UNIVERSITÀ DEGLI STUDI DI MILANO

Dottorato di Ricerca in Scienze Matematiche, ciclo XXIX°

Dipartimento di Matematica

Doctoral Thesis

Algorithms for the large-scale Unit Commitment Problem in the simulation of Power Systems

Discipline Sector: MAT/09 (Operations Research)

Candidate:

Dott. Andrea Taverna

Tutor:

Prof. Giovanni Righini

Industry Tutor:

Dott. Alberto Gelmini

Coordinator: Prof. Vieri Mastropietro

A.A. 2015/2016

"Operations Research is the securing of improvement in social systems by means of scientific method."

C. Churchman (1970) – *Operations research as a profession.*

To my tutor Giovanni Righini and Alberto Ceselli, whose support and guidance have been fundamental in this work.

To Alberto Gelmini, Michele Benini, Dario Siface and the other researchers at RSE S.p.A., who introduced me to exciting research activities in power systems.

To Thorsten Koch and the other researchers at ZIB, for providing a friendly environment to work on challenging problems.

To my parents Gabriella and Riccardo, and to my brother Giorgio, for encouraging and supporting me in several ways. I hope this new results will provide rewards both for me and you.

To myself, for learning to better cope with uncertainty and the unexpected.

Thank you.

Contents

Chapter 1

Introduction

Modern power systems management relies on solving several advanced mathematical problems at different levels. With the development of mathematical programming techniques more and more complex models have been formulated to support decisions in power systems.

The Unit Commitment Problem (UCP), where a set of power plants needs to be scheduled to satisfy energy demand and other system-wide constraints, has been employed for decades to support operational planning of power plants. The model has been used to simulate whole systems as well, which include transmission networks and energy markets, to support strategic decisions on medium and long term horizons. Compared to the operational case, a medium-term model represents more components of the system on a larger scale. Therefore, to limit the complexity, a coarser representation of the single components, often with a linearised approximation, is adopted for these models. On the other hand, these models are typically used to evaluate long-term decisions on the system, by simulating its ideal behaviour under different scenarios, hence low-level details of the system, that would be otherwise critical for operational decisions, can still be abstracted without impacting on the reliability of the simulation.

Large-scale linear UCP models for medium-term simulations are usually quite expensive to solve to high accuracy, and require either large-scale computing power or long solution times. In this work we tackle the problem of performing accurate medium-term power systems simulations with the additional requirements of employing conventional computing power, such as personal computers, and a solution time of a few hours. The problem is routinely faced by our industry partner, the Energy Systems Development department at RSE S.p.A. (Ricerche Sistema Energetico, Energy System Research), a major industrial research centre on power systems in

Italy. While good heuristic solutions, that is with an optimality gap below 10%, can be found for large-scale UCPs in affordable time with the mentioned setting, more accurate solutions, for example with a gap below 1%, are sought to improve the reliability of the simulations and help domain experts, who may not be familiar with the details of mathematical programming methods, to better support their analysis.

The goal of this thesis is to investigate mathematical programming techniques to efficiently solve these large-scale linear UCPs with high-accuracy. In particular, we seek to obtain the following results: a fast heuristic with a lower bounding procedure, to provide good approximated simulations with a quality estimate, and a refinement procedure, to improve the heuristic solution quality if needed. Among the ideas we explored the following methods are the most promising: a matheuristic to efficiently compute good solutions and two exact bounding methods: column generation and Benders decomposition. These methods decompose the problem by decoupling the commitment of thermal plants, represented by discrete variables, and their level of production, represented by continuous variables. The thesis is organised as follow. In chapter 2 we introduce the general problem of power systems simulation. In chapter 3 we introduce the UCP variant subject of this study. In chapter 4 we propose a matheuristic which employs aggregation, continuous relaxations and spatial decomposition to obtain high quality approximated solutions in affordable time. In chapter 5 we propose a column generation method to compute accurate bounds for the problem; we devised an ad-hoc combinatorial exact pricing algorithm, a hybrid master dual optimisation method and a rounding heuristics. In chapter 6 we propose a Benders decomposition method which uses refined optimality cuts and a two-phase heuristic, which first quickly computes valid Benders cuts by solving the continuous relaxation of the problem, and then restores integrality constraints to continue with the standard Benders decomposition. Finally, in chapter 7 we report an empirical analysis of the complexity and numerical stability of the problem and its relaxations.

From this study we conclude that advanced algorithms are needed to solve these large-scale UCPs to the desired accuracy. On the other hand, degeneracy and flatness in the objective function cause state-of-the-art methods and solvers to converge slowly or to fail to converge due to numerical instability issues. Experimental solvers (Desrosiers et al., 2015; Castro, 2014) have been recently proposed that may help overcome these difficulties by directly tackling sparsity and degeneracy. Unfortunately, these solvers lack an adequate implementation to be directly applied to real world problems like ours.

Degeneracy and flatness in the objective function are inherent properties of the model, derived by its size. In other terms, the decision of changing the schedule of a few power plants over a few hours is unlikely to have a sensible impact over the global objective as several other power plants and periods are involved. Thus a solver that explores the solution space finds candidate solutions embedded in large neighbourhoods of alternatives with similar value, a "plateau" which causes exploration to stall.

Contents from this thesis have already been published in international journals or presented to international conferences.

Publications in international journal or conferences with peer-reviewed published proceedings:

- Taverna, A. (2016), Benders Decomposition on Large-Scale Unit Commitment Problems for Medium-Term Power Systems simulations. In *Gesellschaft für Operations Research 2016 (GOR 2016)* - submitted August 2016.

- Ceselli, A., Gelmini, A., Righini, G., and Taverna, A. (2014). Mathematical programming bounds for large-scale unit commitment problems in medium-term energy system simulations. In *SCOR 2014 – 4th Student Conference on Operational Research*.

Chapter 2

Medium-term Power Systems simulation

This chapter describes the real-world problem that motivated our work: the simulation of electric power systems for strategic decision support. Section 2.1 describes the main characteristics of power systems considered in the simulation. Section 2.2 describe the assumptions and points of view adopted for the simulation.

2.1 Power Systems

A power system supports the production and exchange of electric power. Real-world power systems cover large geographical areas, possibly spanning different countries, and coordinate both large-scale and small-scale production. This study focuses on systems for the bulk production of electricity and its transmission through high-voltage lines. They are assumed to have the following elements:

- power plants, controlled by producers;
- a capacitated power network, to exchange power among different parts of the system;
- a system-wide electricity exchange, where producers bid against one another to share demand.

Electricity can be easily transmitted through power lines across the system in short time, as its speed is comparable to the speed of light, but it is hard to store in large quantities. For most power systems it is not practical to satisfy demand by relying on stored electricity. On the other hand, mismatches between supply and

demand can be likely disruptive as they compromise the stability of the network and lead to discharges or blackouts. Therefore, the system constantly requires an accurate match between consumption and production.

Suppliers and consumers coordinate through bidding in the system-wide exchange to balance demand and supply at minimum cost. Auctions are generally held daily or hourly, with power being traded with hourly resolution or higher. As demand is uncertain (fig. 2.1), each market player needs to bid according to demand forecasts. The system then holds different auctions with different timings and contract types to deal with provisioning errors and secure the overall balance. The most important market is the day-ahead market where power is traded every 24 hours for the next day. The system can have an intra-day market as well, where power is traded at higher frequency, e.g. each hour, allowing to fix unbalances determined by previous market sessions.

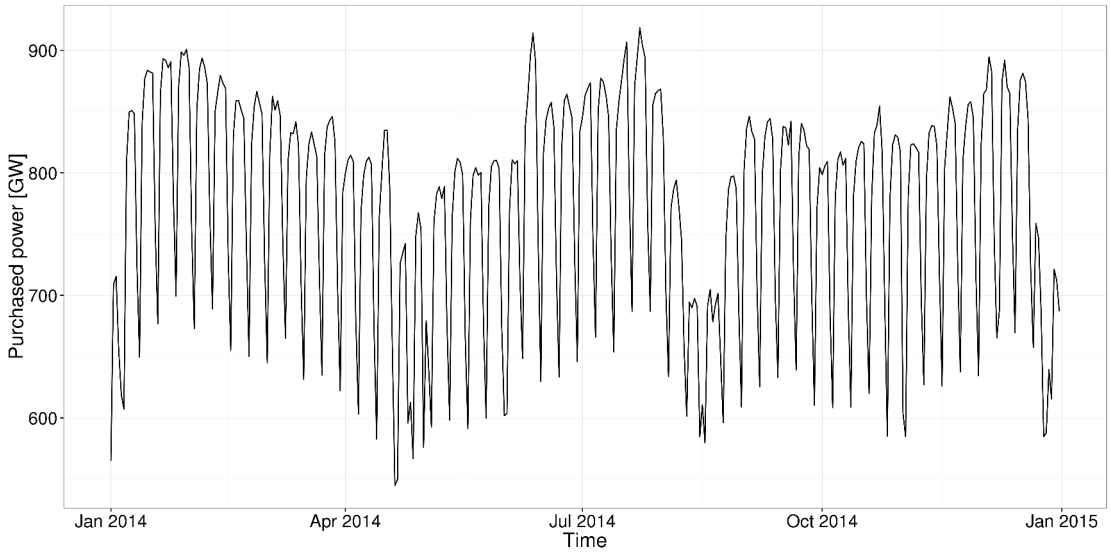


Figure 2.1: Total daily purchased power in the 2014 Italian Day-Ahead market (GME, 2014)

Several types of power plants are available in modern power systems. In most cases, a power plant converts the mechanical energy of a fluid flow, such as water or steam, into electricity through a turbine, in turn connected to an electric generator. In Europe and most other systems the plants that provide the fundamental generation capacity are thermal and hydroelectric power plants.

Thermal power plants produce electricity by consuming fuels, such as coal, lignite, oil, natural gas or nuclear fuels They need a proper mixture of fuels to operate and require complex operations for production and maintenance. In particular, varying the production level or switching the state, from active to inactive or vice

Total net production	3,031,853
of which:	
Conventional thermal	1,442,237
Nuclear	830,842
Hydro	400,647
of which from pumped storage	30,798
Wind	251,265
Solar	96,616
Geothermal	5,817
Other	4,429
Imports	386,930
Exports	371,432
Energy absorbed by pumping	43,051
Energy supplied	3,004,300

Table 2.1: Production breakdown for EU-28 in 2015 (Eurostat, 2015). Values in $[GWh]$

versa, can require significant time and effort depending on the technology employed. Furthermore, with the exception of nuclear plants, during production they emit significant quantities of CO_2 and pollutants, such as nitro-oxides NO_X, which are subjected to fees and regulations. Compared to other plants, they bear high costs of production.

Hydroelectric power plants produce electricity by driving turbines via water streams, originating from rivers or reservoirs. Reservoirs can be fed via an up-stream river, a pumping mechanism (*pumped storage*) or both. Plants with little or no reservoir capacity, called Run-Of-the-River (ROR) depend entirely on the down-stream water flow for their production. Compared to thermal plants, hydroelectric plants provide clean energy with negligible production costs. In particular, plants with reservoir play an important role in balancing the system as they store energy, by collecting water in the reservoir, during low-demand periods and release it, emptying the reservoir to increase production, during high-demand periods.

Other relevant power plants are wind and photo-voltaic (PV) plants. They can provide a considerable amount of renewable energy to the system and with negligible production costs, but their supply is intermittent and volatile and, thus, cannot be relied upon to ensure the stability of the system.

Power plants can be classified according to the reliability of their supply and its ability to follow demand fluctuations. A first distinction is *dispatchability*. A power plant is dispatchable if its production can be timely controlled by system actors,

such as the plant owner or grid operators. Thermal power plants and hydroelectric power plants with reservoir are dispatchable. ROR plant, wind power plants and PV power plants are not dispatchable.

Dispatchable power plants can be further characterised by their *flexibility*, which is the extent they can switch state, from on to off or vice versa, to timely follow changes in the demand. Flexibility can be measured in terms of the number of hours needed or the costs born by the plant operator to switch state. Hydroelectric plants with reservoir offer high flexibility, In thermal power plants depends on the technology used.

Non-dispatchable, or intermittent, power sources can instead be characterised by their *variability*, defined as the degree with which their production level can change in an unexpected way, or, equivalently, the effort required to adequately forecast their production. Production from ROR plants generally exhibits low variability and can be effectively predicted, while wind and PV power productions are normally highly volatile (Brouwer et al., 2014).

The presence of variable and non-dispatchable power supply aggravates the problem of efficiently balancing the system as it increases uncertainty in the net demand, posing a risk to the stability of both market and network (Brouwer et al., 2014) (Lanati et al., 2015). To handle demand uncertainty and ensure profitability, producers adopt different load sharing policies, which can be broadly divided into three classes:

- base-load: for power plants that can continuously satisfy the *base load*, i.e. the minimum level of demand to ensure the stability of the network. Their supply has low variability and they run for most of the year. They require low production costs to be competitive but do not need high flexibility.

- peaking: for power plants that cover *peak demands*, i.e. periods of high demand and high electricity prices. As they run for a limited amount of time during the year, they bid supply at high prices and require the highest flexibility.

- mid-load: for power plants that can follow the variation of demand throughout the day. They have production costs and flexibility which are between the ones offered by base-load and peaking power plants.

In figure 2.2 we illustrate a possible division of the daily load among plants with different strategies. According to this classification, hydroelectric plants with reservoir can adopt all the three strategies as they offer high flexibility and low production

costs. Among the thermal plants, nuclear plants and older coal-powered plants adopt most often the base-load scheme. Combined-Cycle Gas Turbine plants (CCGT) are a well-known class of peaking thermal plants and are usually fuelled by natural gas to ensure low production costs and higher efficiency. Other thermal plants fit into either the base-load or the mid-load class.

Finally, the system can exchange energy with outer systems as well, by importing or exporting power through cross-border lines. Exchanges between the system and the external world are decided outside the global exchange.

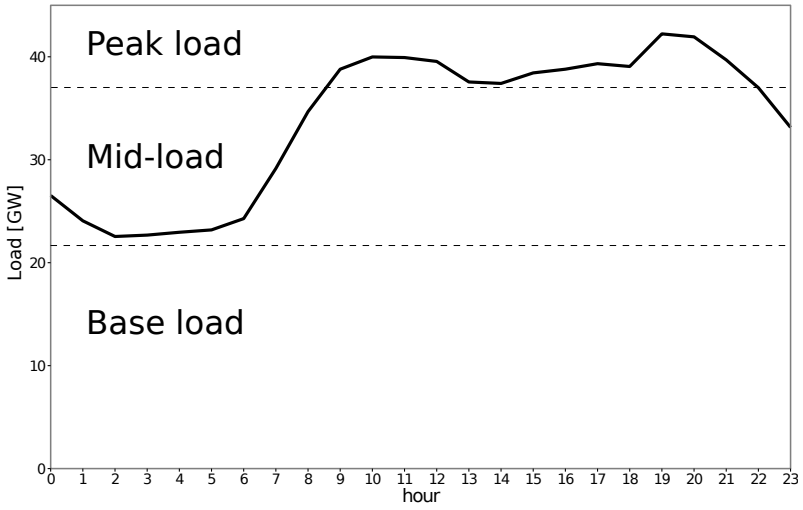


Figure 2.2: Illustration of daily load and possible corresponding strategies for power plants.

2.2 Power Systems simulation

In this section we frame our work by describing the kind of power system simulation subject of this study.

In the literature this approach has been used extensively by domain experts (Burstedde, 2012; Leuthold et al., 2010; Lienert and Lochner, 2012; Spieker et al., 2015; Lanati et al., 2015; Zhang et al., 2016) to directly simulate and compare different scenario hypothesis accounting for variability of demand, fuel prices and other exogenous variables over the year. Simulations are expected to return plausible estimates for aggregated descriptors of the system, such as overall fuel consumption or energy production, and thus provide an immediate comparison with corresponding long-term scenario hypothesis for the same system (Spieker et al., 2015).

2.2.1 Objective: Market and System simulation

In this setting the simulation of the system is used to compute its ideal behaviour under a specific scenario hypothesis. Simulation results are then used as benchmarks to evaluate different scenarios and to support long-term decisions.

The "ideal behaviour" corresponds here to the maximisation of global welfare by satisfying demand at the lowest achievable costs. A well-known result in microeconomics is the *theorem of welfare* which ensures the welfare-maximising solution corresponds to the outcome of a perfectly competitive market, in which prices are determined solely by the free interaction between supply and demand. In this sense, the definition of a global ideal behaviour corresponds to the point of view of both a system planner, such as a system-wide authority that oversees the exchange, and the market players, who want to participate in a fair market.

We study *System Simulation* or *Fundamental Electricity Market Simulation* (Burstedde, 2012), whose goal is to highlight and analyse the fundamental drivers of market outcome, e.g. technical and normative limits on generation and transmission capacity. As such, the simulation uses a detailed representation of the technical assets of the system but simplifies the electricity market by assuming the following:

- the market reaches a perfectly-competitive equilibrium, i.e. imperfect competition is not considered;
- producers bid at production cost, i.e. strategic behaviour is not considered;
- demand is inelastic.

The ideal behaviour of the system can then be simulated by computing the scheduling of the plants that minimises the global production costs. The simulated market is not guaranteed to be close to the real market behaviour, suitable, for example, to forecast actual prices in the exchange, but it can still be used to support strategic decisions.

More sophisticated market models would allow to perform a more realistic simulation with imperfect competition and strategic behaviours, but they are, in general, much more complicated to implement and solve than models for system simulation. It is outside the scope of this work to provide a review of the problems of advanced electricity market simulation. For a detailed reference, we point the reader to (Ventosa et al., 2005).

2.2.2 Network: zonal and nodal models

Our model uses a zonal representation of the network as opposed to a nodal one. In practice transmission networks are composed of several nodes and lines. A nodal model allows to represent these components explicitly. Zonal models, instead, aggregate network elements in homogeneous zones to obtain a coarser and simpler representation of the system.

Zonal models are first of all used in operations to better organise the system and improve its efficiency (ACER, 2011). A zone is defined as a network area where congestion is unlikely to occur. Equivalently, zones are a partition of the network induced by the lines that are most likely to experience congestion, i.e. the bottlenecks of the network. During a congestion links in the network saturate and prevent the transmission of further power to satisfy demand, causing the market to split in areas where prices converge to different values. Zones are then areas with uniform prices for which system actors can define policies and mechanisms while abstracting from the underlying network details. For example, exchanges can implements zonal pricing, where participants submit offers specifically for each zone. Congestions can also be resolved by pricing transmission or generation capacity between zones.

2.2.3 Time scope

Simulations are performed on medium-term horizons, i.e. over one year, with hourly resolution. The length of the horizon allows to account for seasonalities of demand and supply (fig. 2.3), which are known to be influenced by the weather, temperatures and human activities. Furthermore it allows a direct comparison with medium-term forecasts of power systems, which often report annual aggregates for indicators such as demand, supply, and costs (ENTSO-E, 2013) ; (EC, 2013). The high temporal resolution, instead, ensure to adequately represent the response of the system to demand fluctuations, in particular peaking hours.

Compared to short-term ones, models for medium-term simulation are larger and afford a lesser level of detail since, on one hand, low-level decisions with short-term impact become less important as the horizon grows and, on the other hand, the length of the horizon makes impractical to consider more details in the model.

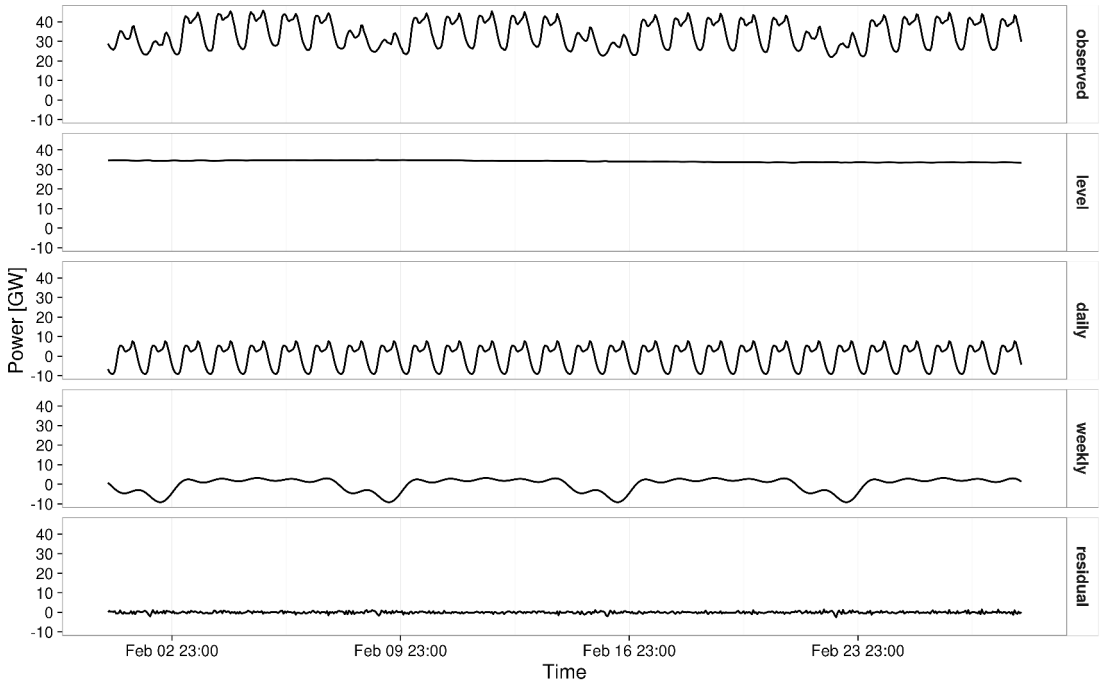


Figure 2.3: Hourly purchased power on the Italian Day-Ahead market on February 2014 (GME, 2014) decomposed by TBATS additive model (Livera et al., 2011). Five time series are shown: original data and four additive components: level, daily seasonality, weekly seasonality and the residual. The daily component shows two positive peaks on every midday and evening. The weekly component shows a decrease on every weekend.

2.2.4 Handling uncertainty

In real systems agents face uncertainty on data such as prices of electricity, costs of fuels and demand, which they handle by using statistics and stochastic modelling to guide operational and tactical decisions. For medium-term simulation both stochastic and deterministic models are used in the literature.

Compared to deterministic ones, stochastic models allow to directly represent uncertainty and yield richer simulations, but they are larger and more expensive to solve. On the other hand, deterministic simulations, in which actors are assumed to possess perfect foresight, provides a plausible representation of the ideal behaviour of the system and, thus, can still offer valuable support to strategic decisions.

In our work we address deterministic simulation.

2.3 Example: simulation of European scenarios at 2020

In this section we report a study from RSE S.p.A.in (Benini et al., 2014) as an example of system simulation. The study considers Italy and neighbouring countries on the northern frontier at year 2020 under four different scenario hypotheses. The system is composed by six countries (fig. 2.4): Italy (IT), Germany (DE), France (FR), Switzerland (CH) and Slovenia (SL). The network is represented with one zone for each country, except for Italy which is described with six market zones: North (NO), Center-North (CN), Center-South (CS), South (SU), Sardinia (SA) and Sicily (SI).

Each scenario hypothesis correspond to an instance of large-scale Unit Commitment Problem with one year horizon and hourly resolution. The instances have been solved with the heuristic implemented in the simulator sMTSIM (Siface et al., 2014) from RSE S.p.A.. The goal of this section is to explain the assumptions behind the development of scenarios and to show how a system simulation provides useful insights on the behaviour of the system. For more details we refer the reader to the original publication.

