existing solutions, and it is often beneficial for the convergence speed if a reasonable initial solution is constructed and provided as input. To do this for the SC problem, we use a cheap heuristic which attempts to repair a conflicting schedule obtained from the scenario subproblems. It generates a non-anticipative solution by committing more than the required generation capacity. This over-commitment heuristic works as follows:

1. Estimate values for λ^α and λ^p, e.g. by solving LPR(μ) or setting them to zero
2. Solve the column generator subproblems with these dual estimates
3. Consider the obtained schedule: for each generator g, proceed in chronological order $t = 1, \ldots, T$:
   - Find bundles b covering period t, whose members $s \in \mathcal{S}_b$ disagree on the commitment α_{gts}
   - For all members $s \in \mathcal{S}_b$ of these bundles, set $\alpha_{g(t+k)s} := 1$ with $k = 0, \ldots, T_g^u - 1$, where T_g^u is the min up-time
4. The schedule is now non-anticipative. Solve a dispatch with fixed binaries to check feasibility.

[1] It is possible to rely on the solver's presolve phase to do this, but we choose not to.

The results of this heuristic can violate minimum down-times, in which case the solution is repaired by eliminating too short down-times and keeping the generators on instead. The solution quality depends on the multiplier estimate. For $\lambda^\alpha = \lambda^p = 0$, many schedule adjustments are necessary and we obtain expensive solutions, while estimating the multipliers from LPR(μ) leads to few adjustments and nearly optimal solutions. Steps 1. and 2. are required before solving the SC heuristic anyway, so applying the over-commitment heuristic adds only a negligible computational cost. For multi-stage UC problems we obtain reasonable starting solutions with both, $\lambda^\alpha = \lambda^p = 0$ and multipliers estimated from LPR(μ). The quality of the obtained solution depends on the EVPI: for problems with non-negligible EVPI zero is not a good multiplier estimate, and estimating them from LPR(μ) gives better solutions, while for problems with negligible EVPI zero works well, too. For two-stage problems many adjustments are necessary when using the over-commitment heuristic, and the following approach usually performs better.

Constructing Solutions for Two-Stage Problems. In the two-stage setting it is straigtforward to construct a solution which can be used as starting point for the SC heuristic. It is possible to solve a deterministic problem, e.g. with an average or low wind scenario, and find the recourse action for the obtained schedule by solving scenario-specific dispatch problems. Feasibility of the approach is guaranteed if the model has relatively complete recourse. In the GB model which we use for our tests, this is warranted by the fact that load shedding and not satisfying reserve are viable options on the second stage. For our test runs, we initialise the SC problem with a solution obtained from a deterministic problem which was augmented with a few additional power output variables. The additional variables are used to approximate the recourse cost: they inform the problem that sufficient capacity must be scheduled to cope even with the lowest wind scenario, or additional costs will be incurred to satisfy the remaining demand via standby reserve. Despite the additional variables, the computational effort is similar to a simple deterministic problem.

These cheaper heuristics are used to provide an initial solution to the SC heuristic. In the following we refer to the SC heuristic including initialisation from one of these approaches simply as the heuristic.

3.2.5 The Stabilised Scenario Decomposition Algorithm

Our scenario decomposition scheme for SUC problems is summarised in Figure 3.4. It solves the DW master problem at the root node of a Branch & Price tree. Dual stabilisation and initialisation techniques are included, as well as the heuristic used to find integer feasible solutions. In all numerical tests the gap between the best integer SUC solution and the RMP objective value was within our required optimality tolerance after solving the root node, so no branching was performed. If branching is necessary, the ColGen procedure can be repeated at every node of the Branch & Price tree, with appropriate modifications so that the master- and subproblems comply with the branching decisions.

3.3 Numerical Experiments

We implemented the ColGen procedure shown in Figure 3.4 and tested it on two-stage and multi-stage versions of our central scheduling model based on the GB power system, with 24 hour-long time steps and up to 50 wind power scenarios. The model has a 30% wind penetration in terms of installed capacity, 130 thermal units and four pump storage plants with a total of 16 pump-turbines. It uses a loss-free real power transmission network model with 17 zones and 27 transmission links between them. Additional restrictions limit the sum of transmissions across a pre-defined set of 17 boundaries. A more detailed discussion of the model and corresponding data sources can be found in Chapter 4. In this model all the uncertainty arises from the unpredictability of wind generation. However, the decomposition technique applies equally well to cases with unpredictable demands or contingency scenarios. The following sections give details of our implementation and summarise our experience with the method.

3.3.1 Details of the Implementation

The decomposition method shown in Figure 3.4 is implemented as a script in AMPL version 20120629 [57]. All LPs and MIPs are solved with CPLEX version 12.4 [58]. We perform the tests on a Dual 8 Core Dell Power Edge C6220 machine with 128 Gb RAM, running 64 bit Linux. The pricing problems are solved in serial, as parallel

Input: iteration counter $k \leftarrow 0$, optimality tolerance Δ, stabilisation levels ρ, μ, upper bound $\bar{z}_0 \leftarrow \infty$, lower bound $\underline{z}_0 \leftarrow -\infty$, gap $\delta_0^{MP} \leftarrow \infty$, iterations until repeating heuristic r, maximum number N of cuts added to dRMP(ρ) per iteration per subproblem;

solve LPR(μ) to find $\hat{\lambda}^\alpha$, $\hat{\lambda}^p$ or set $(\hat{\lambda}^\alpha, \hat{\lambda}^p) \leftarrow 0$;
for $s \in \mathcal{S}$ **do**
  solve subproblem (3.9) with $\hat{\lambda}^\alpha$, $\hat{\lambda}^p$ and $\sigma = 0$;
  for $g \in \mathcal{G}$ **do** add new schedules from opt. subproblem solution to heuristic;
  for *the best subproblem solutions* $i = 1, \ldots, N$ **do**
    calculate $\bar{c}_{si}$ from (3.8);
    if $\bar{c}_{si} < 0$ **then** generate a cut and add it to dRMP(ρ);
  end
end
calculate lower bound $\underline{z}_k$ from (3.17);
repeat
  if $k \mod r \equiv 0$ **then**
    run the heuristic to find $\tilde{\bar{z}}_k$ and $(\tilde{\alpha}^k,\ \tilde{\gamma}^k,\ \tilde{\eta}^k,\ \tilde{p}^k)$;
    if $\tilde{\bar{z}}_k < \bar{z}_k$ **then**
      set current best solution $(\bar{z}_k, \alpha^*, \gamma^*, \eta^*, p^*) \leftarrow \left(\tilde{\bar{z}}_k, \tilde{\alpha}^k, \tilde{\gamma}^k, \tilde{\eta}^k, \tilde{p}^k\right)$;
      calculate δ_k^{MP} from (3.18);
      if $\delta_k^{MP} < \Delta$ **then** terminate;
      for $s \in \mathcal{S}$ **do** generate a cut from $\tilde{\alpha}_s^k,\ \tilde{p}_s^k$ and add it to dRMP(ρ);
    end
  end
  set $k \leftarrow k + 1$;
  solve dRMP(ρ) to find $(\tilde{\lambda}^\alpha,\ \tilde{\lambda}^p,\ \tilde{\sigma})$, $\bar{z}_k$ and $(\alpha^k,\ \gamma^k,\ \eta^k,\ p^k)$;
  if α^k *is integer* **then** set current solution $(\alpha^*, \gamma^*, \eta^*, p^*) \leftarrow \left(\alpha^k, \gamma^k, \eta^k, p^k\right)$;
  for $s \in \mathcal{S}$ **do**
    solve subproblem (3.9) with $\tilde{\lambda}^\alpha$, $\tilde{\lambda}^p$, $\tilde{\sigma}$;
    for $g \in \mathcal{G}$ **do** add new schedules from opt. subprob. solution to heuristic;
    for *the best subproblem solutions* $i = 1, \ldots, N$ **do**
      calculate $\bar{c}_{si}$ from (3.8);
      if $\bar{c}_{si} < 0$ **then** generate a cut and add it to dRMP(ρ);
    end
  end
  calculate $\underline{z}_k$ from (3.17) and δ_k^{MP} from (3.18);
  if $\underline{z}_k > \underline{z}_{k-1}$ **then** set $\hat{\lambda}^\alpha \leftarrow \tilde{\lambda}^\alpha$ and $\hat{\lambda}^p \leftarrow \tilde{\lambda}^p$;
until $\delta_k^{MP} < \Delta$;

Output: dual solution $\hat{\lambda}^\alpha$, $\hat{\lambda}^p$, primal solution $(\alpha^*, \gamma^*, \eta^*, p^*)$, MP objective $\bar{z}$;

Figure 3.4: Scenario decomposition method. The algorithm consists of an initialisation loop and a main loop. The main loop contains the schedule combination heuristic which is repeated every r iterations, and the stabilised ColGen logic.

implementations are not supported by AMPL. However, we use the parallel option for CPLEX, with a maximum of 16 threads. Up to 50 cuts are added to dRMP(ρ) per iteration per subproblem if multiple solutions were found. We set the overall optimality tolerance to $\Delta = 0.1\%$ and the subproblem tolerance to 0.05%, so that at most half of the overall gap is due to the subproblem MIP gaps. In the following, we refer to solutions as optimal if they satisfy the tolerance Δ. The overall gap strikes a balance between computational effort and a sensible accuracy for scheduling under wind uncertainty: on average, 0.1% of the daily cost corresponds to increasing the output by 36MW for the duration of a day. In comparison, the uncertainty in a 3h wind forecast is 840MW, while 24h ahead it is already 2.8GW.

When solving the SC heuristic problem, we relax the integrality of all variables apart from the SOS1 weights, as this works best for CPLEX. The optimality tolerance is left at its default of 0.01%.

3.3.2 Multi-Stage Results

Figure 3.5 shows timings for solving multi-stage stochastic problems with the ColGen method, and solving the extensive form of the same model with CPLEX. We vary the number of scenarios between 3 and 50 and the number of stages between 2 and 4, where each stage covers 3 hours and the final stage covers the remainder of the 24 hours. On small problems, CPLEX and our decomposition work similarly well, while on larger instances the decomposition is superior. The behaviour of the decomposition method suggests that the set of multi-stage test problems can be separated in two groups on which different strategies lead to faster convergence:

1. Problems with negligibly small EVPI. For these 'easy' problems, $\hat{\lambda}^{\alpha} = \hat{\lambda}^{p} = 0$ is a good estimate, so LPR(μ) is not solved. The SC heuristic finds optimal primal solutions after being populated with schedules from a ColGen pass with zero multipliers, and since the first lower bound is the expected value under perfect information, the method terminates after a single iteration.

2. Problems with larger EVPI. If attempted to solve with zero initial multipliers, these 'hard' problems initially yield gaps between 0.5% and 1%. However, if we solve them with multiplier estimates from LPR(μ), we still achieve the required

tolerance of 0.1% in one iteration.

In the second case, the gap with zero multipliers is firstly due to a worse lower bound, i.e. the EVPI being non-negligible, and secondly due to a non-optimal primal solution obtained from the SC problem. The feasible region of the SC problem does not contain an optimal primal solution unless a good multiplier estimate is provided. However, estimating this with LPR(μ) is time consuming for large instances, and where possible we avoid it by applying the following rule to separate easy problems from hard ones: we first solve the subproblems with zero multipliers and use the overcommitment heuristic to produce a primal solution. This provides an upper bound and a Lagrangian lower bound, and if the gap between them exceeds 1% then the problem is hard, otherwise it is easy. For easy problems we continue with the ColGen algorithm by solving the SC heuristic, while for hard problems we first solve LPR(μ) and re-solve the subproblems before continuing. The separation works well on all test cases: on hard problems the gap between overcommitment solution and first lower bound was always at least 3%, while easy problems seldomly reached 0.6%.

To estimate the proportion of hard and easy problems in our GB model, we evaluate a pool of 3,000 multi-stage problems obtained from the long term rolling horizon study described in Chapter 4. This showed that roughly 25% of the problems are hard, while 75% are easy. The timings in Figure 3.5 are weighted accordingly: every data point corresponds to the average solution time of one hard problem and three easy ones, i.e. they represent an expected solution time for an 'average' problem. The same examples were used to obtain CPLEX's timings, but here the solution times are unaffected by whether a problem is easy or hard for the decomposition.

Solving hard problems via decomposition requires solving LPR(μ) with CPLEX's barrier solver, whose CPU timings scale unfavourably in the number of scenarios (cf. next section), and the overall solution times are affected by this. The other major factor that contributes to the time spent in the decomposition is the SC problem, and solution times for that are shown separately in Figure 3.5. The amount of parallelism in the decomposition increases slightly in the number of scenarios: the CPU to elapsed time ratio is between 3:1 on small problems and 4:1 on larger ones, and this increase is due to the barrier solver which is used to solve LPR(μ) in the hard cases. Further parallel

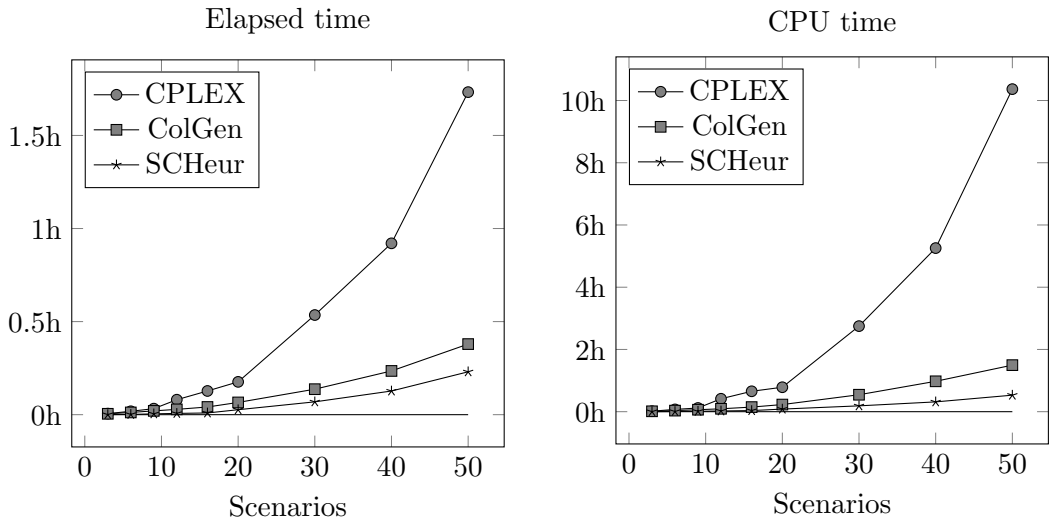


Figure 3.5: Elapsed times (left) and CPU times (right) to solve the multi-stage stochastic problem to within 0.1% of optimality, plotted over the included number of scenarios. We solved four different instances of each size of problem and show their average results. The different curves show timings for solving the extensive formulation via CPLEX and applying ColGen. Additionally, we show timings for solving the SC problem (SCHeur).

speedups are possible if the subproblems are solved in parallel. Without decomposition, CPLEX achieves a larger increase in parallelism with the number of scenarios: the CPU to elapsed time ratio increases from 3:1 to 6:1, so the superiority of the decomposition is more evident on the CPU time graph.

The fact that an optimal solution is provided by the first SC problem suggests another way of proving optimality: we pass the SC solution to CPLEX and ask it to solve the extensive form with the given starting point, by solving an LP relaxation at the root node and adding MIP cuts to tighten the bound. The resulting timings, however, were not better than when solving the whole problem via CPLEX without an initial solution, so we omit them here.

3.3.3 Two-Stage Results

Two-stage SUC problems are harder to solve than multi-stage problems. Figure 3.6 shows elapsed times and CPU times required to solve test cases with 3 to 50 scenarios. As above, we compare the scenario decomposition method to solving the extensive form via CPLEX. For the extensive form we use a single set of first stage variables instead of scenario-specific copies and non-anticipativity constraints. Solution times are generally

higher than in the multi-stage setting.

In our test set there were no two-stage cases with negligible EVPIs, so all cases are 'hard' for the decomposition. Our ColGen method still converged in the first iteration, however, to achieve this it was not sufficient to initialise the duals to zero. With zero duals, the optimality gap achieved after solving the RMP for the first time is typically 1.5% or worse. To find good initial dual estimates, we solve LPR(μ), and Figure 3.6 includes separate timings for that and for solving the SC heuristic as well. Both contribute significantly to the overall time required by the decomposition, and so the time saving resulting from using decomposition instead of CPLEX is smaller than in the multi-stage case. On the other hand, the ratio of decomposition CPU time to elapsed time is now higher: it varies between 4:1 in the smallest case and 6:1 in the largest case. This is due to CPLEX's barrier solver which achieves high parallel speed-ups when used to solve LPR(μ). Most of the time in the decomposition algorithm is spent solving LPR(μ), and we experiment with the following alternative method to estimate initial duals by solving the relaxation in a decomposed way:

1. Apply ColGen with $\hat{\lambda}^\alpha = \hat{\lambda}^p = 0$ initially, but relax integrality restrictions of the subproblems. Solve this relaxation to a gap of $\Delta = 0.05\%$.

2. Use the final multipliers as initial stability center $\hat{\lambda}$ for the integer decomposition.

The resulting duals are a better initial point for the integer decomposition than zero: the gap obtained after solving the first master problem is typically 0.3%. Smaller gaps can be achieved by solving the relaxation to a better accuracy than 0.05%, however the convergence of both, relaxed and integer decomposition is slow, and eventually it takes longer to solve the relaxation by ColGen than with CPLEX's barrier solver. Overall, we achieve better performance with LPR(μ), and the results in Figure 3.6 use the duals from that.

As before, the SC heuristic scales well in the number of scenarios. However, it requires good dual estimates: with schedules generated from a ColGen pass with $\lambda = 0$, the SC heuristic solution is up to 0.2% worse than with dual estimates from LPR(μ). After finding an optimal solution with the SC heuristic, we also try passing it to CPLEX to prove optimality at the root node. Again, this approach is not competitive with out-of-the-box CPLEX, so we omit the results here.

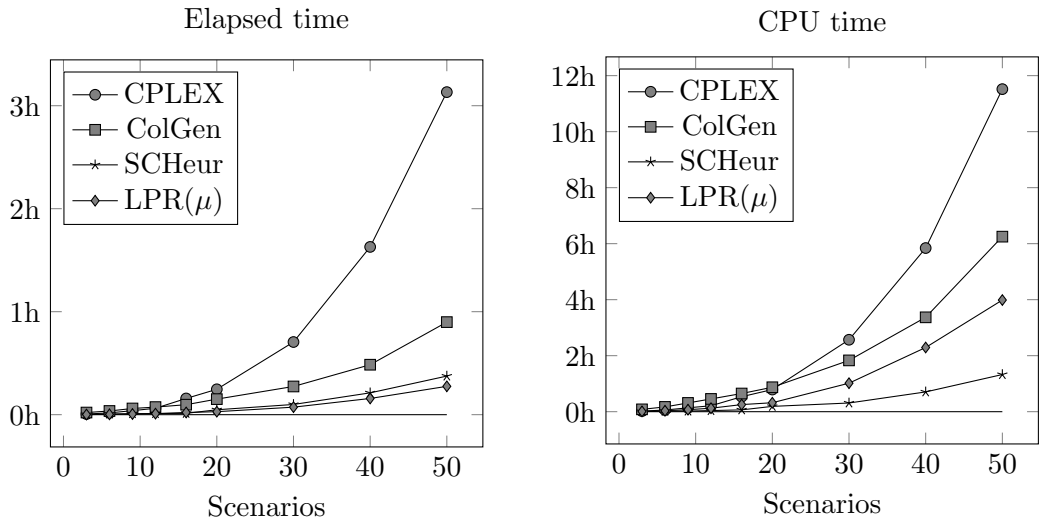
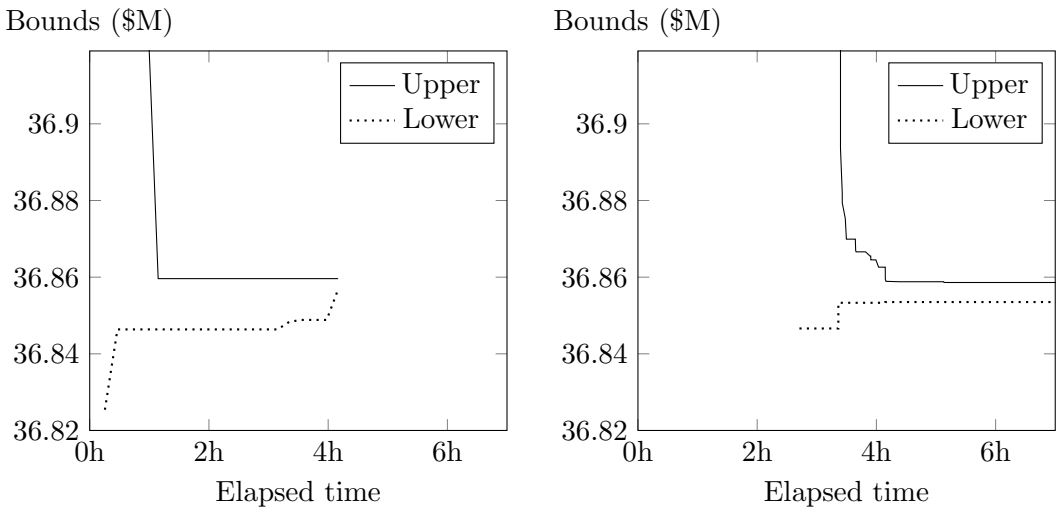


Figure 3.6: Elapsed times (left) and CPU times (right) to solve the two-stage stochastic problem to within 0.1% of optimality, plotted over the included number of scenarios. We solved three different instances of each size of problem and show average performance results. The different cases show timings for CPLEX and applying ColGen. We also show time requirements for solving LPR(μ) and the SC heuristic (SCHeur), which cause the majority of the time spent in the decomposition (besides ColGen passes). For LPR(μ) we used $\mu = 10^{-6}$.

3.3.4 Convergence of Bounds

In the tests described in the previous sections, all problems were solved after a single ColGen pass. The major computational work was in estimating a dual solution from LPR(μ) and constructing a primal optimal solution with the heuristic. While this is sufficient to obtain 0.1% gaps, more work is required to achieve smaller gaps. In the following we briefly discuss convergence properties of CPLEX and the decomposition method to a gap of 0.01%. In the decomposition this requires that subproblems are also solved to a gap of 0.01%. Figure 3.7 shows the convergence of upper and lower bounds as a function of elapsed time on a typical 50 scenario example.

The decomposition obtains the first lower bound after 15 minutes by solving LPR(μ) with CPLEX's barrier method. The lower bound is improved swiftly through a pass of ColGen, i.e. an evaluation of the Lagrangian dual. The first upper bound is obtained immediately thereafter, when an initial solution is constructed for the SC heuristic. This bound is not shown on the graph since it exceeds \$36.9M. The upper bound is first improved when the SC heuristic finds a solution of roughly \$36.86M which is also



(a) Convergence of bounds in ColGen. (b) Convergence of bounds in CPLEX.

Figure 3.7: Convergence of upper and lower bounds in decomposition (3.7a) and in CPLEX (3.7b), as a function of elapsed time. The shown example is one of the two-stage 50 scenario cases that were evaluated for the performance comparison in Figure 3.6, however in this case we solve it to a much smaller gap.

the final solution the decomposition terminates with. At that stage the gap is below 0.1%, but it takes multiple iterations of column generation before the lower bound is raised further. Finally, a gap of 0.007% is achieved after 4 hours, and the method terminates.

CPLEX achieves the first lower bound after 2.5 hours, by solving the root relaxation with a simplex method and adding the first set of MIP cuts. After another hour of root node processing, the cut generation procedure improves the lower bound, and the MIP heuristics produce the first good solution, which is then gradually improved. After 4 hours, CPLEX achieves a solution that is within 0.013% of optimality, however the lower bound cannot be raised any further by adding MIP cuts. Branching is required to close the gap, and this takes a very long time. After 12 hours, we have interrupted CPLEX, and at this stage the bounds have not changed in comparison to those shown on the graph: the procedure has stalled.

The example demonstrates the strengths of the decomposition in comparison to CPLEX's standard Branch & Cut procedure. The first bound and a near-optimal solution are obtained early in the process, resulting in an initial gap below 0.1%. CPLEX, on the other hand, takes a lot longer to produce the first solution and lower bound. To

achieve the 0.1% tolerance, it requires roughly 3.5 hours. Following the initial effort, the decomposition takes a while to converge to the stricter 0.01% tolerance, and CPLEX achieves almost the same accuracy in the same overall time (4 hours), by further improving the primal solution. However, no more progress in the lower bound is achieved, so that it stalls with a gap of 0.013%. At any time the gap of the decomposition method is smaller than that of CPLEX.

We further explore the behaviour of CPLEX and the decomposition method when requiring them to converge to very small gaps, by applying them to one of the test problem sets used to obtain the results in Figure 3.6, with a target tolerance of 0.005%. Both, CPLEX and the decomposition achieve the target for all but the two largest cases. On the large cases, CPLEX stalls at an average gap of 0.01% and the decomposition at 0.006% due to subproblem non-optimality. In cases where the methods did not stall, the elapsed time in the decomposition algorithm was on average half the time spent in CPLEX. On average, 10 ColGen iterations were required.

Lower Bounds and optimal Cutting Planes. One of the major steps in the convergence process of the decomposition is to construct an optimal primal solution, and doing this is useful in its own right, since ultimately it is that solution which we are after. Besides that, however, we also explore the effect of this optimal cutting plane on the lower bound obtained with the duals produced by dRMP(ρ). To do this, we run the ColGen method without providing dRMP(ρ) with a cutting plane derived from a heuristically constructed primal solution. We compare the resulting lower bound to the bound that would have been obtained from the first solve of dRMP(ρ) if the optimal cutting plane had been included. The same stability center is used both times. For two-stage problems, the bounds without the optimal cutting plane are between 0.5% and 3% worse than the ones obtained with it. In multi-stage problems, excluding the optimal plane leads to even worse lower bounds, some of which are negative. This demonstrates that knowing an optimal primal solution is also beneficial to the convergence of dual bounds.