2.3.1 Scenario construction

For the study reported here, each scenario contains the following system descriptors:

Figure 2.4: Countries involved in the study

- thermal power plants: minimum and maximum production bounds, flexibility, coefficients of the cost curve and fuel emissivity coefficients for CO_2;
- hydroelectric power plants: minimum and maximum production bounds, pumping efficiency and hourly pumping limit, reservoir capacity, hourly lateral inflow from the upstream;
- market zones;
- power network: hourly transmission capacity between zones;
- gross zonal demand, at hourly resolution;
- injections: power flow determined by intermittent sources such as wind and PV plants and import/export flows between outer systems, i.e. neighbouring countries not included in the simulation.

To construct future scenarios there are several sources that provide forecast for descriptors such as fuel prices and system assets, e.g. generation or transmission capacity (ENTSO-E, 2013) ; (EC, 2013).

Demand and injections can most often be estimated only as annual totals. It is not possible to reliably forecast hourly profiles for these data further than a few days. As an alternative, domain experts can generate realistic hourly profiles by

combining historic time series of the same data, and then scale the result to obtain the expected annual total.

2.3.2 Simulation

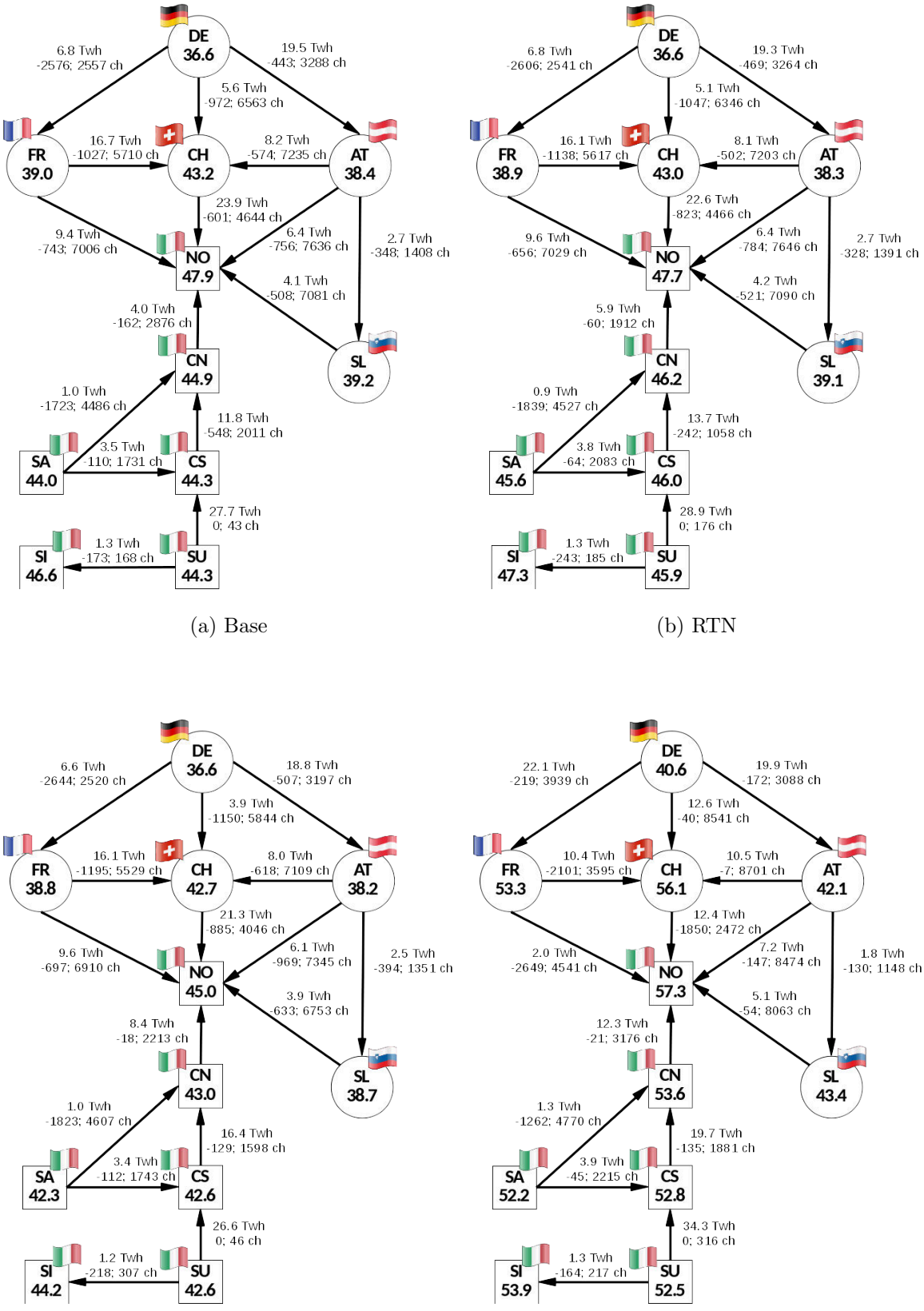
The study considers a "Base" scenario, in which the system evolves according to current expectations, and then defines three other scenarios

- RTN ("Rete Trasmissione Nazionale", National Transmission Network): network capacity between zones (NO,CN) and (CN,CS) are expanded;
- Import: an additional import flow from Montenegro through the CN zone is added to Italy;
- LessNuke: nuclear capacity is partially reduced in Germany, France and Switzerland. In France new flexible gas-powered power plants, e.g. CCGT plants, are added back to compensate for the capacity reduction.

We briefly present the results of the study. In figure 2.5 we report a map for each scenario. Square nodes represent Italian market zones and circle ones the neighbouring countries. Each node is labelled with the zone identifier and the average hourly electricity price, in $[€/MWh]$, computed as average shadow prices of the hourly zonal demand. The latter ones are obtained from the economic dispatching solution computed as a last step by the MTSIM program. Arrows represent the links between the zones, and are labelled with two lines of text: the first line reports the total amount of flow, in $[TWh]$, whose sign determines the direction of the arrow, and the second line reports the total number of congestion hours $[ch]$ for the link in each direction, with a negative value for the congestions in the direction opposite to the arrow. In table 2.2 we report the thermal production by fuel and country and the overall increase in CO_2 emissions with respect to the Base scenario.

Simulation results suggest the following observations

- in the RTN scenario, thanks to the network expansion, power flow through the links (CS,CN) and (CN,NO) increases, while the number of congested hours decreases. Production from southern regions in Italy can then reach the northern ones. This causes a reduction in the price difference between the Italian regions, with the NO zone finding cheaper electricity and the Southern regions selling more power at higher prices. Furthermore, the overall amount of imported flow in Italy from the other countries decreases by $1.1TWh$;



(a) Base (b) RTN

(c) Import (d) LessNuke

Figure 2.5: Maps for four scenarios from (Benini et al., 2014).

Scenario	Italy		Other countries		CO_2 emissions change w.r.t. "Base" scenario
	Coal	Natural gas	Coal	Natural gas	
	$[TWh]$	$[TWh]$	$[TWh]$	$[TWh]$	$[Mt_{CO_2}]$
Base	67.3	29.1	234.1	2.1	0
RTN	68.4	28.9	233.3	1.8	-0.2
Import	65.5	27.8	231.2	1.7	-5.4
LessNuke	74.9	35.8	328.0	12.8	+105.6

Table 2.2: Conventional thermal production and CO_2 emissions in different scenarios

- in the Import scenario the additional electricity from Montenegro causes prices in the NO region to decrease considerably;
- in the LessNuke scenario the reduction of nuclear capacity causes an overall increase of prices. France stops being the largest exporter of the system, in favour of Germany. A sharp increase in conventional thermal production from coal and natural gas is also measured, which yields a significant increase of CO_2 emissions.

Chapter 3

The Unit Commitment Problem (UCP)

The UCP, in its general form, requires to schedule the activation (commitment) of a set of power plants to satisfy system-wide constraints and to optimise a given objective. Indeed, a medium-term electric system simulation can be modelled as a large-scale UCP.

In this chapter we introduce the basic version of the UCP along with a review of the related literature.

3.1 Literature review

The UCP is a well-known combinatorial optimisation problem for which a huge literature is available, spanning both theory and applications. We refer the reader to (Saravanan et al., 2013) and (S. Salam, 2007) for more detailed reviews.

In its general case, the UCP includes integer variables to represent the state of plants and non-linear constraints and objectives to represent cost functions or flows in the transmission networks. One major source of non-linearity for the models surveyed in the literature is thermal power plants, especially their cost function. They are often modelled with quadratic convex cost functions and mixed-integer linear constraints. The problem can then be formulated as a Mixed-Integer Non-Linear Problem (MINLP), for which the exact optimisation is usually prohibitively expensive. The convex UCP is known to be $\mathcal{NP}$-hard (Guan et al., 2003). The convex UCP with a single thermal unit (1-UCP) has polynomial complexity and can be solved with an exact dynamic programming algorithm (Frangioni, 2006).

The UCP has originally been studied to support operational decisions for small systems composed of few thermal power plants on a daily or weekly time horizon. Then most of the published literature covers short-term UCP models, with a horizon from few hours to a week. These models are designed to represent operational decisions with high level of detail, including complicating elements such as non-linear costs, to accurately represent the system components.

For the small-scale, short-term UCP several heuristics have been proposed. Well-known constructive method is "priority lists" (Tingfang and Ting, 2008), where the order of commitment and decommitment of different plants, i.e. the "lists", are determined heuristically via a greedy approach. Other methods include soft computing like neural networks, genetic algorithms and swarm intelligence, and meta-heuristic approaches like tabu search (Padhy, 2004) (S. Salam, 2007) These methods can be easily extended to handle more complicated models as well, by adequately encoding additional constraints and goals as penalties or rewards in the solution process, but the approach is inherently heuristic and sensitive to the specifics of the instances being solved.

The most promising techniques for solving larger UCPs employ Lagrangian relaxation (Belloni et al., 2003; Borghetti et al., 2003; Finardi and da Silva, 2006) to decompose the original problem into smaller and simpler subproblems. These methods usually solve the continuous the Lagrangian model via non-smooth optimisation techniques, bundle methods being the most popular ones, and then they compute feasible integer primal solutions heuristically via rounding techniques, cutting planes, dynamic programming and meta-heuristics.

Recent applications have increasingly expanded both the simulation time horizon and the complexity of the system, by explicitly representing the power network and including heterogeneous plants, such as hydroelectric, nuclear, waste-to-energy and biomass plants.

Large UCP models with several generation units or a long time horizon usually rely on a linearisation of the thermal units cost functions and are solved via mixed-integer linear programming (MILP), which allows to obtain high quality solutions more efficiently than conventional MINLP techniques. UCPs with a weekly time horizon and less than one hundred units have been effectively solved via commercial MILP solvers (Chang et al., 2004). More sophisticated techniques for larger models include branch-and-cut algorithms, e.g. the perspective cuts (Frangioni et al., 2009) which use MILP approximations to support the solution of more accurate

MINLP models. In (Frangioni et al., 2007) a comparison is made between a MINLP approach, including inter-temporal constraints, solved via Lagrangian relaxation and its MILP approximation built with the perspective-cuts described in (Frangioni et al., 2009) and solved with commercial Branch-&-Bound solvers. The paper compares the two algorithms on a weekly scenario with at most 100 thermal units and 200 hydroelectric plants showing that the two approaches are complementary, i.e. the Lagrangian approach manages to compute high-quality lower bounds with great efficiency, while the MILP approach excels at computing good upper bounds from the first branching iterations, but fails at closing the dual gap within practical time for standard levels of accuracy. The paper then combines the two approaches by feeding the MILP solver with the lower-bound computed by the Lagrangian algorithm, obtaining, in particular, a significant improvement on the scalability of the method on the most difficult tests, solving larger instances with higher accuracy levels.

Fewer publications deal with long-term UCP, spanning several months or a whole year. In (Kjeldsen and Chiarandini, 2012) the annual energy production in Denmark is simulated with a MILP model. Exact approaches are known to be prohibitively expensive (Taverna, 2011; Siface et al., 2014); therefore the authors develop a set of constructive heuristics and iterative improvement methods which they combine to obtain a feasible heuristic solution. In this way instances involving up to twenty thermal plants and a time horizon of a whole year could be solved in ten to twenty minutes.

3.2 Mathematical formulation

The Unit Commitment Problem (UCP) we consider in this study is formulated as a large-scale Mixed-Integer Linear Problem (MILP). Here we recall the problem description, highlighting its components and the modelling choices employed to obtain a linear model.

3.2.1 Problem description

The UCP model adopts a zonal representation of the network, where the network is partitioned into price zones hosting different power sources. Zones exchange power through a capacitated power network. Realistic power flow models use Alternate Current (AC) non-linear formulations to account for electrical properties of the

network. As these models can be quite challenging to solve in the general case, linear approximations are often preferred, especially when modelling larger systems which include further elements, as in our case. We use a simpler transportation model, which can be described by a much simpler capacitated linear network flow model, while still yielding a valuable characterisation of network flows. More realistic linear approximations could be achieved using a DC model or a PTDF formulation. For a review of power flow models for UCP we refer the reader to (Van den Bergh and Delarue, 2014).

Time is represented with hourly resolution. Each zone is characterised by a given hourly demand, that must be satisfied by either plants in the same zone or by energy import from other zones.

In the system there are dispatchable and non-dispatchable power sources. The former include thermal and hydroelectric plants with reservoir, and are modelled with decision variables. The latter include wind plants, PV plants, ROR hydroelectric plants and import flows from outer systems, and are assumed to be known and subtracted from the gross demand. The resulting net demand represents the actual load traded in the market. Hydroelectric power plants are represented with linear network flow models that prescribe the balancing between the inbound flow collected by the reservoir and the outbound flow used for production. Their production in each period linearly depends on the outbound flow from the reservoir and is assumed to be costless. We adopt a coarse representation, in which river systems are represented as a single, equivalent hydroelectric plant with reservoir.

Thermal plants need to be ignited and heated to be active and offer limited "flexibility", which here denotes the ease with which a power plant can change state or the production level from one period to the next one. Highly flexible plants, like gas-fuelled CCGT, can change their state or their production level from one period to the next one with few limitations. Others are technically unable to do so, or incur additional costs for it. Flexibility constraints on the plant commitment can be expressed as *minimum up/down constraints*, which force the power plant to maintain the current state for a fixed amount of periods after switching, or as *start-up* and *shut-down costs*, i.e. additional costs the power plant incurs for switching state (see Morales-Espana et al. (2013) for reference). Constraints on the production level are often described as *ramping constraints*, which bounds the change of production level from one period to the next one.

In our formulation we model flexibility by using the minimum up/down con-

straints only. More realistic models would have start-up and shut-down costs, and they would possibly describe the internal state transitions of the power plants. These constraints and costs can be time-dependent, i.e. they can vary depending on the time passed since the last switch, and the costs can be non-linear. However, we remark the following:

- implementing these constraints would require knowing the details of the power plants, which are often not available to the analyst, especially considering long-term scenarios, in which thermal plants are deemed to become more flexible (SETIS, 2015) to adjust to the higher penetration of non-dispatchable RES;

- the objective of the simulation is to represent realistic commitment decisions of the power plants managers. To achieve this it is not necessary to describe the inner working of the power plant in detail but just to have a schedule where commitment and decommitment decisions happen with credible timing. For instance, this can be represented by constraining power plants to remain active for at least 30 hours after switching on. To achieve this, RSE S.p.A. considers estimates of minimum up/down times to be sufficient and, under mild assumptions, equivalent to consider realistic start-up and shut-down costs.

As for ramping constraints, they are usually modelled as linear constraints and, while not present in our formulation, they can be easily added with little or no impact on our methods, as shown in section 5.2.2, where we adapt our algorithms to RSE S.p.A.'s SMTSIM model, which includes ramping constraints.

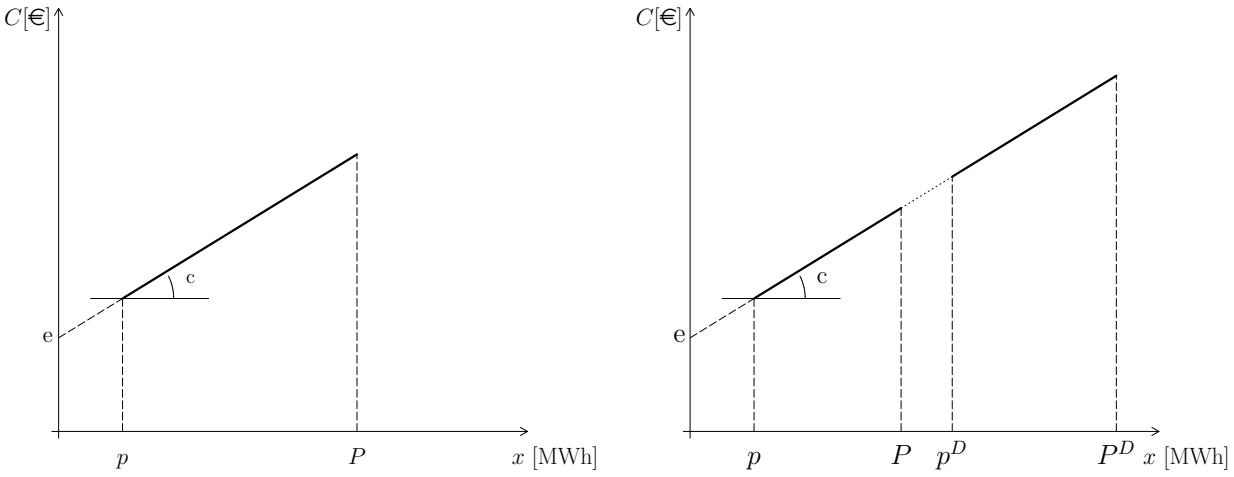
The cost curve for thermal plants is modelled via linear functions, as shown in Figure 3.1 (a)). By "fixed cost" we mean the value of intercept (the value indicated by e in Figure 3.1); by "variable cost" we mean the product between the marginal production cost and the production level (c and x respectively in Figure 3.1). Similarly to other studies from economics (Burstedde, 2012) and engineering (Spieker et al., 2015), we choose a linear continuous approximation of the non-convex cost curve in place of a piece-wise or quadratic one, for the following reasons:

- the use of piece-wise or non-linear formulations could significantly increase the complexity of the problem;

- according to our partners at RSE S.p.A., a quadratic approximation for the cost curves would have a quadratic coefficient one and two orders of magnitude smaller than the linear and constant coefficients respectively; thus, for

the purpose of simulation, it would improve accuracy by a negligible amount compared to a completely linear model.

Thermal plants are therefore represented by a binary activation state, fixed and variable costs, a non-zero technical minimum, indicated by p in Figure 3.1 and reduced flexibility. Variable costs linearly depend on production levels and include pollution penalties. Some thermal plants employ double-shaft technology: they can switch between two working states, with one of them employing more power units and higher production levels (see Figure 3.1 (b)). Therefore the management of thermal plants requires two types of decisions: commitment decisions concern turning the plants on or off and dispatch decisions concern the production level to be attained at each point in time.



(a) (b)

Figure 3.1: Generation cost models for thermal plants in single (a) and double (b) shaft mode.

To deal with such a complex system, we first consider some simplifying observations.

Observation 1. *Thermal plants of the same zone having identical marginal cost, fixed costs and technical minima and maxima are identical from both viewpoints of commitment and dispatch.*

Observation 2. *Thermal plants of the same zone having identical marginal cost (although different fixed costs and technical minima and maxima) are identical from the viewpoint of dispatch.*

Therefore, thermal plants in each zone are clustered in groups, each one including plants characterised by the same marginal costs across the simulation horizon. Each group is partitioned into subgroups, each one including plants characterised by the same fixed costs and same technical minimum and maximum across the simulation horizon.

We remark that the features used for clustering are determined by several characteristics of power plants, in particular production technology and oldness. In a real system the features are likely to vary across similar power plants in different periods, to the point that only trivial singleton clusters could be determined in order to represent power plants with sufficient accuracy. Nevertheless, in the context of large-scale simulations a coarser representation of power plants features is used, which allows for significant clustering.

3.2.2 Notation

Parameters

We define the following sets:

- T : set of time periods;
- Z : set of zones;
- $A \subset Z \times Z$: set of oriented links between zones;
- H : set of hydroelectric plants;
- G : set of thermal plants.

Hydroelectric plants

For each zone $z \in Z$ let H_z be the set of hydroelectric plants in z. Each hydroelectric plant $h \in H_z$ is characterised by the following data:

- P_{zh}: maximum power produced;
- P^{β}_{zh}: maximum pumping power;
- q_{zh}: available volume of water in the reservoir at the beginning of the time horizon;
- Q_{zh}: required volume of water in the reservoir at the end of the time horizon;

- V_{zh}: capacity of the reservoir;
- α_{zh}: conversion factor between energy and water volume units;
- β_{zh}: pumping efficiency factor;
- f_{zh}: maximum water spillage, i.e. water released from the reservoir and not used for energy production;
- n_{zh}: hourly inflow of water from natural sources.

Thermal plants

For each zone $z \in Z$ the following sets are defined

- G_z : set of thermal plants groups in z;
- M_{zg} : set of subgroups of thermal plants in z for $g \in G_z$ such that $\exists M_{zg} \neq \emptyset \forall z \in Z \forall g \in G_z$;
- $M^D_{zg} \subseteq M_{zg}$: set of of subgroups for $g \in G_z$ with double-shaft technology;
- K_{zgm}: set of plants in subgroup $m \in M_{zg}$;
- K^D_{zgm}: set of plants in subgroup $m \in M^D_{zg}$.

For each zone $z \in Z$, group $g \in G_z$ and subgroup $m \in M_{zg}$ thermal plants are characterised by the following data:

- c_{tzg} and e_{tzgm}: marginal and fixed production cost at time $t \in T$, resp.;
- p_{zgm} and P_{zgm}: minimum and maximum power that can be produced by plants in subgroup $m \in M_{zg}$ when they are active;
- p^D_{zgm} and P^D_{zgm}: technical minimum and maximum for plants in subgroup $m \in M^D_{zg}$ when the double-shaft technology is active. In the following, let $p^\Delta_{zgm} = p^D_{zgm} - p_{zgm}$ and $P^\Delta_{zgm} = P^D_{zgm} - P_{zgm}$.
- on_{zgm} and off_{zgm}: minimum number of periods for which plants in subgroup $m \in M_{zg}$ have to stay active or inactive once turned on or off.

Let

$$T^{on}_{tzgm} = \{t' \in T : t \leq t' \leq \min(|T|, t + on_{zgm} - 1)\}$$

be the set of periods in which a plant $m \in M_{zg}$ has to remain active if turned on at time $t \in T$ and

$$T^{off}_{tzgm} = \{t' \in T : t \leq t' \leq \min(|T|, t + off_{zgm} - 1)\}$$

be the set of periods in which it has to remain inactive if turned off at time $t \in T$.

Network

For each oriented link $(i, j) \in A$ and time period $t \in T$ let b_{tij} be the maximum energy transfer capacity of link $(i, j) \in A$. .

Zones

For each zone $z \in Z$ and period $t \in T$ let

- d_{tz} be the demand;
- E_t be the value of lost load, i.e. the cost per unit of unsatisfied demand. It corresponds to the price paid for disconnecting willing customers when the production is insufficient to satisfy the demand;
- F_t be the value of exceeding energy, i.e. the cost per unit of excess production. It corresponds to the cost producers encounter in case of energy curtailment when the production exceeds the demand.

Variables

Hydroelectric plants

To model hydroelectric plants, for each period $t \in T$, zone $z \in Z$ and plant $h \in H_z$, we use the following continuous non-negative variables:

- l_{tzh}: energy production;
- m_{tzh}: energy equivalent of water pumped into the reservoir as water;
- s_{tzh}: spillage from the basin;
- o_{tzh}: volume of water in the reservoir at the beginning of time period $t \in T$.

Network

For each period $t \in T$ and link $(i, j) \in A$ we define continuous variables

- w_{tij}: amount of energy flowing through the link from i to j.

Thermal plants

We introduce, for each period $t \in T$, zone $z \in Z$ and group $g \in G_z$ the following variables:

- x_{tzg}: overall production level for the group.

Then, for each subgroup $m \in M_{zg}$ we introduce the following integer variables:

- y_{tzgm}: number of active plants in subgroup $m \in M_{zg}$;
- ${v^+}_{tzgm}$, ${v^-}_{tzgm}$ integer: number of plants switched on and off respectively.

Analogously we introduce for each $m \in M^D_{zgm}$ and $t \in T$ the following integer variables

- y^D_{tzgm}: number of active plants in double-shaft mode;
- ${v^{D+}}_{tzgm}$, ${v^{D+}}_{tzgm}$: number of plants which enter and exit double-shaft mode respectively.

Zones

For each period $t \in T$ and zone $z \in Z$ we define the continuous non-negative variables

- ENP_{tz}: Energy Not Provided (ENP);
- EIE_{tz}: Energy In Excess (EIE).

These variables represent disruptive events, such as discharges or curtailment, and allow to detect and evaluate problems in the simulated system which can cause a mismatch between supply and demand

The resulting UCP model is the following mixed-integer linear program.

$$\min \quad \phi = \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} c_{tzg}x_{tzg} + \sum_{\substack{t\in T, z\in Z,\\ g\in G_z, m\in M_{zg}}} e_{tzgm}y_{tzgm} + \sum_{t\in T, z\in Z} ENP_{tz}E_t$$

$$+ \sum_{t\in T, z\in Z} EIE_{tz} F_t \tag{3.1}$$

$$\text{s.t. } x_{tzg} \geq \sum_{m\in M_{zg}} p_{zgm} y_{tzgm} + \sum_{m\in M^D_{zg}} p^{\Delta}_{zgm} y^D_{tzgm} \ \forall t \in T, z \in Z, g \in G_z \tag{3.2}$$

$$x_{tzg} \leq \sum_{m\in M_{zg}} P_{zgm} y_{tzgm} + \sum_{m\in M^D_{zg}} P^{\Delta}_{zgm} y^D_{tzgm} \ \forall t \in T, z \in Z, g \in G_z \tag{3.3}$$

$$y^D_{tzgm} \leq y_{tzgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{3.4}$$

$${v^+}_{tzgm} \geq y_{tzgm} - y_{(t-1)zgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{3.5}$$

$${v^-}_{tzgm} \geq y_{(t-1)zgm} - y_{tzgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{3.6}$$

$${v^+}^D_{tzgm} \geq y^D_{tzgm} - y^D_{(t-1)zgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{3.7}$$

$${v^-}^D_{tzgm} \geq y^D_{(t-1)zgm} - y^D_{tzgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{3.8}$$

$$y_{tzgm} \geq \sum_{\tau\in T: t\in T^{on}_{\tau}} {v^+}_{\tau zgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{3.9}$$

$$y_{tzgm} \leq |K_{zgm}| - \sum_{\tau\in T: t\in T^{off}_{\tau}} {v^-}_{\tau zgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{3.10}$$

$$y^D_{tzgm} \geq \sum_{\tau\in T: t\in T^{off}_{\tau}} {v^+}^D_{\tau zgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{3.11}$$

$$y^D_{tzgm} \leq |K^D_{zgm}| - \sum_{\tau\in T: t\in T^{off}_{\tau}} {v^-}^D_{\tau zgm} \ \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{3.12}$$

$$o_{1zh} = q_{zh} \ \forall z \in Z, h \in H_z \tag{3.13}$$

$$o_{(|T|+1)zh} = Q_{zh} \ \forall z \in Z, h \in H_z \tag{3.14}$$

$$o_{tzh} + n_{th} + \beta_h \cdot m_{tzh} = o_{(t+1)zh} + s_{tzh} + \alpha_{zh} l_{tzh} \ \forall t \in T, z \in Z, h \in H_z \tag{3.15}$$

$$\begin{aligned} &\sum_{h\in H_z} l_{tzh} + \sum_{g\in G_z} x_{tzg} + \sum_{(i,z)\in A} w_{tiz} + \sum_{z\in Y} ENP_{tz} \geq \\ &\quad d_{tz} + \sum_{h\in H_z} m_{tzh} + \sum_{(z,j)\in A} w_{tzj} + \sum_{z\in Y} EIE_{tz} \end{aligned} \qquad \forall t \in T, z \in Z \tag{3.16}$$

$$y_{tzgm}, {v^+}_{tzgm}, {v^-}_{tzgm} \in [0, |K_{zgm}|] \cap \mathbb{Z}^+_0 \ \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{3.17}$$

$$y^D_{tzgm}, {v^+}^D_{tzgm}, {v^-}^D_{tzgm} \in [0, |K^D_{zgm}|] \cap \mathbb{Z}^+_0 \ \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{3.18}$$

$$w_{tij} \in [0, b_{ij}] \ \forall t \in T, (i,j) \in A \tag{3.19}$$

$$s_{tzh} \in [0, f_{zh}], o_{tzh} \in [0, V_{zh}], l_{tzh} \in [0, P_{zh}], m_{tzh} \in [0, P^{\beta}_{zh}] \ \forall t \in T, z \in Z, h \in H_z \tag{3.20}$$

$$ENP_{tz} \geq 0, EIE_{tz} \geq 0 \ \forall t \in T, z \in Z \tag{3.21}$$

The objective (3.1) is to minimise the sum of production costs and the cost of the energy not provided and in excess. Constraints (3.2) and (3.3) impose that production level is null for inactive plants and within the prescribed production bounds for active ones. Constraints (3.4) impose that only active plants can enter double-shaft mode. Constraints (3.5)–(3.8) enforce consistency between variables describing activation patterns. Constraints (3.9)–(3.12) impose that activation patterns respect minimum on and off times after switching. Finally (3.13)–(6.6) are flow conservation constraints. Constraints (3.16) ensure energy balance between zones and consistency with thermal and hydroelectric productions inside each zone. A more detailed discussion on modelling issues is presented in (G. Migliavacca, 2009) and (Taverna, 2011).

The usual formulation found in the literature specifies one production variable and one state variable for each thermal power plant. The introduction of groups and subgroups allows to represent the production level and state of power plants with one variable for each group and one variable for each subgroup respectively, reducing symmetries and yielding a more compact formulation.

3.2.3 Minimum up/down constraints

One of the main sources of complexity of the UCP we consider is the presence of minimum up-down constraints. In fact different formulations have been proposed in the literature; a comparison among them is presented in (Hedman et al., 2009). In particular, although being aggregated, our constraints (3.9)–(3.12) turn out to be as strong as those defined in (Deepak and Takriti, 2005), which are in turn known to be stronger than the popular *alternating up/down* inequalities (Hedman et al., 2009). That is, we achieve the same tightness as (Deepak and Takriti, 2005) with a more compact formulation. In fact, consider the formulation of constraints (3.5)–(3.12) for single-shaft plants. For $k \in K_{zgm}, t \in T$ let $u_{tk} \in \{0,1\}$ be the commitment variable, and ${\omega^+}_{tk}, {\omega^-}_{tk} \in \{0,1\}$ the "up" and "down" variables such that ${\omega^+}_{tk} = 1 \Leftrightarrow u_{tk} - u_{(t-1)k} = 1$ and ${\omega^-}_{tk} = 1 \Leftrightarrow u_{(t-1)k} - u_{tk} = 1$.

The single-unit up/down constraints presented in (Deepak and Takriti, 2005), called "turn on/off" inequalities, are

$$u_{tk} \geq \sum_{\tau \in T : t \in T_\tau^{on}} {\omega^+}_{\tau k} \quad \forall k \in K_{zgm} \quad \forall t \in T \tag{3.22}$$

$$u_{tk} \leq 1 - \sum_{\tau \in T : t \in T_\tau^{off}} {\omega^-}_{\tau k} \quad \forall k \in K_{zgm} \quad \forall t \in T \tag{3.23}$$

Let $\mathbf{y} = (y_t)^\top{}_{t\in T}$ and $\mathbf{u} = (u_{tk})^\top{}_{t\in T, k\in K_{zgm}}$ be respectively the aggregated and decomposed representation of a schedule for plants $k \in K_{zgm}$ of subgroup $m \in M_{zg}$ such that $y_t = \sum_{k\in K_{zgm}} u_{tk}\ \forall t \in T$.

Proposition 1. *Constraints* (3.9)–(3.12) *hold for* $\mathbf{y}$ *if and only if constraints* (3.22)-(3.23) *hold for* $\mathbf{u}$.

Proof. The implication (3.22)-(3.23) $\Rightarrow$ (3.9)–(3.12) directly follows from the definition of $\mathbf{y}$.

To prove the implication (3.22)-(3.23) $\Leftarrow$ (3.9)–(3.12) , instead, consider the following algorithm to map $\mathbf{y}$ values to $\mathbf{u}$.

For $t = 1$ choose a schedule such that $\sum_{k\in K_{zgm}} u_{1k} = y_1$. For each period $t > 1$, turn on v_t^+ plants, starting from those which have been inactive for the longest time up to $t-1$ and turn off v_t^- plants starting from those which have been active for the longest time up to $t-1$. Assume that for some $t \in T$ a plant selected to be turned on has not completed the required minimum downtime yet, violating constraint (3.23). Then there would be more active plants at time t than the ones who completed their minimum downtime requirement in the periods $\tau \in T : t \in T_\tau^{off}$ and this would violate constraint (3.12), which contradicts the hypothesis of $\mathbf{y}$ being feasible with respect to constraints (3.12). The same argument can be made for plants turned off at time t. □

The proof can be trivially extended to double-shaft units as the corresponding minimum up/down constraints are identical.

3.3 Complexity of the medium-term UCP

Formulation (3.1)-(3.21) contains a polynomial number of variables and constraints and therefore optimising it by means of general purpose solvers might be an option. RSE S.p.A.uses sMTSIM (Siface et al., 2014) to compute a heuristic solution on a 32GB 8GB 8 Core computer. The heuristic is expected to complete within less than one hour.

As a benchmark, we used scenarios constructed by RSE S.p.A.for the Italian energy system in 2011, consisting of $|Z| = 7$ zones in a tree network, three of which connected to external markets, 148 thermal plants partitioned in $|G| = 98$ groups and $|M| = 103$ subgroups, and $|H| = 34$ groups of hydroelectric plants. Thermal plants are split in three types, the first one including 68 plants with minimum on and

off times of 12 and 6 hours, resp., the second one including 48 plants with minimum on and off times of 60 and 20 hours, resp., and the third one including 32 plants with minimum on and off times of 1 hour (i.e. maximum flexibility). The penalty E_t for the energy not provided was set to a very large value, while the penalty F_t for the energy in excess was set to 0.

Demand data are given and planning decisions are required for the full year with a hour-by-hour resolution, which yields problem instances with $|T| = 8760$ time periods of one hour each. Besides testing our algorithms on the full twelve months horizon, we also extracted instances corresponding to single "months", (twelve instances, $|T| = 730$), pairs of consecutive months (six instances, $|T| = 1460$), trimesters (four instances, $|T| = 2190$), quarters (three instances, $|T| = 2290$) and semesters (two instances, $|T| = 4380$).

In earlier experiments we used IBM CPLEX 12.4 MIP and LP solvers on the instances through AMPL on a PC with 4GB of RAM and a Intel Core 2 Duo 1.2GHz processor. In table 3.1 we report the sizes of the instances after AMPL and CPLEX presolving phases.

In later experiments we used CPLEX solvers through CPLEX's own C++ Concert API on the same hardware. In table 3.2 we report two sets of results obtained from the CPLEX C++ Concert library implementation of the model. The first column is the estimated gap from the MIP solver within three hours of computation time. The second is the time needed by the barrier algorithm, with no crossover, to converge on the continuous relaxation of the formulation. Missing values in the table indicate the respective solver failed due to insufficient memory.

The model requires at least one million variables or more beyond the two-months horizon after the presolving phases of both AMPL and CPLEX. CPLEX, on the raw formulation submitted through the Concert API, is able to handle only the smallest instances for both MIP and LP solvers. The MIP solver determines primal solutions within 60% gap from the optimum after three hours of computation time. The LP solver requires 20 minutes alone for the 3 months instances. Larger instances cause out-of-memory errors.

Experimental results show ad-hoc methods are indeed needed to solve our problem in practical time.

Id	Size	Problem size after presolving				
		Constraint matrix			Integer variables	
		Rows	Columns	Non-zeros	Binaries	Generals
1.1	1 month	303527	263360	1584041	132130	27740
1.2		303352	263228	1583612	132126	27740
1.3		302703	262626	1581453	132108	27740
1.4		302638	262596	1581451	132092	27740
1.5		302688	262621	1581484	132104	27740
1.6		303401	263260	1583707	132130	27740
1.7		303274	263154	1583343	132130	27740
1.8		303478	263325	1583907	132130	27740
1.9		303170	263044	1582930	132121	27740
1.10		303230	263099	1583118	132122	27740
1.11		303315	263174	1583443	132121	27740
1.12		302803	262803	1582165	132123	27740
2.1	2 months	608719	528457	3179094	264256	55480
2.2		606904	526856	3173494	264206	55480
2.3		607521	527341	3175822	264234	55480
2.4		608617	528370	3178790	264260	55480
2.5		608306	528058	3177676	264242	55480
2.6		607236	527138	3175219	264231	55480
3.1	3 months	914632	794332	4774952	396361	83220
3.2		913936	793704	4772787	396330	83220
3.3		915083	794717	4776273	396387	83220
3.4		914092	793897	4773492	396365	83220
4.1	4 months	1222780	1062439	6375423	528458	110960
4.2		1223149	1062751	6376415	528494	110960
4.3		1222809	1062466	6375497	528476	110960
6.1	6 months	1842638	1602087	9583956	792696	166440
6.2		1843365	1602764	9586271	792744	166440

Table 3.1: Instances sizes after presolving

Id	Size	CPLEX MIP solution Gap %	CPLEX LP relaxation Time [s]
1.1	1 month	59.7%	244
1.2		56.9%	245
1.3		64.6%	264
1.4		70.9%	220
1.5		70.4%	237
1.6		71.9%	229
1.7		62.5%	224
1.8		73.4%	230
1.9		61.5%	220
1.10		64.0%	231
1.11		60.3%	235
1.12		64.7%	241
2.1	2 months	58.3%	805
2.2		67.8%	689
2.3		71.2%	850
2.4		68.0%	776
2.5		62.7%	710
2.6		62.5%	728
3.1	3 months	–	1566
3.2		–	1538
3.3		–	1590
3.4		–	1609
4.1	4 months	–	–
4.2		–	–
4.3		–	–
6.1	6 months	–	–
6.2		–	–

Table 3.2: Results for CPLEX MIP within three hours of computing time and for CPLEX LP Barrier solver.

Chapter 4

A fast matheuristic for the medium-term UCP

In this chapter we present a matheuristic algorithm for model (3.1)-(3.21) which uses aggregation, continuous relaxations and spatial decomposition to yield high quality approximated solutions in affordable time.

4.1 The Commit&Dispatch algorithm

The Commit&Dispatch algorithm is initialised with the optimal solution of a relaxation of model (3.1)–(3.21). The relaxation is obtained by aggregation and it is described in subsection 4.1.1. Its optimal solution is exploited by the matheuristic described in subsection 4.1.2; it alternates commit and dispatch optimisation up to convergence.

4.1.1 Aggregated Continuous Relaxations

As discussed in section 3.2.1, for each zone $z \in Z$, plants in each group $g \in G_z$ have identical marginal production costs. For each period $t \in T$ the overall operating cost of a group g is represented by a piece-wise linear function

$$f_{tzg}(x_{tzg}, \mathbf{y}_{tzg}) = c_{tzg}x_{tzg} + \sum_{m \in M_{zg}} e_m y_{tzgm}$$

where x_{tzg} is the overall production level of the group and $\mathbf{y}_{tzg} = (y_{tzgm})_{m \in M_{zg}}$ is the number of active plants in each subgroup $m \in M_{zg}$ such that they satisfy equations 3.2-3.12.

Depending on which plants are active (i.e., the *activation pattern*), the same amount of production can have a different cost. Assuming to select the most convenient activation pattern for each production level, the resulting cost function f_{tzg} is piece-wise linear: all the segments have the same slope and discontinuities occur when the optimum activation pattern changes. Owing to the constraints on minimum allowed production, the function is not necessarily monotone. An example is shown in Figure 4.1.

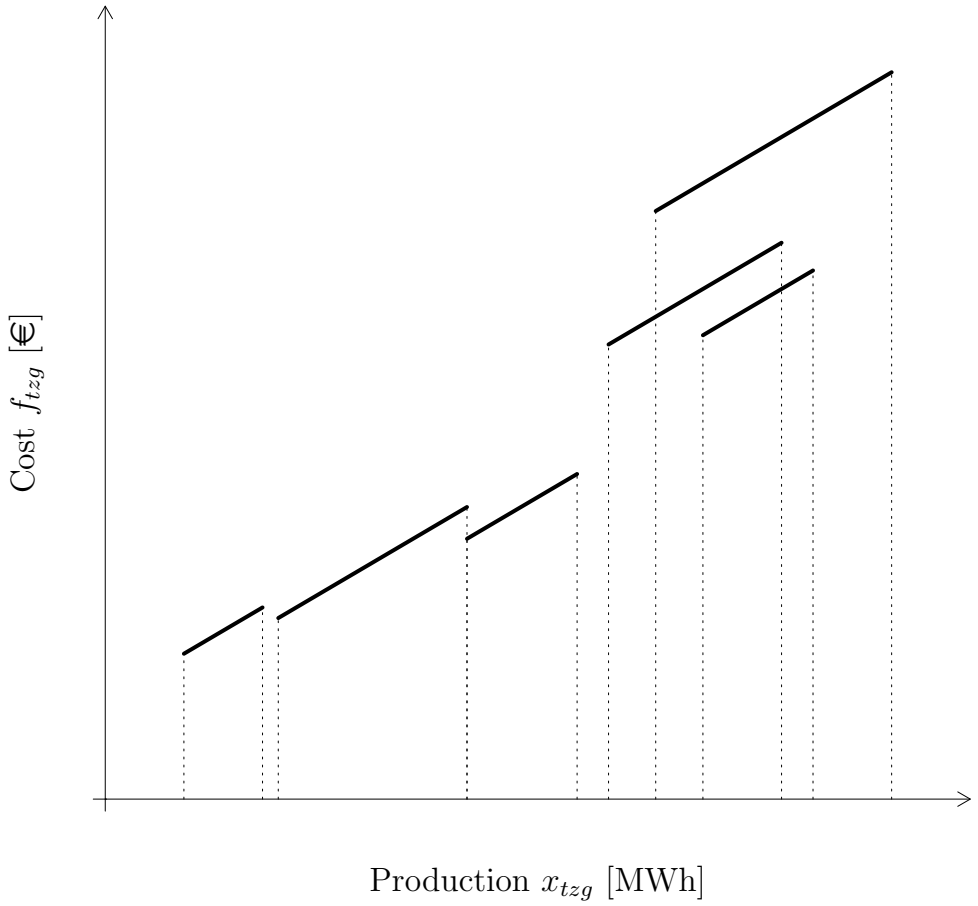


Figure 4.1: Piece-wise linear cost function f_{tzg} as a parametric function of production x_{tzg} with parameters $\mathbf{y}_{tzg}$. The plot shows different cost curves for different activation patterns of the group $g \in G_z$.

To get rid of the dependency on the y variables representing the activation pattern, we underestimate f_{tzg} with a linear function of x_{tzg}, yielding a lower bound to the cost. This gives an aggregated continuous approximation of the group cost function.

We devised two lower bounds for the group cost functions. The first, called Aggregated Continuous Approximation (ACA) is obtained as follow.

Proposition 2. *Let* $\tilde{e}_{tzg} = \min_{m \in M_{zg}} \{e_{tzgm}\}$ *be the smallest fixed cost of plants in group* g *and* $\tilde{P}_{zg} = \sum_{m \in M_{zg}} P_{zgm}|K_{zgm}| + \sum_{m \in M^D_{zg}} P^{\Delta}|K^D_{zgm}|$ *be the overall production capacity of group* g*. For each single plant* m *in group* g *producing at level* x_{tzgm}*, a lower bound to its production cost is given by* $\tilde{c}_{tzgm}x_{tzgm}$ *with*

$$\tilde{c}_{tzgm} = c_{tzg} + \frac{\tilde{e}_{tzgm}}{\tilde{P}_{zgm}}.$$

Proof. For $x_{tzg} = 0$ both the original cost and the lower bound are zero for any value of $\tilde{c}_{tzg}$. Consider then $x_{tzg} \in (0, \tilde{P}_{zgm}]$. There must be at least one active plant in the group, therefore $\sum_{m \in M_{zg}} y_{tzgm} \geq 1$, from which follows

$$c_{tzg}x_{tzg} + \sum_{m \in M_{zg}} e_m y_{tzgm} \geq c_{tzg}x_{tzg} + \tilde{e}_{tzg} \quad \forall x_{tzgm} > 0$$

which can be rewritten has

$$c_{tzg}x_{tzg} + \sum_{m \in M_{zg}} e_m y_{tzgm} \geq \left(c_{tzg} + \frac{\tilde{e}_{tzg}}{x_{tzgm}} \right) x_{tzgm}$$

Then, to have a valid lower bound we impose that

$$\tilde{c}_{tzg} = \min_{x_{tzg} \in (0, \tilde{P}_{zgm}]} \{c_{tzg} + \frac{\tilde{e}_{tzg}}{x_{tzgm}}\} = c_{tzg} + \frac{\tilde{e}_{tzg}}{\tilde{P}_{tzg}}$$

□

We then propose a tighter lower bound, called Improved Aggregated Continuous Approximation (IACA), observing that the slope of $\tilde{f}_{tzg}(x)$ in Figure 4.2 can be increased while still obtaining a valid lower bound to $f_{tzg}(x)$.