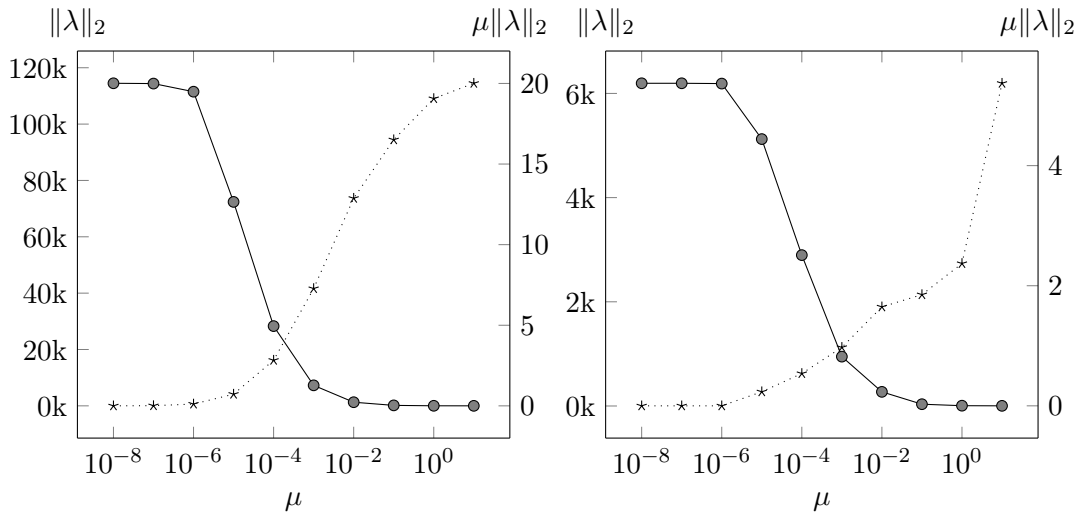
3.3.5 Initial Multiplier Estimates

Due to the redundancy of non-anticipativity constraints, their dual optimal solutions are degenerate in the SUC problem and its relaxation. In Section 3.2.3 we introduced LPR(μ) with $\mu \geq 0$ as a means of controlling the size of the initial multipliers. For any small enough μ the obtained dual solution is *optimal* for the relaxation of the SUC problem, i.e. if we solve the *relaxed* subproblems with these duals, we are guaranteed to obtain a tight Lagrangian lower bound for the SUC relaxation. The results described in the previous sections suggest that, at least with the employed value of $\mu = 10^{-6}$, the optimal duals from LPR(μ) are also optimal for the *integer* SUC problem, and the duality gap vanishes or is negligibly small. In this section we explore whether this is the case for other values of μ.

Figure 3.8 shows the norm of the optimal duals of LPR(μ) as a function of μ for a multi-stage and a two-stage example. On the secondary axis, we also map the non-anticipativity constraint violation resulting from the same value of μ. The duals are stable for various μ, but approach zero as the perturbation level is increased and the non-anticipativity violation increases. The duals obtained with $\mu = 0$ are not included on the graphs for scaling issues: in the two-stage case their norm is twice as large as the largest shown value, and in the multi-stage case it is 50 times larger. The duals in the two cases shown on the graph are quite different: there are more duals in the two-stage problem, covering all scenarios and time steps, and individual entries typically take larger values, resulting in a much larger norm than in the multi-stage case.

In the two-stage example, we observed that any of the duals obtained with $\mu \in [0, 10^{-5}]$ provide a good enough lower bound to terminate after the first ColGen pass. For any values larger than that, the non-anticipativity violation increases swiftly, and the obtained bounds are inferior, so that the decomposition does not terminate immediately. Additionally, values of μ larger than 10^{-5} lead to increasingly worse primal solutions found by the SC heuristic, because the non-anticipativity constraints are partially relaxed and essential information is removed from the multipliers which are needed to encourage the subproblems to produce useful schedules.

In contrast to the two-stage example, the shown multi-stage example is one of the 'easy' cases for which zero duals produce a sufficiently tight lower bound, due to a



(a) Effect of μ on λ in LPR(μ) (two-stage). (b) Effect of μ on λ in LPR(μ) (multi-stage).

Figure 3.8: Further results concerning the choice of μ in LPR(μ) in the two-stage case (3.8a) and the multi-stage case (3.8b). The graphs show the effect of μ on the magnitude of the dual solution $\|\lambda\|_2$ (left axes) and on the violation of non-anticipativity constraints $\mu\|\lambda\|_2$ (right axes) in LPR(μ). The values shown here are taken from a nine scenario example.

negligible EVPI. In this case any of the values for μ shown on the graph provided a dual solution which proved optimality of the primal solution in the next ColGen pass. The same is also true for duals obtained with $\mu = 0$. The degenerate dual solutions of vastly different size extracted from LPR(μ) all produce optimal bounds for the integer SUC problem.

Since the duals estimated from LPR(0) were sufficient to obtain a gap of 0.1% after the first ColGen iteration for both, two-stage and multi-stage problems, it may appear unnecessary to control their size by choosing $\mu > 0$. However, we observed that on the larger 30 to 50 scenario cases, CPLEX's barrier solver was roughly by a third faster when using $\mu = 10^{-6}$ rather than $\mu = 0$. Furthermore, the duals obtained with $\mu > 0$ appear to be a better initial stability center if additional ColGen iterations are necessary: we observed that the first iterations with dRMP(ρ) can be unstable even with $\rho > 0$, if duals obtained with $\mu = 0$ are used as initial stability center and the dRMP contains an unfavourable selection of cutting planes. When attempted to solve to a smaller gap than 0.1%, some of the cases shown in Figure 3.6 obtained very low Lagrangian bounds after solving dRMP(ρ) for the first time, and it took about ten

iterations to find the next bound which was within 0.1%. We were able to remedy this effect by forcing CPLEX not to estimate a MIP start from partial integer solutions when solving the subproblems, i.e. forcing it to generate a different, more favourable set of cutting planes. However, with initial duals estimated with $\mu > 0$ the problem never occurred – independent of the set of cutting planes. For these reasons we prefer solving LPR(μ) with $\mu > 0$.

4

Stochastic vs Deterministic Scheduling

In this chapter we discuss the added value of stochastic programming over deterministic programming in UC applications. To characterise properties of stochastic and deterministic schedules and quantify the potential savings, we conduct a computational study with a UC model based on the British power system. After defining and discussing the model formulation, we explain briefly how the data was obtained for it and what assumptions were made in the modelling process. Wind power forecasts used in this study were synthesised, and we explain the approach used to do that. We use a time series model to sample wind power scenarios and a technique based on distances between probability measures to reduce the number of scenarios and construct a tree. Finally, we explain the rolling horizon approach that was used to evaluate the different scheduling strategies, and then discuss their performance.

4.1 A Model of the British Power System

The UC model used in our rolling horizon evaluation is a central scheduling model based on the British National Grid, including all transmission-connected hydro-thermal and wind power supplies. The objective of this central planning model is to minimise the cost of electricity generation to the economy. Although the UK currently follows a practice of decentral or self-scheduling (cf. section below), we believe that for a study of the value of stochastic scheduling, a central scheduling model is a good starting point: it captures the effects of wind uncertainty on a local and national level and puts them in relation to pump storage capacities and transmission capacities across the whole system. The model can be used to assess the effects of wind power variability

and forecast uncertainty on the power system as a whole, and how stochastic scheduling can help to mitigate them. The centrally scheduled situation can provide a reference model when comparing different market stuctures.

4.1.1 Production Cost Considerations

Electricity markets are operated in various different ways, depending on the regulatory regime: there is a wide variety of market layouts, ranging from regulated monopolies to free markets such as the European Energy Exchange (EEX). The current market layout in Britain is decentral: generation companies schedule their assets individually, so as to satisfy their demand contracts with maximum profitability. The schedule is reported to National Grid, the nationwide transmission system owner and balancing authority, who adjusts schedules for network feasibility and acquires reserve, response and other ancillary services to balance the system. Power plants are charged for grid connection, according to their location relative to transmission bottlenecks and demand centres. On the other hand, they are recompensated if their schedules are altered because of network issues. This results in a complicated multi-layer market structure with national ancillary service markets and individual demand and generation contracts. Market imperfections can be expected to exist in many of these layers, and the result of a national, central schedule can be quite different from the outcome of the self-scheduling process.

When approximating GB power systems operations with a central scheduling model, we need to assign the production cost estimates with care. The simplest way of modelling a central schedule is to assume perfect competition, in which case the offer price made by generation companies is well represented by the marginal cost of generation. This would result in relatively low prices and a situation in which generation companies struggle to recover the capital cost of their assets. Since electricity is a capital intensive product such a market setup is problemtic in the long term, as it discourages investments in new generation capacity. A possible workaround for this problem which has been subject of political discussions in Britain is to establish a capacity market which allows investors in new generation capacity to recover their capital cost. In a setup with separate markets for generation capacity and electric energy one can argue that, if there is sufficient competition the short-term energy price may largely correspond to

the marginal cost of generation. However, electric energy is also a non-homogeneous product: most large generation units require notification and startup periods before they become available to generate, and only few can offer flexible generation that can become available at short notice. At peak demand times these generators can be expected to have significantly higher market power than slow units, so if marginal cost figures are used they are likely to underestimate the price of flexible generation. In our evaluation of stochastic and deterministic scheduling techniques the relation between the price of flexible and inflexible generation is of major importance since it determines the cost of dealing with unforeseen situations and therefore has a direct effect on the total cost achieved with the different scheduling techniques. Underestimating the price of flexibility means to underestimate the added value of stochastic scheduling.

In our evaluation we use levelised cost estimates, which, besides marginal cost, contain a markup for capacity cost, operation and maintenance cost and decommissioning cost. This assumes that generation companies need to recover these costs on the short-term electricity market and have the market power to do so. Like the marginal cost model discussed above this will not lead to quite the same outcome as the current GB self scheduling practice. However, due to the different long term average load factors of slow base-load and flexible peak-load generators this produces a more realistic spread between prices for different generation technologies and is less likely to underestimate the price of flexibility. Since that is a major point in our evaluation we choose this approach over the marginal cost approach.

4.1.2 Overview of the Model

Besides hydro, thermal and wind power generation, our UC model includes pump storage capabilities and an aggregated representation of the transmission system with generation zones and transmission links between them. There are limits on the power flow under normal operation. These are expressed in terms of individual transmission links and additional boundaries, each of which splits the network in two and imposes a real power flow limit on the sum of transmissions crossing it in each direction. The limits are derived by the network operator, using physical network feasibility criteria, n-1 security and fault analyses [59]. The model contains pump storages which can be used for providing ancillary services and storing wind energy. Each pump storage

scheme is modelled as a closed reservoir system, connected to a single plant which contains multiple pump-turbines. Wind power availability is treated as uncertain and a scenario model is used to approximate its possible realisations. Excess wind power can be curtailed at no cost. Load shedding is also permitted, but at a high cost. Our formulation contains more technical detail than the models used in other studies such as Tuohy et al [29] and Sturt and Strbac [31]. We model storage, flexible generation capabilities and wind uncertainty at their relative location to network bottlenecks. This allows us to capture both, local and system-wide effects of wind uncertainty and relate them to the flexibility of the system in potentially congested situations.

In terms of thermal generation units we distinguish fast-start units from slow units. Fast-start units are open-cycle gas turbines (OCGT) which can be started within the hour. All other thermal units are categorised as slow and must be notified at least an hour before they can become available to generate.

Following British practice, we distinguish between frequency response and reserve. Response is fast-acting and is used to stabilise the frequency within seconds, e.g. in the immediate aftermath of a fault, for up to 15 minutes. Reserve is used for two separate reasons: to deal with errors in wind forecasts and to restore response capability by freeing up used response after a failure. Reserve is required to be available for at least an hour. While dedicated variables are needed for frequency response provided by part-loaded generators, reserve can be modelled without additional variables. To do this we formulate one quantity for response, and another quantity for the sum of response and reserve. For pump storage units we use both, dedicated response variables and combined reserve and response variables. Reserve and response are treated as soft constraints, and we include piecewise linear (PWL) functions to penalise for providing insufficient amounts of them. Since the boundary limits were set under contingency considerations, reserve and response are modelled as system-wide services which are not affected by them, i.e. we assume that the boundaries can be overloaded in a post-contingency state where reserve and response are required.

Our day-ahead planning model is a two-stage stochastic model, while the intraday model is a multi-stage stochastic model. We present a single model here, which can be adapted to represent both situations, depending on the choice of non-anticipativity constraints. A single-scenario version of the same model is used to perform deterministic

scheduling and to evaluate existing schedules by solving a dispatch problem. A detailed description of how we handle scheduling and dispatch with this model is given along with the algebraic model description below and in Section 4.3.

An overview of our notation is given below, followed by an algebraic model statement. We use the same conventions as before: sets are in calligraphic font, parameters are latin and greek capitals, and variables are lower case latin or greek letters. Superscripts are used to extend variable names, while subscripts are indices. Reserve and response quantities are distinguished by a hat: for any quantity associated with response, say r, the corresponding quantity for response plus reserve is denoted by $\hat{r}$. The planning horizon is $t = 1, \ldots, T$, and where the statement shows or implies variables for $t \leq 0$, they are fixed input data rather than actual variables.

4.1.3 Notation

Sets

$\mathcal{B}, \mathcal{B}^{01}$: set of scenario bundles. Bundles in B^{01} are for binary decisions of slow units.

$\mathcal{D}$: set of transmission boundaries in the network

$\mathcal{F}$: set of fast start units, $\mathcal{F} \subset \mathcal{G}$. Slow units are in $\mathcal{G} \setminus \mathcal{F}$.

$\mathcal{G}$: set of generation units, $\mathcal{G}_n$ is the set of generators at node $n \in \mathcal{N}$

$\mathcal{L}$: set of transmission lines

$\mathcal{N}$: set of network nodes (transmission areas)

$\mathcal{P}$: set of pump storage plants, $\mathcal{P}_n$ is the subset at node $n \in \mathcal{N}$

$\mathcal{S}$: set of wind power scenarios, $\mathcal{S}_b$ is the scenario subset of bundle $b \in \mathcal{B}$

$\mathcal{W}$: set of wind farms, $\mathcal{W}_n$ is the subset at node $n \in \mathcal{N}$

Parameters

Ψ: minimum proportion of response to be met by part-loaded generators

B_{ld}: line-boundary adjacency matrix. 1 if line l crosses boundary d in one direction, -1 if it crosses in the other direction, 0 otherwise

$C(r^{tot})$: PWL penalty function for keeping too little response r^{tot}

$\hat{C}(\hat{r}^{tot})$: PWL penalty function for keeping too little reserve plus response $\hat{r}^{tot}$

C_g^{nl}: no load cost of generator g [\$/h]

$C_q^{H_2O}$: end-of-day water value in the reservoir of pump storage plant q [\$/MWh]

C_g^m: marginal cost of generator g [\$/MWh]

C_g^{st}: startup cost of generator g [\$]

C^{voll}: value of lost load [\$/MWh]

D^t: time granularity of the model [h]

D^{res}: time for which response must be served [h], with $D^{res} \leq D^t$

E_q: pump-generator cycle efficiency at storage $q \in \mathcal{P}$ [proportion]

H_q^{max}: storage capacity at plant $q \in \mathcal{P}$ in MWh of dischargeable energy

N_q^{pum}: number of (identical) pumps in pump storage plant $q \in \mathcal{P}$

$N_l^{st,end}$: start (end) nodes of line l

P_q^{cap}: capacity of a single pump in pump storage plant $q \in \mathcal{P}$ [MWh]

P_{nt}^{dem}: real power demand at node n in period t [MW]

$P_{g,q}^{min,max}$: min (max) generation limit of generator $g \in \mathcal{G}$ (storage $q \in \mathcal{P}$) [MW]

$\bar{P}_{l,d}$: maximum power transmission on line l / across boundary d [MW]

π_s: probability of scenario s

$P_g^{ru,rd}$: operating ramp up (down) limits of generator g [MW/D^t]

$P_g^{su,sd}$: startup (shutdown) ramp limits of generator g [MW/D^t]

P_{wts}^{win}: wind power available from wind farm w in period t, scenario s [MW]

R_g^{max}: maximum response available from generator g [MW]

T: last time period of the planning horizon

T_g^{nt}: startup notification time of generator g [h]

$t_b^{st,end}$: start (end) periods of scenario bundle b

$T_g^{u,d}$: minimum uptime (downtime) of generator g [h]

Variables

$\alpha_{gts} \in \{0,1\}$: 1 if thermal unit g is on in period t, scenario s, and 0 if it is off

$\gamma_{gts} \in \{0,1\}$: 1 if thermal unit g is started up in period t, scenario s, and 0 otherwise

$\eta_{gts} \in [0,1]$: 1 if thermal unit g is shut down in period t, scenario s, and 0 otherwise

$\delta_{qits} \in \{0,1\}$: 1 if pump i of storage q is pumping in period t, scenario s, 0 otherwise

$\zeta_{qts} \in \{0,1\}$: 1 if storage q is generating in period t, scenario s, and 0 otherwise

$h_{qts} \in [0, H_q^{max}]$: level of storage q *after* period t, scenario s (dischargeable MWh)

$p_{qts}^{dis} \in [0, P_q^{max}]$: real power discharged from storage q in period t, scenario s [MW]

$p_{lts}^{flo} \in [-\bar{P}_l, \bar{P}_l]$: real power flow on line l in period t, scenario s [MW]

$p_{gts}^{gen} \in [0, P_g^{max}]$: real power output of generator g in period t, scenario s [MW]

$p^{pum}_{qts} \geq 0$: real power pumped into storage q in period t, scenario s [MW]

$p^{shed}_{nts} \geq 0$: load shed at node n in period t, scenario s [MW]

$r^{gen}_{gts} \in [0, R^{max}_{g}]$: response provided by generator g in period t, scenario s [MW]

$r^{pum}_{qts} \geq 0$: response provided by pump storage q in period t, scenario s [MW]

$\hat{r}^{pum}_{qts} \geq 0$: reserve plus response provided by storage q in period t, scenario s [MW]

$r^{tot}_{ts} \geq 0$: total available response in period t, scenario s [MW]

$\hat{r}^{tot}_{ts} \geq 0$: total available reserve plus response in period t, scenario s [MW]

$u^{win}_{wts} \in [0, P^{win}_{wts}]$: used wind power from farm w in period t, scenario s [MW]

4.1.4 Algebraic Statement

Objective function

$$\begin{aligned} \min \quad & \sum_{s\in\mathcal{S}} \pi_s \left[\sum_{t=1}^{T} \sum_{g\in\mathcal{G}} \left(C^{st}_{g} \gamma_{gts} + D^t C^{nl}_{g} \alpha_{gts} + D^t C^{m}_{g} p_{gts} \right) \right. \\ & + \left. \sum_{q\in\mathcal{P}} C^{H_2O}_{q} \left(h_{q0s} - h_{qTs} \right) + \sum_{t=1}^{T} \left(\sum_{n\in\mathcal{N}} D^t C^{voll} p^{shed}_{nts} + C(r^{tot}_{ts}) + \hat{C}(\hat{r}^{tot}_{ts}) \right) \right]. \end{aligned} \tag{4.1}$$

The objective is to minimise the expected cost of supplying electricity to the economy, including expected losses due to underserved reserve and response and a penalty for lost load. The generation cost consists of startup, no-load and marginal cost terms. These contain fuel and carbon emission costs and a levelised contribution from capital cost, operation and maintenance cost and decommissioning cost [60]. The water level after the last period is treated as variable, and we apply a linear water value to the reservoir level difference created over the course of the planning horizon.

We use penalty functions to model the cost of underserved reserve to the economy. The penalties represent the expected cost of lost load due to generator failure(s) at times where the system lacks sufficient response and reserve to deal with them. The penalty function $C(r^{tot}_{ts})$ models the expected cost of single generator failures at a response level of r^{tot}_{ts}, while $\hat{C}(\hat{r}^{tot}_{ts})$ models the *additional* expected cost of double generator failures at a level $\hat{r}^{tot}_{ts}$ of response plus reserve. To obtain the correct penalty for single and double generator failures, both are applied.

The penalty function for underserved response is calculated as follows. In any time period, consider a given generator g which is operating at its full capacity, while the

amount of available response in the system is x. In case of a failure of generator g, we assume that $D^t \max\{0, P_g^{max} - x\}$ MWh of demand are lost and the system can recover after a time period D^t. The cost of lost load due to the failure of generator g is a random variable

$$F_g^c(x) := \begin{cases} C^{voll} D^t \max\{0, P_g^{max} - x\} & \text{if generator } g \text{ fails} \\ 0 & \text{otherwise.} \end{cases}$$

If the probability that generator g fails within a given period is p_g, then its expected failure cost is $C_g(x) = p_g C^{voll} D^t \max\{0, P_g^{max} - x\}$. We define the total failure cost $F^c(x) := \sum_{g \in \mathcal{G}} F_g^c(x)$. Then the expected total failure cost when operating the system with x MW response in any time period is

$$C(x) := C^{voll} D^t \sum_{g \in \mathcal{G}} p_g \max\{0, P_g^{max} - x\}. \tag{4.2}$$

This is used as penalty function for underserved response and is shown in Figure (4.5a) in Section 4.2. In our implementation we approximate the function with seven PWL pieces.

After the failure of a single generator, reserve is used to restore the response level. A subsequent failure in the same period D^t will lead to a loss of load unless the combined amount of response and reserve cover the loss of both generators. Using the same approach as above, we derive an additional penalty function $\hat{C}(y)$ for insufficient levels y of response plus reserve. The cost of lost load due to the failure of a generator tuple (g_1, g_2) in the same time period is given by

$$\hat{F}^c_{(g_1,g_2)}(y) := \begin{cases} C^{voll} D^t \max\{0, P_{g_1}^{max} + P_{g_2}^{max} - y\} & \text{if generators } g_1 \text{ and } g_2 \text{ fail} \\ 0 & \text{otherwise.} \end{cases}$$

Let $\hat{\mathcal{G}}$ denote the set of all combinations of generators and define the total failure cost $\hat{F}^c(y) := \sum_{(g_1,g_2) \in \hat{\mathcal{G}}} \hat{F}^c_{(g_1,g_2)}(y)$. Assuming that generators fail independently, we obtain the following expression for the expected total cost of double failures while operating

at a combined response and reserve level y:

$$\hat{C}(y) := C^{voll} D^t \sum_{(g_1,g_2)\in\hat{\mathcal{G}}} p_{g_1} p_{g_2} \max\{0, P^{max}_{g_1} + P^{max}_{g_2} - y\}. \tag{4.3}$$

This is used as additional penalty function for underserved response plus reserve and is shown in Figure (4.5b) in Section 4.2. In our implementation, we approximate this function with five PWL pieces.

We assume that all generators fail with equal probability, $p_g = p \ \forall g \in \mathcal{G}$. Further, the formulae build on the assumption that all generators are operating at their maximum output level, so the penalties tend to overestimate the expected cost of lost load due to failures. Additional losses due to quick successive failures (before response can be restored) and failures of more than two generators within one hour are not taken into account. However, the cost of single failures is a small percentage of overall cost (cf. Figure 4.12), and the cost of double failures is an order of magnitude smaller than that (cf. Figure 4.5). Thus the approximation error can be expected to be small. For alternative reserve pricing approaches we refer to Ortega-Vazquez and Kirschen [61] and the PJM method, which is described in Billinton and Allan [62]. The minimisation of objective (4.1) is subject to the following constraints:

Load balance equations for all $n \in \mathcal{N},\ s \in \mathcal{S},\ t = 1, \ldots, T$:

$$\begin{aligned} 0 \ &= \ \sum_{g\in\mathcal{G}_n} p_{gts} + \sum_{w\in\mathcal{W}_n} u^{win}_{wts} + \sum_{l\in\mathcal{L}:N^{end}_l=n} p^{flo}_{lts} + \sum_{q\in\mathcal{P}_n} p^{dis}_{qts} + p^{shed}_{nts} \\ &- \ P^{dem}_{nt} - \sum_{l\in\mathcal{L}:N^{st}_l=n} p^{flo}_{lts} - \sum_{q\in\mathcal{P}_n} p^{pum}_{qts}. \end{aligned} \tag{4.4}$$

These ensure that power input and output are equal at all times at all network nodes.

Transmission boundary limits for all $t = 1, \ldots, T,\ d \in \mathcal{D},\ s \in \mathcal{S}$:

$$-\bar{P}_d \leq \sum_{l\in\mathcal{L}} B_{ld} p^{flo}_{lts} \ \leq \ \bar{P}_d. \tag{4.5}$$

These inequalities impose restrictions on the transmission across pre-defined boundaries, by limiting the sum of power flows on lines crossing the boundary in each direc-

tion. They are used in addition to the limits on individual lines' power flow variables to model network congestion.

Generator bounds for all $s \in \mathcal{S},\ t = 1, \ldots, T,\ g \in \mathcal{G}$:

$$p^{gen}_{gts} \geq P^{min}_{g} \alpha_{gts} \tag{4.6}$$

$$p^{gen}_{gts} + r^{gen}_{gts} \leq P^{max}_{g} \alpha_{gts}. \tag{4.7}$$

Constraints (4.6) and (4.7) establish the connection between power output, response and on-off variables. When a generator is on ($\alpha_{gts} = 1$), it must generate between the minimum and maximum stable limits, and the response it can provide is limited by its spare headroom (beside the upper limit R^{max}_{g}). When it is off ($\alpha_{gts} = 0$), the generator's generation and response levels are at zero.

Ramp rate constraints for all $g \in \mathcal{G},\ s \in \mathcal{S},\ t = 1, \ldots, T$:

$$p^{gen}_{gts} - p^{gen}_{g(t-1)s} \leq P^{ru}_{g} \alpha_{g(t-1)s} + P^{su}_{g} \gamma_{gts} \tag{4.8}$$

$$p^{gen}_{g(t-1)s} - p^{gen}_{gts} \leq P^{rd}_{g} \alpha_{gts} + P^{sd}_{g} \eta_{gts}. \tag{4.9}$$

These work in the same way as the ramp rate constraints in the basic UC formulation explained in Section 2.3: constraints (4.8) limit the *increase* in generation level between two successive periods $t-1$ and t in the case where a generator is on in both periods ($\alpha_{g(t-1)s} = 1,\ \gamma_{gts} = 0$), and in the case where it is started up in the second period ($\alpha_{g(t-1)s} = 0,\ \gamma_{gts} = 1$). Similarly, constraints (4.9) limit the *decrease* in two successive periods during continuous operation ($\alpha_{gts} = 1,\ \eta_{gts} = 0$) and shutdown ($\alpha_{gts} = 0,\ \eta_{gts} = 1$).

Switching constraints for all $s \in \mathcal{S},\ t = 1, \ldots, T,\ g \in \mathcal{G}$:

$$\alpha_{gts} - \alpha_{g(t-1)s} = \gamma_{gts} - \eta_{gts} \tag{4.10}$$

$$1 \geq \gamma_{gts} + \eta_{gts}. \tag{4.11}$$

Logical constraints (4.10) and (4.11) establish the relationship between on-off, startup and shutdown variables. Like the ramp rate formulation, these are identical to the constraints used in the basic UC model, and we refer to Section 2.3 for further explanation.

Minimum up- and downtime constraints for all $s \in \mathcal{S},\ g \in \mathcal{G},\ t = 1, \dots, T$:

$$\sum_{i=t-T_g^u+1}^{t} \gamma_{gis} \quad \leq \quad \alpha_{gts} \tag{4.12}$$

$$\sum_{i=t-T_g^d+1}^{t} \eta_{gis} \quad \leq \quad 1 - \alpha_{gts}. \tag{4.13}$$

To model minimum up- and downtimes, we use the facet-defining minimum up-down cuts (4.12) and (4.13) by Rajan and Takriti [11] (cf. Section 2.3).

Pump storage operation constraints for all $q \in \mathcal{P},\ t = 1, \dots, T,\ s \in \mathcal{S}$:

$$\delta_{q1ts} \quad \leq 1 - \zeta_{qts} \tag{4.14}$$

$$\delta_{q(i+1)ts} \quad \leq \delta_{qits} \quad \forall i = 1, \dots, N_q^{pum} - 1 \tag{4.15}$$

$$p_{qts}^{pum} \quad = \sum_{i=1}^{N_q^{pum}} \delta_{qits} P_q^{cap} \tag{4.16}$$

$$\zeta_{qts} P_q^{min} \leq \quad p_{qts}^{dis} \quad \leq \zeta_{qts} P_q^{max}. \tag{4.17}$$

Pump storage plants are useful for providing reserve and response, meeting peak demand and storing excess wind power. Different pump storage plants are pre-qualified to provide different ancillary services, namely primary or secondary response and fast reserve. However, we do not distinguish between responses or reserves at different time scales but only between response and reserve in general. Binary variables ζ_{qts} determine whether a plant is discharging or not, and constraints (4.17) link them to continuous discharge variables with lower and upper limits. Within one plant, the pumps all have identical capacities, and they can only be pumping when the plant is not discharging (4.14). After switching on the first pump, the others are switched on in order from lowest to highest (4.15) to avoid symmetric solutions. The pumping level is decided by the number of active pumps, since they can only run at full capacity (4.16). The bina-

ries for each pump, δ_{qits}, could be replaced by a single integer variable $\bar{\delta}_{qts}$ indicating the number of active pumps. However, we use binaries because then δ_{q1ts} can be used in constraints (4.14), (4.24) and (4.25) to indicate whether a plant is pumping or not.

Reservoir constraints for all $q \in \mathcal{P},\ s \in \mathcal{S},\ t = 1, \ldots, T$:

$$h_{qts} = h_{q(t-1)s} + D^t E_q p^{pum}_{qts} - D^t p^{dis}_{qts}. \tag{4.18}$$

Reservoir levels are tracked by constraints (4.18). They are expressed in terms of MWh of electrical energy that would be generated using the contained water. A constant cycle efficiency is applied to incoming energy, thus keeping the model linear by neglecting the head effect which is small. The plants are located at separate sites with no hydrological connection. Also, exogenous inflows are small and lower reservoirs are large, so the water cycle of each pump storage plant is modelled as a single reservoir system with a given storage capacity. This is a good approximation for GB pump storage schemes.

Reserve and response definitions for all $t = 1, \ldots, T,\ s \in \mathcal{S}$:

$$\sum_{g \in \mathcal{G}} \left(\alpha_{gts} P^{max}_g - p_{gts}\right) + \sum_{q \in \mathcal{P}} \hat{r}^{pum}_{qts} = \hat{r}^{tot}_{ts} \tag{4.19}$$

$$\sum_{g \in \mathcal{G}} r^{gen}_{gts} + \sum_{q \in \mathcal{P}} r^{pum}_{qts} = r^{tot}_{ts} \tag{4.20}$$

$$\sum_{g \in \mathcal{G}} r^{gen}_{gts} \geq \Psi r^{tot}_{ts}. \tag{4.21}$$

Equations (4.20) and (4.19) define system-wide levels of response and reserve plus response, respectively. Part-loaded generators contribute all spare headroom $P^{max}_g - p_{gts}$ to the slow-acting reserve quantity (4.19), while, due to ramp limits, they can only contribute a limited amount of their headroom $r^{gen}_{gts} \leq \min\{R^{max}_g, P^{max}_g - p_{gts}\}$ to the fast-acting response quantity (4.20). The contributions from pump storages are defined in the next paragraph. Constraints (4.21) require a minimum amount of response to be met by part-loaded generators to avoid relying too much on pump storage units.

Pump storage reserve constraints for all $q \in \mathcal{P},\ t = 1, \ldots, T,\ s \in \mathcal{S}$:

$$\hat{r}^{pum}_{qts} + p^{dis}_{qts} \leq P^{max}_{q} + p^{pum}_{qts} \tag{4.22}$$

$$D^t \hat{r}^{pum}_{qts} + D^t p^{dis}_{qts} \leq h_{q(t-1)s} + D^t p^{pum}_{qts} \tag{4.23}$$

$$r^{pum}_{qts} + p^{dis}_{qts} \leq P^{max}_{q} (1 - \delta_{q1ts}) + p^{pum}_{qts} \tag{4.24}$$

$$D^{res} r^{pum}_{qts} + D^t p^{dis}_{qts} \leq h_{q(t-1)s} + D^t P^{max}_{q} \delta_{q1ts}. \tag{4.25}$$

Pump storage plants can provide different levels of reserve and response, depending on whether they are currently discharging, pumping or spinning in air. Constraints (4.22) and (4.23) impose limits on the sum of response and reserve $\hat{r}^{pum}_{qts}$, and constraints (4.24) and (4.25) limit the available response r^{pum}_{qts}.

When the plant is in **discharge mode** ($\zeta_{qts} = 1$, $\delta_{q1ts} = 0$), pump variables p^{pum}_{qts} are all zero. Then constraints (4.22) and (4.23) state that the current discharge plus reserve and response can exceed neither the maximum discharge nor the remaining energy level in the storage. Constraint (4.24) states that the response provided during discharge is limited by the headroom available in the turbine, and constraint (4.25) makes sure that there is sufficient energy stored in the reservoir to meet the discharge during the hour and provide response over a fraction of D^{res} of an hour.

In **pump mode** ($\zeta_{qts} = 0$, $\delta_{q1ts} = 1$) a plant can provide reserve by turning off the pumps *and* starting to discharge. The demand reduction through turning off the pumps is fast enough to meet response standards, while subsequent discharge only qualifies as reserve. The discharge level p^{dis}_{qts} is zero, and constraint (4.22) limits the provided reserve plus response to be at most the current pumping level plus maximum discharge. Now constraint (4.23) states that the reserve plus response is bounded above by the amount of energy left in the reservoir plus the current pumping level. Equation (4.24) says that the available response is upper bounded by the pumping level, while (4.25) is removed by increasing the right-hand side term by P^{max}_{q}.

Finally, if the plant has its turbines **spinning in air** ($\zeta_{qts} = 0$, $\delta_{q1ts} = 0$), pump and discharge variables p^{pum}_{qts} and p^{dis}_{qts} are both zero, and reserve plus response is simply bounded above by the maximum discharge (4.22) and the available energy level (4.23). The same is true for response and is achieved by equations (4.24) and (4.25), only here the energy level contained in the reservoir need only be sufficient to maintain response

for a fraction D^{res} of an hour. The energy consumption required to keep the turbines spinning in air is small and not taken into account here. The model does not need a separate **idle mode**: the spinning in air mode also covers situations in which no reserve or response is provided.

Non-anticipativity constraints determine the structure of the decision tree underlying our optimization model. To keep the notation minimal we show a chain formulation of the constraints here. However, when applying the decomposition method to this model we resort to the formulation with redundant common target variables. For binary decisions of slow units we use a specific set of bundles, denoted by $\mathcal{B}^{01}$. The following constraints are included for all $b \in \mathcal{B}^{01},\ j,k \in \mathcal{S}_b : k = j+1$:

$$\alpha_{gtj} = \alpha_{gtk} \quad \forall g \in \mathcal{G} \setminus \mathcal{F},\ t = t_b^{st}, \ldots, t_b^{end} \tag{4.26}$$

$$\gamma_{gtj} = \gamma_{gtk} \quad \forall g \in \mathcal{G} \setminus \mathcal{F},\ t = t_b^{end}+1, \ldots, t_b^{end}+T_g^{nt}. \tag{4.27}$$

Constraints (4.26) make commitment decisions of slow units unique across all bundled scenarios. They are required for both, two-stage and multi-stage stochastic problems.

Constraints (4.27) are non-standard and are included in multi-stage problems to model startup notification times. When scheduling generators with a deterministic model or a day-ahead stochastic model, a sufficient notification period for generator startups is implicit. However, if commitments are updated in the course of the day, as is done in the multi-stage model, then after a scenario split and decision update we must allow for a minimum notification period to pass before additional startups can become effective. To achieve this, non-anticipativity of startup variables is extended for a notification time after the split of a bundle. During time periods $t_b^{st}, \ldots, t_b^{end}$, constraints (4.27) are implied by (4.26) together with (4.10). Thus, to avoid redundancy we only include them for the time periods $t_b^{end}+1, \ldots, t_b^{end}+T_g^{nt}$. Further, we use the following additional non-anticipativity constraints for recourse variables of the multi-stage problem. They are included for all $b \in \mathcal{B},\ j,k \in \mathcal{S}_b : k = j+1$ and $t = t_b^{st}, \ldots, t_b^{end}$:

$$\alpha_{gtj} = \alpha_{gtk} \quad \forall g \in \mathcal{F} \tag{4.28}$$

$$\delta_{qitj} = \delta_{qitk} \quad \forall q \in \mathcal{P},\ i = 1, \ldots, N_q^{pum} \tag{4.29}$$

$$\zeta_{qtj} = \zeta_{qtk} \quad \forall q \in \mathcal{P} \tag{4.30}$$

$$p^{gen}_{gtj} = p^{gen}_{gtk} \quad \forall g \in \mathcal{G} \tag{4.31}$$

$$p^{dis}_{qtj} = p^{dis}_{qtk} \quad \forall q \in \mathcal{P}. \tag{4.32}$$

In the rolling horizon evaluation we use deterministic, two-stage stochastic and multi-stage stochastic problems, and with slight data modifications this model represents all of them. Figure 4.1 shows how we use the data structures to shape two-stage and multi-stage decision trees.

The simplest model is a **deterministic** one with a single wind power scenario and no non-anticipativity constraints. It is used for deterministic scheduling and dispatch:

1. In the scheduling model the wind scenario is equal to a central forecast and we use a fixed margin for reserve plus response, i.e. we ask for $\hat{r}^{tot}_{ts}$ to be greater than or equal to some fixed margin.

2. In the dispatch model we fix a given schedule for the slow units and evaluate it against the actual wind outturn. The dispatch model uses all available recourse actions to compensate for the error in the wind forecast that was used to create the schedule. It decides optimal output levels of committed generators, operation of fast-start units and pump storage plants, available response and reserve, and the amount of shed load.

The interaction between scheduling and dispatch models in the rolling horizon context is further described in Section 4.3.

For day-ahead scheduling we use a **two-stage stochastic** model with multiple wind power scenarios as shown in Figure 4.1 (right). In this setting, the first stage decisions are day-ahead commitments of slow units for the whole 24h planning period. All remaining variables are recourse variables. The two-stage model has non-anticipativity constraints (4.26), while (4.27) to (4.32) are dropped.

For intraday scheduling we use a **multi-stage stochastic** model as shown in Figure 4.1 (left). In this model, one stage covers either 3 or 6 hours of the 24-hour planning horizon, depending on how often commitments of slow units can be updated. The first stage decisions are commitments of these units between t_1 and t_2 and startup decisions for a notification time thereafter, which are non-anticipative due to (4.26) and (4.27).

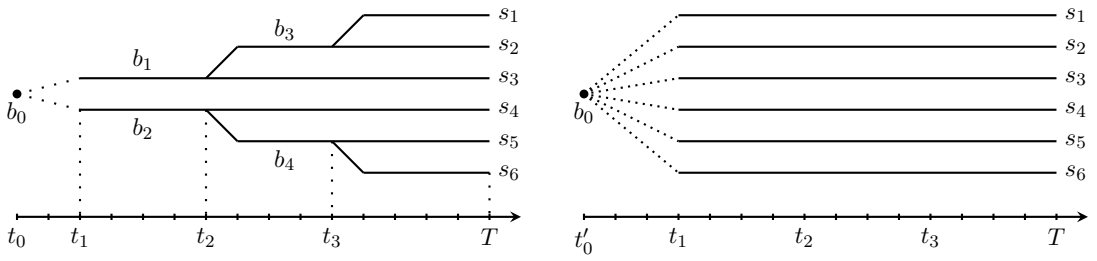


Figure 4.1: Right: two-stage decision tree with 6 scenarios. First stage commitment decisions are made at time t'_0 and are unique for the whole planning horizon $\{t_1, \ldots, T\}$. From t_1 onwards, recourse decisions in every scenario are made under perfect information. The bundles are $\mathcal{B} = \emptyset$, $\mathcal{B}^{01} = \{b_0\}$ with $\mathcal{S}_{b_0} = \mathcal{S}$ and $t^{st}_{b_0} = 1$, $t^{end}_{b_0} = T$. Constraints (4.27) are dropped. Left: multi-stage decision tree with two scenarios on the second stage and 6 leaves. The bundles are $\mathcal{B} = \{b_1, \ldots, b_4\}$, $\mathcal{B}^{01} = \{b_0, b_3, b_4\}$. On the first stage we have bundle b_0 with $\mathcal{S}_{b_0} = \mathcal{S}$ and $t^{st}_{b_0} = t_1$, $t^{end}_{b_0} = t_2$. Like the first stage, the second stage also covers periods $\{t_1, \ldots, t_2\}$, with bundles b_1 and b_2 containing $\mathcal{S}_{b_1} = \{s_1, s_2, s_3\}$ and $\mathcal{S}_{b_2} = \{s_4, s_5, s_6\}$. There is a third stage with bundles b_3, covering $\mathcal{S}_{b_3} = \{s_1, s_2\}$ and b_4, covering $\mathcal{S}_{b_4} = \{s_5, s_6\}$, and a fourth stage with no bundles.

We use *multiple* wind power scenarios between times t_1 and t_2 to make the commitment decisions robust. This is not standard in the UC literature, where multi-stage trees are typically restricted to a *single* scenario for the first few hours. Within each of the bundles $\mathcal{B} = \{b_1, \ldots, b_4\}$, we seek a non-anticipative solution by including all constraints (4.28) to (4.32). For bundles b_3 and b_4 we also require constraints (4.26) and (4.27), while for bundles b_1 and b_2 those constraints are not required because between t_1 and t_2 they are already included for all scenarios, due to bundle b_0. Hence the choice $\mathcal{B}^{01} = \{b_0, b_3, b_4\}$.

We call a schedule non-anticipative when all variables appearing in constraints (4.8) to (4.13) and (4.18) have identical solutions across all subsets of scenarios that are identical at any given time t. Constraints (4.8) to (4.13) and (4.18) are the only constraints which introduce variable interdependence between subsequent time steps, and all variables not appearing therein can be re-evaluated independently at each time step. However, not all variables appearing in these constraints require explicit non-anticipativity constraints if that property can be deduced from other variables linked with them. For pump level variables p^{pum}_{qts}, non-anticipativity follows directly from constraints (4.16) and (4.29). Then for reservoir levels it follows from constraints (4.18), (4.32) and the fact that initial reservoir levels are fixed. Finally, for startup and shutdown variables

it follows from constraints (4.10) and (4.11), together with (4.26) or (4.28). To avoid unnecessary redundancy, we omit non-anticipativity constraints for those variables for which this property can be deduced from sets of other constraints.

4.2 Input Data and Scenario Generation

In this section we briefly describe the data sources for our GB model and for the rolling horizon evaluation. For the evaluation process, we require historic wind data and historic wind forecast data. While actual wind speed data was available for this study, historic forecasts were not. To synthesise wind speed forecasts with the desired statistical properties, we use a pattern matching technique, which is also explained in the following sections. Having synthesised the forecasts, we obtain a time series of wind forecast errors, to which we fit an auto-regressive moving average model with one auto-regression term and one moving average term [ARMA(1,1)]. This model is used to sample forecast error scenarios which are subsequently translated to wind speed scenarios by adding them to a central forecast. The wind speed scenarios are converted to representative regional load factors using an aggregated wind speed to power curve, and then reduced to a smaller number of representative scenarios. In the two- stage stochastic case, the load factors are then applied directly to the wind farms in the model, while in the multi-stage case the scenarios are condensed into a tree first.

4.2.1 Data Sources

Data on thermal generation units, pump storage, wind farms, and transmission system topology and capacity are taken from National Grid's 2013 Electricity Ten Year statement (ETYS) [59]. The list of generation units and their maximum capacities can be found there, as well as their location. Approximate minimum stable limits for the different generation technologies were obtained from the Federal Energy Regulatory Commission (FERC) report [10]. The figures for installed wind capacity and transmission capabilities correspond to those under the Gone Green Scenario described in the ETYS. Historic demand time series are taken from National Grid's website [63], but scaled to meet National Grid's 2020 average demand expectation. Demand is treated net of interconnector imports and exports to and from connected countries, i.e. inter-

connector exchanges are set according to historic time series rather than co-optimizing them. A graph of the system topology is shown in Figure 4.2. Additionally, Figures 4.3 and 4.4 show the installed generation capacity by fuel type and study zone, and the demand distribution by study zone, respectively. Figure 4.4 also indicates the boundary transmission capabilities. Conventional power generation sources are located close to the large demand centres in England, and large transmission capacities are available between those areas. However, a lot of wind power generation capacity has been installed in the north of Scotland (Z1, Z2), where demand is very low. Consequently, these areas export significant amounts of power when winds are high. Transmission capacities, however, are low, and with the increased wind supplies this leads to network congestion in Scotland.

Technical information like ramp rates and minimum up/down times for thermal units were obtained through the Balancing Mechanism Report System [64]. Startup notification times for different types of generators were taken from [65, 66]. For our generation cost figures we use levelised cost estimates from the Department of Energy and Climate Change (DECC) [60]. They contain carbon cost and fossil fuel cost predictions and a levelised contribution from capital cost and decommissioning cost. Historic response figures, and the proportion Ψ of minimum response provided by part-loaded generators are calculated from the monthly Balancing Services Summaries, using data for Mandatory Frequency Response and Firm Frequency Response [67].

Approximate historic wind speed data is available from a reanalysis with a mesoscale weather model [68], and historic forecasts were synthesised by applying the pattern matching forecast technique described in the following section. The wind speed data is aggregated by regions, and we use equivalent regional wind speed to power conversion curves from [69] to translate wind speeds to representative regional load factors that can be applied to the corresponding wind farms.

Lost Load and Underserved Reserve. The GB value of lost load (VOLL) was estimated to be $27,104 (£16,940) per MWh in a publication by London Economics, the Department of Energy and Climate Change (DECC) and the Office of Gas and Electricity Markets (Ofgem) [70]. We use this to model the cost of lost load to the economy and to estimate the penalty functions $C(r_{ts}^{tot})$ and $\hat{C}(\hat{r}_{ts}^{tot})$ for the expected

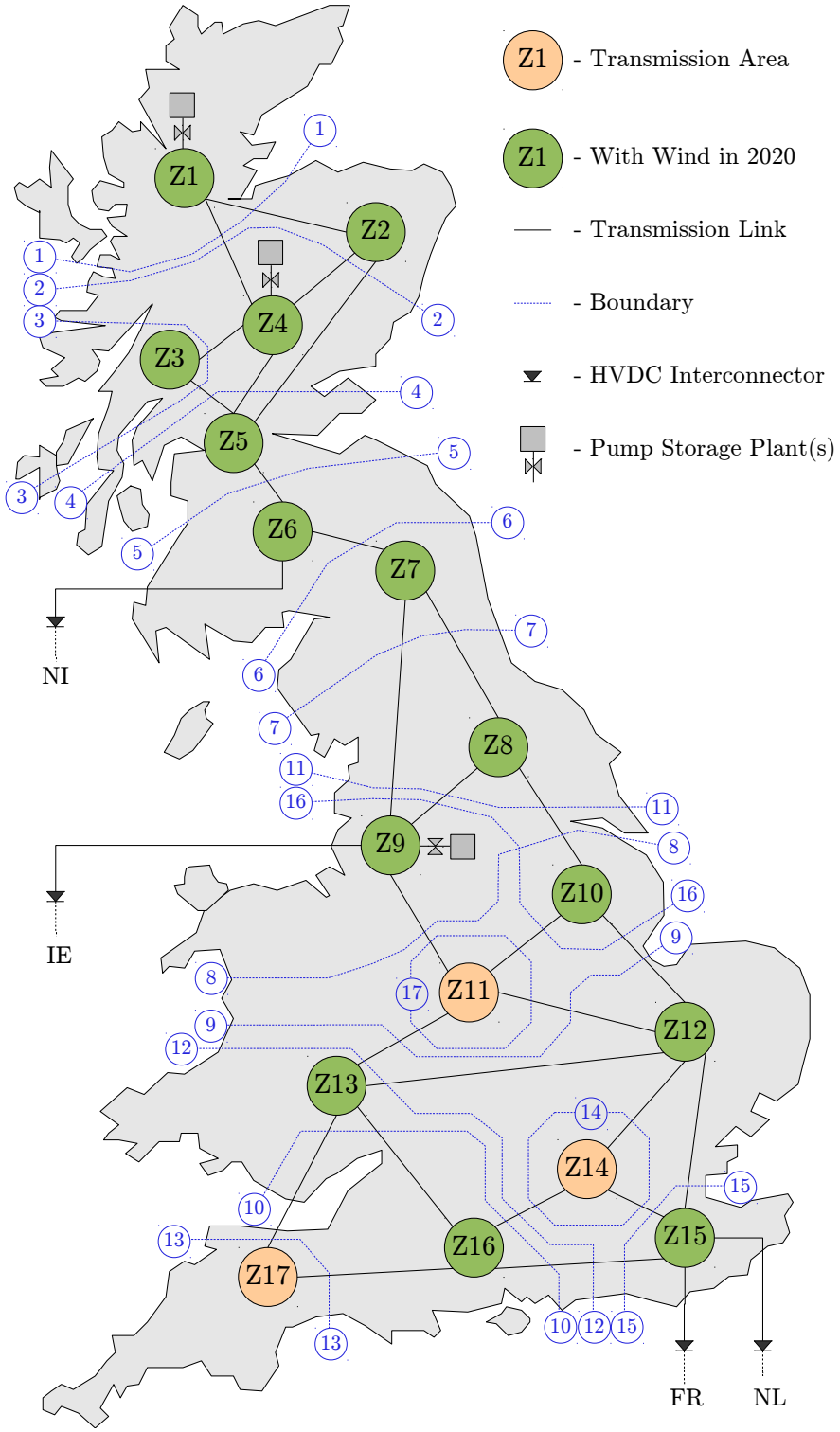


Figure 4.2: Aggregated GB transmission system with 17 areas and 27 links between them. There are two pump storage plants in Scotland (zones one and four) and two in Wales (zone nine), and interconnectors to Ireland, France and the Netherlands. The 17 boundaries are shown in blue and impose limits on the sum of transmissions on all lines that they cross.