Proposition 3. *For each single plant* m *in group* g *producing at level* x_{tzgm}*, a lower bound to its production cost is given by* $\hat{c}_{tzgm}x_{tzgm}$ *with*

$$\hat{c}_{tzgm} = \frac{e_{tzgm} + c_{tzg}P_{zgm}}{P_{zgm}}.$$

Proof. Since $e_{tzgm} \geq 0$, then $c_{tzg} \leq \hat{c}_{tzgm}$. If $x_{tzgm} = P_{zgm}$, then $\hat{c}_{tzgm}x_{tzgm}$ is equal to the production cost of P_{zgm}, i.e. $e_{tzgm} + c_{tzg}P_{zgm}$; we indicate it by C_{max}. If $x_{tzgm} < P_{zgm}$, then define $\delta = P_{zgm} - x_{tzgm}$ and observe that the actual production

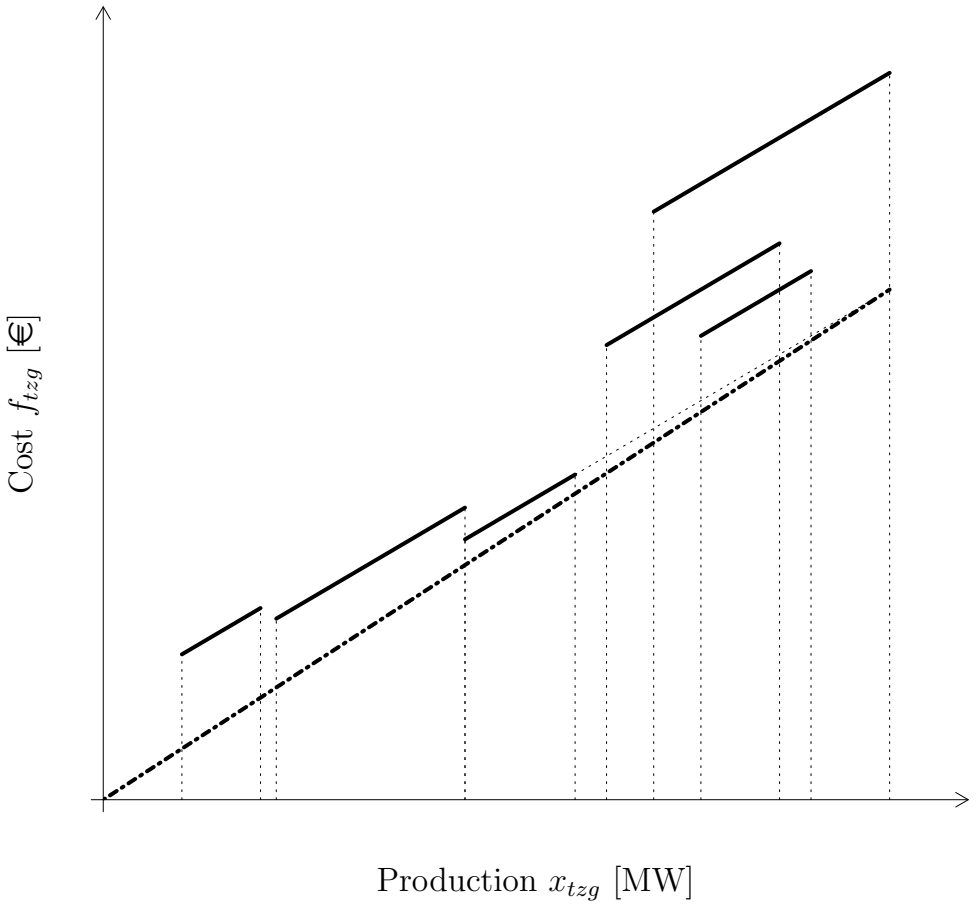


Figure 4.2: The ACA lower bound for f_{tzg} (dot-dashed line).

cost is $C_{max} - c_{tzg}\delta$, while the lower bound is $C_{max} - \hat{c}_{tzgm}\delta$ and the former is larger than the latter because $c_{tzg} \leq \hat{c}_{tzgm}$ and $\delta \geq 0$. □

For plants with double-shaft technology the same proof holds, using P^D_{zgm} instead of P_{zgm}.

Proposition 4. *For each subset Q of plants in group g yielding an overall production x_{tzg}, a lower bound to its production cost is given by $\hat{c}_{tzgQ}x_{tzg}$ with*

$$\hat{c}_{tzgQ} = c_{tzg} + \min_{m \in Q}\{\frac{e_{tzgm}}{P_{zgm}}\}.$$

Proof. We indicate by $C(x_{tzg})$ the production cost of the amount x_{tzg} and by x_{tzgm}

the amount of production of each plant $m \in Q$. Then we have

$$C(x_{tzg}) = \sum_{m \in Q} (e_{tzgm} + c_{tzg} x_{tzgm})$$

and

$$x_{tzg} = \sum_{m \in Q} x_{tzgm}.$$

The following equalities hold:

$$\frac{C(x_{tzg})}{x_{tzg}} = \frac{\sum_{m \in Q} (e_{tzgm} + c_{tzg} x_{tzgm})}{\sum_{m \in Q} x_{tzgm}} = \sum_{m \in Q} \frac{(e_{tzgm} + c_{tzg} x_{tzgm})}{x_{tzgm}} \frac{x_{tzgm}}{\sum_{m \in Q} x_{tzgm}}.$$

This is a convex combination of non-negative values $\frac{(e_{tzgm} + c_{tzg} x_{tzgm})}{x_{tzgm}}$, each one weighted by a coefficient $\frac{x_{tzgm}}{\sum_{m \in Q} x_{tzgm}}$ whose value is between 0 and 1. The value of such a convex combination is obviously larger than or equal to the smallest of the values in the combination, i.e.

$$\sum_{m \in Q} \frac{(e_{tzgm} + c_{tzg} x_{tzgm})}{x_{tzgm}} \frac{x_{tzgm}}{\sum_{m \in Q} x_{tzgm}} \geq \min_{m \in Q} \{ \frac{(e_{tzgm} + c_{tzg} x_{tzgm})}{x_{tzgm}} \}.$$

The minimum with respect to the production x_{tzgm} is attained when $x_{tzgm} = P_{zgm}$. Hence for any production level x_{tzg}

$$\frac{C(x_{tzg})}{x_{tzg}} \geq c_{tzg} + \min_{m \in Q} \{ \frac{e_{tzgm}}{P_{zgm}} \}.$$

□

Therefore, a lower bound to the production cost of all subsets of plants Q is given by $\hat{c}_{tzg} x_{tzg}$ with

$$\hat{c}_{tzg} = c_{tzg} + \min_{Q} \min_{m \in Q} \{ \frac{e_{tzgm}}{P_{zgm}} \} = c_{tzg} + \min_{m \in M_{zg}} \{ \frac{e_{tzgm}}{P_{zgm}} \},$$

that is, $\hat{c}_{tzg}$ can be computed by checking only a linear number of terms.

Relying upon the IACA bound illustrated above, we define an improved aggregated continuous relaxation (IACR), given by the following linear programming model:

$$\min \quad \hat{\phi} = \sum_{\substack{t \in T, z \in Z, \\ g \in G_z}} \hat{c}_{tzg} x_{tzg} + \sum_{t \in T, z \in Z} ENP_{tz} E_t + \sum_{t \in T, z \in Z} EIE_{tz} F_t \tag{4.1}$$

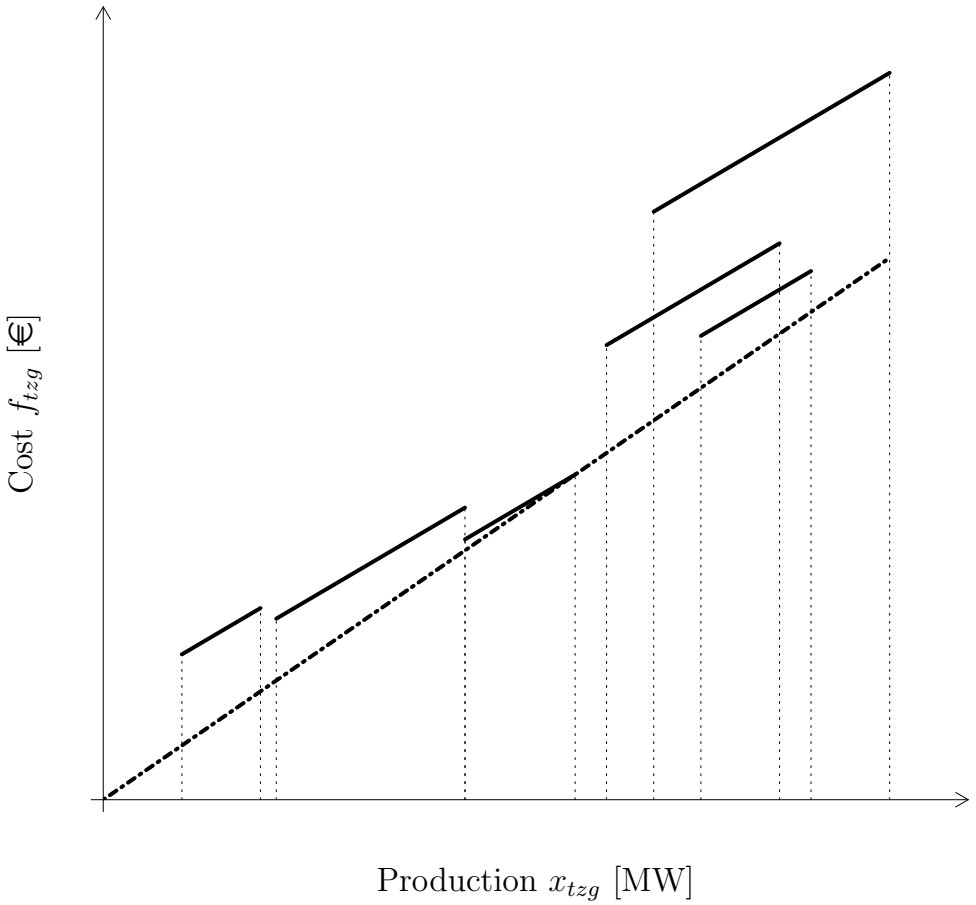


Figure 4.3: The IACA lower bound of f_{tzg} (dot-dashed line).

$$\text{s.t. } 0 \leq x_{tzg} \leq \tilde{P}_{zg} \ \forall t \in T, z \in Z, g \in G_z \tag{4.2}$$

$$o_{1zh} = q_{zh} \ \forall z \in Z, h \in H_z \tag{4.3}$$

$$o_{(|T|+1)zh} = Q_{zh} \ \forall z \in Z, h \in H_z \tag{4.4}$$

$$o_{tzh} + n_{th} + \beta_h \cdot m_{tzh} = o_{(t+1)zh} + s_{tzh} + \alpha_{zh} l_{tzh} \ \forall t \in T, z \in Z, h \in H_z \tag{4.5}$$

$$\sum_{h \in H_z} l_{tzh} + \sum_{g \in G_z} x_{tzg} + \sum_{(i,z) \in A} w_{tiz} + \sum_{z \in Y} ENP_{tz} \geq d_{tz} + \sum_{h \in H_z} m_{tzh} + \sum_{(z,j) \in A} w_{tzj} + \sum_{z \in Y} EIE_{tz} \quad \forall t \in T, z \in Z \tag{4.6}$$

$$w_{tij} \in [b_{ij}^-, b_{ij}^+] \ \forall t \in T, (i,j) \in A \tag{4.7}$$

$$s_{tzh} \in [0, f_{zh}], o_{tzh} \in [0, V_{zh}], l_{tzh} \in [p_{zh}, P_{zh}], m_{tzh} \in [0, P_{zh}^{\beta}] \ \forall t \in T, z \in Z, h \in H_z \tag{4.8}$$

$$ENP_{tz} \geq 0, EIE_{tz} \geq 0 \ \forall t \in T, z \in Z \tag{4.9}$$

Equivalently, an Aggregated Continuous Relaxation (ACR) can be obtained by replacing the coefficients $\hat{c}_{tzg}$ of the IACA formulation in 4.1 with ACA coefficients $\tilde{c}_{tzg}$ $\forall t \in T, z \in Z, g \in G_z$.

ACR and IACR are indeed relaxations of the original formulation because they are obtained by aggregation of constraints, replacement of the cost function with a lower bounding function and by the relaxation of integrality constraints. Both models can be shown to provide weaker bounds than CR. On the other hand the aggregated continuous relaxations can be solved more efficiently than CR as they are smaller and can also be formulated as network flow problems, for which well-known exact polynomial time algorithms can be used.

4.1.2 Commit&Dispatch

Besides giving a valid lower bound, IACR solutions provide a tentative dispatch plan. Therefore, we use them as a starting point for an alternating Commit&Dispatch (CD) matheuristic that first searches for activation patterns of minimum cost to satisfy the estimated production levels (*Commit*) and then optimises the production levels for fixed activation patterns (*Dispatch*). Commit and Dispatch phases are iterated until no more improvements are obtained in either phase. Hereafter we describe the details of each phase.

Commit. Let the values

$$\tilde{x}_{tzg} \in \left[0, \sum_{m \in M_{zg}} P_{zgm}|K_{zgm}| + \sum_{m \in M^D_{zg}} P^{\Delta}_{zgm}|K^D_{zgm}|\right]$$

for each $t \in T, z \in Z, g \in G_z$ describe a feasible production plan. A minimum cost activation plan complying with such production plan is computed by solving the following integer linear program:

$$\min \phi_C = \sum_{\substack{t \in T, z \in Z, \\ g \in G_z, m \in M_{zg}}} e_{tzgm} y_{tzgm} \tag{4.10}$$

$$\text{s.t. } \tilde{x}_{tzg} \leq \sum_{m \in M_{zg}} P_{zgm} y_{tzgm} + \sum_{m \in M^D_{zg}} P^{\Delta}_{zgm} y^D_{tzgm} \quad \forall t \in T, z \in Z, g \in G_z \tag{4.11}$$

$$(3.4) - (3.12), (3.17), (3.18) \tag{4.12}$$

Model equations (4.10)-(4.12) decomposes into independent subproblems for each

$t \in T, z \in Z, g \in G_z$; these subproblems are solvable in pseudo-polynomial time and experimentally they are optimised very quickly by ILP solvers.

Dispatch. Let $(\tilde{y}_{tzgm}, \tilde{y}^D_{tzgm}, \tilde{v}^+_{tzgm}, \tilde{v}^{+D}_{tzgm}, \tilde{v}^-_{tzgm}, \tilde{v}^{-D}_{tzgm})$ for each $z \in Z, g \in G_z, m \in M_{zg}$ be a feasible activation plan of thermal plants, satisfying constraints (3.4) – (3.12), (3.17) – (3.18).

A minimum cost feasible production plan complying with such an activation plan is computed by optimising the dispatching problem, modelled as a linear program:

$$\min \phi_D = \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} c_{tzg}x_{tzg} + \sum_{t\in T, z\in Z} ENP_{tz}E_t \sum_{t\in T, z\in Z} EIE_{tz}F_t + \sum_{\substack{z\in Z, g\in G_z,\\ m\in M_{zg}}} e_{tzgm}\tilde{y}_{tzgm} \tag{4.13}$$

$$\text{s.t. } x_{tzg} \leq \sum_{m\in M_{zg}} P_{zgm}\tilde{y}_{tzgm} + \sum_{m\in M^D_{zg}} P^{\Delta}\tilde{y}^D_{tzgm} \ \forall t \in T, z \in Z, g \in G_z \tag{4.14}$$

$$x_{tzg} \geq \sum_{m\in M_{zg}} p_{zgm}\tilde{y}_{tzgm} + \sum_{m\in M^D_{zg}} p^{\Delta}_{zgm}\tilde{y}^D_{tzgm} \ \forall t \in T, z \in Z, g \in G_z \tag{4.15}$$

$$(3.13) - (3.16), (3.19) - (3.21) \tag{4.16}$$

This model is comparable in complexity to IACR (4.1)-(4.9).

Proposition 5 (Monotonicity)**.** *The objective function value computed at each iteration of the Commit&Dispatch algorithm is monotone non-increasing.*

This property directly follows from the following observation: every time either phase of the algorithm is executed its input and its output are respectively feasible and optimal with respect to a same set of constraints.

Compared to well-known classical Lagrangian relaxation heuristics, based on an analogous decoupling between plant production and commitment decisions (Redondo and Conejo, 1999) , which where previously studied in Taverna (2011) as well, our heuristic guarantees, thanks to properties of the commit model (4.10)-(4.12), to satisfy demand via constraints (4.12), ensuring $ENP_{zt} = 0 \ \forall z \in Z, t \in T$, since the first iteration. Lagrangean methods, instead, can yield dual solutions with positive, albeit relatively small, amounts of energy not provided, which, due to the heavy penalisation via coefficients E_t, cause the Lagrangean dual solution to oscillate and converge slowly to the optimum. Furthermore, the commit phase minimises the

amount of periods in which power plants are active, as it always incurs a positive cost, thus limiting the amount of energy produced in the dispatching phase. This would then allow to minimise also the value of energy in excess (EIE_{zt}) in the dispatching phase.

4.2 Computational results

In this section we report the results for the aggregated continuous relaxations and the Commit&Dispatch heuristic on the instances from section 3.3. The tests were run on the same hardware.

4.2.1 Solving continuous relaxations

In table 4.1 we compare the three different continuous relaxations proposed for problem (3.1)-(3.21): the continuous relaxation of the original formulation (CR) and the two aggregated continuous relaxations ACR and IACR in equations (4.1)-(4.9). For CR we report the computing time. For ACR and IACR we report the computing time and the relative gap with CR. Missing values are due to out-of-memory errors.

All the algorithms are implemented with CPLEX 12.4 and AMPL. We used the barrier algorithm with no crossover phase for all the three models as in earlier experiments it proved to be the most efficient solution method. Instances are obtained from a scenario for the Italian system at 2011.

We remark that, due to differences in implementation and data, these results can hardly be compared with results in 3.2. In particular, the AMPL version passes to CPLEX a presolved version of the same model which results in a smaller CPLEX model object. The solver can then handle bigger instances than the C++ CPLEX Concert implementation, which is passed raw and larger formulations of a similar model.

It can be seen that both ACR and IACR can be 50 times more efficient to solve than CR while providing comparable lower bounds. In the worst case, the ACR and IACR solutions are 3.8% and 0.4% lower than CR respectively. Furthermore, the aggregated continuous relaxations can be solved for the largest instances of 6 and 12 months, while CPLEX encounters out-of-memory errors on the same instances for the CR formulation.

Id	Size	CR	ACR		IACR	
		Time [s]	Time [s]	Gap % with CR	Time [s]	Gap % with CR
1.1	1 month	244	5	2.9%	6	0.1%
1.2		164	5	2.7%	6	0.1%
1.3		149	5	2.8%	5	0.4%
1.4		152	5	3.0%	5	0.3%
1.5		163	5	2.8%	6	0.3%
1.6		158	5	2.9%	6	0.3%
1.7		171	5	2.8%	6	0.1%
1.8		149	6	3.8%	6	0.3%
1.9		167	6	2.9%	6	0.2%
1.10		156	6	2.8%	5	0.3%
1.11		165	5	2.7%	6	0.1%
1.12		167	5	2.8%	6	0.2%
2.1	2 months	755	10	2.7%	12	0.1%
2.2		583	11	2.8%	11	0.4%
2.3		576	11	2.7%	11	0.3%
2.4		595	11	3.1%	13	0.2%
2.5		605	10	2.8%	13	0.2%
2.6		612	11	2.6%	11	0.2%
3.1	3 months	933	16	2.7%	19	0.2%
3.2		1048	17	2.7%	18	0.2%
3.3		900	18	3.0%	20	0.2%
3.4		889	16	2.6%	19	0.2%
4.1	4 months	1380	27	2.8%	27	0.2%
4.2		1391	25	3.0%	26	0.3%
4.3		1310	22	2.8%	26	0.2%
6.1	6 months	-	51	-	54	-
6.2		-	40	-	42	-
12.1	12 months	-	112	-	118	-

Table 4.1: Continuous relaxations for UCP

4.2.2 Commit&Dispatch

We implemented both IACR and the Commit&Dispatch algorithm in C++ using IBM ILOG CPLEX 12.4 to optimise the mathematical models.

We initialise the Commit&Dispatch algorithm with the IACR relaxation. The Commit&Dispatch scheme is then run once. Earlier experiments showed further iterations yielded negligible improvement over the primal bound. We used the CPLEX barrier algorithm to solve the IACR and the dispatching model.

In Table 4.2 we report the experimental results. For each instance we report the value of the IACR lower bound, the value of the IACR+CD heuristic, the gap and the computing time for the lower bound and the total time for the heuristic.

Id	Size	Gap %	IACR Time [s]	IACR+CD Time [s]
1.1	1 month	2.6 %	4.3	57.5
1.2		3.0 %	4.3	48.1
1.3		3.4 %	3.3	47.2
1.4		3.4 %	3.9	58.7
1.5		2.7 %	3.3	43.2
1.6		2.9 %	4.0	44.6
1.7		2.2 %	3.6	48.8
1.8		3.3 %	3.9	55.9
1.9		2.3 %	3.6	52.4
1.10		3.7 %	3.9	44.5
1.11		3.7 %	3.4	43.1
1.12		3.2 %	4.0	48.9
2.1	2 months	2.9 %	9.4	91.7
2.2		3.6 %	8.5	115.8
2.3		3.0 %	8.2	91.7
2.4		2.6 %	8.9	94.4
2.5		3.0 %	8.6	108.0
2.6		3.2 %	8.4	90.7
3.1	3 months	3.2 %	16.1	163.6
3.2		3.2 %	15.8	193.4
3.3		2.5 %	18.9	197.5
3.4		3.4 %	14.5	150.0
4.1	4 months	3.3 %	24.2	225.7
4.2		2.8 %	22.1	243.5
4.3		3.0 %	41.9	290.9
6.1	6 months	3.3 %	57.9	890.6
6.2		3.2 %	43.4	906.3

Table 4.2: Results for IACR+CD

The IACR+CD shows indeed to be a practical tool for solving our UCP, always

producing duality gaps around 3% in minutes of computation. IACR requires little time to be solved for all the instances. The MIP solver was able to solve most of the Commit subproblems (4.10)-(4.12) via presolving or by performing a few LP iterations at the root node.

Chapter 5

A Column Generation algorithm for the medium-term UCP

Column generation has been applied to several optimisation problems to derive both high-quality dual and primal bounds (Lübbecke and Desrosiers, 2005). In this chapter we present a *Column Generation* algorithm to compute high-quality primal solutions for the long-term UCP.

5.1 The Column Generation algorithm

From model (3.1)-(3.21) we define

$$S_{zgm} = \{(\tilde{y}_{tzgm}, \tilde{y}^D_{tzgm}, \tilde{v}^+_{tzgm}, \tilde{v}^{+D}_{tzgm}, \tilde{v}^-_{tzgm}, \tilde{v}^{-D}_{tzgm}) | (3.4) - (3.12), (3.17) - (3.18)\}$$

to be the set of all feasible activation patterns for each $z \in Z, g \in G_z, m \in M_{zg}$. For each pattern $s_u \in S_{zgm}$, let y^u_{tzgm} and $y^{u^D}_{tzgm}$ be the integer coefficients representing the number of active plants in normal and double-shaft mode respectively, and $\gamma^u_{zgm} \in \{0, 1\}$ be the binary variable such that $\gamma^u_{zgm} = 1$ if and only if the pattern is selected.