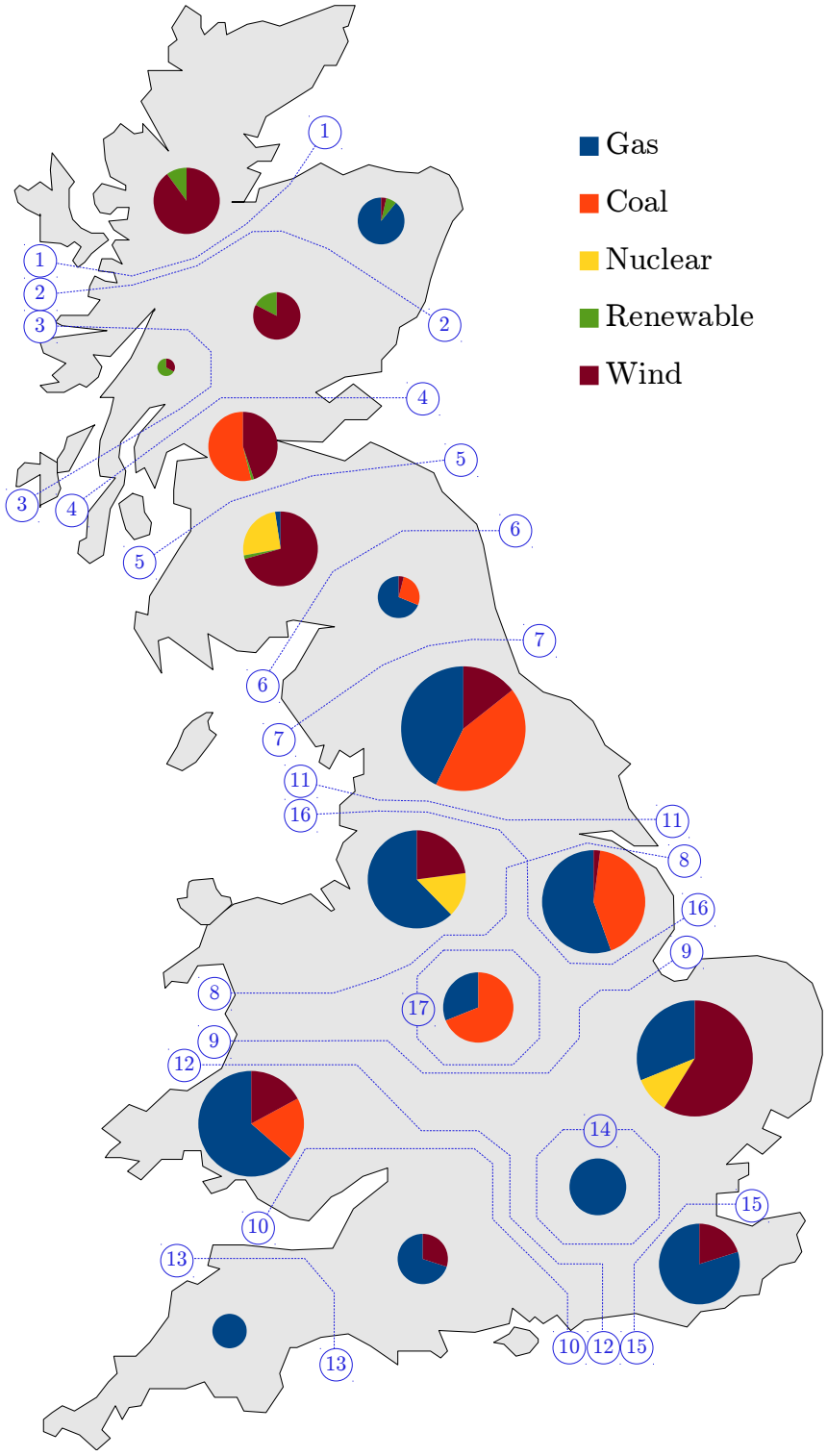


Figure 4.3: GB power supply capacity by study zone and fuel type. We distinguish Gas (open cycle, combined cycle and combined heat and power), Coal, Nuclear, Renewable (Biomass and hydro) and Wind power supplies. The area of each pie chart is representative of the production capacity in the corresponding zone.

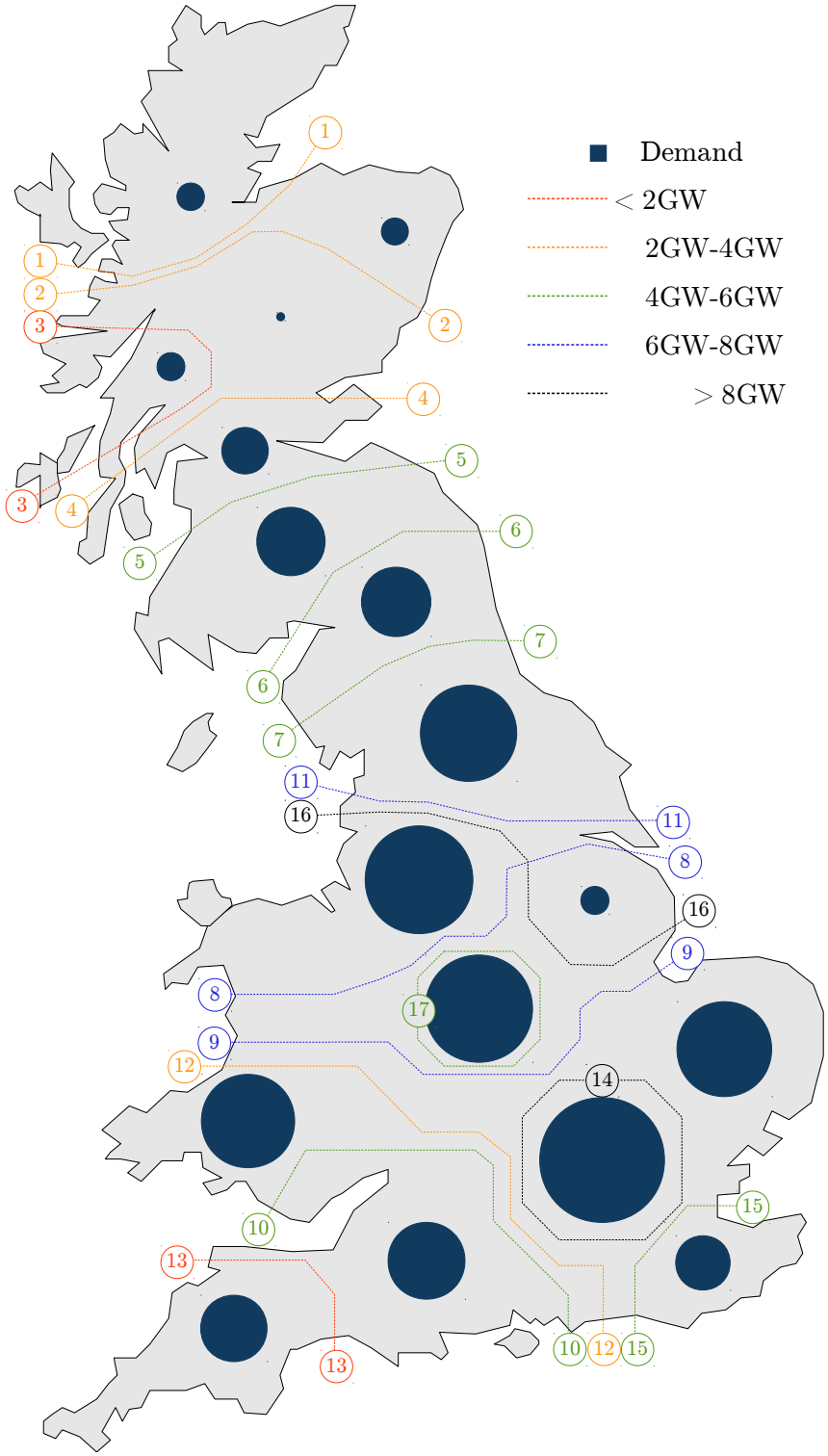
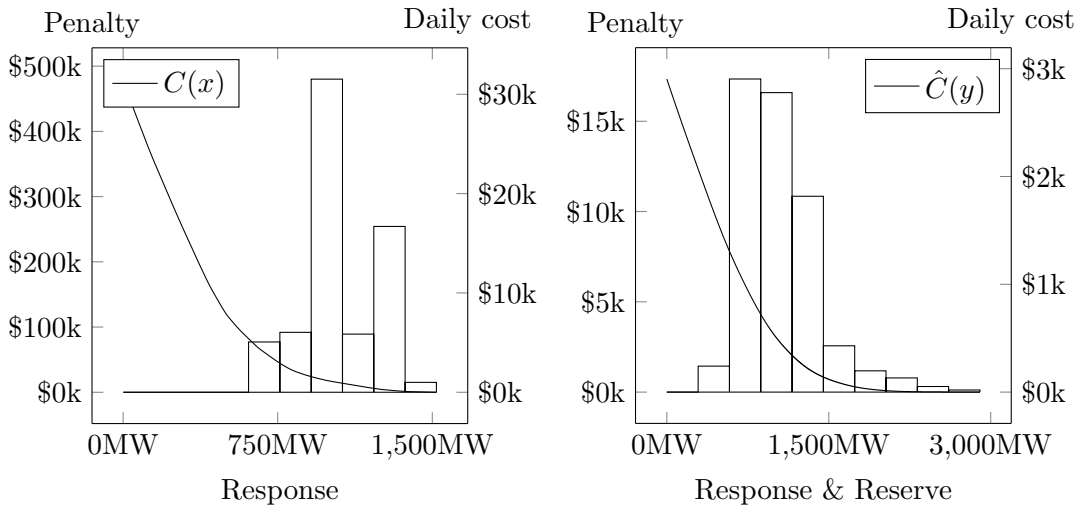


Figure 4.4: Breakdown of GB power demand by study zones, and boundary transmission capabilities. The graph shows the average power demand in each study zone over the two-year evaluation period. The area of the demand circles is representative of the demand proportion in the corresponding study zone. Boundary transmission capabilities are color coded.



(a) Penalty for single generator failures.

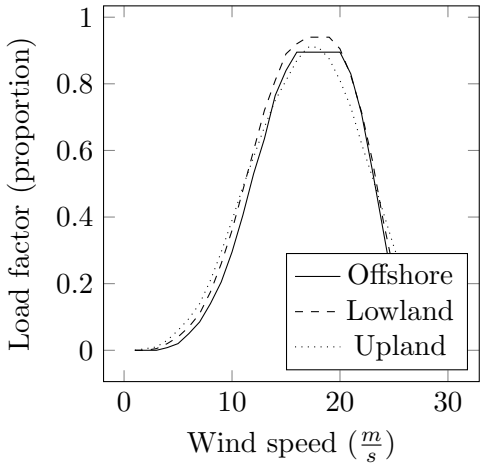
(b) Penalty for double generator failures.

Figure 4.5: Response (a) and reserve plus response (b) penalty functions, based on expected loss due to single and double generator failures. The superimposed bar charts in (a) and (b) show an exemplary outcome of the total cost of operating at different levels of response and reserve plus response, respectively. Discretised in ten bins, the charts show the daily average penalty cost of operating at the given levels of response and reserve. The values are taken from the most successful two-year evaluation of 6-hour deterministic scheduling reported in Section 4.4.

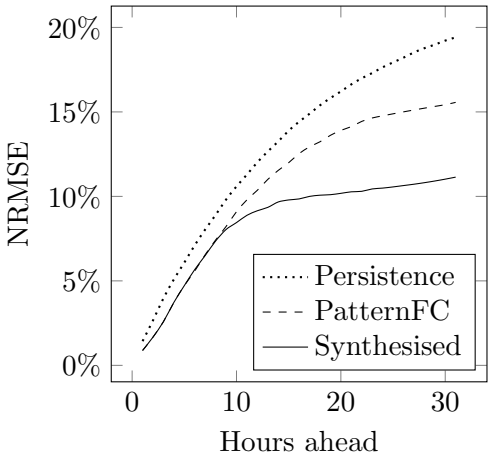
cost of lost load in the case of generator failures. To calculate the cost functions we assume a generator failure probability p which is equivalent to an average of one failure per generator in 150 days. The cost curves are shown in Figure 4.5.

4.2.2 Synthesising Wind Power Forecasts

Historic wind speed or wind power forecasts are not available to us, so for the purpose of this evaluation we synthesise them. Our synthetic forecasts are a weighted average of historic wind and a forecast made by pattern matching. The weights in this are adapted so as to achieve a root mean square error (RMSE) of 10% of installed capacity at the 24 hour ahead mark while matching the shape of typical forecast error curves as shown in [71, 72]. The RMSE of a sample of m forecasts is a function of the forecast horizon t. It can be calculated for single sites, regions, or a whole country. Assume that there are $k = 1, \ldots, N$ regions. Then the RMSE in a t hour ahead forecast for



(a) Regional equivalent power curves.



(b) NRMSE of wind power forecasts.

Figure 4.6: Left (4.6a): Regional equivalent power curves for wind speed to power conversion. We use these to translate wind speeds to regional load factors, which are then multiplied with wind farm capacities to give individual outputs. The curves are taken from [69]. Right (4.6b): Normalised root mean square error (NRMSE) of wind power persistence forecasts, forecasts made by pattern matching, and our final synthesised forecasts which are a weighted average of pattern forecasts and real wind. Forecast error curves are shown in % of installed capacity and up to 32 hours ahead.

region k is defined as

$$\mathrm{RMSE}_{kt} := \sqrt{\frac{1}{m}\sum_{i=1}^{m} e_{ikt}^2}. \tag{4.33}$$

where e_{ikt} is the t hour ahead error in forecast $i \in \{1, \ldots, m\}$ for region k. When calculating the RMSE for a whole country, the errors are first aggregated, $e_{it} := \sum_{k=1}^{N} e_{ikt}$, before applying (4.33). The RMSE of a whole country is therefore significantly smaller than the RMSE of a region. In this study, we work with wind *power* rather than wind *speed* forecast errors: regional forecasts are synthesised in terms of wind speed and then translated to regional load factors using the power curves shown in Figure 4.6a before calculating the wind power forecast RMSE. The RMSE is typically expressed as a percentage of the overall installed capacity, and this is referred to as normalised root mean square error (NRMSE). The graph in Figure (4.6b) shows the NRMSE of synthesised forecasts as a function of the forecast horizon, along with the NRMSE of persistence forecasts and forecasts made by pure pattern matching. Persistence forecasts assume that the current wind conditions will remain unchanged, i.e. persist, and their NRMSE is included for reference only.

To generate forecasts for a given year via pattern matching, we use historic wind speed patterns from the preceding year. In this context, a pattern is an hourly progression of wind speeds for a given region, obtained by averaging observed wind speed progressions which satisfy initial criteria for the first two periods. To generate a T hour ahead forecast via pattern matching, we collect patterns of length $T+2$. Details of the procedure are outlined below.

1. Discretise the domain of observed wind speeds into $k = 1, \ldots, K$ equidistant intervals and create $3K$ bins $b_{ki},\ i \in \{u, l, d\}$ to hold historic wind progressions of length $T+2$. A wind progression $w_j,\ j = 1, \ldots, T+2$ is assigned to bin b_{ku} if the wind in the first hour, w_1 is inside wind interval k and the wind is picking *up*, that is, $w_2 \geq w_1 + \epsilon$ with a fixed threshold ϵ. Analogously, bins b_{kl} and b_{kd} hold wind progressions that stay *level*, that is, $w_2 \in (w_1 - \epsilon, w_1 + \epsilon)$ or point *down*, so that $w_2 \leq w_1 - \epsilon$, respectively.

2. Iterate over historic wind data of the previous year in one-hour steps. Extract wind progressions of length $T+2$ starting at the current hour and assign them to the bins. Then find a representative wind pattern $\hat{w}_{jki}$ for every bin b_{ki}, with $j = 1, \ldots, T+2$, by averaging all wind patterns assigned to that bin.

3. Use the $\hat{w}_{jki}$ to make forecasts starting in any hour of the current year: to make a forecast starting in hour t with wind history $w_{t-1} \geq w_{t-2} + \epsilon$, select the bin b_{ku} such that wind speed w_{t-2} lies in interval k. The forecast for hours $t, \ldots, t+T-1$ is given by $\hat{w}_{jku}$ with $j = 3, \ldots, T+2$. Analogously, use forecasts $\hat{w}_{jkl}$ from bins b_{kl} if $w_{t-1} \in (w_{t-2} - \epsilon, w_{t-2} + \epsilon)$, and forecasts $\hat{w}_{jkd}$ from bins b_{kd} if $w_{t-1} \leq w_{t-2} - \epsilon$. Shift the entire forecast $\hat{w}_{jk*},\ j = 3, \ldots, T+2$ by $w_{t-1} - \hat{w}_{2k*}$, so that real wind and forecast are equal in hour $t-1$.

The forecasts are wind patterns which had similar wind speeds in hour $t-2$ and developed similarly in hour $t-1$. They are shifted to match the real wind in hour $t-1$. These pattern forecasts are made individually for every region. They are better than persistence forecasts, but for 6 or more hours ahead they are significantly worse than numerical weather predicition models. Thus, from 6 hours ahead we use a weighted combination of pattern forecasts and actual wind to adapt the RMSE to the level

shown in Figure (4.6b). The weight parameters used in this are the same throughout the regions, but dependent on the time horizon of the forecast. They are adapted so that for the first six hours the forecasts consist only of pattern-matching forecasts, and after that an increasing amount of real wind data is used to improve the forecast quality. At the 32 hour ahead mark, roughly 70% of the forecast originate from the pattern matching process, while 30% are real wind data.

4.2.3 Generating Scenarios

The scenario generation, reduction and tree construction methodology that we use in this study is based on techniques used in the WILMAR study [30, 73]. To simulate wind speed forecast errors we use a method described by Söder in [74], which is based on a multivariate ARMA time series model with one autoregressive term and one moving average term. This is commonly abbreviated as ARMA(1,1). This approach captures the correlation between wind speed forecast errors at different sites. Unpredicted wind conditions occuring at one site are likely to also affect nearby sites, resulting in a geographical correlation of the wind speed forecast errors that is essential to capture when modelling wind power uncertainty. The model is fitted to wind forecast errors in the year previous to the evaluation. For instance, to evaluate the scheduling model on 2010 wind data, we collect patterns from 2008 and synthesise forecasts for 2009. Then we use 2009 wind data to calculate forecast errors to which we fit the time series model. The evaluation is then performed on out-of-sample 2010 wind data, with new forecasts generated from 2009 patterns. We evaluate the model on wind speeds from 2009 and 2010, so the wind data used for this is from 2007 to 2010. The scenario generation, reduction and tree construction procedures described below are implemented in AMPL [57] to integrate seamlessly with our GB model.

Assume that there are $k = 1, \ldots, N$ areas and that a wind speed forecast trajectory consists of $t = 1, \ldots, T$ (here $T = 32$) consecutive hourly average wind speeds. Both, regional wind speed forecast errors and the correlation between them can be expected to increase as the forecast horizon increases [74]. It is therefore necessary to make them dependent on the forecast length t.

ARMA Model for a Single Region. The time series model for forecast errors in region $k = 1, \ldots, N$ is

$$\begin{aligned} X_{k0} &= 0 && (4.34)\\ Z_{k0} &= 0 && (4.35)\\ X_{kt} &= \alpha_k X_{k(t-1)} + Z_{kt} + \beta_k Z_{k(t-1)} && (4.36) \end{aligned}$$

with

$$\begin{aligned} X_{kt} &:= \text{wind speed forecast error in a } t \text{ hour ahead forecast for region } k\\ Z_{kt} &:= \text{independent zero-mean normal random variables with standard dev. } \sigma_{Z_k} \end{aligned}$$

Let V_{kt} be the variance of X_{kt}, i.e. the variance in a t hour ahead wind speed forecast for region k. The variance is a nonlinear function of the model parameters, $V_{kt}\left(\alpha_k, \beta_k, \sigma_{Z_k}\right)$, given by

$$\begin{aligned} V_{k0} &= 0 && (4.37)\\ V_{k1} &= \sigma^2_{Z_k} && (4.38)\\ V_{kt} &= \alpha_k^2 V_{k(t-1)} + (1 + \beta_k^2 + 2\alpha_k\beta_k)\sigma^2_{Z_k} \quad \forall t \geq 2 && (4.39) \end{aligned}$$

The recursion (4.39) follows directly from the observation that

$$\begin{aligned} V_{kt} &= \text{Var}\left(\alpha_k X_{k(t-1)} + Z_{kt} + \beta_k Z_{k(t-1)}\right)\\ &= \alpha_k^2 V_{k(t-1)} + \sigma^2_{Z_k} + \beta_k^2\sigma^2_{Z_k} + 2\text{Cov}\left(\alpha_k X_{k(t-1)}, \beta_k Z_{k(t-1)}\right), && (4.40) \end{aligned}$$

where the covariance term can be rewritten as follows

$$\begin{aligned} \text{Cov}(\alpha_k X_{k(t-1)}, \beta_k Z_{k(t-1)}) &= \mathbb{E}(\alpha_k[X_{k(t-1)} - \underbrace{\mathbb{E}(X_{k(t-1)})}_{=0}] \cdot \beta_k[Z_{k(t-1)} - \underbrace{\mathbb{E}(Z_{k(t-1)})}_{=0}])\\ &= \mathbb{E}(\alpha_k\beta_k Z_{k(t-1)}[\alpha_k X_{k(t-2)} + Z_{k(t-1)} + \beta_k Z_{k(t-2)}])\\ &= \mathbb{E}(\alpha_k\beta_k Z^2_{k(t-1)}) = \alpha_k\beta_k[\text{Var}(Z_{k(t-1)}) + \underbrace{\mathbb{E}(Z_{k(t-1)})^2}_{=0}]\\ &= \alpha_k\beta_k\sigma^2_{Z_k} && (4.41) \end{aligned}$$

We can use an induction argument to rewrite (4.39) in closed form, without recursion:

$$V_{kt} = \sigma_{Z_k}^2 \left(\alpha_k^{2(t-1)} + (1 + \beta_k + 2\alpha_k\beta_k) \sum_{i=1}^{t-1} \alpha_k^{2(i-1)} \right) \quad \forall t \geq 2. \tag{4.42}$$

To find the model paramters for each region, we solve N unconstrained nonlinear least squares (NLS) problems

$$\min_{\alpha_k, \beta_k, \sigma_{Z_k}} \sum_{t=1}^{T} \left(V_{kt}(\alpha_k, \beta_k, \sigma_{Z_k}) - RMSE_{kt} \right)^2 \quad \forall k = 1, \ldots, N \tag{4.43}$$

where V_{kt} is the ARMA model's variance according to (4.42) and $RMSE_{kt}$ is the observed RMSE in a t hour ahead forecast in region k, calculated from a set of training data by applying (4.33) for each region. These are small scale NLS problems which can be solved by any general purpose nonlinear solver. We use the MINOS [75] solver through its AMPL [57] interface, with the starting point $\alpha = \beta = 1$, $\sigma_Z = 0.05$.

Multiple Regions with Correlated Wind. In a second step we correlate the regional forecast error time series with one another. The regional one-step errors in model (4.34) to (4.36) are now assumed to be weighted sums of region specific independent standard normal random variables $\hat{Z}_{jt}$:

$$Z_{kt} = \sum_{j=1}^{N} c_{kj} \hat{Z}_{jt} \quad \forall t = 0, \ldots, T. \tag{4.44}$$

Here $c = c_{kj},\ k, j = 1, \ldots, N$ is a matrix of connection parameters between regions k and j. For a non-diagonal matrix c the errors in different regions are now dependent on one another. The variance of the one-step errors can be calculated as

$$\mathrm{Var}(Z_{kt}) = \sigma_{Z_k}^2 = \sum_{j=1}^{N} c_{kj}^2, \tag{4.45}$$

while the covariance is given by

$$\mathrm{Cov}(Z_{kt}, Z_{it}) \quad = \quad \mathbb{E}\left(\left[\sum_{j=1}^{N} c_{kj} \hat{Z}_{jt} \right] \left[\sum_{j=1}^{N} c_{ij} \hat{Z}_{jt} \right] \right)$$

$$= \sum_{j=1}^{N} c_{kj} c_{ij}. \tag{4.46}$$

This allows us to calculate the covariance of the ARMA models in all time steps. If we define $C_{kit} := \mathrm{Cov}(X_{kt}, X_{it})$, then the covariance can be expressed recursively as

$$\begin{aligned}
C_{ki0} &= 0 && (4.47)\\
C_{ki1} &= \mathrm{Cov}\left(Z_{k1}, Z_{i1}\right) = \sum_{j=1}^{N} c_{kj} c_{ij} && (4.48)\\
C_{kit} &= \mathrm{Cov}\left(X_{kt}, X_{it}\right) = \mathbb{E}\left(X_{kt} X_{it}\right) \\
&= \mathbb{E}\left(\left[\alpha_k X_{k(t-1)} + Z_{kt} + \beta_k Z_{k(t-1)}\right]\left[\alpha_i X_{i(t-1)} + Z_{it} + \beta_i Z_{i(t-1)}\right]\right) \\
&= \alpha_k \alpha_i C_{ki(t-1)} + (1 + \beta_k\beta_i + \alpha_k\beta_i + \beta_k\alpha_i) \underbrace{\mathrm{Cov}(Z_{kt}, Z_{it})}_{=\sum_{j=1}^{N} c_{kj} c_{ij}} \quad \forall t \geq 2. && (4.49)
\end{aligned}$$

In analogy to the variance calculations above, an induction argument can be used to obtain a closed form expression of the covariance recursion (4.49):

$$C_{kit} = \left[\sum_{j=1}^{N} c_{kj} c_{ij}\right]\left[(\alpha_k\alpha_i)^{t-1} + (1 + \beta_k\beta_i + \alpha_k\beta_i + \beta_k\alpha_i) \sum_{l=1}^{t-1} (\alpha_k\alpha_i)^{l-1}\right] \quad \forall t \geq 2. \tag{4.50}$$

Finally this is used to calculate the correlation coefficients between the region-specific ARMA models:

$$\rho_{kit} := \frac{C_{kit}}{\sqrt{V_{kt} V_{it}}}. \tag{4.51}$$

With the standard deviations in the denominator, the correlation is a nonlinear function of all model parameters, i.e. $\rho_{kit}\left(\alpha, \beta, \sigma_Z, c\right)$, where α, β and σ_Z have already been fitted for every region. The connection parameters c_{ki} are found by solving the NLS problem

$$\min_{c} \sum_{t=1}^{T} \sum_{k=1}^{N} \sum_{i=1}^{N} \left(\rho_{kit}^{mod}(\alpha, \beta, \sigma_Z, c) - \rho_{kit}^{obs}\right)^2 \tag{4.52}$$

with fixed α, β and σ_Z. Here ρ_{kit}^{mod} is the model correlation according to (4.51) and ρ_{kit}^{obs} is the observed correlation of forecast errors between regions k and i in a t hour ahead

forecast as observed in the training dataset, that is,

$$\rho_{kit}^{obs} := \frac{\sum_{j=1}^{n}(e_{jkt} - \bar{e}_{kt})\sum_{j=1}^{n}(e_{jit} - \bar{e}_{it})}{\sqrt{\sum_{j=1}^{n}(e_{jkt} - \bar{e}_{kt})^2 \sum_{j=1}^{n}(e_{jit} - \bar{e}_{it})^2}}. \tag{4.53}$$

In this, e_{jkt} is the error in the j-th t hour ahead forecast for region k, and $\bar{e}_{kt} := \frac{1}{n}\sum_{j=1}^{n} e_{jkt} \approx 0$ is the average forecast error in a t hour ahead forecast for region k.

This NLS problem is larger and takes longer to solve with general purpose nonlinear optimization software than the small problems described above. Again, we use the MINOS [75] solver, with the starting point $c_{ki} = \frac{\sigma_{Z_k}}{\sqrt{N}}$ which satisfies (4.45). After solving problems (4.43) to obtain α, β and σ_Z, and problem (4.52) to obtain c, we can simulate a single scenario of multivariate wind speed forecast errors by drawing standard normal variates for $\hat{Z}_{kt}$ and substituting into (4.44) and the ARMA model (4.34) to (4.36) to find the forecast error vectors X_{kt}. Finally, a wind speed scenario is obtained by adding the simulated forecast error scenario X to a central forecast. By repeating this procedure multiple times, we obtain a fan of wind speed scenarios, all of which agree on the wind speed in hour $t = 0$ and then split.

4.2.4 Constructing Scenario Trees

Our rolling horizon evaluation scheme requires multi-stage scenario trees for the intraday problems and sets of distinct scenarios for the two-stage day-ahead problems. Given the parameters obtained from fitting the model described above, we construct the scenarios as follows. We generate 600 wind speed scenarios by drawing realisations of forecast errors from the ARMA model and adding them to a synthesised forecast. Wind speeds are then translated to regional load factors by applying the nonlinear power curve shown in [69]. Finally, to obtain the power output of a wind farm under a given scenario, we multiply its generation capacity with the load factor of the region it is located in. To keep the stochastic optimization problems tractable, we reduce the number of scenarios with a technique based on Römisch et al [38, 39], which we describe below.

Kantorovich Distances. Let $\mathcal{S}^o$ be the set of *original* scenarios, $\mathcal{S}^r$ the *remaining* scenarios, and P^o and P^r the corresponding probability measures with probabilities p^o

and p^r, respectively. By $\mathcal{S}^d := \mathcal{S}^o \setminus \mathcal{S}^r$ we denote the *deleted* scenarios. We seek $\mathcal{S}^r$ such that the Kantorovich distance between P^o and P^r, calculated as

$$D(P^o, P^r) = \sum_{i \in \mathcal{S}^d} p_i^o \min_{j \in \mathcal{S}^r} d_T(i,j) \tag{4.54}$$

is minimal. In this context, we measure the difference between scenarios i and j up to time t as

$$d_t(i,j) := \sum_{w \in \mathcal{W}} \sum_{k=1}^{t} |P_{wki}^{win} - P_{wkj}^{win}|. \tag{4.55}$$

Here P_{wts}^{win} is the wind power from wind farm w at time t in scenario s. To keep the notation simple, we write $d_t(i,j)$ with the scenario indices rather than the realisation of the uncertain parameter. The probability is redistributed among the remaining scenarios $j \in \mathcal{S}^r$ according to the rule

$$p_j^r := p_j^o + \sum_{i \in \mathcal{S}_j^d} p_i^o \tag{4.56}$$

$$\mathcal{S}_j^d := \left\{ i \in \mathcal{S}^d : j \in \operatorname*{argmin}_{k \in \mathcal{S}^r} d_T(i,k) \right\}. \tag{4.57}$$

The new probability of a preserved scenario is equal to the sum of its former probability and the probability of all deleted scenarios that are closer to it than to any other preserved scenario. An optimal reduction in the sense of minimum Kantorovich distance can be achieved by solving the problem

$$\min \quad \sum_{i \in \mathcal{S}^d} p_i^o \min_{j \in \mathcal{S}^r} d_T(i,j) \tag{4.58}$$

$$\text{s.t.} \quad \mathcal{S}^d \subset \mathcal{S}^o \tag{4.59}$$

$$|\mathcal{S}^d| = N^d \tag{4.60}$$

with a fixed number N^d of scenarios to be deleted. This can be formulated as a set covering problem.

Scenario Reduction Heuristics. To approximate solutions of this problem efficiently, Gröwe-Kuska et al [39] propose two greedy heuristics named simultaneous backward reduction and fast forward selection. They are based on the observation

that the optimal selection problem can be solved easily by enumeration if only one scenario is deleted, $N^d = 1$ or only a single scenario is preserved, $N^d = |\mathcal{S}^o| - 1$. In the former case the problem reduces to

$$\min_{l \in \mathcal{S}^o} p_l^o \min_{j \in \mathcal{S}^o \setminus \{l\}} d_T(l, j), \tag{4.61}$$

and if the minimum is attained at $l^* \in \mathcal{S}^o$ then l^* is deleted. In the latter case the problem reduces to

$$\min_{u \in \mathcal{S}^o} \sum_{j \in \mathcal{S}^o} p_j^o d_T(j, u), \tag{4.62}$$

and if the minimum is attained at $u^* \in \mathcal{S}^o$ then u^* is selected to be kept. In the former case we delete a scenario which is closest to another one, whilst in the latter case we preserve a scenario which is closest to all other ones. Note that the solution of these problems is not necessarily unique. The idea behind simultaneous backward reduction is to successively delete single scenarios until the desired number of scenarios is reached, while the idea behind fast forward selection is to repeatedly select single scenarios until the desired number of scenarios is reached. Following the scenario selection, probabilities are adjusted according to rule (4.56). The algorithms are outlined below.

Algorithm 4.1 (Simultaneous Backward Reduction) *Input: scenarios $\mathcal{S}^o$, probabilities p^o, number of scenarios to delete N^d, and time t up to which distance $d_t(\cdot)$ is measured*

1. *Calculate $d_{kj} := d_t(k, j), \quad \forall k, j \in \mathcal{S}^o$.*

2. *Calculate*

$$\begin{aligned} c_{ll}^{[1]} &:= \min_{j \neq l} d_{lj}, \quad \forall l \in \mathcal{S}^o \\ z_l^{[1]} &:= p_l^o c_{ll}^{[1]}, \quad \forall l \in \mathcal{S}^o \\ l^1 &\in \operatorname*{argmin}_{l \in \mathcal{S}^o} z_l^{[1]} \\ \mathcal{S}^{d[1]} &:= \{l^1\}. \end{aligned}$$

3. For $i = 2, \ldots, N^d$ calculate

$$\begin{aligned} c_{kl}^{[i]} &:= \min_{j \notin \mathcal{S}^{d[i-1]} \cup \{l\}} d_{kj}, \quad \forall l \notin \mathcal{S}^{d[i-1]},\ k \in \mathcal{S}^{d[i-1]} \cup \{l\} \\ z_l^{[i]} &:= \sum_{k \in \mathcal{S}^{d[i-1]} \cup \{l\}} p_k^o c_{kl}^{[i]}, \quad \forall l \notin \mathcal{S}^{d[i-1]} \\ l^i &\in \operatorname*{argmin}_{l \notin \mathcal{S}^{d[i-1]}} z_l^{[i]} \\ \mathcal{S}^{d[i]} &:= \mathcal{S}^{d[i-1]} \cup \{l^i\}. \end{aligned}$$

4. $\mathcal{S}^d := \mathcal{S}^{d[N^d]}$ is the set of deleted scenarios. Set $\mathcal{S}^r = \mathcal{S}^o \setminus \mathcal{S}^d$ and redistribute the probabilities of deleted scenarios according to (4.56).

The backward reduction algorithm uses auxiliary variables c and z to calculate Kantorovich distances. In Step 2 we choose the first deletable scenario l^1 such that its distance to the closest scenario among all others is minimal. In Step 3 we find l^i for $i = 2, \ldots, N^d$ such that deleting l^i leads to the smallest possible Kantorovich distance between the probability measures associated with the original and remaining scenario sets. To find this l^i, we iterate over all remaining candidates for deletion, $l \notin \mathcal{S}^{d[i-1]}$, and calculate the Kantorovich distance $z_l^{[i]}$ that would result if l was deleted, that is, if $\mathcal{S}^{d[i-1]} \cup \{l\}$ was the set of deleted scenarios.

Algorithm 4.2 (Fast Forward Selection) *Input: scenario set $\mathcal{S}^o$, probabilities p^o, number of scenarios to delete N^d, and time t up to which distance $d_t(\cdot)$ is measured*

1. Calculate

$$\begin{aligned} c_{ku}^{[1]} &:= d_t(k, u), \quad \forall k, u \in \mathcal{S}^o \\ z_u^{[1]} &:= \sum_{k \in \mathcal{S}^o \setminus \{u\}} p_k^o c_{ku}^{[1]}, \quad \forall u \in \mathcal{S}^o \\ u^1 &\in \operatorname*{argmin}_{u \in \mathcal{S}^o} z_u^{[1]} \\ \mathcal{S}^{d[1]} &:= \mathcal{S}^o \setminus \{u^1\}. \end{aligned}$$

2. For $i = 2, \ldots, |\mathcal{S}^o| - N^d$ calculate

$$c_{ku}^{[i]} := \min\{c_{ku}^{[i-1]}, c_{ku^{i-1}}^{[i-1]}\}, \quad \forall k, u \in \mathcal{S}^{d[i-1]}$$

$$
\begin{aligned}
z_u^{[i]} &:= \sum_{k \in \mathcal{S}^{d[i-1]} \setminus \{u\}} p_k^o c_{ku}^{[i]}, \quad \forall u \in \mathcal{S}^{d[i-1]} \\
u^i &\in \operatorname*{argmin}_{u \in \mathcal{S}^{d[i-1]}} z_u^{[i]} \\
\mathcal{S}^{d[i]} &:= \mathcal{S}^{d[i-1]} \setminus \{u^i\}.
\end{aligned}
$$

3. *$\mathcal{S}^d := \mathcal{S}^{d[|\mathcal{S}^o|-N^d]}$ is the index set of deleted scenarios. Redistribute the probabilities of deleted scenarios according to (4.56).*

The forward selection algorithm also uses auxiliary variables c and z to calculate Kantorovich distances. In Step 1 we choose the first scenario to be retained, u^1, such that the resulting Kantorovich distance is minimal. In Step 2 we find u^i for $i = 2, \ldots, |\mathcal{S}^o| - N^d$ such that retaining u^i leads to the smallest possible Kantorovich distance between the probability measures associated with the original and remaining scenario sets. For all candidates $u \in \mathcal{S}^{d[i-1]}$ we calculate the distance $z_u^{[i]}$ that would result if u was retained rather than deleted. Before $z_u^{[i]}$ can be calculated, this requires us to update the distance of all deleted scenarios $k \in \mathcal{S}^{d[i-1]}$ from the potential new set of selected scenarios containing u according to $\min\{c_{ku}^{[i-1]}, c_{ku^{i-1}}^{[i-1]}\}$, because scenario k could be closer to some other previously selected scenario than it is to u. Finally, u^i is selected so as to give the minimum Kantorovich distance $z_u^{[i]}$.

Tree Construction. With these reduction algorithms it is possible to derive various scenario tree construction procedures. The technique described in [39] allows the scenarios to split at any time period in the planning horizon and uses a threshold for the maximum Kantorovich distance between the origial distribution and the final tree to decide on the number of retained scenarios. However, we aim for a technique that allows us to specify the tree structure and the number of retained scenarios in advance. Our approach is inspired by the algorithm described in [73]. Our tree construction method begins on the last stage of the tree, by reducing the scenarios to the number of desired leaves. Then it proceeds recursively from stage to stage until it reaches the first stage. On every stage, the number of scenarios is reduced further, until the desired number for the current stage is reached. When scenarios are merged into a bundle, they are replaced with their average wind scenario rather than choosing one representative scenario. This reduces the variance at the beginning of the planning horizon,

but maintains the expectation of the initially selected scenarios.

During the recursive tree construction process, the algorithm produces the data structure for bundles $b \in \mathcal{B}$, with member scenarios $\mathcal{S}_b$ and start and end times t_b^{st} and t_b^{end}, respectively. The probabilities for individual scenarios are determined by the initial reduction. The algorithm requires the user to specify a vector of split points τ, which contains all time periods where the scenarios are allowed to split, sorted in decreasing order. Additionally, it requires an input vector ν whose i-th entry specifies the number of permitted scenarios from split point τ_i onwards. For instance, a possible input would be $\tau = [10, 7, 4, 1]^T$ and $\nu = [8, 4, 2, 1]^T$, meaning that the desired tree has three stages of three hours each, and a final stage that covers the remainder of the day. There is one scenario on the first stage, two on the second, and four and eight scenarios on stages three and four, respectively. The scenario tree construction method works as follows.

Algorithm 4.3 (Scenario Tree Construction) *Input: scenarios $\mathcal{S}^o$, probabilities p^o, input vectors τ and ν and their number of entries m*

1. *Perform forward selection with inputs $\mathcal{S}^o$, p^o, $t = T$ for distance measure d_t, and $N^d = |\mathcal{S}^o| - \nu_1$. This results in ν_1 remaining scenarios, stored in $\mathcal{S}^{r[1]}$, with probabilities p^r re-distributed according to (4.56). Make a copy $\hat{p}^{r[1]} := p^r$.*

2. *For $i = 2, \ldots, m$*

   - *Perform steps (1)–(3) of the backward reduction algorithm with input $\mathcal{S}^{r[i-1]}$, $\hat{p}^{r[i-1]}$, $t = \tau_{i-1} - 1$ for distance measure d_t, and $N^d = \nu_{i-1} - \nu_i$. This results in deletable scenario set $\mathcal{S}^{d[i]}$ and remaining scenario set $\mathcal{S}^{r[i]}$. For all scenarios $l \in \mathcal{S}^{d[i]}$, find a remaining scenario $u_l \in \mathcal{S}^{r[i]}$ which is closest to it, i.e. $u_l \in \operatorname{argmin}_{u \in \mathcal{S}^{r[i]}} d_t(l, u)$.*
   - *For $u \in \mathcal{S}^{r[i]}$, introduce a new bundle b with start time $t_b^{st} := \tau_i$ and end time $t_b^{end} := \tau_{i-1} - 1$ and member scenarios $\mathcal{S}_b := \{u\} \cup \{l \in \mathcal{S}^{d[i]} : u_l = u\}$. Remember the bundle owner $o(b) = u$ and extend the members $\mathcal{S}_b = \mathcal{S}_b \cup \left(\bigcup_{\bar{b}:o(\bar{b}) \in \mathcal{S}_b} \mathcal{S}_{\bar{b}}\right)$ until no more scenarios can be added this way. For $t = 1, \ldots, \tau_{i-1} - 1$ and all wind farms $w \in \mathcal{W}$, calculate the bundle's average*

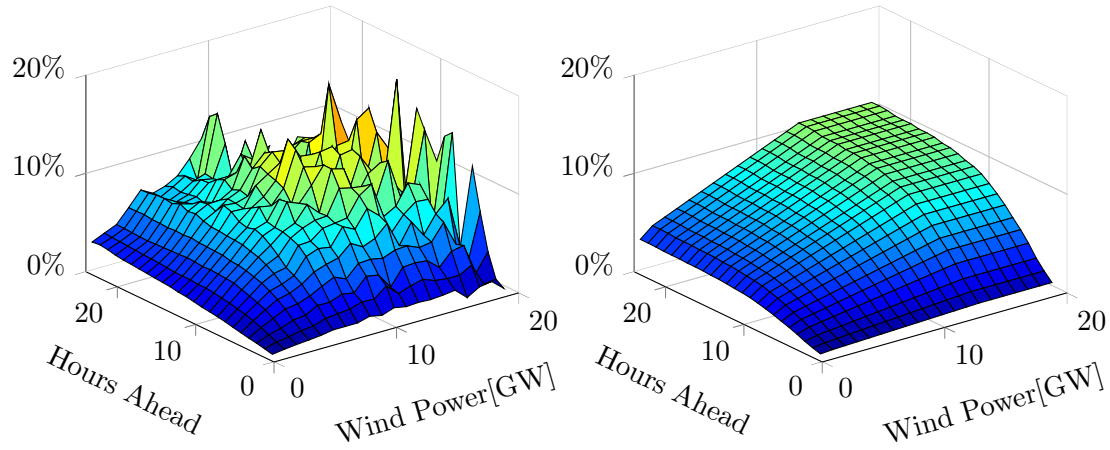
wind $\bar{P}^{win}_{wtb} := \sum_{s\in\mathcal{S}_b} p^r_s P^{win}_{wts} / \sum_{s\in\mathcal{S}_b} p^r_s$ *and set* $P^{win}_{wts} := \bar{P}^{win}_{wtb}$ *for all* $s \in \mathcal{S}_b$.
Update $\hat{p}^{r[i]}_u := \sum_{s\in\mathcal{S}_b} p^r_s$.

The tree construction algorithm may produce bundles b which are singletons, $|\mathcal{S}_b| = 1$, and these can simply be removed afterwards. Upon completion of the tree construction procedure, the reduced set of scenarios forms a tree, i.e. it satisfies property (2.74).

Level-Dependent Forecast Errors. The scenario generation methodology described above is based on techniques used in the WILMAR study [73]. The scenarios drawn from the ARMA model represent the correlation of wind forecast errors in different regions, but are independent of the forecast wind level. Mauch et al [76] point out that wind power forecast errors are strongly dependent on the forecast levels, so efficient scheduling strategies require wind dependent reserve margins. In order to make stochastic strategies dependent on the forecast level, the variance of the scenario generator must vary with it. Since this is not reflected in the WILMAR scenario generation method, we use a simple scaling approach to adapt the trees used in this study. In each scenario s we replace the original wind at all times t, P^{win}_{wts}, by $(\beta_t P^{win}_{wts} + (1-\beta_t)\bar{P}^{win}_{wt})$ with $\beta_t \in [0,1]$. Here $\bar{P}^{win}_{wt}$ is the average wind under all scenarios and β_t depends linearly on $\bar{P}^{win}_{wt}$. The resulting variance is shown in Figure 4.7, alongside the root mean square of errors where the forecast overestimated the actual wind. We choose β_t so that the variance of the scenarios matches the RMSE of situations where the actual wind was overestimated because those cases can result in significantly increased cost, due to lost load or the use of expensive fast-start units. On the other hand, cases where the wind was underestimated can be dealt with by curtailing it at no extra cost. The results described in Section 4.4 show that the scaling leads to a significant cost reduction ($100k per day) in comparison to scheduling with scenario trees which are independent of the forecast level. The scenario tree construction procedure completes the input data generation process for our stochastic UC models. The overall process can be summarised in the following steps

1. Synthesise wind forecasts with the desired RMSE

2. Fit a multi-variate ARMA model to the regional forecast error time series



(a) NRMSE of wind power overestimates.

(b) NRMSE of scenario trees.

Figure 4.7: Left: NRMSE of synthesised forecasts, as function of forecast horizon (hours ahead) and forecast wind level (FC level). Only wind power overestimates were included in the error calculation. Right: NRMSE of the generated scenario trees. The error is scaled in the forecast wind level: for a fixed forecast level above 15GW, any one-dimensional slice through the surface is equal to the NRMSE function shown in Figure 4.6b, while for lower levels the same function is scaled by a linear factor.

3. Simulate wind scenarios for every stochastic UC problem (day-ahead or intraday) and reduce them to a small subset of representative scenarios

4. Merge the remaining scenarios into a suitable tree structure for a multi-stage decision problem (intraday only) and adapt the scenario spread depending on the wind forecast expectation

We use this approach to generate scenario data for a two year rolling horizon evaluation of day-ahead and intraday stochastic unit commitment. The following sections describe the use of this data in the evaluation process and the results we obtained with it.

4.3 Rolling Horizon Evaluation

We compare multi-stage stochastic and deterministic scheduling in the intraday setting, and two-stage stochastic and deterministic scheduling in the day-ahead setting. The evaluation is done in a rolling horizon manner, where 24-hour schedules are made for a central wind forecast or a set of wind scenarios, and then evaluated against the actual wind by solving a set of dispatch problems. After the evaluation step, the planning horizon is moved forward to decide the next schedule. We repeat the procedure until a

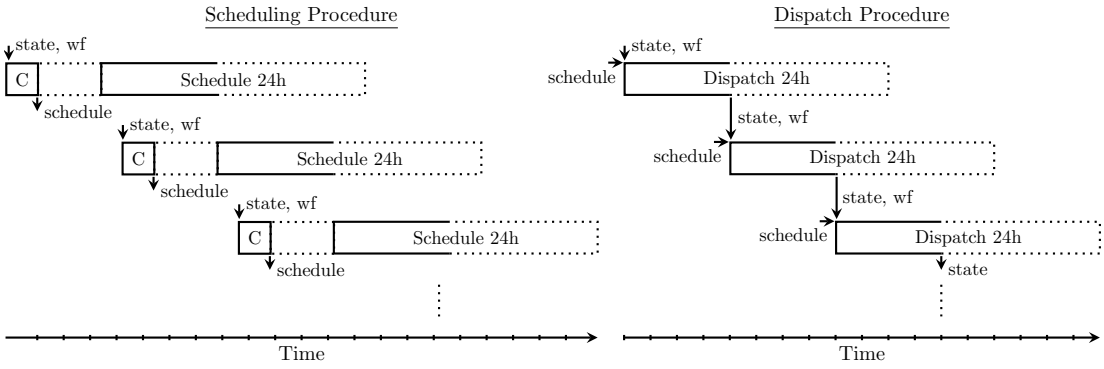


Figure 4.8: Rolling horizon evaluation procedure. The scheduling steps (left) obtain the current system state (state) and a wind forecast (wf). They calculate (C) a new schedule which becomes active a few hours later (dotted lines) and is valid for 3, 6 or 24 hours (solid part of box). The dispatch steps (right) obtain the current state and the schedule, and evaluate it against the actual wind 3 hours at a time. The model is formulated for 24 hours to avoid emptying the reservoirs towards the end of the 3 hours. A wind forecast (wf) is used for the additional 21 hours. The rolling procedure alternates between a scheduling step and (potentially multiple) dispatch steps.

period of two years is covered and compare the average cost of the different planning techniques. Intra-day UC is performed with 3-hour and 6-hour steps, i.e. the binary decisions for large generators can either be updated every 3 hours or every 6 hours. Day-ahead commitments can only be updated once per day. Other than the update frequency, there are no fundamental differences between the rolling horizon procedures for 3-hour, 6-hour and 24-hour scheduling. In the following, we describe a generic procedure which is applicable to all of them. The process is visualised in Figure 4.8. The scheduling and dispatch steps are shown separately on the graphic, but are interlaced in the implementation where the rolling procedure alternates between them.

Scheduling Steps. Calculating a schedule in practice is not instantaneous and must be done a few hours in advance of its implementation. In Figure 4.8 (left) we show how a schedule is calculated, using the current system state and a wind forecast. The current state is required to estimate the system state immediately before the implementation of the schedule. After calculating the schedule, it is reported to the dispatch procedure and becomes active a few hours later. We assume that the time between calculating and implementing a schedule is 3, 6 and 8 hours in 3-hour, 6-hour and 24-hour planning, respectively. When implemented, the schedule is active for 3, 6 or 24 hours, and the

next one is made in time to become active as soon as its predecessor expires.

Dispatch Steps. The dispatch steps are used to estimate operational costs of implementing a schedule. The dispatch model uses the same hourly granularity as the scheduling model. It is formulated as a 24h problem, but only the first 3 hours are used to estimate the costs. The dispatch model has a single wind scenario with 3 hours corresponding to the actual wind and a forecast for the remainder. Inside the model, the active schedule is fixed, and all recourse decisions are made cost-minimally, that is, use of OCGT, levels of reserve and response, pump storage operation and shed load. We record the resulting operational cost for a 3-hour period, including penalties for underserved reserve and response and lost load. Additional time periods after the first 3 hours are included to avoid reservoirs being emptied towards the end of 3 hours. The calculated system state is used as initial state for the next dispatch problem. The rolling horizon procedure alternates between scheduling steps and dispatch steps. While schedules are made for 3, 6 or 24 hours at a time, the dispatch model always evaluates 3 hours to keep the results consistent and comparable. Thus, multiple dispatch steps are required after a single scheduling step if the schedule is valid for more than 3 hours. For the dispatch we choose 3-hour steps instead of one-hour steps to save computing time.

Overview of Test Runs. We evaluate seven different types of scheduling: for each of the three approaches with updates every 3, 6 or 24 hours, we run a stochastic and a deterministic version. For reference, we also perform one additional run with perfect foresight, by solving a single-scenario combined scheduling and dispatch model in which future wind power is known in advance. The amount of reserve plus response (4.19) in this is treated as a soft constraint, unlike in the other deterministic scheduling models which use a fixed margin for reserve plus response.

The stochastic models have the following structure: for 3-hour scheduling we use a multi-stage scenario tree with 3 stages of 3 hours each and a final stage that covers the remainder of the day. There are 3 scenarios on the first stage, then 6, 9 and 12 on subsequent stages. For 6-hour scheduling we use trees which have 4 stages of 6 hours each, with 4 scenarios on the first stage, and then 8, 10, and 12 scenarios on subsequent

stages. The two-stage day-ahead model has 12 senarios on the second stage.

4.4 Evaluation Results

The following two sections explain the results of our two-year rolling horizon evaluation. Section 4.4.1 discusses the effects of wind variability and wind uncertainty on network congestion and locational marginal prices in the system, and on the way pump storage plants are operated. The purpose of this section is to demonstrate how the model uses its flexible storage assets to overcome adverse effects of wind variability and uncertainty in a setting with restrictive transmission constraints. Hence it provides an a posteriori motivation for modelling the power system in as much detail as was done for this study. Finally, Section 4.4.2 presents our main results concerning the economic comparison of deterministic and stochastic scheduling approaches. In our evaluations, reserve and response margins for *deterministic* scheduling strategies were set by the formula 'capacity of the largest generator plus r% of the forecast wind'. A range of cases were evaluated with r taking values between 0% and 50%. Due to the scenario scaling approach discussed in Section 4.2, the *stochastic* strategies also depend on wind forecast levels.

4.4.1 Pump Storage Operation and Network Congestion

In this section we demonstrate the interaction of the different model elements, i.e. thermal generation, pump storages and the transmission model, and show what influence variable and uncertain wind power supply has on them. We show how the model operates pump storage plants under normal circumstances and how they are used to deal with specific wind situations. To do this we look at both, the pump storage operation under perfect foresight and under wind forecast uncertainty. This allows us to explore how the forecast uncertainty affects the way in which pump storages are operated and how they contribute towards using more of the uncertain wind power to satisfy demand. Furthermore, we look at examples of network congestion due to specific wind situations and show how LMPs fluctuate when it appears.