We consider the following extended formulation of the UCP:

$$\begin{aligned} \min \quad \phi_{Ext} = & \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} c_{tzg} x_{tzg} + \sum_{\substack{z\in Z, g\in G_z,\\ m\in M_{zg}, u\in S_{zgm}}} \gamma^u_{zgm} \Big(\sum_{t\in T} \tilde{y}^u_{tzgm} e_{tzgm}\Big) \\ & + \sum_{t\in T, z\in Z} ENP_{tz} E_t + \sum_{t\in T, z\in Z} EIE_{tz} F_t \end{aligned} \tag{5.1}$$

$$\text{s.t. } x_{tzg} \geq \sum_{\substack{m \in M_{zg}, \\ u \in S_{zgm}}} (\tilde{y}^u_{tzgm} p_{zgm} + \tilde{y}^{u^D}_{tzgm} p^{\Delta}_{zgm}) \gamma^u_{zgm} \ \forall t \in T, z \in Z, g \in G_z \tag{5.2}$$

$$x_{tzg} \leq \sum_{\substack{m \in M_{zg}, \\ u \in S_{zgm}}} (\tilde{y}^u_{tzgm} P_{zgm} + \tilde{y}^{u^D}_{tzgm} P^{\Delta}_{zgm}) \gamma^u_{zgm} \ \forall t \in T, z \in Z, g \in G_z \tag{5.3}$$

$$\sum_{u \in S_{zgm}} \gamma^u_{zgm} = 1 \ \forall z \in Z, g \in G_z, m \in M_{zg} \tag{5.4}$$

$$\gamma^u_{zgm} \in \{0, 1\} \ \forall z \in Z, g \in G_z, m \in M_{zg}, u \in S_{zgm} \tag{5.5}$$

$$(3.13) - (3.21) \tag{5.6}$$

Constraints (5.2) and (5.3) are the reformulated counterparts of constraints (3.2) and (3.3), respectively. Constraints (5.4) enforce that a single pattern is selected for each $z \in Z, g \in G_z, m \in M_{zg}$.

In this extended model the region defined by constraints (3.4)-(3.12), (3.17)-(3.18) of model (3.1)-(3.21) is replaced by the set of integer points lying in its convex hull. Such a region is, in turn, the intersection of UCP feasibility regions, one for each zone, group and sub-group. UCPs for each subgroup are purely integer commitment problems consisting of $k \geq 1$ identical and independent power plants being scheduled at the same time. This UCP variant, called k-UCP in the remainder, has thus a special structure compared to a general UCP, which can be exploited to achieve the following result:

Proposition 6. *k-UCP can be reduced in polynomial time to the shortest-path problem.*

We refer to section 5.1.1 for the proof. From that it follows the integrality property holds for the k-UCP formulation given by (3.4)-(3.12), (3.17)-(3.18) and the continuous relaxation of the extended formulation yields the same bound as CR. However, it can be computed more efficiently, by exploiting column generation techniques.

In our method we start with a restricted master model where each set S_{zgm} is replaced by a subset $\overline{S}_{zgm}$ containing only patterns generated by heuristics. Then we iteratively solve the restricted master model, we obtain a vector of dual variables and we search for columns of negative reduced cost by solving the following pricing problem, for each $z \in Z$, $g \in G_z$ and $m \in M_{zg}$:

$$\min \pi^u_{zgm} = \sum_{t\in T}(e_{tzgm} + p_{zgm}\lambda_{tzg} - P_{zgm}\mu_{tzg})y^u_{tzg} + \sum_{t\in T}(p^{\Delta}_{zgm}\lambda_{tzg} - P^{\Delta}_{zgm}\mu_{tzg})y^{u^D}_{tzg} - \eta_{zgm} \quad (5.7)$$

$$\text{s.t. } (y^u_{tzgm}, y^{u^D}_{tzgm}, v^{+u}_{tzgm}, v^{+D^u}_{tzgm}, v^{-u}_{tzgm}, v^{-D^u}_{tzgm})_{t\in T} \in S_{zgm} \quad (5.8)$$

where $\lambda_{tzg} \geq 0$, $\mu_{tzg} \geq 0$ and η_{zgm} are the dual variables associated with constraints (5.2), (5.3) and (5.4), respectively. If any column of negative reduced cost is found, then the corresponding pattern is added to $\overline{S}_{zgm}$, and the process is iterated; otherwise, the optimal solution of the restricted model is optimal also for the full model, and therefore the process is halted.

5.1.1 Solving the pricing problem

The pricing problem decomposes into independent sub-problems (5.7)-(5.8) for each zone, group and subgroup. In order to solve each sub-problem to proven optimality, we extend a dynamic programming algorithm described in (Frangioni, 2006), originally designed for the single-unit UCP. We extend the algorithm to the problem with several identical units. It is worth noting that the pricing problem does concern neither the continuous variables x nor demand satisfaction constraints. Its only purpose is to select promising activation patterns, i.e. activation patterns with negative reduced cost, described by the binary variables y.

Proposition 7. *For each subproblem and zone there exists an optimal integer solution of the pricing problem (5.7)-(5.8) in which, at each time $t \in T$, all plants are simultaneously either off or on. Furthermore it is not optimal for double-shaft plants to work in single-shaft mode.*

Proof. The second part of the proposition is trivially proved: since the coefficients associated with double-shaft variables are always less than or equal to those associated with single-shaft variables, it is always optimal to employ double-shaft mode, whenever double-shaft plants are active. The first part of the proposition is proved by contradiction. Assume that a feasible solution exists, in which each plant follows a potentially different activation and deactivation pattern, and let y^* be the most profitable pattern among those. A non-worse feasible solution can be found

by using pattern y^* for all plants. This argument applies, in particular, to optimal solutions. □

Owing to the proposition above, we restrict our search to the solutions with the particular structure mentioned in the proposition. Let ε_t be the partial cost for activating all plants at time t:

$$\varepsilon_t = \begin{cases} |K_{zgm}| \cdot e_{tzgm} + (p_{zgm} + p^{\Delta}_{zgm}) \cdot \lambda_{tzg} - (P_{zgm} + P^{\Delta}_{zgm}) \cdot \mu_{tzg} & \text{if } m \in M^D_{zg} \\ |K_{zgm}| \cdot e_{tzgm} + p_{zgm} \cdot \lambda_{tzg} - P_{zgm} \cdot \mu_{tzg} & \text{otherwise} \end{cases}$$

For each zone $z \in Z$, group $g \in G_z$ and subgroup $m \in M_{zg}$, we define an arc-weighted directed graph $\Gamma_{zgm} = (V_{zgm}, A_{zgm})$ with weights on arcs. V_{zgm} includes one layer of vertices for each $t \in T$. Each layer contains two vertices v^{on}_t and v^{off}_t, corresponding to the all-on and all-off states of plants, respectively. It also includes an initial node σ and a final node ω. A_{zgm} includes two arcs between each pair of vertices (v^{on}_{t-1}, v^{on}_t) with weight ε_t and each pair $(v^{off}_{t-1}, v^{off}_t)$ with weight 0, for each $t \in T, t \neq 1$. These arcs represent the decision of keeping the plants in their state from time $t-1$ to time t. Let $E(t) = \min(|T|, t)\ \forall t \in \mathbb{N}$. We also include an arc between each pair of vertices $(v^{on}_t, v^{off}_{E(t+\tau^{off})})$ with weight 0, representing the decision of turning off plants after time t, and keeping them off for τ^{off} time units, in order to fulfil the flexibility constraints. Similarly, we include an arc between each pair of vertices $(v^{off}_t, v^{on}_{E(t+\tau^{on})})$ with weight $\sum_{t'=t+1}^{t'=E(t+\tau^{on})} \varepsilon_{t'}$, representing the decision of turning on plants after time t, and keeping them on for τ^{on} time units. Finally, we connect σ to both v^{off}_1 and v^{on}_1, and both $v^{off}_{|T|}$ and $v^{on}_{|T|}$ to ω with arcs of weight 0. The structure of the digraph Γ_{zgm} is shown in Figure 5.1.

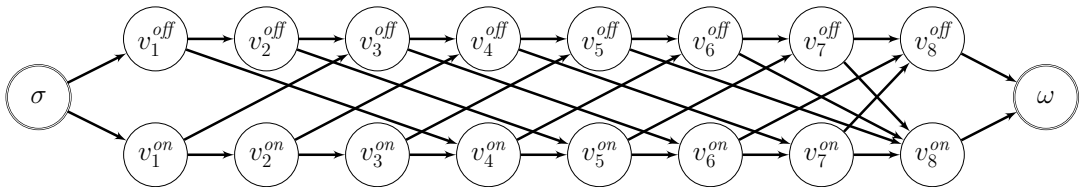


Figure 5.1: The digraph Γ_{zgm} with $\tau^{on} = 3$ and $\tau^{off} = 2$ on 8 periods.

Observation 3. *Any $\sigma - \omega$ path in Γ_{zgm} represents a feasible sequence of activation and deactivation patterns of plants. In particular, the minimum weight path represents an optimal solution to the pricing problem.*

Therefore, an optimal pricing solution can be found by solving a shortest path

problem for each $z \in Z, g \in G_z$ and $m \in M_{zg}$. Since Γ_{zgm} is acyclic, this can be done in $O(|T|)$ time by dynamic programming.

We note the graph construction procedure assumes the power plant has to wait its respective minimum up/down time at period 1 before switching to a new state. In short-term models for operational support the initial state of power plants at period 1 is known in advance and is thus fixed. For our model the initial state has to be assumed as part of the simulated scenario.

5.1.2 Solving the master problem

Preliminary experiments revealed that, given the efficiency of the pricing algorithm, the bottleneck of the method is the solution of the master problem. This is a large scale LP and suffers for numerical instability and degeneracy which occur *in the LP solver* during the solution of the master problem *at each iteration* of the CG algorithm. These problems differ from the "dual instability" problems encountered in literature (Ben Amor et al., 2009; Briant et al., 2008; Gondzio et al., 2013) as they do not cause either the dual bound or the value of dual variables to oscillate significantly; instead they consist of numerical issues during the solution of the master problem that cause the LP solver to either "stall" (IBM, 2011), i.e. have the bounds not improving for several iterations, or fail to converge completely and halt. We refer the reader to sec.7.1 for a deeper analysis of the issue.

In order to overcome this difficulty we devised a hybrid method for computing optimal dual solutions. The method quickly computes additional columns at each iteration in order to provide a better description of the master's polytope, in the hope of improving its numerical properties.

We consider the following Lagrangian relaxation of the extended formulation (5.1)-(5.6):

$$\begin{aligned}\min \quad \mathcal{L} = & \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} c_{tzg}x_{tzg} + \sum_{t\in T, z\in Z} ENP_{tz}E_t + \sum_{t\in T, z\in Z} EIE_{tz}F_t \\ & + \sum_{\substack{z\in Z, g\in G_z,\\ m\in M_{zg}, u\in S_{zgm}}} \gamma^u_{zgm}\Big(\sum_{t\in T} \tilde{y}^u_{tzgm}e_{tzgm}\Big) \\ & - \sum_{t\in T, z\in Z, g\in G_z} \lambda_{tzg}\Big(x_{tzg} - \sum_{\substack{m\in M_{zg},\\ u\in S_{zgm}}} (\tilde{y}^u_{tzgm}p_{zgm} + \tilde{y}^{u^D}_{tzgm}p^{\Delta}_{zgm})\gamma^u_{zgm}\Big)\end{aligned}$$

$$- \sum_{t\in T, z\in Z, g\in G_z} \mu_{tzg}(-x_{tzg} + \sum_{\substack{m\in M_{zg},\\ u\in S_{zgm}}} (\tilde{y}^u_{tzgm} P_{zgm} + \tilde{y}^{u^D}_{tzgm} P^{\Delta}_{zgm})\gamma^u_{zgm}) \quad (5.9)$$

$$\text{s.t.} \sum_{u\in S_{zgm}} \gamma^u_{zgm} = 1 \ \forall z \in Z, g \in G_z, m \in M_{zg} \quad (5.10)$$

$$\gamma^u_{zgm} \in \{0,1\} \ \forall z \in Z, g \in G_z, m \in M_{zg}, u \in S_{zgm} \quad (5.11)$$

$$(3.13) - (3.21) \quad (5.12)$$

Constraints (5.2) and (5.3) are dualised; their violation is penalised in the objective function by multipliers λ and μ. When the values of multipliers are fixed, problem (5.9)-(5.12) decomposes in two parts. The first one ($\mathcal{L}^x$) includes only x_{tzg} variables and constraints (3.13)-(3.21):

$$\min \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} (c_{tzg} - \lambda_{tzg} + \mu_{tzg})x_{tzg} + \sum_{t\in T, z\in Z} ENP_{tz}E_t + \sum_{t\in T, z\in Z} EIE_{tz}F_t \quad (5.13)$$

$$\text{s.t. } (3.13) - (3.21) \quad (5.14)$$

This turns out to be a LP subproblem involving a polynomial number of variables.

The second one $\mathcal{L}^y$ includes γ_u variables and constraints (5.10) and (5.11):

$$\min \sum_{\substack{z\in Z, g\in G_z,\\ m\in M_{zg}, u\in S_{zgm}}} \gamma^u_{zgm} \cdot \varpi^u_{zgm} \quad (5.15)$$

$$\text{s.t.} \sum_{u\in S_{zgm}} \gamma^u_{zgm} = 1 \ \forall z \in Z, g \in G_z, m \in M_{zg} \quad (5.16)$$

$$\gamma^u_{zgm} \in \{0,1\} \ \forall z \in Z, g \in G_z, m \in M_{zg}, u \in S_{zgm} \quad (5.17)$$

where

$$\begin{aligned} \varpi^u_{zgm} = (&\sum_{t\in T} \tilde{y}^u_{tzgm} e_{tzgm}) + \\ &+ \sum_{t\in T} (\tilde{y}^u_{tzgm} p_{zgm} + \tilde{y}^{u^D}_{tzgm} p^{\Delta}_{zgm})\lambda_{zgm} \\ &- \sum_{t\in T} (\tilde{y}^u_{tzgm} P_{zgm} + \tilde{y}^{u^D}_{tzgm} P^{\Delta}_{zgm})\mu_{zgm} \end{aligned}$$

is a partial cost for selecting pattern $u \in S_{zgm}$. This is an ILP problem including

an exponential number of variables, but it further decomposes into independent subproblems $\mathcal{L}^y_{zgm}$ for each $z \in Z, g \in G_z$ and $m \in M_{zg}$, each requiring (due to constraints (5.10)) to select a single pattern of minimum pseudo-cost. These are actually the same subproblems that need to be solved by the pricing algorithm.

Observation 4. *The dual bound that can be obtained by solving the Lagrangian dual problem*

$$\max_{\lambda,\mu} \mathcal{L}$$

to optimality is equivalent of that given by the LP relaxation of (5.1)-(5.6).

The integrality property holds for the second part of the decomposed problem (5.9)-(5.12), i.e. model (5.15)-(5.17). In fact the polytope of model (5.15)-(5.17) is, by definition, the convex hull of integer feasible solutions of model (3.1)-(3.21); hence it always admits at least one extreme integral solution. For the sake of completeness, we also mention that this bound can also be achieved by the Lagrangian relaxation of constraints (3.2) and (3.3) in model (3.1)-(3.21).

Preliminary experiments revealed that a standard linear search for good multipliers along subgradient directions can show significant oscillations and slow convergence. This is likely to be due to the flatness of the objective function. The systematic selection of small steps, at the same time, is not sufficient to eventually improve dual solutions (Taverna, 2011).

However, preliminary experiments also highlighted that the dual solution provided by the *Restricted Master Problem* (RMP) in early column generation iterations is often close to the Lagrange dual optimum and, starting from it, the direction given by subgradients is sufficient for improving, provided that the steps are small.

Therefore, we designed the following specific master dual optimisation algorithm, in which we alternate subgradient iterations with RMP re-optimisation.

At each column generation iteration, the RMP solution and the best known Lagrangian dual solution provide an upper $\bar{\mathcal{L}}$ and a lower $\underline{\mathcal{L}}$ bound to the optimal Lagrangian dual solution, respectively. During subgradient iterations, each step is classified as follows:

- *bad*, if the value of the produced solution is "too far" from $\bar{\mathcal{L}}$;
- *poor*, if the value of the produced solution is "too close" to $\underline{\mathcal{L}}$;
- *good*, otherwise.

When good and poor steps are detected, we respectively decrease and increase the step length; when bad steps are detected, instead, the algorithm backtracks re-initialising the dual multipliers to the values corresponding to the last non-bad step.

We define the following parameters:

- α the step length in multiplier update;
- $\alpha_0 > 0$ an initial step length;
- $\beta^+ \in (0,1)$ the minimum improvement factor to classify a poor step;
- $\beta^- \in (0,1)$ the minimum improvement factor to classify a bad step (a different parameter $\beta_0^- \in (0,1)$ is used in the first step);
- $\delta \in (0,1)$ the decay factor applied to α if a good step is performed;
- $\theta \in (0,1)$ the adjustment factor applied to α if a poor step is performed;
- $\xi \in (0,1)$ the adjustment factor applied to α if a bad step is performed;
- $N_{bad} > 0$ the maximum number of consecutive bad steps that can occur before the algorithm is stopped;
- $N > 0$ the maximum number of allowed iterations;
- $\rho > 0$ optimality tolerance.

The algorithm works as follows:

1. Initialise the S_{zgm} sets with the patterns obtained from the Commit&Dispatch algorithm.
2. Set $\underline{\mathcal{L}} := -\infty$; set $\alpha := \alpha_0$;
3. Iteratively:
   (a) Solve the RMP; let $\lambda^{\mathcal{M}}$, $\mu^{\mathcal{M}}$ and $\eta^{\mathcal{M}}$ be the values of λ, μ and η in an optimal RMP dual solution; set $\bar{\mathcal{L}}$ to its value.
   (b) Initialise the Lagrangian multipliers $\lambda^{\mathcal{L}} := \lambda^{\mathcal{M}}$, $\mu^{\mathcal{L}} := \mu^{\mathcal{M}}$ and the backup multipliers $\lambda_+^{\mathcal{L}} := \lambda^{\mathcal{L}}$ $\mu_+^{\mathcal{L}} := \mu^{\mathcal{L}}$.
   (c) Let $c_{bad} := 0$. For i in $1..N$:

i. Solve the Lagrangian subproblems using $\lambda^{\mathcal{L}}$ and $\mu^{\mathcal{L}}$, and compute the corresponding Lagrangian bound $\mathcal{L}'$; let $\mathbf{x}^{\mathcal{L}}$ be the optimal dispatching of subproblem $\mathcal{L}_D$ and for each subgroup $m \in M_{zg}$ let $\mathbf{y}^{\mathcal{L}}{}_{zgm}$ be the optimal scheduling for subproblem $\mathcal{L}_{\mathcal{C}}$.

ii. If $\left(i = 1 \ \wedge \ \frac{|\mathcal{L}' - \bar{\mathcal{L}}|}{\bar{\mathcal{L}}} \geq \beta_0^-\right)$ or $\left(i > 1 \ \wedge \ \frac{|\mathcal{L}' - \bar{\mathcal{L}}|}{|\underline{\mathcal{L}} - \bar{\mathcal{L}}|} \geq (1 + \beta^-)\right)$ then i is a bad step.

Let $c_{\text{bad}} := c_{\text{bad}} + 1$. If $c_{\text{bad}} > N_{\text{bad}}$ then exit the loop.
Otherwise backtrack by setting $\lambda^{\mathcal{L}} := \lambda^{\mathcal{L}}_+$, $\mu^{\mathcal{L}} := \mu^{\mathcal{L}}_+$ and update $\alpha := \xi\alpha$.

iii. Otherwise update backup multipliers $\lambda^{\mathcal{L}}_+ := \lambda^{\mathcal{L}}$ and $\mu^{\mathcal{L}}_+ := \mu^{\mathcal{L}}$ and reset the bad step counter $c_{\text{bad}} := 0$.

- If $\frac{|\mathcal{L}' - \underline{\mathcal{L}}|}{|\underline{\mathcal{L}}|} \leq \beta^+$ then i is a poor step. Increase the step: $\alpha := (2 - \theta)\alpha$.
- Otherwise i is a good step: let $\alpha := \delta\alpha$.

iv. Set $\underline{\mathcal{L}} = \max(\mathcal{L}', \underline{\mathcal{L}})$.

v. For each subgroup compute the reduced cost of scheduling $\mathbf{y}^{\mathcal{L}}_{zgm}$ with multipliers $\lambda^{\mathcal{M}}$, $\mu^{\mathcal{M}}$, $\eta^{\mathcal{M}}$ according to the objective function (5.7). If the reduced cost is negative then add $\mathbf{y}^{\mathcal{L}}_{zgm}$ to $\bar{S}_{zgm}$.

vi. Compute the subgradients of constraints (3.2) and (3.3):

$$\Delta^L_{zgm} = -x^{\mathcal{L}}_{tzg} + \sum_{m \in M_{zg}} y^{\mathcal{L}}_{tzgm} P_{zgm} + y^{\mathcal{L}^D}_{tzgm} P^{\Delta}_{zgm}$$

$$\Delta^U_{zgm} = x^{\mathcal{L}}_{tzg} - \sum_{m \in M_{zg}} y^{\mathcal{L}}_{tzgm} p_{zgm} + y^{\mathcal{L}^D}_{tzgm} p^{\Delta}_{zgm}$$

and update multipliers $\lambda^{\mathcal{L}}$ and $\mu^{\mathcal{L}}$ as follows:

$$\lambda^{\mathcal{L}}_{zgm} := \min(0, \lambda^{\mathcal{L}}_{zgm} - \alpha \cdot \frac{\Delta^U_{zgm}}{\|\Delta^U\|})$$

$$\mu^{\mathcal{L}}_{zgm} := \min(0, \mu^{\mathcal{L}}_{zgm} - \alpha \cdot \frac{\Delta^L_{zgm}}{\|\Delta^L\|})$$

(d) If $\frac{\bar{\mathcal{L}} - \underline{\mathcal{L}}}{\underline{\mathcal{L}}} < \rho$ then return $\underline{\mathcal{L}}$; else go to step 3a.

We remark that, from the theoretical point of view, the solution returned in step 3d is guaranteed to be within a factor ρ from a Lagrangian dual optimum, that is in turn equivalent to the optimal solution of (5.1)-(5.6). Besides improving the bound, the subproblem solutions produced during subgradient iterations are encoded as

columns and inserted into the RMP; they help to provide a better description of the master dual feasibility region, and to overcome LP stability issues. At the same time, the dual solutions obtained by RMP LP optimisation allow to escape from subgradient local minima.

From the practical point of view, such a master optimisation algorithm requires many parameters to be tuned. In Section 5.2 we report on experiments about tuning the most critical parameters.

5.1.3 Column Generation rounding

During column generation we search for good integer solutions with the following Rounding Heuristic (RH). After pricing, we consider model (5.1)-(5.6): for each $z \in Z$, $g \in G_z$ and $m \in M_{zg}$, we fix to 1 the γ^u_{zgm} variable of highest fractional value, and we fix to 0 all the remaining variables. Ties are broken according to the lexicographic order. In this way, no more integer variables are left free, and model (5.1)-(5.6) becomes a LP whose optimisation yields a full UCP solution.

5.2 Computational results

In this section we report experimental results for our algorithms. In subsection 5.2.1 we report the results for Column Generation methods in terms of dual and primal bounds. In subsection 5.2.2 we report a comparison between our algorithms and the sMTSIM simulator from RSE S.p.A.with RSE S.p.A.'s own environment.

5.2.1 Column Generation on Italy 2011

Tests in this section were performed on instances for Italy 2011 used in chapter 4.1.2 on the same hardware: a 4GB RAM 2 Core 1.2GHz processor PC. We implemented the column generation algorithm in C++ with SCIP 3.1.0 and CPLEX 12.4 and the full dual optimisation algorithm with CPLEX Concert API.

Dual bounds

In a first round of experiments we focus on finding best dual bounding strategies. We compared three column generation strategies:

- Crossover, Pure column generation (CP) : solve master LPs with barrier algorithm followed by crossover phase, price new columns, skip step 3c of the

master dual optimisation algorithm.

- Barrier only, Pure column generation (BP) : solve master LPs with barrier algorithm, price new columns, skip step 3c of the master dual optimisation algorithm.

- Barrier only, Full master dual optimisation (BF) : solve LPs with barrier and include step 3c, exploiting the full master dual optimisation algorithm.

Each strategy uses IACR+CD to initialise the RMP. In the pricing problem it is assumed power plants have to wait the respective minimum up/down time at the first period before switching state, as depicted in figure 5.1.

Parameter tuning. For the master dual optimisation algorithm, after preliminary experiments, we set: $\alpha_0 = 5 \times 10^{-10}$, $\delta = 0.95$, $\xi = 0.3$, $\theta = \delta^2 = 0.9025$ $\beta^- = 0.1$, $\beta_0^- = 0.3$, $\beta^+ = 0.01$ $N_{bad} = 3$, $\rho = 0.001$. The number of steps to be performed on each run N is set as follows: $N = 0$ (i.e. no subgradient step is performed) for the first two column generation iterations, $N = 5$ for iterations between 3 and 5, $N = 18$ for iterations between 6 and 10 and $N = 25$ for iterations beyond 10. The rationale for the choice of N is that on one hand the subgradient steps are more effective at generating good columns starting from good dual solutions, which are computed in the latest pricing iterations; on the other hand, in the latest pricing iterations it is more convenient to price new columns during subgradient steps instead of performing a new column generation iteration, as the complexity of each step of the Lagrangian relaxation remains constant throughout the algorithm, while the complexity of solving the RMP grows due to the increase in the number of columns.

In Tables 5.1, 5.2 and 5.3 we report the results for column generation strategies CP, BP and BF respectively. In each table we detail, in turn, the number of RMP optimisation iterations performed and the improvement, i.e. the increase, over the IACR dual bound; we also include the gap between the final value of the RMP and the best dual bound, giving an upper estimate on the improvement that could be obtained by allowing more computing resources to the master dual optimisation process. Finally, we mark with an asterisk '*' the instances for which SCIP detects numerical issues. When this happens the corresponding dual solution cannot be proved to be either optimal or feasible, and therefore any dual optimality guarantee is lost.

Id	Size	Iterations	Improvement over IACR %	Gap RMP/Dual %
1.1	1 month	10	-0.1%	0.8%
1.2		10	0.2%	0.4%
1.3		15	0.3%	0.3%
1.4		11	0.2%	0.6%
1.5		15	0.4%	0.2%
1.6		15	0.2%	0.2%
1.7		12	-0.1%	0.5%
1.8		9	0.0%	0.4%
1.9		10	0.0%	0.5%
1.10		12	0.1%	0.7%
1.11		11	0.2%	0.5%
1.12		11	-0.1%	0.9%
2.1	2 months	5	-0.4%	1.2%
2.2		6	-0.5%	1.4%
2.3		6	-0.2%	0.9%
2.4		6	-1.5%	2.2%
2.5		6	-1.3%	2.1%
2.6		4	-1.6%	2.6%
3.1	3 months	5	-0.8%	1.6%
3.2		7	0.1%	0.7%
3.3		5	-0.9%	1.6%
3.4		5	-0.4%	1.4%
4.1	4 months	7	-0.1%	0.9%
4.2		5	-1.8%	2.6%
4.3		6	-0.6%	1.4%
6.1	6 months	4	-1.5%	2.4%
6.2		4	-1.5%	2.5%

Table 5.1: Dual bounds for CP column generation strategy

Id	Size	Iterations	Improvement over IACR %	Gap RMP/Dual %
1.1	1 month	45	0.4%	0.2%
1.2		39	0.4%	0.1%
1.3		41	0.5%	0.1%
1.4		43	0.6%	0.1%
1.5		40	0.5%	0.2%
1.6		54	0.4%	0.1%
1.7		38	0.2%	0.2%
1.8		44	0.2%	0.1%
1.9		37	0.2%	0.2%
1.10		41	0.6%	0.1%
1.11		42	0.6%	0.1%
1.12		43	0.5%	0.2%
*2.1	2 months	10	–	–
*2.2		13	–	–
*2.3		7	–	–
*2.4		15	–	–
*2.5		8	–	–
*2.6		4	–	–
*3.1	3 months	6	–	–
*3.2		5	–	–
*3.3		5	–	–
*3.4		6	–	–
*4.1	4 months	6	–	–
*4.2		8	–	–
*4.3		6	–	–
6.1	6 months	6	-0.4%	1.2%
6.2		6	-0.4%	1.2%

Table 5.2: Dual bounds for BP column generation strategy

Id	Size	Iterations	Improvement over IACR %	Gap RMP/Dual %
1.1	1 month	18	0.1%	0.5%
1.2		18	0.2%	0.4%
1.3		19	0.3%	0.3%
1.4		28	0.5%	0.2%
1.5		31	0.5%	0.1%
1.6		36	0.4%	0.1%
1.7		18	-0.1%	0.6%
1.8		33	0.2%	0.2%
1.9		17	0.0%	0.5%
1.10		18	0.2%	0.6%
1.11		18	0.4%	0.4%
1.12		19	0.1%	0.6%
2.1	2 months	8	-0.2%	0.9%
2.2		8	0.1%	0.7%
2.3		15	0.1%	0.6%
2.4		16	-0.1%	0.6%
2.5		8	-0.2%	1.0%
2.6		8	-0.2%	1.0%
3.1	3 months	7	-0.2%	1.0%
3.2		11	0.0%	0.8%
3.3		12	-0.2%	0.7%
3.4		7	-0.4%	1.2%
4.1	4 months	5	0.0%	0.9%
4.2		7	-0.3%	1.0%
4.3		6	-0.5%	1.4%
6.1	6 months	5	-0.7%	1.7%
6.2*		–	–	–

Table 5.3: Dual bounds for BF column generation strategy

The BP strategy provides bounds that are on average 0.7% from the optimum, but suffers numerical troubles on several instances. CP strategy, by using the crossover algorithm, allows to solve these numerical troubles at the expense of speed: the number of RMP LP re-optimisation steps is more than halved, the final accuracy estimate doubles and the dual bounds are worse. The BF strategy allows to effectively solve these issues: it yields better bounds with no numerical issues and a higher accuracy guarantee with fewer LP optimisation steps. The subgradient steps allow to produce columns that improve significantly the stability and the effectiveness of the column generation process.

We remark we found an exception in instance 6.2, for which the LP solver yielded unreliable bounds with the BF strategy.

Heuristic solutions

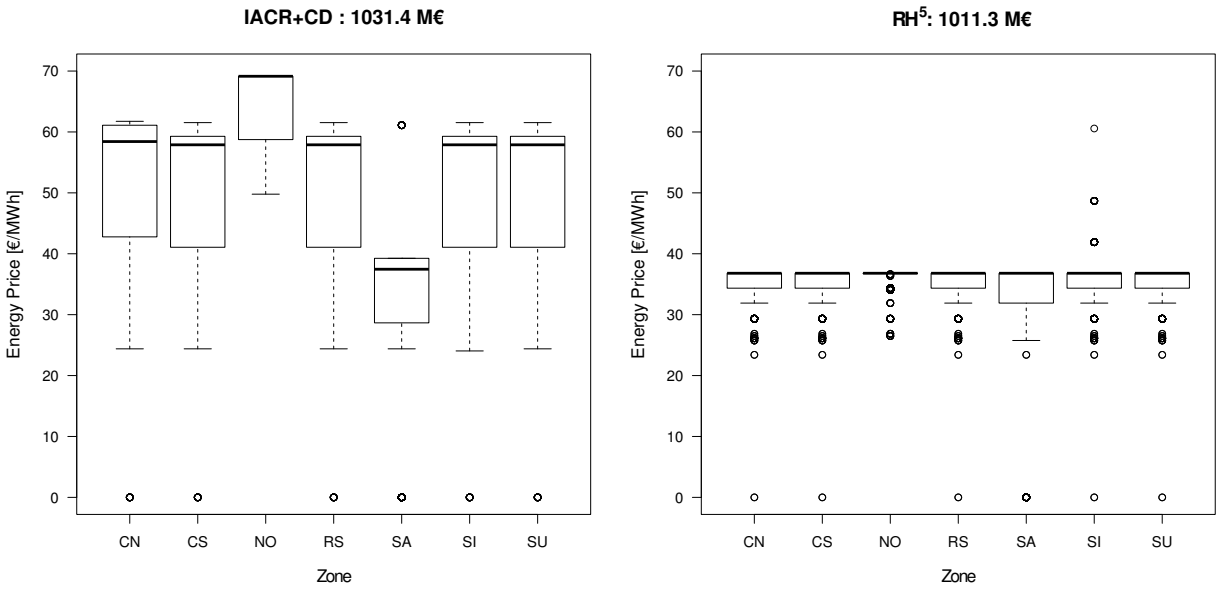
In a second round of experiments we focused on finding good integer solutions. In particular we compared the primal solution found by the CD matheuristic initialised with an optimal solution of the IACR relaxation (IACR+CD) against the solution found by the rounding heuristic, RH^k, run at the k^{th} iteration of the BF column generation algorithm. In our tests we selected $k = 5$ as a benchmark, i.e. we ran the rounding heuristic RH on the set of columns in the RMP after five column generation iterations. In Table 5.4 we report, for each instance, the IACR+CD solution value, its duality gap, and the corresponding required CPU time, the RH^5 value, its duality gap, the time required to perform the five column generation iterations and the time required to run RH itself. The duality gaps have been computed by considering the best dual bound provided by the BF strategy. The IACR+CD heuristic algorithm yields a good compromise between accuracy and speed, being able to produce solutions with an average duality gap of 2.9% within a few minutes. Its accuracy tends to be stable and the computing time grows slowly with the instance size. The RH^5 heuristic algorithm provides even higher accuracy, allowing to reach a duality gap of 0.4% on average and below 1% in all cases but two; however it requires substantially larger computing time, especially for instances of large size. As a side result, this allows to conclude also that IACR dual bounds are very tight, that in turn explains the relatively small dual bound improvement offered by the column generation strategies.

We also tried to assess the impact of the improvement in the objective value achievable with the RH^5 heuristic on the solution structure. We found out that

even small relative improvements yield relevant differences between primal solution structures, i.e. it is not determined by small numerical adjustments but by a different match between supply and demand. The latter can be synthesised by zonal energy prices, given by the dual prices $(\pi_{tz})^{\top}_{t\in T, z\in Z}$ of constraints (3.16). In figure 5.2 zonal prices distributions for instance 4.2 in the two primal solutions are reported using boxplot charts. The difference between objectives of the primal solutions for this instance are 20.1 $M€$, i.e. less than 2.1% of the best known dual bound; still significant differences can be observed on theprices distributions. We note that in the RH solution there are three hours, in the 'SI' region, with $ENP_{tz} > 0$ that signal a loss-of-load event and with an hourly energy price above 400 $\frac{€}{MWh}$. They are not included in the chart to ease the representation.

Id	Size	IACR+CD		RH		
		Gap %	Time [s]	Gap %	Time RH [s]	Time 5^{th} iteration [s]
1.1	1 month	2.5%	57.5	0.7%	86.8	407.0
1.2		2.8%	48.1	0.5%	101.0	445.0
1.3		3.0%	47.2	0.5%	88.1	406.0
1.4		2.8%	58.7	0.4%	85.6	466.0
1.5		2.1%	43.2	0.3%	54.7	406.0
1.6		2.5%	44.6	0.2%	37.0	282.0
1.7		2.2%	48.8	0.6%	106.9	488.0
1.8		3.0%	55.9	0.2%	59.1	247.0
1.9		2.3%	52.4	0.6%	147.5	503.0
1.10		3.5%	44.5	0.6%	117.0	447.0
1.11		3.3%	43.1	0.5%	123.1	478.0
1.12		3.2%	48.9	0.7%	83.5	436.0
2.1	2 months	3.0%	91.7	0.8%	476.9	1817.0
2.2		3.5%	115.8	0.8%	899.2	1790.0
2.3		2.9%	91.7	0.7%	159.8	1092.0
2.4		2.7%	94.4	0.7%	262.6	1171.0
2.5		3.1%	108.0	0.8%	1499.6	2252.0
2.6		3.2%	90.7	0.9%	526.7	1870.0
3.1	3 months	3.2%	163.6	0.8%	1231.3	3326.0
3.2		3.2%	193.4	0.9%	552.5	2058.0
3.3		2.6%	197.5	0.7%	417.9	2611.0
3.4		3.4%	150.0	0.9%	802.9	2866.0
4.1	4 months	3.2%	225.7	2.9%	1810.5	9433.0
4.2		2.8%	243.5	0.8%	1209.8	4633.0
4.3		3.0%	290.9	1.0%	1531.3	7652.0
6.1	6 months	3.3%	890.6	0.9%	2561.3	8817.0
6.2*		2.4%	906.3	–	–	–

Table 5.4: Primal bounds. Column generation with BF strategy.



	IACR+CD	RH^5
Zero-price hours	142	84

Figure 5.2: Zonal energy prices distributions in different primal dual bounds for instance 2.4. Bold lines in the chart indicate the median of the distribution. Zero-price hours signal overabundant production (*overgeneration*) and, thus, an inefficient match between supply and demand.

5.2.2 Comparison with sMTSIM

In this section we compare the behaviour of the heuristic IACR+CD illustrated in chapter 4 and the Column Generation algorithm presented here with the sMTSIM algorithm available at RSE S.p.A.. sMTSIM (Siface et al., 2014; G. Migliavacca, 2009) is a zonal market model simulator developed by RSE S.p.A.. It is derived from MTSIM, another zonal market simulator from RSE S.p.A., and provides, among others, an extensions to perform stochastic optimisation and simulate an optimal policy for the system which accounts for uncertainty in the data.

The deterministic solver in sMTSIM has a structure similar to the Commit&Dispatch heuristic (section 4.1.2): a first step solves a linear problem to compute a tentative dispatching for power plants, neglecting minimum production levels and minimum up/down times. The approximated dispatching is then used to determine a heuristic commitment for thermal plants. Finally, a linear dispatching problem computes the optimal plant production levels given the heuristic commitments. The main differences between IACR+CD and sMTSIM are:

- power plant grouping: sMTSIM does not group thermal plants.

- first step: sMTSIM uses different coefficients to approximate the piece-wise linear cost of thermal plants in the continuous linear model at the first step. In particular, the model is not a proper lower bound of the original model and cannot return a valid lower estimate of the optimum value.

- commitment: sMTSIM uses a polynomial-time heuristic to compute a feasible commitment for each power plant. First the procedure tries to determine a commitment that satisfies the tentative dispatching computed at the first step. Such commitment can violate minimum up/down time constraints (equations (3.5)-(3.12)) in some time intervals. Then, the procedure fixes those violations by changing the commitment in each infeasible interval. The procedure relies on user-defined thresholds on the overall production and commitment time of an interval to generate new commitments. When there are more than one commitment to choose for an interval, the procedure computes an estimate of the production profit for each possible commitment, using zonal energy prices from the tentative dispatching, and then it chooses the commitment that maximises the estimated profit.

Additionally, sMTSIM implements constraints such as ramping (Frangioni, 2006) and reserve constraints (Kirby, 2003; Zani et al., 2015) which are not included in the model (3.1)-(3.21).

We remark that, in general, these constraints, especially the ones included in sMTSIM, can be effortlessly added to our models since they are usually formulated as linear inequalities. In particular, pricing problems can be extended with switching costs, intermediate states (warming-up, cooling-down), and other linear constraints while still being solvable in polynomial time by the same algorithm we adapted from (Frangioni, 2006).

Experimental results

We performed two tests on a yearly instance for Europe 2020, spanning Italy and neighbouring countries: Slovenia, France, Germany, Switzerland and Austria.

sMTSIM defines both an algorithm and a model, hence a first test compares the different algorithms on the model (3.1)-(3.21) used in this work; a second test compares the same algorithms using version 8.0 of the sMTSIM model, which includes additional ramping and reserve constraints. We refer the reader to (Siface et al., 2014) for further details.

sMTSIM is implemented in GAMS and uses the CPLEX 12.1 MILP solver. We implemented the IACR+CD model and the Column Generation algorithm with the same technology. For Column Generation we perform three iterations using the BP pricing strategy and computing a primal bound with the rounding heuristic (section 5.1.3) at each iteration. The pricing problems (5.7)-(5.8) are solved with the CPLEX MILP solver. We remark we do not use the Full dual optimisation procedure for the BF strategy as it would require appropriate precaution to be implemented in GAMS without incurring in excessive overhead.

For the first test, on model (3.1)-(3.21), we report in table 5.5 for each algorithm the difference in percentage between the dual bound and the best known dual bound, computed by the CG algorithm, the gap in percentage between the primal bound and the best known dual bound, and the computation time.

Results show the IACR+CD slightly improves over the sMTSIM heuristic for the primal bound and provides an estimate of the optimality gap while still requiring comparable computation time. The CG algorithm, despite requiring more time, effectively reduces the optimality gap with just three iterations and within affordable time, i.e. less than one hour.

Computation time, however, does not include the overall time, and memory, required by the GAMS implementation to generate and parse the necessary data to communicate with the CPLEX solvers. For sMTSIM and IACR+CD heuristic the total time amounts to between 10 and 20 minutes. For CG it reaches up to 1 hour. Hence, while the CG algorithm allows to significantly increase the quality of both bounds for the instance in reasonable time, its actual usage in production environment could easily become quite expensive with larger instances or more iterations.

	sMTSIM	IACR+CD	CG-BP with 3 iterations
Dual improvement %	–	-7.9%	0%
Gap %	16.5%	12.6%	2.4%
Time [s]	248	402	1841 (of which 402 for IACR+CD)

Table 5.5: Comparison for sMTSIM and IACR+CD for model (3.1)-(3.21).

For the second test we performed a few experiments with the latest version of sMTSIM using its own extended model, with the same instance for Europe 2020. The sMTSIM model is larger than eq.(3.1)-(3.21) due to the additional constraints and variables. We first present the experimental results using only the first two months (1460 hours) of the year. Furthermore, to avoid numerical instability problems, we use the CP pricing strategy for the CG algorithm.

We considered two scenarios:

- *NoExcess*, with heavy penalisation of mismatches between supply and demand. For the scenario we set $E_t = F_t = 10,000€/KWh \, \forall \, t \in T$.

- *OptimalThermal*, with no penalisation of excessive energy. The objective then considers only the cost of the thermal production and the penalty for unsatisfied demand. Compared to the other scenario, the model focuses on minimising the cost of the thermal production schedule. We set $E_t = 3,000€/KWh$ and $F_t = 0€/KWh \; \forall t \in T$.

In tables 5.6 and 5.7 we report the results for the three algorithms in the *NoExcess* and *OptimalThermal* scenarios respectively. The sMTSIM heuristic computed slightly better primal bounds than the IACR+CD heuristic in both scenarios. The primal heuristic in the CG algorithm, on the other hand, managed to significantly reduce the gap in both cases by computing almost optimal solutions within few iterations. Further inspection of the two solutions showed that the CG solution, compared to the sMTSIM one

- in the *NoExcess* scenario reduces excess production by 30% and pumping power by 18%.

- in the *OptimalThermal* scenario reduces thermal production by 0.1% and pumping power by 16%.

In both cases the CG algorithm computes a better schedule for thermal and hydroelectric productions, optimizing the main objective drivers and yielding a lower global cost.

On the other hand, the CG algorithm is way more expensive than the other methods. We remark as well that the computing time in the CG algorithm is clearly dominated by the solution time of the master problem. In tables 5.8 and 5.9 we report the computation time for master and pricing problems at each iteration for the *NoExcess* and *OptimalThermal* scenarios respectively. It can be noticed the solution time of the master problem grows at each iteration in both cases, while the pricing problem takes roughly the same amount of time at each iteration. At the end of the CG algorithm solving the master problem required rougly ten and five times more time than the pricing problem in the *NoExcess* and *OptimalThermal* scenarios respectively. It can also be noticed that the LP solver manages to solve

the master problem more quickly in the *OptimalThermal* scenario as the objective function has fewer terms.

We performed further experiments on the full-year scenario with the IACR+CD and CG algorithms as well, using both the CP and the BP pricing strategy. We noticed the IACR+CD algorithm usually fails to improve over the sMTSIM solution and the CG algorithm either requires long execution times with the CP strategy, e.g. more than twenty minutes for solving the master once, or fails due to numerical instabilities with the BP strategy.

In general the problem of finding an approximation method that guarantees improvements over heuristic solutions from sMTSIM and CD while requiring comparable complexity is non trivial. Quick heuristic methods have to estimate optimal values of variables to efficiently compute feasible solutions and cannot thus provide guarantees about the quality of their solutions with respect to the original problem. On the other hand, the size of the model and the presence of degeneracy can cause more powerful methods like CG to converge slowly or not at all due to numerical issues with state-of-the-art linear solvers.

We refer to chapter 7 for an empirical analysis on the computational difficulties inherent to these large-scale UCPs.

	sMTSIM	IACR+CD	CG-BP with 3 iterations
Dual improvement %	–	0%	0%
Gap %	17.8%	22.4 %	2.3%
Time [s]	31	59	1419 (of which 59 for IACR+CD)

Table 5.6: Comparison for sMTSIM and IACR+CD for sMTSIM model in the *NoExcess* scenario.

	sMTSIM	IACR+CD	CG-BP with 3 iterations
Dual improvement %	–	0%	0%
Gap %	1.4%	1.5 %	0.6%
Time [s]	29	60	1088 (of which 60 for IACR+CD)

Table 5.7: Comparison for sMTSIM and IACR+CD for sMTSIM model in the *OptimalThermal* scenario.

Problem	Iteration			Total time
	1	2	3	
master	211	392	633	1292
pricing	40	44	43	129

Table 5.8: Computing time, in seconds, for master and pricing problems in *NoExcess* scenario.

Problem	Iteration			Total time
	1	2	3	
master	108	160	210	534
pricing	43	45	44	132

Table 5.9: Computing time, in seconds, for master and pricing problems in *OptimalThermal* scenario.

Chapter 6

Benders Decomposition

In this chapter we report our results for the Benders decomposition (BD) (Benders, 1962) algorithm we devised for the long-term UCP. Similarly to Column Generation in chapter 5, BD is a technique used to solve large-scale MILP to high accuracy. We adapt the standard BD algorithm to our problem and use two techniques to improve its convergence: the Magnanti-Wong (MW) or Pareto-optimal cuts, a family of refined Benders optimality cuts, and a two-phases heuristic, which quickly computes optimality cuts in the first phase by applying BD to the continuous relaxation of the problem and then restores integrality constraints to continue with the standard BD algorithm.

In section 6.1 we report the main elements of BD algorithms and the specific adaptation for our UCP. In section 6.2 we report the experimental results.

6.1 Benders Decomposition for the UCP

The standard BD method decomposes a MIP model into an integer master problem and a continuous slave problem. By exploiting dual information from the slave solution, the algorithm iteratively constructs an approximation of the optimal facets of the slave problem into the master problem as a reduced set of optimality cuts (*Benders cuts*). The method can be seen as a dual approach of CG as it "swaps" the definitions of master and slave problems, having the slave providing dual information to the master. We note that recent studies (Hooker, 2009; Fischetti et al., 2009) developed techniques to generate Benders cuts from purely integer subproblems (*Logic Benders cuts*) as well. An even more recent technique (Zeng and Zhao, 2011; Long Zhao and Bo Zeng, 2012) proposes to employ the slave problem to provide

primal information to the master problem, generating new columns and constraints at each iteration. While this method is reported to be an order of magnitude faster than classical Benders decomposition on sample stochastic programs, it also tends to require much more memory. Since our model can already require a significant amount of memory, we do not consider this method suitable for our purposes.