Effects of Wind Variability. Figure 4.9 shows how demand and actual wind power supply affect congestion, pump storage operation and the available spare capacity (headroom) in thermal generation units. The numbers shown here were taken from the perfect foresight evaluation. This allows us to eliminate effects of wind power uncertainty completely and focus entirely on aspects of wind power variability: at times of low wind supply we can observe regular operation of the system, while at times of high wind supply we can observe the system's best response to the wind, given its network restrictions and storage capability.

The third graph from the top in Figure 4.9 shows net demand and actual wind power supply, which are the main drivers for how the system is operated. The top graph in Figure 4.9 shows the reservoir levels at the four pump storages sites. Except for the Foyers storage in the north of Scotland, all of them follow the daily demand cycles for most of the month: when demand is low and marginal generation cost is low the storages are filled, and during peak demand they are emptied again. The Welsh storages (Z9) have large pump-turbines and achieve steeper pump and discharge ramps than the Scottish storages (Z1, Z4). When wind variability has an influence on the pump storage operation, e.g. because there is a lot of wind power available that must be stored because it cannot be consumed immediately, the Welsh storages reach their limits quickly, while the Scottish storages respond slowly. The Scottish reservoirs never reach their upper limits as part of the regular day- night cycle.

At night when demand is low all storages are emptied almost completely and cannot provide large amounts of reserve or response. The second graph from the top in Figure 4.9 shows the total headroom of part-loaded thermal units and explains why it is not necessary to provide large amounts of reserve or response via pump storages: at night the thermal units are ramped down, resulting in an increase in the headroom, so most of the reserve and response can be provided by thermal units.

The wind variability has a noticeable effect on pump storage operation, in particular if both the available wind and the storage capacity are on the same side of a transmission constraint. The bottom graph shows LMPs for zones Z1 and Z2 in the north of Scotland, and for the rest of the system. During the particular month shown on the graphs, network congestion only led to different LMPs in these zones, while the rest of the network was not congested. Zones Z1 and Z2 contain a significant amount of wind

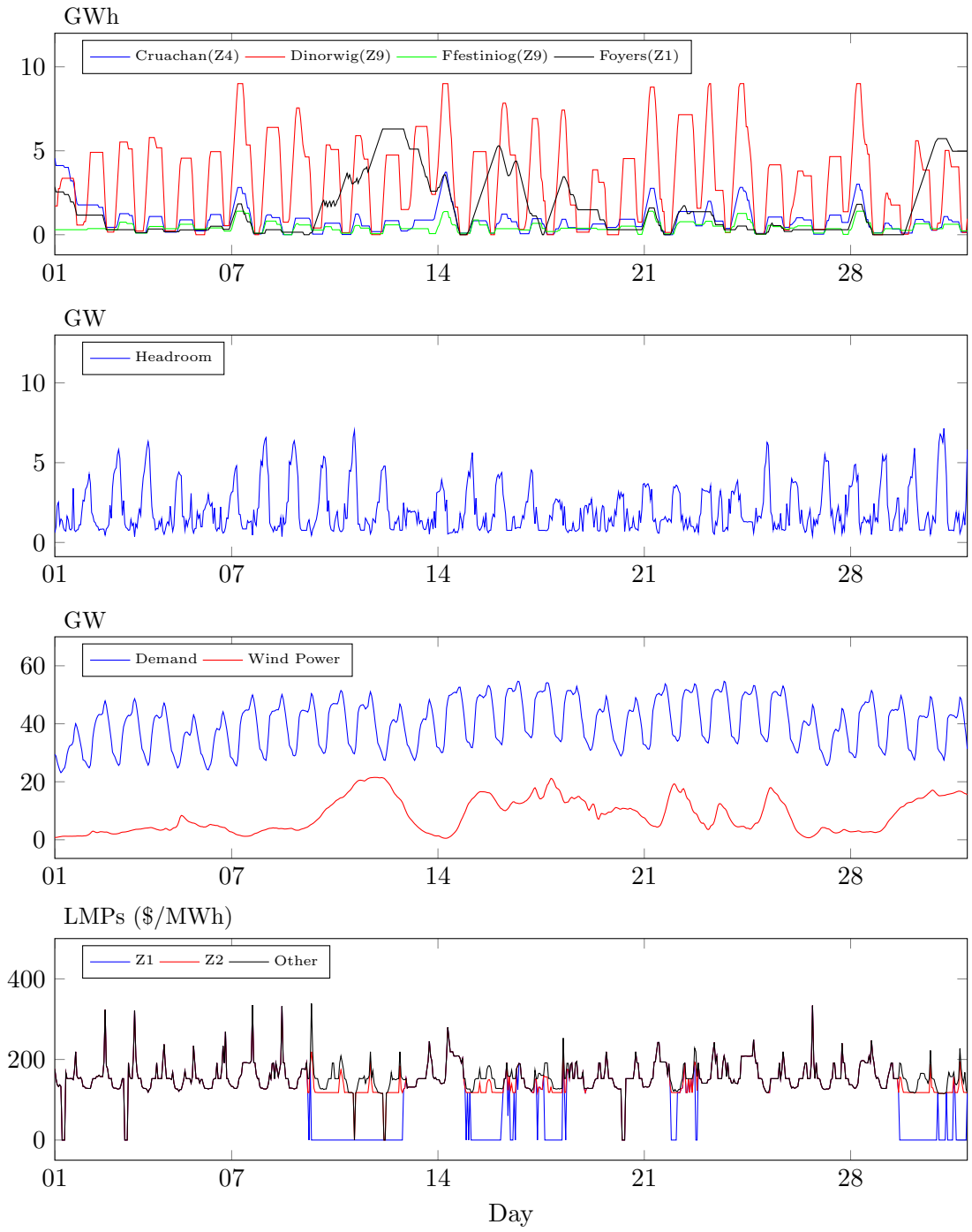


Figure 4.9: Pump storage operation and network congestion and the effect of wind power fluctuations on them. The graphs show the first month (January) of the evaluation. The numbers are taken from a scheduling and dispatch evaluation under perfect foresight. The top graph shows reservoir levels of the four pump storage schemes. The graph below shows the total spare capacity (headroom) in part-loaded generators. The third graph shows demand (net of interconnection) and available wind power. Finally the bottom graph shows LMPs (dual solutions of (4.4)) for zones Z1, Z2 and all others. During the first month, only zones Z1 and Z2 had LMPs different from the other ones.

power generation capacity, and at times of high wind power availability the LMPs drop (bottom graph), and the Foyers storage gives up its regular day-night cycle in favour of storing excess wind power that cannot be transported to the rest of the system. On Days 12 and 13 the reservoir is completely full, and on day 14 it is being emptied again since the wind power supply decreased. Thus the model uses pump storage to alleviate consequences of network congestion at times of high wind availability and achieves higher wind penetration levels and lower costs than would be possible without storage.

Effects of Wind Uncertainty. In the previous paragraph we discussed regular daily pump storage cycles at times of low wind and showed how these change at times of high wind power outputs. The model uses pump storage as a means of storing wind power which is otherwise unavailable because it cannot be transported and consumed at the time it is available. This effect was purely due to wind power *variability*, since the evaluation was done under perfect foresight. In this section we explore the effect of wind forecast *uncertainty* on pump storage operation, by examining the reservoir levels in the evaluation of deterministic and stochastic scheduling strategies. Pump storage operation is a recourse decision made by the dispatch model under knowledge of the actual wind power, but optimal pump and discharge decisions are a consequence of the thermal schedule which is decided under uncertainty. Figure 4.10 shows the operation of pump storage reservoirs and the total capacity of scheduled thermal generators under deterministic and stochastic scheduling strategies and under perfect foresight.

The bottom graph in Figure 4.10 shows the total thermal generation capacity committed under the various scheduling schemes. The amount of available wind power has a significant influence on this: at times of high wind the thermal capacity reduces significantly. At these times the deterministic and stochastic strategies schedule more thermal capacity than the perfect foresight procedure, since both are aware of the higher forecast uncertainty in comparison to times of low wind. During these peak wind times the deterministic strategy is more extreme than the stochastic one: in peak demand hours it tends to schedule more capacity, and at low demand hours it schedules less capacity than the stochastic strategy. At times of low wind power availability, both, deterministic and stochastic strategies commit nearly the same capacity as the perfect foresight procedure.

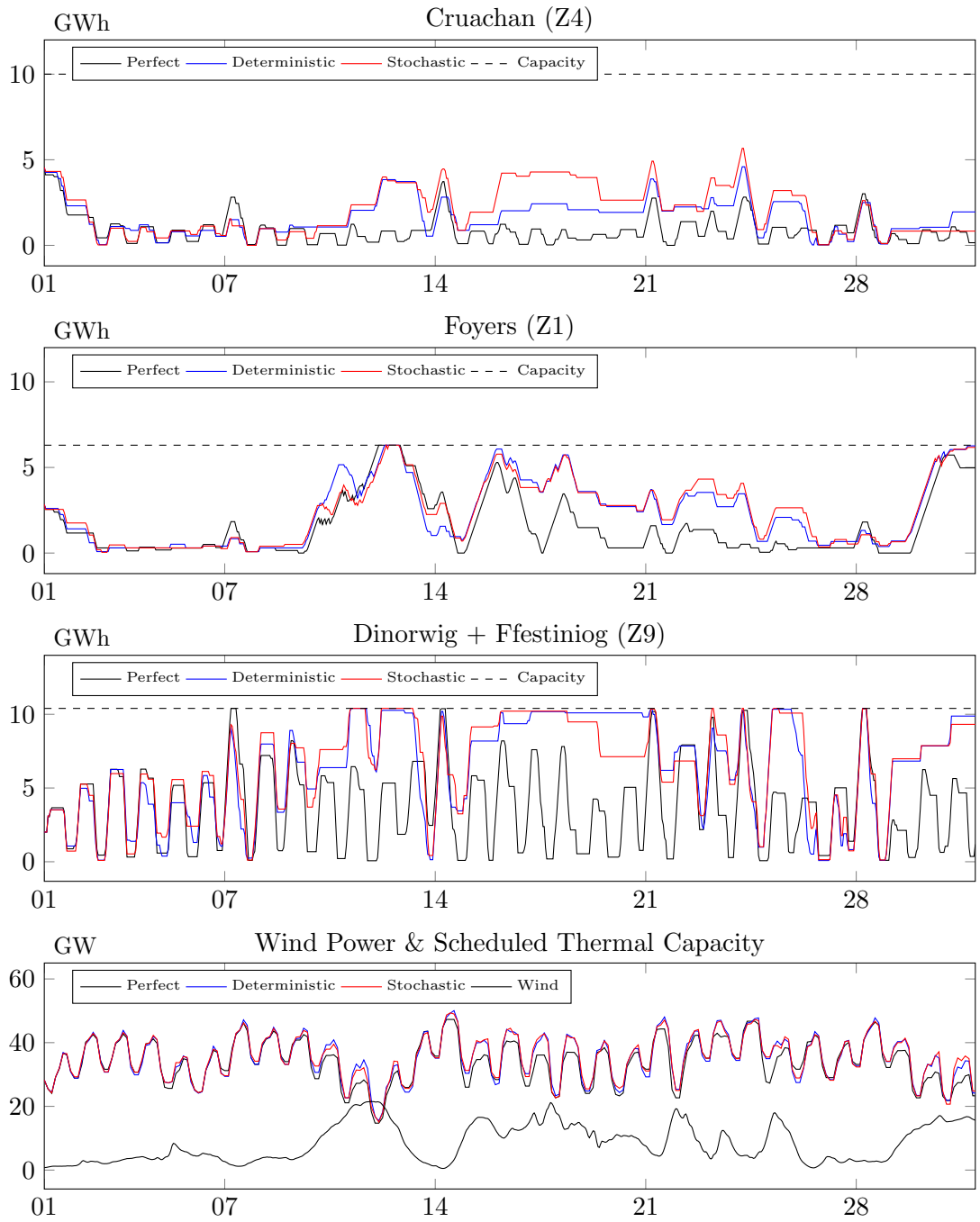


Figure 4.10: Additional graphs on pump storage operation under various scheduling strategies. We show the first month (January) of the stochastic, perfect foresight, and the most successful deterministic 3-hour evaluation. The top two graphs show the reservoir levels at the Scottish pump storage plants (Z1 and Z4). The third graph shows the sum of the reservoir levels at the Welsh storages (Z9). Finally the bottom graph shows the scheduled conventional generation capacity and the available wind power supply. All numbers are taken from the dispatch stage of the evaluation process, where reservoir levels are optimal recourse decisions under actual wind availability but conventional capacity is scheduled under uncertainty (except in perfect foresight case).

The top graph in Figure 4.10 shows the reservoir level at the Cruachan storage scheme in zone Z4. Its operation is affected significantly by the wind forecast uncertainty: due to the higher levels of committed thermal capacity at times of high wind, the typical day-night cycle disappears while the reservoir is used to store wind energy that cannot be consumed immediately. The Cruachan scheme has a large reservoir in comparison to its pump capability and its day-night cycles are relatively small. It is well suited to absorb and store unused wind power, but may be more effective with additional pumping capacities. The economic value of installing additional machines is explored in the next section, in case Z4 in Figure 4.13.

The second graph from the top in Figure 4.10 shows the reservoir level at the Foyers storage scheme in the north of Scotland (Z1). This zone has very low demand but large amounts of installed wind power generation capacity and lies behind a restricitve transmission constraint. The day-night cycles at the Foyers storage are very small. However, at times of high wind it is required to absorb unused wind that cannot be exported to the rest of the system, and this is the case even under perfect foresight scheduling. When the reservoir reaches its upper limit it is necessary to curtail wind. Thus there is a value to adding additional storage capacity in this part of the system, and this is explored in the next section, in case Z1 in Figure 4.13. Similar to Cruachan, the Foyers storage is required to store more energy for a longer period of time when wind forecasts are uncertain, i.e. in the stochastic and deterministic cases.

The third graph from the top in Figure 4.10 shows the combined reservoir levels of the Dinorwig and Ffestiniog storages in Wales (Z9). Dinorwig is the most powerful storage in the system, with both, a large reservoir and powerful pump-turbines. Both storages are located in a zone that is well connected with areas of large demand and conventional generation capacity. The day-night cycles are very expressed, and without wind uncertainty these reservoirs are not used for long term wind energy storage. However, in the deterministic and stochastic evaluations with uncertain wind, they are also used as wind energy storages.

4.4.2 Deterministic and Stochastic Performance

The results of our two-year evaluation are shown in the graphs in Figure 4.11. A range of *deterministic* cases were evaluated with the forecast-dependent reserve and response

margin r taking values between 0% and 50%. The resulting average margin is what is shown on the x-axes in Figures 4.11 and 4.12. Stochastic models allocate reserve for forecast errors based on their scenarios, while being aware of the recourse cost of keeping too little additional reserve and response for potential failures. Hence they determine optimal reserve and response levels internally and only need to be evaluated once, unlike the deterministic cases which we evaluated for multiple values of r. On the graphs in Figure 4.11, the horizontal dotted lines show the values achieved by the stochastic cases with 3-hour, 6-hour and 24-hour schedule updates.

Average Cost. The total cost consists of no-load, startup and marginal generation costs and various recourse costs. It comes to roughly $134 per MWh. Recourse costs include the cost of lost load, underserved reserve and response, and OCGT usage. For the 6-hour deterministic cases, the graph in Figure 4.12 shows a detailed breakdown of these costs. The total generation cost of slow units is determined by the demand and the average marginal cost of the committed generators: in Figure 4.12 they increase from left to right as the amount of OCGT usage decreases and more demand is satisfied from cheaper, slow units. No-load and startup costs of slow units also increase from left to right, while recourse costs decrease from left to right, resulting in cost minima between average reserve and response (R&R) margins of 2.7GW and 3.1GW.

The top left graph in Figure 4.11 shows an overview of the daily average cost achieved with all scheduling strategies. The deterministic procedures all have cost minima between average R&R margins of 2.5GW and 4GW, where very little or no load is shed and the gradients of increasing no-load, startup and generation costs cancel out with decreasing OCGT and R&R costs. In an area around these minima the cost curve is flat: evaluations with different reserve margins give similar cost. The total cost decreases if commitments of slow units can be revised more regularly: 24-hour scheduling is more expensive than 6-hour scheduling, which in turn is more expensive than 3-hour scheduling. The maximum room for improvement through better forecasts or better (e.g. stochastic) scheduling methods is indicated by the cost under perfect foresight. The average costs with stochastic scheduling models are lower than the minimum costs achieved with the corresponding deterministic models: the gaps are $100k ($approx 0.1\%$) per day in the 3-hour and 6-hour cases, and $300k ($\approx 0.3\%$) per

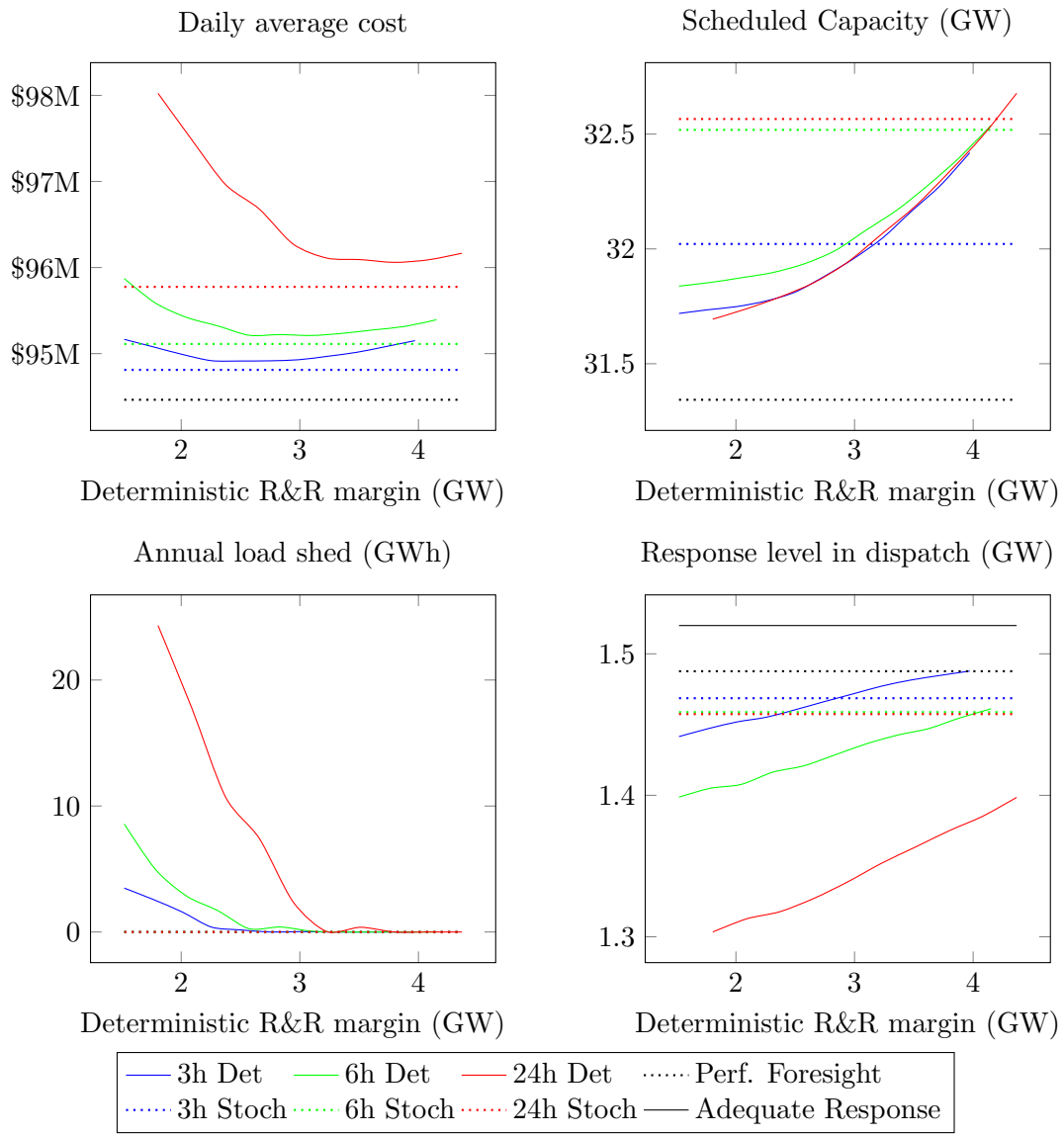


Figure 4.11: Results of a two year evaluation of deterministic and stochastic 3-hour and 6-hour intraday and 24-hour day-ahead scheduling. For reference, the performance of perfect foresight scheduling is also included. The value on the x-axes is the average set margin for reserve and response in deterministic scheduling problems. Graphs of stochastic results are dotted to indicate that they were not solved with different set margins. The top left graph shows the average daily cost of the different scheduling strategies, including penalties for load shedding and not keeping enough response. The bottom left graph shows the total proportion of load shed over the two years. The top right graph shows the average conventional generation capacity scheduled by the various approaches. The bottom right graph shows the average amount of response available at the dispatch stage. Here, the 'adequate' level indicates the level below which a penalty is incurred for not keeping enough response.

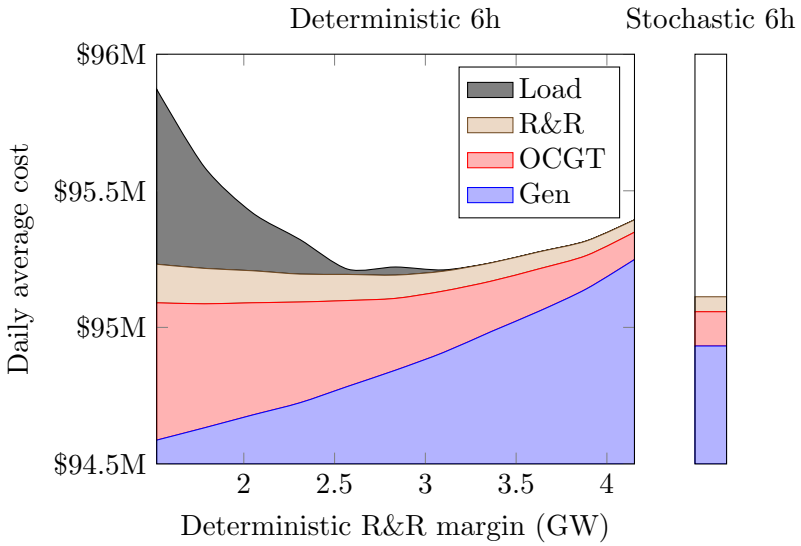


Figure 4.12: Breakdown of daily average cost into startup, no-load and generation costs of slow units (Gen), and recourse costs. The recourse costs are for OCGT usage (OCGT), underserved reserve and response (R&R) and shed load (Load). The generation cost portion of (Gen) increases from \$91.5M in the leftmost case to \$91.9M in the rightmost case (+\$0.4M). The remainining increase in (Gen) is due to startup and no-load costs which increase from \$3.1M to \$3.34M (+\$0.24M). The graph shows the deterministic 6-hour rolling cases with various fixed reserve and response margins. For the case with average set margin of 2.7GW, Figure 4.5 in Section 4.2 shows how the R&R penalty cost is accrued at different levels of underserved reserve and response. For comparison, the average daily costs in the corresponding stochastic case are shown on the bar to the right of the graph.

day in the 24-hour case. While the stochastic cases have higher generation costs than the best deterministic cases, the recourse costs are lower (cf. example in Figure 4.12).

The cost in the 3-hour stochastic case is \$350k ($\approx 0.35\%$) per day higher than in the perfect foresight case. The gap between these is the value of perfect information. To see if it can be reduced by including more than 12 scenarios in the stochastic model, we performed the same evaluation again with 20 Scenarios. However, the resulting cost did not change significantly ($< 0.01\%$). The stochastic evaluations were also performed without the scenario scaling approach that makes the scenario spread dependent on the forecast level: due to higher generation cost this led to a worse overall performance of stochastic scheduling, which eliminated the gap between the stochastic procedure and the best deterministic procedure. Achieving minimal operational cost requires a careful balance of committed spare capacity and the use of costly recourse actions, and stochastic models tend to over-commit conventional capacity in situations with

low wind if the correlation between wind speed and forecast errors is not taken into account.

Load Shedding. The bottom left graph in Figure 4.11 shows the average annual load shed over the two years. While all stochastic models avoid shedding any load, deterministic models shed load if the set R&R margin is not high enough. At \$27k per MWh, the cost of load shedding is large enough to dominate the shape of the deterministic cost curves to the left of their minima in the top left graph of Figure 4.11. The impact of the cost of load shedding on these cost graphs is also visualised in Figure 4.12. Increasing the deterministic R&R margin does not always lead to a reduction in shed load: in the deterministic model spare capacity is allocated based on cost only, ignoring potential network congestion, which sometimes leads to situations where it is lumped behind a transmission constraint and unavailable elsewhere. Stochastic models, on the other hand, allocate generation capacity based on correlated wind scenarios and are aware of the network restrictions in the potential operational states. Hence spare capacity is allocated where it is needed to deal with critical wind situations.

Scheduled Capacity. The top right graph in Figure 4.11 shows the average committed conventional generation capacity. For deterministic cases, the capacity is a consequence of the average wind power level in the forecasts and the set R&R margin. The capacity curves for 3-hour, 6-hour and 24-hour scheduling are all relatively close together. The committed capacity increases with the R&R margin, and this drives the cost increase to the right-hand side of the cost minima in the top left graph.

In stochastic problems, the scheduled capacity is a consequence of the average wind power forecast level and the scenario variation at times for which scheduling decisions are made. The relevant forecast horizon is 4 to 6 hours, 7 to 12 hours, and 8 to 32 hours ahead in 3, 6 and 24-hour scheduling, respectively. Figure 4.7b shows the variation of generated scenarios for these varying forecast horizons: while there is only a small difference between the average errors relevant for 6 and 24-hour scheduling, those relevant for 3-hour scheduling are notably lower. Consequently the 6 and 24-hour rolling procedures committed similar capacity levels, while 3-hour rolling committed a lower level.