As with Column Generation, the method is used to decouple power dispatching from unit commitment. From model (3.1)-(3.21) the method yields a slave dispatching model (6.1)-(6.10), defined for a feasible commitment $(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^D)$ and identical to the dispatching model (4.13)-(4.16).

$$\min \phi_{\mathcal{S}} = \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} c_{tzg}x_{tzg} + \sum_{t\in T, z\in Z} EIE_{tz}F_t + \sum_{t\in T, z\in Z} ENP_{tz}E_t + \sum_{\substack{z\in Z, g\in G_z,\\ m\in M_{zg}}} e_{tzgm}\tilde{y}_{tzgm} \tag{6.1}$$

$$\text{s.t. } x_{tzg} \leq \sum_{m\in M_{zg}} P_{zgm}\tilde{y}_{tzgm} + \sum_{m\in M^D_{zg}} P^{\Delta}_{zgm})\tilde{y}^D_{tzgm} \ \forall t \in T, z \in Z, g \in G_z \tag{6.2}$$

$$x_{tzg} \geq \sum_{m\in M_{zg}} p_{zgm}\tilde{y}_{tzgm} + \sum_{m\in M^D_{zg}} p^{\Delta}_{zgm}\tilde{y}^D_{tzgm} \ \forall t \in T, z \in Z, g \in G_z \tag{6.3}$$

$$o_{1zh} = q_{zh} \ \forall z \in Z, h \in H_z \tag{6.4}$$

$$o_{(|T|+1)zh} = Q_{zh} \ \forall z \in Z, h \in H_z \tag{6.5}$$

$$o_{tzh} + n_{th} + \beta_h \cdot m_{tzh} = o_{(t+1)zh} + s_{tzh} + \alpha_{zh}l_{tzh} \ \forall t \in T, z \in Z, h \in H_z \tag{6.6}$$

$$\sum_{h\in H_z} l_{tzh} + \sum_{g\in G_z} x_{tzg} + \sum_{(i,z)\in A} w_{tiz} + \sum_{z\in Y} ENP_{tz} \geq d_{tz} + \sum_{h\in H_z} m_{tzh} + \sum_{(z,j)\in A} w_{tzj} + \sum_{z\in Y} EIE_{tz} \quad \forall t \in T, z \in Z \tag{6.7}$$

$$w_{tij} \in [0, b_{ij}] \ \forall t \in T, (i,j) \in A \tag{6.8}$$

$$s_{tzh} \in [0, f_{zh}], o_{tzh} \in [0, V_{zh}], l_{tzh} \in [0, P_{zh}], m_{tzh} \in [0, P^{\beta}_{zh}] \ \forall t \in T, z \in Z, h \in H_z \tag{6.9}$$

$$ENP_{tz} \geq 0, EIE_{tz} \geq 0 \ \forall t \in T, z \in Z \tag{6.10}$$

and a purely integer linear master problem

$$\min \phi_{\mathcal{M}} = \psi \tag{6.11}$$

$$\text{s.t. } y^D_{tzgm} \leq y_{tzgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{6.12}$$

$$v^+{}_{tzgm} \geq y_{tzgm} - y_{(t-1)zgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{6.13}$$

$$v^-{}_{tzgm} \geq y_{(t-1)zgm} - y_{tzgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{6.14}$$

$$v^{+D}_{tzgm} \geq y^D_{tzgm} - y^D_{(t-1)zgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{6.15}$$

$$v^{-D}_{tzgm} \geq y^D_{(t-1)zgm} - y^D_{tzgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{6.16}$$

$$y_{tzgm} \geq \sum_{\tau \in T : t \in T^{on}_\tau} v^+{}_{\tau zgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{6.17}$$

$$y_{tzgm} \leq |K_{zgm}| - \sum_{\tau \in T : t \in T^{off}_\tau} v^-{}_{\tau zgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{6.18}$$

$$y^D_{tzgm} \geq \sum_{\tau \in T : t \in T^{off}_\tau} v^{+D}_{\tau zgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{6.19}$$

$$y^D_{tzgm} \leq |K^D_{zgm}| - \sum_{\tau \in T : t \in T^{off}_\tau} v^{-D}_{\tau zgm} \; \forall t \in T, z \in Z, g \in G_z, m \in M^D_{zg} \tag{6.20}$$

$$\psi \geq \sum_{\substack{t \in T, z \in Z, \\ g \in G_z, m \in M_{zg}}} \Big((e_{tzgm} + \mu_{tzgb} p_{zgm} - \lambda_{tzgb} P_{zgm}) y_{tzgm} + (\mu_{tzgb} p^\Delta_{zgm} - \lambda_{tzgb} P^\Delta_{zgm}) y^D_{tzgm} \Big) + \eta_b \quad \forall b \in B \tag{6.21}$$

Inequalities 6.21 are Benders optimality cuts for a solution $(\tilde{\mathbf{y}}_b, \tilde{\mathbf{y}}^D_b)$ with index $b \in B$ and coefficients $\boldsymbol{\xi}_b = (\boldsymbol{\lambda}_b, \boldsymbol{\mu}_b, \eta_b)$ where $\boldsymbol{\lambda}_b = (\lambda_{tzgb})^\top$ and $\boldsymbol{\mu} = (\mu_{tzgb})^\top$ $\forall t \in T, z \in Z, g \in G_z$ are the duals of constraints 6.2 and 6.3 respectively and $\eta_b = \phi_{\mathcal{S}}(\tilde{\mathbf{y}}_b, \tilde{\mathbf{y}}^D_b) - \sum_{\substack{t \in T, z \in Z, \\ g \in G_z, m \in M_{zg}}} e_{tzgm} \tilde{y}_{tzgmb}$.

In this setting Benders feasibility cuts are not needed because the slave problem is feasible for any commitment returned by the master problem. Indeed, in the worst case a valid commitment causes a mismatch between supply and demand in the system (constraints (6.7)) by limiting production of thermal plants in some periods and some zones. This mismatch is then absorbed by the corresponding EIE or ENP variables and causes the solution value to worsen considerably. The solution, however, is still feasible.

In general the master problem has a combinatorial number of valid optimality cuts, one for each feasible commitment of thermal plants. However, only a few of them are going to be active in an optimal solution. Hence, the BD algorithm starts from a Restricted Master Problem (RMP) where the set of cut indexes is replaced by a non-empty subset $\overline{B}$ containing only cuts generated from heuristic solutions. Then it iteratively adds new cuts to the restricted set by solving the slave problem, until the solutions of master and slave problems converge to the same value.

The rationale for trying BD on the UCP model (3.1)-(3.21) is overcoming the numerical difficulties encountered in the CG algorithm (chapter 5), while solving the RMP (5.1)-(5.6), to obtain a more efficient and effective solution method. Indeed, the continuous model (6.1)-(6.10) is more stable and easier to solve than the RMP from CG (5.1)-(5.6) and, at the same time, the commitment problem (6.11)-(6.21) maintains most of the structure of the pricing problem (5.7)-(5.8). On the other hand, the latter is more complex than the pricing problems (5.7)-(5.8), and $\mathcal{NP}$-hard in the general case, because of Benders cuts, which couple the commitment of different plants in different periods.

We can further prove the master problem to be reducible to a *weakly-$\mathcal{NP}$*-hard variant of the shortest path problem.

Proposition 8. *The master problem (6.11)-(6.21) is reducible to the Absolute Robust Shortest Path Problem (ARSPP) (Yu and Yang, 1998).*

First we introduce the ARSPP. Let $\Gamma = (V, A)$ be a directed and connected graph with $\sigma, \omega \in V$ being the source and sink nodes respectively and S be a set of scenarios. For each arc $(i, j) \in A$ and scenario $s \in S$ a non-negative cost c_{ij}^s is defined. The ARSPP is the problem of finding the path from σ to ω that minimises the maximum cost over all scenarios $s \in S$.

Let $x_{ij} \in \{0, 1\}$ be the flow on arc $(i, j) \in A$. The ARSPP problem is thus defined as

$$\min_{x} \max_{s \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^s x_{ij} \right\} \tag{6.22}$$

s.t.

$$\sum_{j \in V} x_{ij} - \sum_{k \in V} x_{ki} = \begin{cases} 1 & i = \sigma \\ -1 & i = \omega \\ 0 & \text{otherwise} \end{cases} \tag{6.23}$$

$$x_{ij} \in \{0, 1\} \tag{6.24}$$

The ARSPP is known to be $\mathcal{NP}$-hard (Yu and Yang, 1998), but admits a pseudo-polynomial time exact algorithm, described in (Yu and Yang, 1998), and a Fully Polynomial-Time Approximation Scheme (FPTAS) (Aissi et al., 2005).

We then prove that the master problem (6.11)-(6.21) can be polynomially re-

duced to the ARSPP with the following procedure. To simplify the proof we use the disaggregated formulation of the UCP, in which commitment variables for each power plants are represented separately from others. From a disaggregated solution an aggregated one can be computed in linear time by summing the commitment variables of plants in each subgroup and period, as seen in theorem 1.

Let K be the overall set of power plants in the system and let it be ordered according to some criterion, (e.g. lexicographical).

1. For each thermal plant $k \in K$ generate a graph as done in section 5.1.1 or, equivalently, in (Frangioni, 2006). Set costs to be zero for every arc. The graph encodes all the feasible commitments of each power plant as acyclic paths between a source and a sink node.

2. Define the global graph $\hat{\Gamma} = (\hat{V}, \hat{A})$ with $\hat{V} = \{\sigma_0, \omega_0\}$ and $\hat{A} = \{(\sigma_0, \omega_0)\}$ as a graph with one source σ_0 and one sink ω_0 and one arc (σ_0, ω_0) connecting the two.

3. Extract each plant $k \in K$ in order, let $\Gamma_k = (V_k, A_k)$ be the corresponding graph and let σ_k and ω_k be its source and sink respectively. $\hat{\Gamma}$ is updated as follows:

   (a) $\hat{V} = \hat{V} \cup V_k$

   (b) $\hat{A} = \hat{A} \cup A_k$

   (c) Consider the arc $a_{\omega_0} = (h, \omega_0)$. Connect h to σ_k, connect ω_k to ω_0 and remove a_{ω_0}.

   When all the thermal plants have been extracted from K, $\hat{\Gamma}$ is an acyclic graph, where the original graphs $\Gamma_k \, \forall k \in K$ are connected in series sink-to-source. The foremost source is σ_0, connected to the first source of the first extracted graph, and the last sink is ω_0, connected to the sink of the last extracted graph (for example, see fig. 6.1).

4. Let $S = B$. The different scenarios are determined by the cut indices set B. For each cut index $b \in B$ a cost function is defined for the commitment of power plants $(\boldsymbol{u}, \boldsymbol{u}^D) = ((u_{tk})_{TK}, (u^D_{tk})_{TK})$ with $u_{tk}, u^D_{tk} \in \{0, 1\} \; \forall t \in T, k \in K$.

$$f_b(\mathbf{u}, \mathbf{u}^D) = \sum_{t \in T, k \in K} [(e_{tzgm} + \mu_{tzgb} p_{zgm} - \lambda_{tzgb} P_{zgm}) u_{tk} + (\mu_{tzgb} p^{\Delta}_{zgm} - \lambda_{tzgb} P^{\Delta}_{zgm}) u^D_{tk}] + \eta_b$$

Then, using the labelling algorithm described in section 5.1.1, for each scenario $b \in B$ and plant $k \in K$ a cost is computed for arcs belonging to $\sigma_k - \omega_k$ paths in $\hat{\Gamma}$, such that the overall cost of each path is equal to the cost of the corresponding commitment. Finally for every scenario set the cost of every arc in $\hat{\Gamma}$ connecting sinks to sources and of the arcs incident on σ_0 and ω_0 to zero.

It can be shown the procedure to construct $\hat{\Gamma}$ has linear complexity $O(|B||K||T|)$.

Proposition 9. *For each $b \in B$ any $\sigma_0 - \omega_0$ path on $\hat{\Gamma}$ represents a feasible commitment of thermal plants and its total cost is equal to the overall cost of the commitment under scenario b.*

Finally, we note arcs in $\hat{G}$ can have negative costs but the graph is acyclic; therefore ARSP instances are well-defined.

Proposition 10. *Any feasible solution to the master problem (6.11)-(6.21) can be represented as a path on $\hat{\Gamma}$, and vice versa. In particular, every optimal solution of the master problem is an ARSP over $\hat{\Gamma}$, and vice versa.*

Indeed, for both the master problem (6.11)-(6.21) and the ARSPP over $\hat{\Gamma}$ the optimal solution is given by a min-max optimisation of the sum of cost functions f_b for $b \in B$.

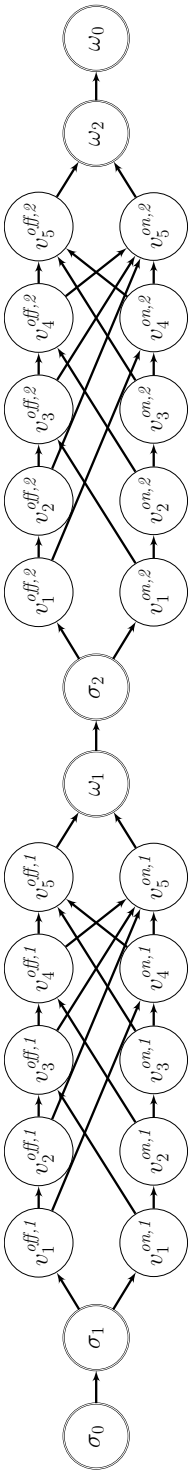


Figure 6.1: The digraph $\hat{\Gamma}$ with for a system of two power plants with $\tau^{on} = 3$ and $\tau^{off} = 2$ on 5 periods.

6.1.1 Accelerating convergence with Magnanti-Wong cuts

Modern applications of BD employ acceleration techniques to improve the convergence of the algorithm. Several ones are available in the literature. We can broadly classify them into two groups:

- *cut-refining*: they compute better cuts by refining the standard ones. Examples are Pareto-optimal cuts (Papadakos, 2008), high-density cuts (Rei et al., 2008) and cuts bundles (Saharidis et al., 2010).

- *local-search*: they explore a neighbourhood of the current optimal solution in the master problem space to obtain stronger cuts (with respect to the local optimum) while solving smaller integer subproblems. Examples are trust-region methods (Maher et al., 2014) and local branching (Rei et al., 2008).

In this work we investigate cut-refining methods, which we consider to be more promising than local-search ones for our problem. The rationale for our decision is that a local search would define a restricted set of moves to operate on a master solution to explore its neighbourhood, i.e. switching the state of some power plants. However, the impact of such moves is unlikely to produce a significant change on the overall solution due to the presence of degenerate solutions and the flatness of the objective function. Furthermore, the master problem provides no information to avoid mismatches between supply and demand in the slave problem, as it only models feasible commitment decisions for power plants. Cut-refining methods instead try to maximise the utility of dual information from the slave solution, which already accounts for more elements of the system than the master solution, including energy balancing constraints.

We consider the cut-refining method of Magnanti-Wong (MW) to compute Pareto-optimal cuts (Magnanti and Wong, 1981; Papadakos, 2008). A MW cut is a Benders optimality cut which is closest to an interior point (*core point*) of the master problem polytope. The cut has the further property of being *Pareto optimal*. Formally, let

$$\min \ \boldsymbol{c}^\mathsf{T}\boldsymbol{x} + \boldsymbol{e}^\mathsf{T}\boldsymbol{y} \tag{6.25}$$

$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{y} = \boldsymbol{b} \tag{6.26}$$

$$\boldsymbol{x} \geq \boldsymbol{0} \tag{6.27}$$

$$\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in Y \subset \mathbb{R}^m \tag{6.28}$$

be a generic linear problem, and let the following be the slave problem on the $\boldsymbol{x}$ variables obtained via BD by dualising constraints 6.26

$$\min \phi_{\mathcal{S}} = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} \tag{6.29}$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} - \tilde{\boldsymbol{y}} \tag{6.30}$$
$$\boldsymbol{x} \geq \boldsymbol{0} \tag{6.31}$$

Let its dual be

$$\max \nu \ = (\boldsymbol{b} - \tilde{\boldsymbol{y}})^{\mathsf{T}}\boldsymbol{\xi} \tag{6.32}$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{\xi} \leq \boldsymbol{c} \tag{6.33}$$

and let $\Xi = \{\boldsymbol{\xi} \in \mathbb{R}^m : \boldsymbol{A}\boldsymbol{\xi} \leq \boldsymbol{c}\}$ be the set of feasible solutions. The master is

$$\min \phi_{\mathcal{M}} = \boldsymbol{e}^{\mathsf{T}}\boldsymbol{y} + \psi \tag{6.34}$$
$$\text{s.t. } \boldsymbol{y} \in Y \tag{6.35}$$
$$\psi \geq (\boldsymbol{b} - \boldsymbol{y})^{\mathsf{T}}\boldsymbol{\xi}_b \quad \forall b \in B \tag{6.36}$$

where equations 6.36 are Benders optimality cut.

We further introduce the following definitions.

Definition 1 (Bertsekas, 1999)**.** *Let $C \subset \mathbb{R}^m$. The relative interior of C is the interior of its affine space:*

$$ri(C) = \{\boldsymbol{u} \in C \ : \ B(\boldsymbol{u}, r) \cap \mathit{aff}(C) \text{ for some } r > 0\}$$

Where $B(\boldsymbol{u}, r)$ is the ball centered in $\boldsymbol{u}$ with radius $r > 0$, and $\mathit{aff}(C) = \{\boldsymbol{v} \in \mathbb{R}^m \ : \ \exists \boldsymbol{\theta} \in \mathbb{R}^m, \boldsymbol{u} \in C :$ is the affine space of C.

In convex optimisation the relative interior $ri(C)$ is often used in place of just the interior $int(C)$ to account for cases in which the set C has a dimension $n < m$ smaller than the embedding space $\mathbb{R}^m$. In those cases the interior of its containing affine space is used. Otherwise we have $\text{aff}(C) = C$ and $ri(C) = int(C)$.

By definition, it follows that

Definition 2 (Bertsekas, 1999)**.** *For a polyhedron $C = \{\boldsymbol{x} \in \mathbb{R}^m_+ \ : \ \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \ \boldsymbol{b} \in$*

$\mathbb{R}^h,\ \boldsymbol{A} \in \mathbb{R}^{h \times m}\}$ *the relative interior is*

$$ri(C) = \{\boldsymbol{x} \in C \ : \ \boldsymbol{A}x < \boldsymbol{b}\}$$

Then let Y^C be the convex hull of the master polytope Y. In the MW method we consider the relative interior of Y^C and call each $y \in ri(Y^C)$ a *core point* of Y.

Let $\boldsymbol{\xi}^*$ be an optimal solution to model (6.32)-(6.33) and $\boldsymbol{y}^C \in ri(Y^C)$ be a core point of Y, the following is the MW separation problem

$$\max \hat{\nu} = (\boldsymbol{b} - \boldsymbol{y}^C)^\mathsf{T}\boldsymbol{\xi} \tag{6.37}$$

$$\text{s.t. } (\boldsymbol{b} - \tilde{\boldsymbol{y}})^\mathsf{T}\boldsymbol{\xi} = \nu(\boldsymbol{\xi}^*) \tag{6.38}$$

$$\boldsymbol{A\xi} \leq \boldsymbol{c} \tag{6.39}$$

Model (6.37)-(6.39) searches for the point in Ξ that lies on the optimal facet of problem (6.32)-(6.33) and is closest to the interior point $\boldsymbol{y}^C \in Y^C$.

Then, a dominance relation between cuts is defined.

Definition 3 (Cut dominance)**.** *A cut (6.21) with coefficients $\boldsymbol{\xi} \in \Xi$ dominates a cut with coefficients $\boldsymbol{\xi}' \in \Xi$ if and only if $(\boldsymbol{b}-\boldsymbol{y})^\mathsf{T}\boldsymbol{\xi} \geq (\boldsymbol{b}-\boldsymbol{y})^\mathsf{T}\boldsymbol{\xi}'$ $\quad \forall y \in Y$ with strict inequality for at least one point $y \in Y$.*

Definition 4 (Pareto-optimal cut)**.** *A cut (6.21) with coefficients $\boldsymbol{\xi} \in \Xi$ is Pareto-optimal if it is not dominated by any other cut.*

In (Magnanti and Wong, 1981) it is proved that MW cuts are Pareto-optimal.

The method requires generating a new core point at each iteration to guarantee Pareto-optimality. In the literature (Papadakos, 2008) one of the most used technique is to compute a convex combination between the current solution and the interior point employed in the last iteration. At the first iteration the initial solution is used as a core point.

The Magnanti-Wong method has one significant limitation: in practice the constraint 6.38 is often numerically unstable and can lead the problem (6.37)-(6.39) to become numerically unbounded (Papadakos, 2008). In our algorithm when this issue is encountered the algorithm just adds the original Benders cuts.

6.1.2 Two-phases heuristic

Another technique used with Benders decomposition is the two-phases heuristic (Mercier et al., 2005). The general idea is to have a standard BD algorithm used in two phases. For MILP problems the scheme goes as follow. In the first phase integrality constraints in the master are relaxed. In the second phase the integrality constraints are restored. In every phase the BD algorithm is executed by adding Benders cuts until the desired convergence is reached. As cuts for the continuous relaxation are valid for the original problem as well, the scheme allows to improve the efficiency of the standard BD algorithm by quickly and cheaply computing cuts in the first phase, when dual information from the slave problem is scarcer, providing a "warm-up" phase to the standard algorithm implemented in the second phase.

6.2 Computational results

We conducted a series of experiments on ten "weekly" instances of 168 hours obtained from a scenario hypothesis from RSE S.p.A. for Italy in 2011. The experiments were implemented with AMPL 20081120 and CPLEX 12.6 on a Linux laptop with 4GB RAM and 2.7GHz quad-core processor. For comparison, we report the instances require around 10 minutes to be solved by CPLEX MIP solver on the same system.

The scheme initialises the Benders decomposition algorithm with the IACR+CD solution sec.4.1.1. The BD algorithm follow the two-phases scheme. We evaluated the algorithm with and without applying the MW cuts at each iteration. The procedure is then described by a triple of parameters (k, c, m) where k is the number of iterations in the 1st phase, with a fractional master problem, c the number of iterations in the 2nd phase, with the integral master problem, and $m \in \{N, Y\}$ is the MW flag such that $m = Y \iff$ MW cuts are computed. We considered 6 possible combinations of parameter values, yielding 60 different tests.

To asses the complexity required to solve each subproblem, for each iteration in the BD scheme we consider the time spent by the CPLEX solvers to fathom each subproblem. Then we express these times as percentages of the total solver time of each iteration. We report these values in figure 6.2 as boxplots for each subproblem. We considered four subproblems: the continuous relaxation of the master problem (in the first phase), the integral master problem (in the second phase), the slave subproblem and, if present, the Magnanti-Wong subproblem. Indeed, the charts

show that the slave subproblem is, by large, the easiest subproblem to solve among the others. The master subproblem is the most expensive subproblem to solve, because of both integrality and the cuts being added at each iteration. Finally, the Magnanti-Wong problem, despite being continuous and sharing most of the structure with the slave subproblem, requires significant effort to solve, probably because of its numerical instability.