System-Wide Response Levels. The bottom right graph in Figure 4.11 shows the average amount of response in the dispatch. The 'adequate' level indicates where the response penalty curve is first different from zero, and levels below that incur the corresponding penalty. On average, perfect foresight scheduling chooses a level where a small penalty applies, and the stochastic scheduling strategies lead to similar levels. Deterministic procedures lead to lower response levels than their stochastic counterparts, but their level increases with the R&R margin. If we take low response levels as an indicator that the power system is exposed to high stress due to forecast uncertainty, then this shows the stress reduction through stochastic scheduling. The gaps between the deterministic curves and their stochastic counterparts differ systematically: in 3-hour scheduling the curves are at a similar level, while in 6-hour and 24-hour scheduling they are further apart. The higher the forecast uncertainty, the larger the stress reduction through stochastic scheduling. The forecast uncertainty also explains the difference between the top and bottom right graphs: the 6-hour and 24-hour scheduling procedures commit a higher capacity level than the 3-hour procedure, but result in less response at the dispatch stage, as the remaining headroom is used towards dealing with the higher forecast uncertainty.

Network Congestion. Table 4.1 shows differences in locational marginal prices (LMPs) between selected zones, averaged over the two-year planning horizon. The LMP values shown here are the dual solutions of constraints (4.4) for each network zone, taken from the dispatch model. In the presence of transmission restrictions, stochastic models have a better awareness of the location where spare capacity is required to deal with forecast uncertainty. The deterministic models are not aware of the spatial correlation of wind forecast errors, which leads to congested situations where neighbouring zones have different LMPs more frequently than with stochastic scheduling. Consequently, the average LMP differences in Table 4.1 are higher in the deterministic cases. Most cases of congestion appear in Scotland (Z1-Z5), while some also appear in the greater London area and in central England. However, these are less frequent so the average LMP differences are lower and we exclude them from the table. The cases shown in Table 4.1 include the stochastic case and one selected deterministic case for 3-hour and 24-hour scheduling. As deterministic cases we chose the cost-optimal

Case	Z1–Z2	Z1–Z4	Z2–Z4	Z2–Z5
3hStoch	130.00	156.44	26.46	26.46
3hDet-15	132.57	395.51	262.93	262.93
24hStoch	117.20	132.67	15.47	15.47
24hDet-30	127.31	146.94	19.63	19.63

Table 4.1: LMP differences between selected zones, averaged over the two-year planning horizon. The LMP values show the average saving per day in $ that can be expected from increasing the transmission capacity between the zones by 1 MW. The shown cases are: stochastic 3-hour scheduling, deterministic 3-hour scheduling with variable R&R margin r=15% of forecast wind (= 2.2GW margin on av.), stochastic 24-hour scheduling, and deterministic 24-hour scheduling with variable R&R margin of r=30% of forecast wind (= 3GW margin on av.).

24-hour case which incurs no load shedding, and a slightly suboptimal 3-hour case with some load shedding. When load shedding occurs, it drives the LMPs up significantly, resulting in very large LMP differences between zones.

Pump Storage and Congestion Cost. We explore the cost of various model alterations concerning the transmission network and pump storage schemes, by performing evaluations with 3-hour deterministic, stochastic and perfect foresight scheduling. The results are summarised in Figure 4.13. We show the cost-optimal deterministic strategy, i.e. the case for which the evaluation showed a posteriori that it had the best variable R&R margin. The following cases have been evaluated for this study:

Z1: This case is motivated by the observation that the system may benefit from additional storage capacity in the north of Scotland, in zone Z1 (cf. previous section). We explore the economic impact of installing two additional pump storage schemes (reservoirs and plants) identical to the existing Foyers scheme which has 300MW pump-turbine capability and 6.3GWh storage capacity.

Z4: This is motivated by the observation that the Cruachan storage scheme in zone Z4 has small pump and turbine capabilities in comparison to its storage capacity. We explore the impact of doubling the number of installed machines, resulting in an increase in pump-turbine capability from 440MW to 880MW.

NoN: This case is included to demonstrate the average cost of network congestion over the two-year evaluation period. We remove all transmission restrictions

from the system and solve a so-called coppper-plate model in which the whole system is represented as a single node.

NoS: In this case we explore the economic value of the pump storage capabilities that are already available. We remove them from the system and re-run the evaluation.

All cases are compared to a 'norm' case, which corresponds to the case shown on the result graph in Figure 4.11. Taking the stochastic results as the base, the gap between stochastic and perfect foresight solutions is between 0.35% and 0.37% ($350k to $370k daily) in all considered cases. The gap between the deterministic and stochastic solutions is 0.11%, 0.08%, 0.06%, 0.08% and 0.14% in the Norm, Z4, Z1, NoN and NoS cases, respectively (left to right in Figure 4.13). The additional storage capabilities in the Z4 and Z1 cases reduce the gap between stochastic and deterministic planning from the Norm case, while removing storage capacity in the NoS case increases the gap. Storage provides a way of compensating for wind forecast uncertainty, so it reduces the advantage of stochastic scheduling over deterministic scheduling.

With the best implementable (stochastic) policy, the cost savings achieved by storage expansion Z4 in comparison to the Norm case is 0.05% ($50k), while storage expansion Z1 gives a 0.08% ($80k) improvement. Removing network congestion (NoN) has a value of 0.05% ($50k), while removing storage capabilities entirely (NoS) leads to a major cost increase of 0.6% ($600k). The different cost in the studied cases can be explained with the system's ability to use more wind and commit less thermal generation: in the Norm case 0.36% of available wind power are curtailed, while in the Z1 and NoN cases only 0.05% are curtailed, and in the NoS case 0.7% need to be curtailed. In the Z1 and NoN cases the total capacity of committed thermal generators is on average 0.1% lower than in the Norm case, while in the NoS case it is 2.5% higher. Pump storages provide a major share of sytem-wide reserve and response, and without them more thermal generators must be switched on and kept off their upper limits to provide reserve and response. The Z4 case does not differ much from the Norm case in terms of wind curtailment, but has 0.06% lower thermal commitment.

In addition to the above cases, we calculate the operational cost in a hypothetical perfect foresight case where infinite lossless storage is available at any node of the

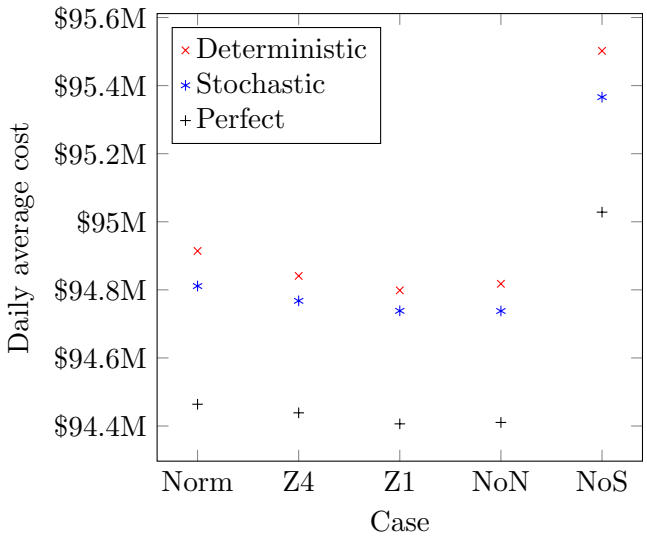


Figure 4.13: A comparison of operational costs in various cases. The graph shows the average daily cost of 3-hour scheduling with the best (a posteriori) deterministic strategy, stochastic strategy, and perfect foresight. In order from left to right the cases are: the 'normal' reference case (Norm, same as Figure 4.11), a case with doubled pump-turbine capability but unchanged storage capacity in zone 4 (Z4), a case with two additional pump storage schemes identical to Foyers in zone 1 (Z1) (increased pump, discharge and storage capacity), a case without transmission network restrictions (NoN) and a case without any pump storages (NoS).

network. In that case all wind and demand variability will be met from storage, and generation levels will be constant. No startup cost will occur, since the generators would be committed according to their merit order and kept on continuously. The cost of satisfying demand in this way is a lower bound to the operational cost that can be achieved by an implementable strategy in a real system, so in particular it provides a (loose) bound on what can be achieved by the optimization. In our GB test system it amounts to a daily average cost of $92.06M.

5

Conclusions

This section summarises the main results of our stochastic unit commitment study and explains our conclusions. Section 5.1 provides an overview of the contents of this study, and Section 5.2 has our conclusions.

5.1 Summary of the Contents

In Chapter 2 we study modelling and solution techniques which are required to solve realistic implementations of stochastic UC problems efficiently. We review two-stage and multi-stage recourse problems, show how the non-anticipativity property can be imposed through constraints, and how these constraints can be relaxed to derive scenario decomposition algorithms. Besides Progressive Hedging, which remains a heuristic when applied to mixed-integer problems, Lagrangian relaxation and Dantzig-Wolfe decomposition can be used for scenario decomposition. We show how LR is used for lower bounding and decomposition, and what solution techniques exist for the dual problem. As a counterpart to this dual viewpoint, we present the primal DW decomposition approach: we introduce Dantzig and Wolfe's decomposition principle and explain generalisations for the mixed-integer case. For a given dual solution, LR and DW decomposition provide the same lower bound on the optimal objective value, and if the Lagrangian dual problem is solved by a cutting plane algorithm, then DW decomposition and LR are equivalent to one another in that the cutting plane problem is the LP dual of the DW master problem. Proximal bundle methods are presented as a way of stabilising the dual in this algorithm, and Branch & Price is briefly explained as a method to close occurring duality gaps. We close this chapter with a brief review

of deterministic and stochastic unit commitment formulations and explain likely causes of redundancy of non-anticipativity constraints.

Chapter 3 develops an efficient solver for multi-stage and two-stage stochastic UC problems of realistic size and detail. Based on the theory explained in Chapter 2, we derive a basic column generation framework for scenario decomposition of stochastic problems that have an explicit non-anticipativity formulation via constraints. We show how valid lower bounds for the optimal stochastic solution can be derived from lower bounds of scenario specific subproblems if these cannot be solved to optimality. To address the well-known dual instability issues that ColGen methods suffer from, the master problem in our approach uses a proximal bundle term. The proximity center is hot-started from a multiplier estimate that we obtain from an LP relaxation. Since the non-anticipativity constraints of the LP relaxation are equally as redundant as those of the original integer SUC problem, we augment the relaxation with a quadratic regularisation term that is similar to the proximal bundle term. This permits us to control the size of the obtained dual solution in the presence of dual degeneracy. We argue that quadratic regularisation improves the solver performance if interior point methods are used to solve the relaxation, and show that controlling the size of the initial dual solution is beneficial for stability of the ColGen method with certain sets of columns. Besides dual hot starts, we employ a novel MIP heuristic to derive near-optimal integer SUC solutions. These are used to provide a primal hot-start to the master problem. The hot-starts accelerate our method significantly and allow for a major reduction of runtime in comparison to CPLEX's out-of-the-box Branch & Bound solver. All numerical tests are performed with our UC model that is based on the British power system.

In Chapter 4 we describe a long term evaluation of hydro-thermal scheduling under wind uncertainty, using deterministic and stochastic UC approaches. We first describe and explain our GB UC model, which includes transmission-connected conventional and renewable power plant, a sophisticated model of pump storage, an aggregated model of the transmission system, and a model for flexible system-wide provision of frequency response and reserve. This model is an expected 2020 version of the GB system with an overall wind penetration of 30% in terms of installed capacity. We perform a two-year evaluation of deterministic scheduling with fixed reserve margins and a central fore-

cast, stochastic scheduling with scenario trees based on the same central forecast, and perfect foresight scheduling for reference. We compare these in different operational contexts, i.e. day-ahead and intraday scheduling with variable time lengths between re-scheduling slow conventional generation units. Wind forecasts in this evaluation were synthesised so as to achieve specific error statistics. We describe the employed forecast error scenario generation approach which is based on multi-variate ARMA series, and a scenario reduction and tree construction method based on Kantorovich distances between probability measures. These are a variation of the scenario generation techniques used in the WILMAR study, where we adapt the scenario spread so that it depends on the forecast wind level. Finally, we explain the rolling horizon methodology and the dispatch approach used to evaluate the schedules against the actual wind, and demonstrate the potential cost savings through stochastic scheduling. The detailed modelling approach with network restrictions and a sophisticated pump storage model is motivated a posteriori by demonstrating the effects of wind variability and uncertainty on the optimal storage operation under network congestion. We demonstrate the economic value of the existing storage schemes under deterministic and stochastic scheduling, explore two alternatives of expanding the storage capabilities and quantify the impact of the expansion on operational costs under both scheduling approaches. Additionally, we explore the average cost of congestion in the transmission network. To sum it up, the two major research topics addressed in this study are

- To develop an efficient scenario decomposition methodology for stochastic unit commitment problems which scales well in the number of scenarios.
- To quantify the added value of stochastic scheduling over deterministic scheduling under wind uncertainty, and to characterise the different schedules and explain their impact on the power system.

Our main findings on these two topics are summarised in the two following sections. Section 5.2.1 draws conclusions on the former topic, while Section 5.2.2 is concerned with the latter.

5.2 Main Findings & Further Research

5.2.1 Efficient Scenario Decomposition

Our mixed integer ColGen algorithm for two-stage and multi-stage stochastic UC problems is based on a DW decomposition of the underlying scenario set. In theory, DW decomposition readily applies to scenario decomposition, but a series of additional issues arise in practical applications. We stabilise the scenario decomposition algorithm with a dual proximal bundle term and estimate dual hot-starts from a perturbed LP relaxation. We also derive a fast, novel, MIP-based schedule combination heuristic which is used to construct primal optimal solutions. Our tests suggest that knowing this solution at the start of the ColGen process also enables quick convergence of lower bounds. The scenario decomposition method can solve SUC problems to optimality if optimal solutions of the deterministic subproblems can be obtained. Where this is not the case, a valid lower bound can be derived from the lower bounds of the subproblems. In all test runs the duality gaps vanish or are well below the 0.1% optimality tolerance. Since optimal primal solutions can be constructed by a heuristic, no branching is necessary.

Our implementation is tested on two-stage and multi-stage instances of our GB power system model. The ColGen method terminates in just one iteration. For small problems the solution times of the decomposition method are similar to those of CPLEX's Branch & Bound solver when applied to the extensive formulation. However, on larger instances the decomposition takes significantly less time. Most of the work is done in the initialisation procedure, to find an optimal primal solution, and to estimate non-anticipativity multipliers for cases with non-zero EVPI. On 'easy' cases with negligible EVPI, the decomposition performs very well even without a dual hot start. Setting the initial duals to zero is sufficient to encourage the subproblems to generate all relevant schedules required by the SC heuristic to construct an optimal solution, and is also sufficient to prove optimality in the next ColGen pass. In order to solve harder cases with non-zero EVPI, a good dual estimate is required for the following two reasons:

1. To populate the SC heuristic with a good subset of schedules, so that its solution space is as small as possible but contains an optimal solution.

2. To guide the cutting plane problem dRMP(ϵ) by stabilising it around an accurate stability center, so that it provides good dual solutions.

The stabilised LP relaxation LPR(μ) can be solved to obtain such multipliers. Our tests showed that interior point methods benefit from the convex quadratic stabilisation term that we include in this problem. Studying the dual solutions at different stabilisation levels revealed that there is a continuum of dual optimal solutions with vastly different magnitudes, and that the ColGen method gains in stability if we control the size of the initial multipliers. In terms of elapsed times this strategy scales acceptably in the number of scenarios, because the linear algebra underlying interior point solvers parallelises well. However, in terms of CPU time CPLEX's barrier solver does not scale well on LPR(μ) if the number of scenarios is increased. An alternative way to obtain approximate dual solutions is by applying scenario decomposition to the LP relaxation. However, achieving high accuracy with the relaxed ColGen method takes very many iterations, and using dual solutions of lower accuracy has an adverse effect on the quality of the lower bounds obtained in the integer decomposition. Overall, the fastest way of achieving the required tolerance in the integer decomposition is by solving LPR(μ) with an interior point method. Alternative ways to solve LPR(μ) quickly *and* accurately are a potential direction for future research.

A central element for the success of our method is the SC heuristic. It can be started from a feasible over-committment solution or a solution obtained with other cheap heuristics, and it scales very well on the problems we solved for this study. However, since it is a MIP itself, it will fail to do so eventually if the problem size is increased sufficiently. In that case cheaper heuristics will be required, and this is another possible direction of future research.

The strategy adopted in this study aims to reduce the work required to solve SUC problems: we seek accurate primal and dual hot starts to minimise the number of ColGen iterations. This results in a favourable reduction of CPU time, which is suitable for environments with moderate computational power. Alternative strategies may seek to reduce elapsed times by increasing parallelism: subproblems can be solved in parallel, and each subproblem can again be solved by parallel Branch & Bound. The resulting ColGen iterations are cheaper, and it may be worthwhile to reduce the accuracy of

the initial duals at the expense of doing more iterations. Less accurate duals can then be obtained via scenario decomposition of the initial LP relaxation, which can again be parallelised. This may be a more suitable strategy if large computer clusters are available, or even if less accurate lower bounds are required.

5.2.2 Stochastic vs Deterministic Evaluation

We have performed a two-year rolling horizon evaluation of stochastic and deterministic unit commitment approaches under wind uncertainty, with periods of varying length between times when the schedules of slow conventional generators can be revised. For the evaluation we use a central scheduling and dispatch model based on the British power system under National Grid's Gone Green scenario for 2020, including a pump storage model, transmission restrictions between network areas and a model of system-wide reserve and response provision. The focus of our study is on the performance comparison of deterministic and stochastic generator scheduling at different time scales. We quantifiy the monetary value of stochastic scheduling models over deterministic ones under a central scheduling hypothesis, and pinpoint other advantages of stochastic schedules.

Some of the effects of wind variability and uncertainty discussed in this study were only possible to observe due to the amount of technical detail included in the model. We observed that wind uncertainty can have major effects on the way in which pump storages are operated if both lie on the same side of a transmission bottleneck which separates them from a demand centre. Additionally, we have seen that the spatial distribution of reserve and response has an impact on whether uncertain wind events can be dealt with cost effectively, since wind forecast errors are spatially correlated. In Section 4.4.2 we showed that modelling the interaction of transmission capabilities, spatially correlated wind uncertainty and flexible generation and storage technologies makes a difference to the overall operational cost. Hence the evaluation results shown here apply to a rather specific model and context, and may turn out to be quite different in other settings, e.g. if a less detailed model is used, a different amount of storage is available, or the transmission network is less congested.

There are significant differences in costs between operating systems that allow the major generators to be rescheduled every three hours, six hours or only once per day.

The best (stochastic) cases of 3-hour, 6-hour and 24-hour scheduling have $350k, $650k and $1.3M higher average daily operational cost than the perfect foresight solution. Similarly, the superiority of stochastic over deterministic scheduling grows with the amount of uncertainty in the relevant wind forecasts: the gaps are $100k per day in the 3-hour and 6-hour cases and $300k per day in the 24h case.

Stochastic models result in minimum operational costs without having to tune reserve margins in advance. In all cases there is a gap between the lowest deterministic cost and the cost of stochastic scheduling. This is despite the fact that we compare their performance with the *best* (a posteriori) setting for deterministic reserve and response which is not known a priori and can be different from one year to another. The superiority of stochastic scheduling grows with the amount of uncertainty in the relevant wind forecasts, but is reduced if additional pump storage capacity is installed.

Realising the full potential of cost improvement through stochastic scheduling requires some care in constructing the tree of input scenarios. Minimal operational cost is achieved through the right balance of committed spare capacity and the use of costly recourse actions. In deterministic problems this balance is determined by the strategy for setting reserve margins, while in stochastic models the scenario trees are the determining factor. Deterministic reserve margins were set in a flexible way, so that there are larger amounts of reserve at times of high wind forecasts and smaller amounts at times of low forecasts. This is consistent with the assumption that the wind forecast error is proportional to the amount of forecast wind. If the stochastic model is driven with scenario trees that are generated from the same time series model at all times of the year, irrespective of the amount of forecast wind, then the deterministic model is given an unfair advantage by allowing it to schedule variable reserve margins depending on the forecast wind. To rectify this, a scenario generator is required which allows the variability of the scenarios to depend on the wind level. In this study we follow a scenario generation and tree construction approach which was developed for the WILMAR study, in which the scenarios are generated from the same ARMA model in all wind forecast situations. On average the stochastic model then schedules significantly more generation capacity than the deterministic model, which increases the overall cost unnecessarily. Therefore the width of the scenario trees in our tests is scaled down at times of low and medium wind forecasts and kept the same at times of

high wind forecasts. Our results show that this approach with scenario trees of adaptive width outperforms deterministic scheduling. There may be potential for further savings through an improved forecast level dependent scenario tree generation methodology. However, developing a new scenario generation method is outside the scope of this study. Simple ways of obtaining forecast level dependent errors are by

- Fitting different time series models for errors in forecasts of different levels. This will require much larger amounts of historic data.
- Logit-transforming wind power and forecasts before calculating the error [76]. The fitting and scenario generation processes can be done with the transformed data before the resulting scenarios are translated back to wind power. However, a disadvantage of this is that the logit transform counteracts the effect that least squares fitting penalises large errors more than small errors.

These can be starting points for further research into wind power scenario generation methodology.

Penalties for keeping too little response and reserve are modelled, and they account for \$50k to \$350k (≈0.05% to 0.35%) of the total daily cost, depending on how well a scheduling method performs. The main cost drivers, however, are generation costs and the recourse cost for shed load and OCGT usage. An analysis of these cost drivers in the different deterministic scheduling strategies showed that the minimum operational cost is obtained in a situation where the cost increase for committing additional capacity is balanced by the resulting cost decrease in recourse costs, i.e. OCGT cost and reserve and response penalties. Due to the high penalty for lost load, the minimum is typically obtained by strategies which manage to avoid load shedding entirely, or only encounter very low levels of load shedding on average. The stochastic models are more successful at finding the right balance of cost drivers than their deterministic counterparts: the resulting total costs are lower than in the best deterministic cases. Additionally, stochastic models tune committed capacity levels internally, and the resulting response levels under wind uncertainty are similar to those achieved with a perfect foresight model.

In the presence of transmission restrictions, stochastic models have a better awareness of spatial wind issues and allocate reserve where it is necessary to deal with the

wind uncertainty. In the deterministic scheduling model, increasing the reserve margin did not always lead to a reduction of lost load. Additional reserve capacity is scheduled based on where it is cheapest rather than based on where it may be required to buffer wind forecast uncertainty. The stochastic model, on the other hand, uses scenarios drawn from a time series model that is aware of the spatial correlation of wind forecast errors to decide on where to commit spare capacity. In cases where there is no load shedding, lower LMPs indicate that the stochastic scheduling models typically lead to less network congestion. Congestion is measured by differences in LMPs, and it is shown that these are less with stochastic scheduling models. An additional evaluation in which network restrictions were removed confirmed that the average cost of network congestion is \$40k per day lower with stochastic scheduling than with deterministic scheduling.

Another interesting question is whether further cost savings can be achieved by including more scenarios in the stochastic scheduling problem. The results shown in the previous chapter are all obtained with twelve wind scenarios to keep the runtime for the evaluation within manageable limits. We performed the evaluations with 20 scenarios as well, but the cost is not sufficiently different from the twelve scenario evaluation to justify showing the results here. However, we discussed above that it may be possible to achieve further cost savings by using better forecast level dependent scenario generation techniques, and it is unclear if a larger number of scenarios, or even a *forecast level dependent* number of scenarios would have a beneficial effect on the cost in that context. In future research projects these alternatives could be explored in conjunction with one another.

We have seen that stochastic UC methodology allows for operational cost savings in a setting where thermal schedules are made under significant wind uncertainty. The savings are a small percentage of the overall cost, but due to the vast cost they are still a noteworthy amount when expressed in total numbers. We have seen that a key issue in making stochastic models profitable and worthwhile is to generate good input scenarios. A moderate number of good scenarios can already lead to notable cost savings, and with the scenario decomposition methodology these stochastic models can be solved in a reasonable timeframe.

A

UC Model with Approximate Recourse

The solution method for two-stage stochastic UC problems described in Chapter 3 relies on a cheap heuristic to find approximate solutions for the problem. These approximate solutions are passed to the SC heuristic as an initial solution which it can improve upon. A simple way of finding such a solution is by solving a deterministic problem for a central or low wind scenario and then solving scenario-specific dispatch problems with fixed schedules to find the recourse action. We find that this approach can be improved notably if the deterministic problem that is used to find the central schedule is augmented with some additional variables and constraints which inform it about the expected cost of potential repercussions under the different wind scenarios. In the following, we describe a model which is deterministic at the core, but knows about the potential wind outcomes P^{win2}_{wts} for every wind farm $w \in \mathcal{W}$ under each scenario $s \in \mathcal{S}$. We call this an approximate recourse model. It has additional power output variables p^{gen2}_{gts} for every scenario. These make the model much larger than a simple deterministic one, especially for large numbers of scenarios. However, we find that if these variables are used carefully in the model, then the computational cost of solving it is comparable to a deterministic model. The model distinguishes between fast start OCGT generators (set $\mathcal{F}$) which are part of the recourse action, and slow conventional generators. In the two-stage stochastic model described in Chapter 4, pump storages are also part of the recourse action. However, in the approximate recourse model this is ignored, and pump storages are part of the initial decision. An algebraic description of the approximate recourse model is given below. This is largely a deterministic version of the British model described in Chapter 4, but with the following modifications to

make it aware of the expected cost of dealing with the different wind scenarios:

- In the objective (A.1) we include an additional term

$$\sum_{s\in\mathcal{S}} P_s^{rob}\left[\sum_{t=1}^{T}\sum_{g\in\mathcal{F}}\left(D^t C_g^m p_{gts}^{gen2}\right)\right]$$

  to account for the expected generation cost of fast start OCGT units. This could be extended to include the expected cost for all generation units rather than having a deterministic cost for large conventional units under a central wind scenario and an expected cost for OCGT units under the different wind scenarios. However, that increases the computational effort significantly, so we settle for this model instead.

- We include additional cuts (A.2) to approximate the load balance equations under every scenario. This is done in a global fashion rather than by transmission areas. The equations require that enough power p_{gts}^{gen2} is available in every scenario to satisfy the global residual demand which remains after subtracting the available wind power under that scenario.

- Generator bounds (A.3) and (A.4) link the additional power output variables p_{gts}^{gen2} to the central commitment decisions: large conventional generators can only contribute to the approximate load balance equations under every scenario if they have been committed to run. These bounds do not apply to fast start generators $g \in \mathcal{F}$ which can be committed as part of the recourse action.

The resulting model is a modified deterministic UC model with an extended continuous part. Its size grows in the number of scenarios, however, in the experiments we conducted, the time required to solve this model via CPLEX was very similar to the time requirement for simple deterministic models.

Sets

$\mathcal{D}$: set of transmission boundaries in the network

$\mathcal{F}$: set of fast start units, $\mathcal{F} \subset \mathcal{G}$. Slow units are in $\mathcal{G} \setminus \mathcal{F}$.

$\mathcal{G}$: set of generation units, $\mathcal{G}_n$ is the set of generators at node $n \in \mathcal{N}$

$\mathcal{L}$: set of transmission lines

$\mathcal{N}$: set of network nodes (transmission areas)

$\mathcal{P}$: set of pump storage plants, $\mathcal{P}_n$ is the subset at node $n \in \mathcal{N}$

$\mathcal{S}$: set of wind power scenarios

$\mathcal{W}$: set of wind farms, $\mathcal{W}_n$ is the subset at node $n \in \mathcal{N}$

Parameters

Ψ: minimum proportion of response to be met by part-loaded generators

B_{ld}: line-boundary adjacency matrix. 1 if line l crosses boundary d in one direction, -1 if it crosses in the other direction, 0 otherwise

$C(r^{tot})$: PWL penalty function for keeping too little response r^{tot}

$\hat{C}(\hat{r}^{tot})$: PWL penalty function for keeping too little reserve plus response $\hat{r}^{tot}$

C_g^{nl}: no load cost of generator g [\$/h]

$C_q^{H_2O}$: end-of-day water value in the reservoir of pump storage plant q [\$/MWh]

C_g^{m}: marginal cost of generator g [\$/MWh]

C_g^{st}: startup cost of generator g [\$]

C^{voll}: value of lost load [\$/MWh]

D^t: time granularity of the model [h]

D^{res}: time for which response must be served [h], with $D^{res} < D^t$

E_q: pump-generator cycle efficiency at storage $q \in \mathcal{P}$ [proportion]

H_q^{max}: storage capacity at plant $q \in \mathcal{P}$ in MWh of dischargeable energy

N_q^{pum}: number of (identical) pumps in pump storage plant $q \in \mathcal{P}$

$N_l^{st,end}$: start (end) nodes of line l

P_q^{cap}: capacity of a single pump in pump storage plant $q \in \mathcal{P}$ [MWh]

P_{nt}^{dem}: real power demand at node n in period t [MW]

$P_{g,q}^{min,max}$: min (max) generation limit of generator $g \in \mathcal{G}$ (storage $q \in \mathcal{P}$) [MW]

$\bar{P}_{l,d}^{flo}$: maximum power transmission on line l / across boundary d [MW]

P_s^{rop}: probability of scenario s

$P_g^{ru,rd}$: operating ramp up (down) limits of generator g [MW/D^t]

$P_g^{su,sd}$: startup (shutdown) ramp limits of generator g [MW/D^t]

P_{wt}^{win}: expected wind power available from wind farm w in period t [MW]

P_{wts}^{win2}: wind power available from wind farm w in period t, scenario s [MW]

R_g^{max}: maximum response available from generator g [MW]

T: last time period of the planning horizon

T_g^{nt}: startup notification time of generator g [h]

$t_b^{st,end}$: start (end) periods of scenario bundle b

$T_g^{u,d}$: minimum uptime (downtime) of generator g [h]

Variables

$\alpha_{gt} \in \{0,1\}$: 1 if thermal unit g is on in period t, and 0 if it is off

$\gamma_{gt} \in \{0,1\}$: 1 if thermal unit g is started up in period t, and 0 otherwise

$\eta_{gt} \in [0,1]$: 1 if thermal unit g is shut down in period t, and 0 otherwise

$\delta_{qit} \in \{0,1\}$: 1 if pump i of storage q is pumping in period t, 0 otherwise

$\zeta_{qt} \in \{0,1\}$: 1 if storage q is generating in period t, and 0 otherwise

$h_{qt} \in [0, H_q^{max}]$: level of storage q *after* period t (dischargeable MWh)

$p_{qt}^{dis} \in [0, P_q^{max}]$: real power discharged from storage q in period t [MW]

$p_{lt}^{flo} \in [-\bar{P}_l, \bar{P}_l]$: real power flow on line l in period t [MW]

$p_{gt}^{gen} \in [0, P_g^{max}]$: real power output of generator g in period t [MW]

$p_{gts}^{gen2} \in [0, P_g^{max}]$: real power output of generator g in period t, scenario s [MW]

$p_{qt}^{pum} \geq 0$: real power pumped into storage q in period t [MW]

$p_{nt}^{shed} \geq 0$: load shed at node n in period t [MW]

$r_{gt}^{gen} \in [0, R_g^{max}]$: response provided by generator g in period t [MW]

$r_{qt}^{pum} \geq 0$: response provided by pump storage q in period t [MW]

$\hat{r}_{qt}^{pum} \geq 0$: reserve plus response provided by storage q in period t [MW]

$r_t^{tot} \geq 0$: total available response in period t [MW]

$\hat{r}_t^{tot} \geq 0$: total available reserve plus response in period t [MW]

$u_{wt}^{win} \in [0, P_{wts}^{win}]$: used wind power from farm w in period t [MW]

Using the notation described above, the deterministic base model with approximate recourse reads as follows:

$$\begin{aligned}
\min \quad & \sum_{t=1}^{T} \sum_{g \in \mathcal{G}} \left(C_g^{st} \gamma_{gt} + D^t C_g^{nl} \alpha_{gt} + D^t C_g^{m} p_{gt} \right) + \sum_{q \in \mathcal{P}} C_q^{H_2O} \left(h_{q0} - h_{qT} \right) \\
+ \quad & \sum_{t=1}^{T} \left(\sum_{n \in \mathcal{N}} D^t C^{voll} p_{nt}^{shed} + C(r_t^{tot}) + \hat{C}(\hat{r}_t^{tot}) \right) \\
+ \quad & \sum_{s \in \mathcal{S}} P_s^{rob} \left[\sum_{t=1}^{T} \sum_{g \in \mathcal{F}} \left(D^t C_g^{m} p_{gts}^{gen2} \right) \right]
\end{aligned} \tag{A.1}$$

subject to these additional constraints for the recourse approximation:

- Second stage global load balance approximation cut for all $t = 1, \ldots, T,\ s \in \mathcal{S}$

$$\sum_{g \in \mathcal{G}} p_{gts}^{gen2} \geq \sum_{n \in \mathcal{N}} \left(P_{nt}^{dem} - p_{nt}^{shed} \right) - \sum_{w \in \mathcal{W}} P_{wts}^{win2} \tag{A.2}$$

- Second stage generator bounds for all $g \in \mathcal{G} \setminus \mathcal{F},\ s \in \mathcal{S},\ t = 1, \ldots, T$

$$\begin{aligned} p_{gts}^{gen2} &\geq P_g^{min} \alpha_{gt} && \text{(A.3)} \\ p_{gts}^{gen2} + r_{gt}^{gen} &\leq P_g^{max} \alpha_{gt} && \text{(A.4)} \end{aligned}$$

All following constraints describe a simple deterministic model with a central wind scenario P_{wt}^{win} for every wind farm $w \in \mathcal{W}$ and time step $t = 1, \ldots, T$.

- Load balance equation for all $n \in \mathcal{N},\ t = 1, \ldots, T$

$$\begin{aligned} 0 \;=\; & \sum_{g \in \mathcal{G}_n} p_{gt} + \sum_{w \in \mathcal{W}_n} u_{wt}^{win} + \sum_{l \in \mathcal{L}: N_l^{end} = n} p_{lt}^{flo} + \sum_{q \in \mathcal{P}_n} p_{qt}^{dis} + p_{nt}^{shed} \\ - \;& P_{nt}^{dem} - \sum_{l \in \mathcal{L}: N_l^{st} = n} p_{lt}^{flo} - \sum_{q \in \mathcal{P}_n} p_{qt}^{pum}. \end{aligned} \tag{A.5}$$

- Reserve and response definitions for all $t = 1, \ldots, T$

$$\begin{aligned} \sum_{g \in \mathcal{G}} \left(\alpha_{gt} P_g^{max} - p_{gt} \right) + \sum_{q \in \mathcal{P}} \hat{r}_{qt}^{pum} &= \hat{r}_t^{tot} && \text{(A.6)} \\ \sum_{g \in \mathcal{G}} r_{gt}^{gen} + \sum_{q \in \mathcal{P}} r_{qt}^{pum} &= r_t^{tot} && \text{(A.7)} \\ \sum_{g \in \mathcal{G}} r_{gt}^{gen} &\geq \Psi r_t^{tot}. && \text{(A.8)} \end{aligned}$$

- Transmission boundary limits for all $t = 1, \ldots, T,\ d \in \mathcal{D}$

$$-\bar{P}_d \leq \sum_{l \in \mathcal{L}} B_{ld} p_{lt}^{flo} \leq \bar{P}_d. \tag{A.9}$$

- Generator bounds for all $t = 1, \ldots, T,\ g \in \mathcal{G}$

$$p_{gt}^{gen} \geq P_g^{min} \alpha_{gt} \tag{A.10}$$

$$p_{gt}^{gen} + r_{gt}^{gen} \quad \leq \quad P_g^{max}\alpha_{gt}. \tag{A.11}$$

- Ramp rate constraints for all $g \in \mathcal{G},\ t = 1, \ldots, T$

$$p_{gt}^{gen} - p_{g(t-1)}^{gen} \quad \leq \quad P_g^{ru}\alpha_{g(t-1)} + P_g^{su}\gamma_{gt} \tag{A.12}$$

$$p_{g(t-1)}^{gen} - p_{gt}^{gen} \quad \leq \quad P_g^{rd}\alpha_{gt} + P_g^{sd}\eta_{gt}. \tag{A.13}$$

- Switching constraints for all $t = 1, \ldots, T,\ g \in \mathcal{G}$

$$\alpha_{gt} - \alpha_{g(t-1)} \quad = \quad \gamma_{gt} - \eta_{gt} \tag{A.14}$$

$$1 \quad \geq \quad \gamma_{gt} + \eta_{gt}. \tag{A.15}$$

- Minimum up- and downtime constraints for all $g \in \mathcal{G},\ t = 1, \ldots, T$

$$\sum_{i=t-T_g^u+1}^{t} \gamma_{gi} \quad \leq \quad \alpha_{gt} \tag{A.16}$$

$$\sum_{i=t-T_g^d+1}^{t} \eta_{gi} \quad \leq \quad 1 - \alpha_{gt}. \tag{A.17}$$

- Pump storage operation constraints for all $q \in \mathcal{P},\ t = 1, \ldots, T$

$$\delta_{q1t} \quad \leq 1 - \zeta_{qt} \tag{A.18}$$

$$\delta_{q(i+1)t} \quad \leq \delta_{qit} \quad \forall i = 1, \ldots, N_q^{pum} - 1 \tag{A.19}$$

$$p_{qt}^{pum} \quad = \sum_{i=1}^{N_q^{pum}} \delta_{qit} P_q^{cap} \tag{A.20}$$

$$\zeta_{qt} P_q^{min} \leq \quad p_{qt}^{dis} \quad \leq \zeta_{qt} P_q^{max}. \tag{A.21}$$

- Pump storage reserve constraints for all $q \in \mathcal{P},\ t = 1, \ldots, T$

$$\hat{r}_{qt}^{pum} + p_{qt}^{dis} \quad \leq \quad P_q^{max} + p_{qt}^{pum} \tag{A.22}$$

$$D^t \hat{r}_{qt}^{pum} + D^t p_{qt}^{dis} \quad \leq \quad h_{q(t-1)} + D^t p_{qt}^{pum} \tag{A.23}$$

$$r_{qt}^{pum} + p_{qt}^{dis} \quad \leq \quad p_{qt}^{pum} + P_q^{max}\left(1 - \delta_{q1t}\right) \tag{A.24}$$

$$D^{res} r_{qt}^{pum} + D^t p_{qt}^{dis} \quad \leq \quad h_{q(t-1)} + D^t P_q^{max} \delta_{q1t}. \tag{A.25}$$

- Pump storage reservoir constraints for all $q \in \mathcal{P},\ t = 1, \ldots, T$

$$h_{qt} = h_{q(t-1)} + D^t E_q p_{qt}^{pum} - D^t p_{qt}^{dis}. \tag{A.26}$$

References

[1] T. Schulze, A. Grothey, and K. I. M. McKinnon, "A stabilised scenario decomposition algorithm applied to stochastic unit commitment problems," Tech. Rep. ERGO 15-009, The University of Edinburgh, School of Mathematics, 2015.

[2] T. Schulze and K. I. M. McKinnon, "The value of stochastic programming in day-ahead and intraday generation unit commitment," Tech. Rep. ERGO 15-010, The University of Edinburgh, School of Mathematics, 2015.

[3] C. C. Carøe and R. Schultz, "A two-stage stochastic program for unit commitment under uncertainty in a hydro-thermal power system," in *Preprint SC 98-11, Konrad-Zuse-Zentrum für Informationstechnik*, pp. 98–113, 1998.

[4] J. Goez, J. Luedtke, D. Rajan, and J. Kalagnanam, "Stochastic unit commitment problem," Tech. Rep. RC24713 (W0812-119), IBM Research Division, 2008.

[5] N. Gröwe-Kuska and W. Römisch, *Stochastic Unit Commitment in Hydro-thermal Power Production Planning*, ch. 30, pp. 605–624. Preprints aus dem Institut für Mathematik, Humboldt Universität Berlin, 2002.

[6] M. P. Nowak and W. Römisch, "Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty," *Annals of Operations Research*, vol. 100, no. 1-4, pp. 251–272, 2000.

[7] S. Takriti, J. R. Birge, and E. Long, "A stochastic model for the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 11, no. 3, pp. 1497–1508, 1996.

[8] M. Carrión and J. M. Arroyo, "A computationally efficient mixed-integer formulation for the thermal unit commitment problem," *IEEE Transactions on Power Systems*, vol. 21, pp. 1371–1378, 2006.

[9] J. Ostrowski, M. F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 39–46, 2012.

[10] E. Krall, M. Higgins, and R. P. O'Neill, "RTO unit commitment test system," tech. rep., Federal Energy Regulatory Commission (FERC), 2012.

[11] D. Rajan and S. Takriti, "Minimum up/down polytopes of the unit commitment problem with start-up costs," Tech. Rep. RC23628 (W0506-050), IBM Research Division, 2005.

[12] R. Jiang, Y. Guan, and J.-P. Watson, "Cutting planes for the multi-stage stochastic unit commitment problem," Tech. Rep. SAND2012-9093J, Sandia National Laboratories, 2012.

[13] S. J. Wang, S. M. Shahidehpour, D. S. Kirschen, S. Mokhtari, and G. D. Irisarri, "Short-term generation scheduling with transmission and environmental constraints using an augmented Lagrangian relaxation," *IEEE Transactions on Power Systems*, vol. 10, pp. 1294–1301, 1995.

[14] T. Shiina and J. R. Birge, "Stochastic unit commitment problem," *International Transactions in Operational Research*, vol. 11, pp. 19–32, 2004.

[15] O. Briant, C. Lemaréchal, P. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck, "Comparison of bundle and classical column generation," *Mathematical Programming*, vol. 113, no. 2, pp. 299–344, 2008.

[16] R. T. Rockafellar and R. J. B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of Operations Research*, vol. 16, no. 1, pp. 119–147, 1991.

[17] S. Ryan, R.-B. Wets, D. Woodruff, C. Silva-Monroy, and J.-P. Watson, "Toward scalable, parallel progressive hedging for stochastic unit commitment," in *Power and Energy Society General Meeting (PES), 2013 IEEE*, pp. 1–5, July 2013.

[18] D. Gade, G. Hackebeil, S. M. Ryan, J.-P. Watson, R. J. B. Wets, and D. L. Woodruff, "Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs," tech. rep., Graduate School of Management, UC Davis, 2014.

[19] J. P. Watson and D. Woodruff, "Progressive hedging innovations for a class of stochastic resource allocation problems," *Computational Management Science*, vol. 8, no. 4, pp. 355–370, 2011.

[20] D. Dentcheva and W. Römisch, "Duality gaps in nonconvex stochastic optimization," *Mathematical Programming*, vol. 101, no. 3, pp. 515–535, 2004.

[21] W. Römisch and R. Schultz, "Multistage stochastic integer programs: An introduction," in *Online Optimization of Large Scale Systems*, pp. 581–600, Springer, 2001.

[22] C. C. Carøe and J. Tind, "L-shaped decomposition of two-stage stochastic programs with integer recourse," *Mathematical Programming*, vol. 83, no. 1-3, pp. 451–464, 1998.

[23] Q. P. Zheng, J. Wang, P. M. Pardalos, and Y. Guan, "A decomposition approach to the two-stage stochastic unit commitment problem," *Annals of Operations Research*, vol. 210, pp. 387–410, 2013.

[24] C. Sagastizábal, "Divide to conquer: decomposition methods for energy optimization," *Mathematical Programming*, vol. 134, pp. 187–222, 2012.

[25] P. A. Ruiz, C. R. Philbrick, E. Zak, K. W. Cheung, and P. W. Sauer, "Uncertainty management in the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 24, pp. 642–651, 2009.

[26] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Ij, R. Mukerij, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour, and C. Singh, "The IEEE reliability test system - 1996," *IEEE Transactions on Power Systems*, vol. 14, pp. 1010–1020, 1999.

[27] A. Papavasiliou and S. S. Oren, "A comparative study of stochastic unit commitment and security-constrained unit commitment using high performance computing," in *2013 European Control Conference (ECC)*, pp. 2507–2512, 2013.

[28] E. Constantinescu, V. Zavala, M. Rocklin, S. Lee, and M. Anitescu, "Unit commitment with wind power generation: Integrating wind forecast uncertainty and stochastic programming," Tech. Rep. ANL/MCS-TM-309, Argonne National Laboratory, 2009.

[29] A. Tuohy, P. Meibom, E. Denny, and M. OMalley, "Unit commitment for systems with signicant wind penetration," *IEEE Transactions on Power Systems*, vol. 24, pp. 592–601, 2009.

[30] C. Weber, P. Meibom, R. Barth, and H. Brand, "Wilmar: A stochastic programming tool to analyze the large-scale integration of wind energy," in *Optimization in the Energy Industry*, pp. 437–458, Springer, 2009.

[31] A. Sturt and G. Strbac, "Efficient stochastic scheduling for simulation of wind-integrated power systems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 323 – 334, 2012.

[32] G. Lulli and S. Sen, "A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems," *Management Science*, vol. 50, no. 6, pp. 786–796, 2004.

[33] J. Higle, B. Rayco, and S. Sen, "Stochastic scenario decomposition for multi-stage stochastic programs," *IMA Journal of Management Mathematics*, vol. 21, no. 1, pp. 39–66, 2009.

[34] K. C. Kiwiel, "Proximity control in bundle methods for convex nondifferentiable minimization," *Mathematical Programming*, vol. 46, no. 1, pp. 105–122, 1990.

[35] G. B. Dantzig, "Linear programming under uncertainty," *Management Science*, vol. 1, no. 3-4, pp. 197–206, 1955.

[36] P. Kall and S. W. Wallace, *Stochastic Programming*. Wiley, 1994.

[37] J. L. Higle, "Stochastic programming: Optimization when uncertainty matters," in *Tutorials in Operations Research*, pp. 30–53, INFORMS, 2005.

[38] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, "Scenario reduction in stochastic programming: An approach using probability metrics," *Mathematical Programming*, vol. 95, no. 3, pp. 493–511, 2003.

[39] N. Gröwe-Kuska, H. Heitsch, and W. Römisch, "Scenario reduction and scenario tree construction for power management problems," in *2003 IEEE Bologna PowerTech Conference Proceedings*, vol. 3, 2003.

[40] C. Lemaréchal, "Lagrangian relaxation," in *Computational Combinatorial Optimization*, pp. 112–156, Springer, 2001.

[41] M. E. Lübbecke and J. Desrosiers, "Selected topics in column generation," *Operations Research*, vol. 53, pp. 1007–1023, 2005.

[42] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, pp. 101–111, January 1960.

[43] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*. Wiley, 1st ed., 1999.

[44] J. Desrosiers and M. E. Lübbecke, "A primer in column generation," in *Column Generation*, pp. 1–32, Springer, 2005.

[45] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Computational Management Science*, vol. 2, no. 1, pp. 3–19, 2005.

[46] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2nd ed., 2006.

[47] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms II*. Springer, 1993.

[48] T. Larsson, M. Patriksson, and A. Strömberg, "Ergodic, primal convergence in dual subgradient schemes for convex programming," *Mathematical Programming*, vol. 86, no. 2, pp. 283–312, 1999.

[49] C. Lemaréchal and C. Sagastizábal, "Variable metric bundle methods: From conceptual to implementable forms," *Mathematical Programming*, vol. 76, no. 3, pp. 393–410, 1997.

[50] F. Vanderbeck and M. W. P. Savelsbergh, "A generic view of Dantzig-Wolfe decomposition in mixed integer programming," *Operations Research Letters*, vol. 34, pp. 296–306, 2006.

[51] G. B. Dantzig, *Linear Programming and Extensions*. Princeton University Press, 1st ed., 1963.

[52] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Operations Research*, vol. 46, pp. 316–329, 1998.

[53] F. Vanderbeck, "Implementing mixed integer column generation," in *Column Generation*, pp. 331–158, Springer, 2005.

[54] G. Hechme-Doukopoulos, S. Brignol-Charousset, J. Malick, and C. Lemaréchal, "The short-term electricity production management problem at EDF." Mathematical Optimization Society Newsletter OPTIMA 84, September 2010.

[55] W. K. Klein Haneveld and M. H. van der Vlerk, "Stochastic integer programming: General models and algorithms," *Annals of Operations Research*, vol. 85, pp. 39–57, 1999.

[56] S. Takriti and J. R. Birge, "Using integer programming to refine Lagrangian-based unit commitment solutions," *IEEE Transactions on Power Systems*, vol. 15, pp. 151–156, 2000.

[57] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL – A Modeling Language for Mathematical Programming*. Brooks/Cole, 2003.

[58] IBM Ilog, *IBM Ilog CPLEX v.12.4 – User's Manual for CPLEX*, 2009.

[59] National Grid plc, "2013 electricity ten year statement (ETYS)." `www2.nationalgrid.com/UK/`, November 2012.

[60] Department of Energy & Climate Change (DECC), "Electricity generation costs 2013." `www.gov.uk/government/publications/electricity-generation-costs`, July 2013.

[61] M. A. Ortega-Vazquez and D. S. Kirschen, "Optimizing the spinning reserve requirements using a cost/benefit analysis," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 24–33, 2007.

[62] R. Billinton and R. N. Allan, *Reliability Evaluation of Power Systems*. Springer, 1996.

[63] National Grid plc, "Data explorer - historic demand data." `www2.nationalgrid.com/UK/`, 2013/14.

[64] Elexon Ltd, "Balancing mechanism reporting system (bmrs)." `www.bmreports.com/`, October 2012.

[65] L. Balling, "Flexible future for combined cycle," *Modern Power Systems*, vol. 30, pp. 61–63, December 2010.

[66] L. Balling, "Fast cycling and rapid start-up: new generation of plants achieves impressive results," *Modern Power Systems*, vol. 31, pp. 35–40, January 2011.

[67] National Grid plc, "Monthly balancing services summaries (MBSS) 2013/14." `www2.nationalgrid.com/UK/`, 2013/14.

[68] S. L. Hawkins, *High resolution reanalysis of wind speeds over the British Isles for wind energy integration*. PhD thesis, The University of Edinburgh – School of Engineering, November 2012.

[69] European Wind Energy Association (EWEA), "Integrating wind – developing Europe's power market for the large-scale integration of wind power." `http://www.trade-wind.eu/`, May 2009.

[70] London Economics, "The value of lost load (voll) for electricity in great britain – final report for OFGEM and DECC." `londoneconomics.co.uk/publications/`, July 2013.

[71] G. Giebel, P. Sørensen, and H. Holttinen, "Forecast error of aggregated wind power," Tech. Rep. Risø-I-2567(EN), Risø National Laboratory, April 2007.

[72] G. Kariniotakis, P. Pinson, N. Siebert, G. Giebel, and R. Barthelmie, "The state of the art in short-term prediction of windpower – from an offshore perspective," in *Proceedings of the 2004 SeaTech Week*, October 2004.

[73] R. Barth, L. Söder, C. Weber, H. Brand, and D. J. Swider, "WILMAR deliverable 6.2 (d) - methodology of the scenario tree tool," tech. rep., Institute of Energy Economics and the Rational Use of Energy, University of Stuttgart, 2006.

[74] L. Söder, "Simulation of wind speed forecast errors for operation planning of multi-area power systems," in *International Conference on Probabilistic Methods Applied to Power Systems*, pp. 723–728, IEEE, 2004.

[75] B. A. Murtagh and M. A. Saunders, "Minos 5.51 user's guide," Tech. Rep. SOL 83-20R, Department of Management Science and Engineering, Stanford University, 2003.

[76] B. Mauch, J. Apt, P. M. S. Carvalho, and M. J. Small, "An effective method for modeling wind power forecast uncertainty," *Energy Systems*, vol. 4, no. 4, pp. 393–417, 2013.

List of Figures

List of Tables