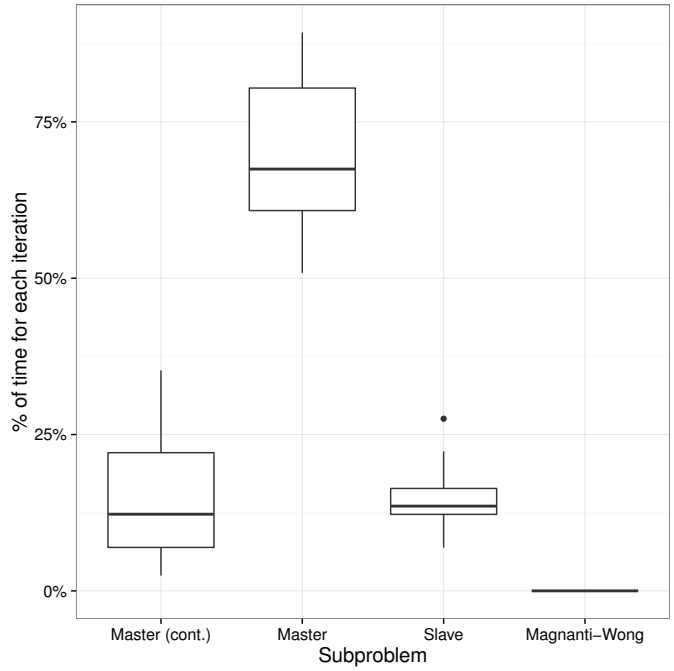
In table 6.1 we report the initial gap obtained from the heuristic and its computation time for the ten instances. In table 6.2 we report for each configuration, across instances, the following indicators: average total computing time, average number of failed MW cut computations (due to numerical unboundness), number of instances for which the upper bound improved, average gap, in percentage, between the best bounds computed by the algorithm, which estimates the distance of the algorithm from termination, and average improvement over the upper bound and the lower bound, in percentage term, compared with the initial values from the heuristic .

For illustration purposes, in figures 6.3–6.8 we report the series of primal and dual bounds during the BD algorithm on the week-10 instance with the six parameter configurations reported in table 6.2. In each figure we report the upper and lower bound computed by the algorithm. The dashed lines represent the initial value of the respective bound from the CD heuristic. Bounds are reported with light grey for the first phase and dark grey for the second one. The charts suggest the algorithm encounters a "plateau" after the first iterations, where the dual bound "stalls", i.e. remains almost constant. Once the plateau is overcome, both bounds start improving again. This agrees with our previous experience with CG, where we encountered numerical issues due to degeneracy in the master problem.

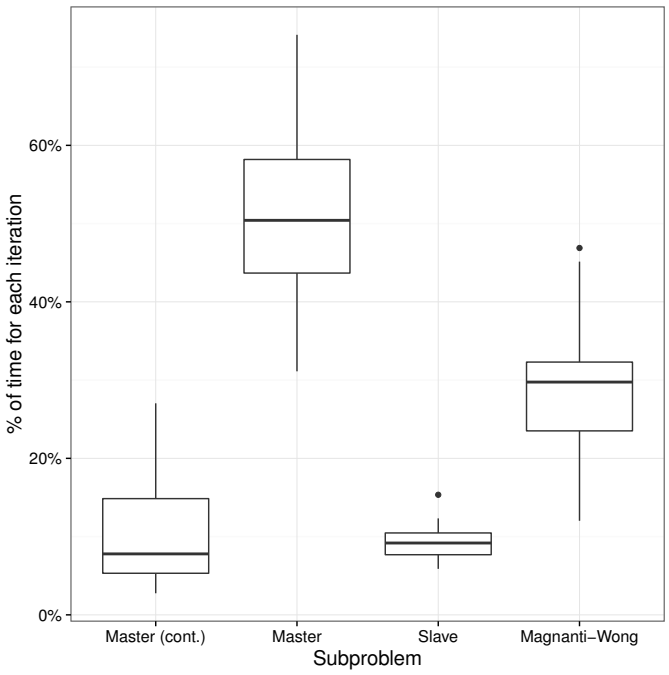
Table 6.1: Initial heuristic: gap and computation time

Instance	Gap %	Time (s)
week-5	1.28	11
week-10	1.42	11
week-15	1.56	11
week-20	2.23	10
week-24	1.03	11
week-30	0.93	11
week-35	0.83	11
week-40	1.27	11
week-45	1.54	11
week-50	1.30	12

Results in tab.6.2 show the most effective configurations for the BD algorithm



(a) Without MW cuts

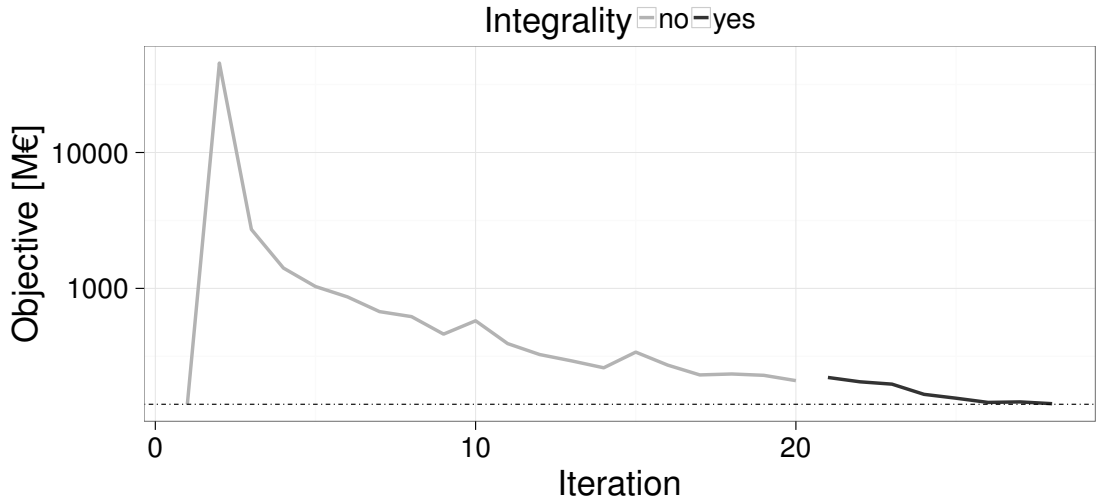


(b) With MW cuts

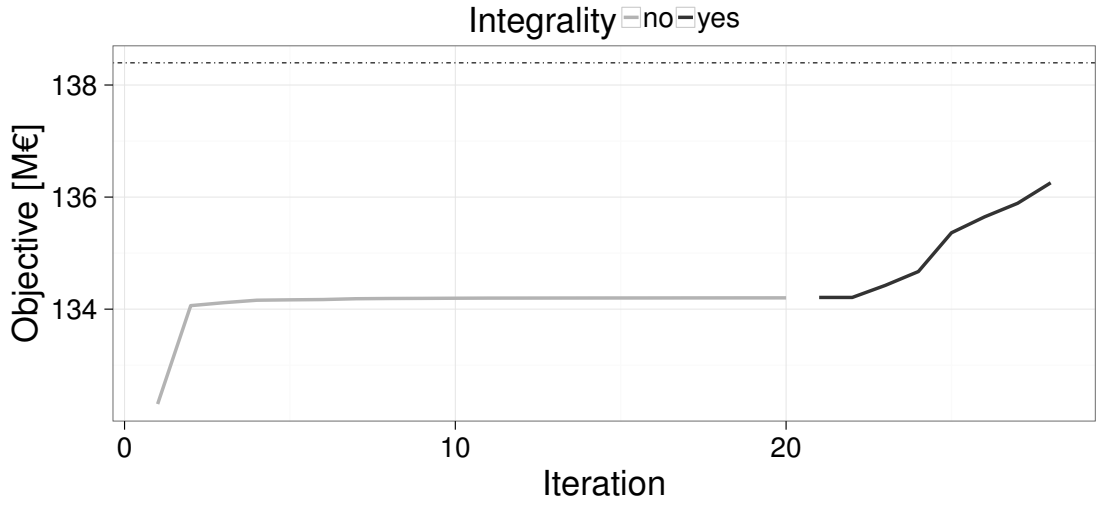
Figure 6.2: Boxplots of solver time required for each subproblem in each iteration, expressed as percentage over the total solver time of the iteration.

Table 6.2: Results for different configurations across instances.

1^{st} phase	2^{st} phase	MW cuts	avg. Time (s)	avg. MW failures	# UB improvements	avg. BD Gap %	avg. UB improvement %	avg. LB improvement %
20	8	N	959	–	1	27.4	-23.3	-2.1
20	8	Y	1460	4.5	1	30.5	-26.2	-2.2
20	16	N	1994	–	2	12.8	-9.5	-1.7
20	16	Y	2787	9.2	4	11.6	-8.4	-1.6
40	8	N	2344	–	3	12.1	-9.6	-1.0
40	8	Y	3264	18	5	4.7	-2.2	-1.0

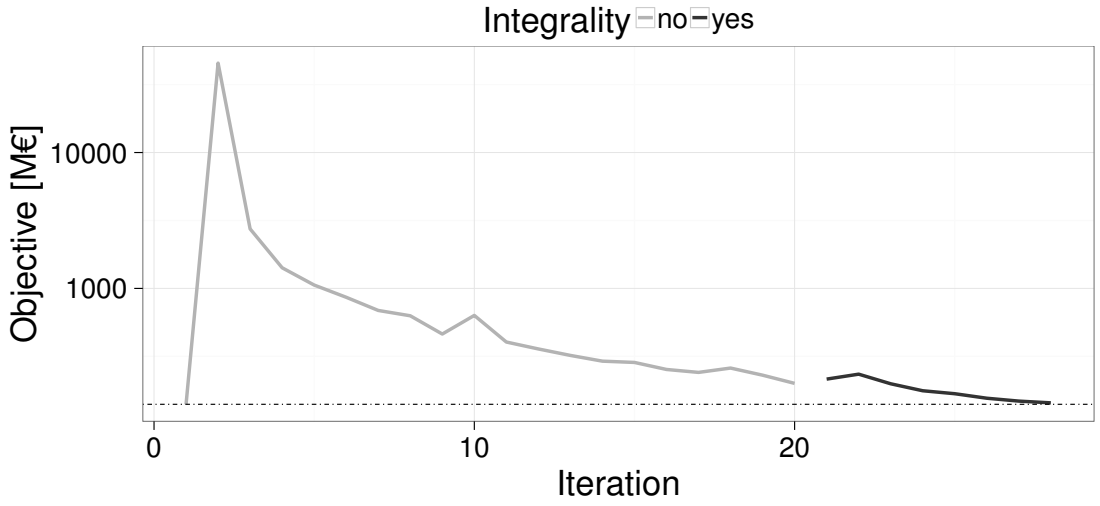


(a) Slave

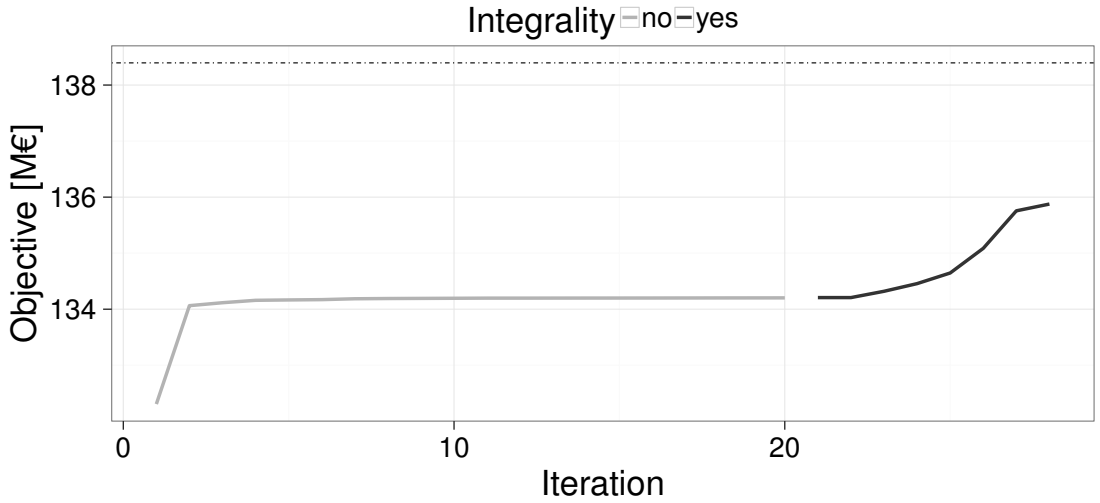


(b) Master

Figure 6.3: Bounds for the BD algorithm on week-10 with configuration $(k, c, m) = (20, 8, N)$.

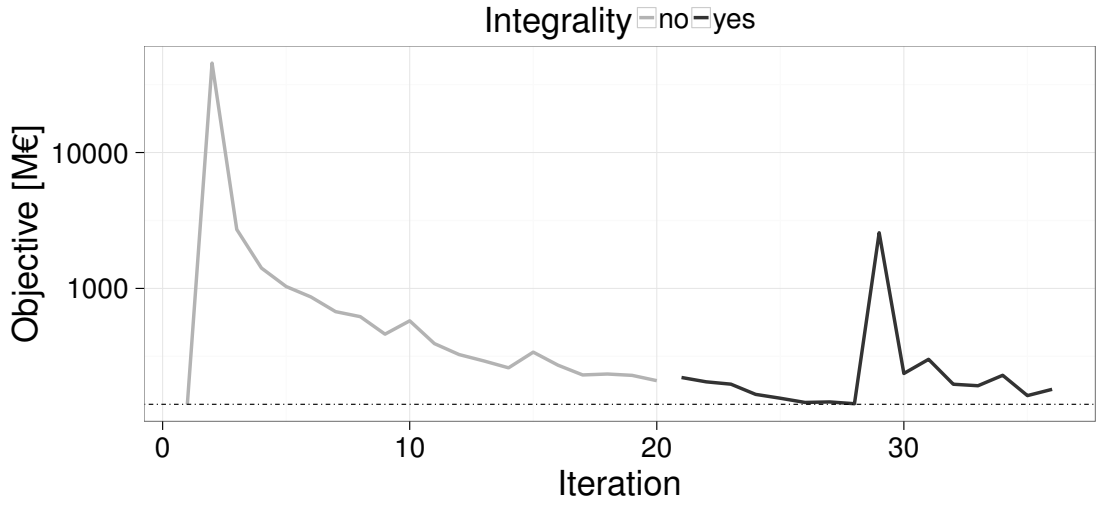


(a) Slave

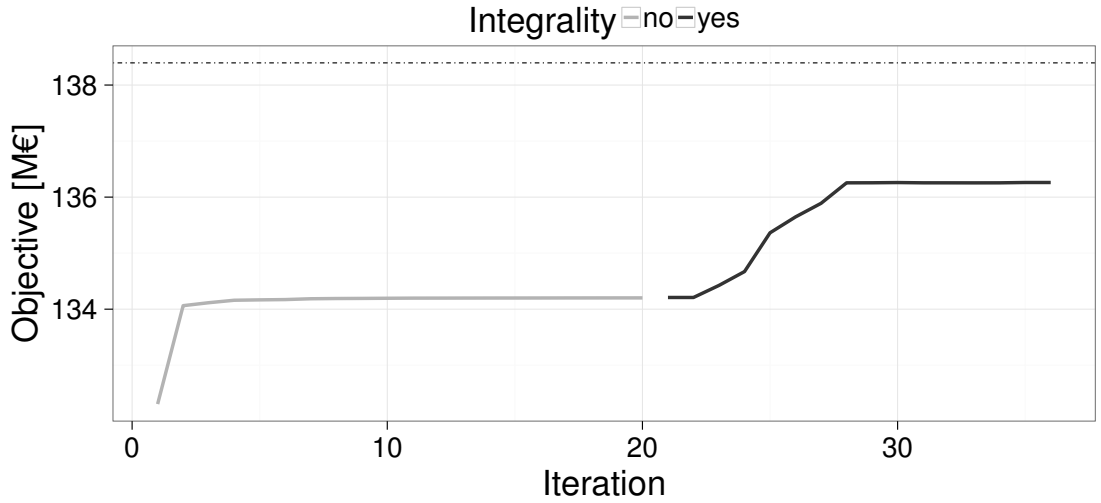


(b) Master

Figure 6.4: Bounds for the BD algorithm on week-10 with configuration $(k, c, m) = (20, 8, Y)$.

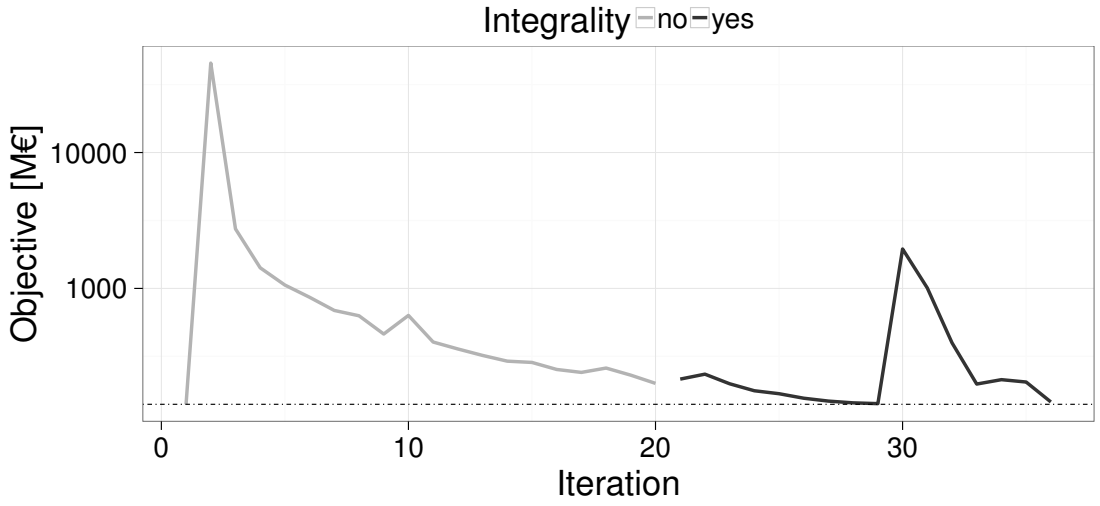


(a) Slave

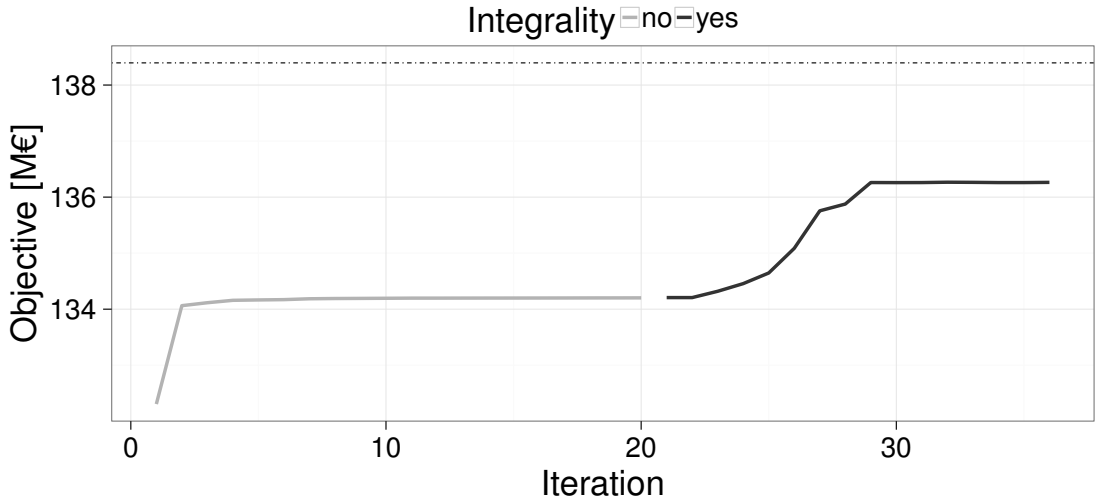


(b) Master

Figure 6.5: Bounds for the BD algorithm on week-10 with configuration $(k, c, m) = (20, 16, N)$.

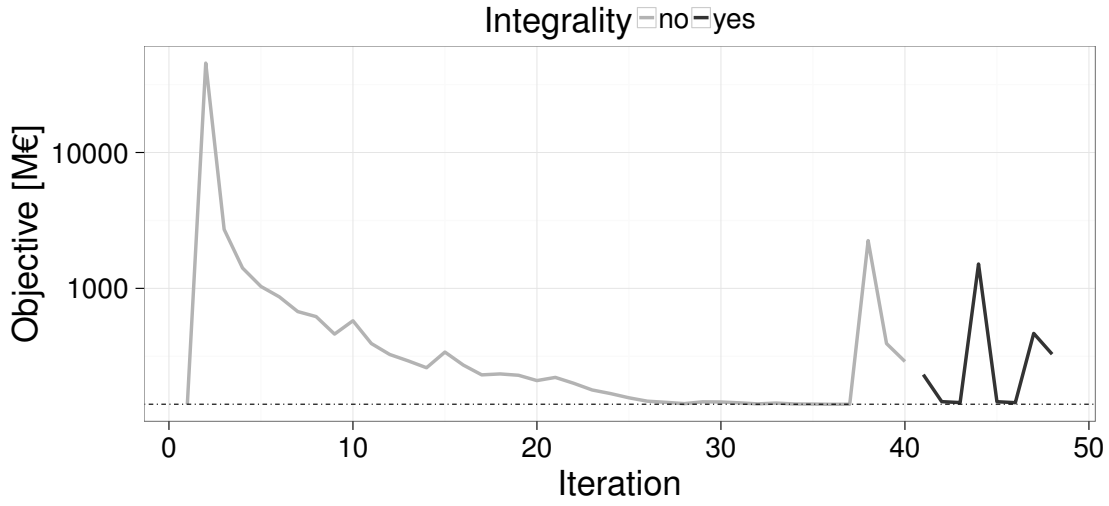


(a) Slave

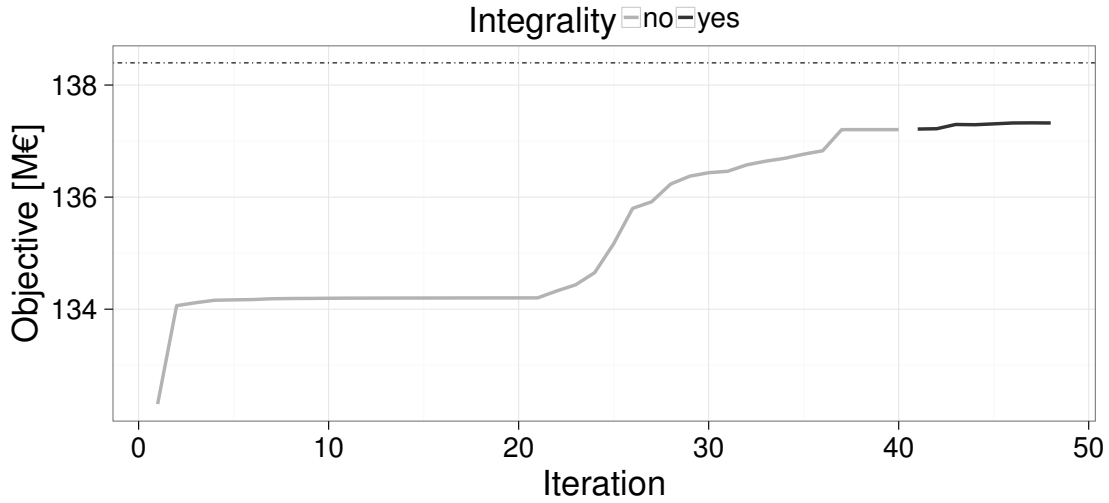


(b) Master

Figure 6.6: Bounds for the BD algorithm on week-10 with configuration $(k, c, m) = (20, 16, Y)$.

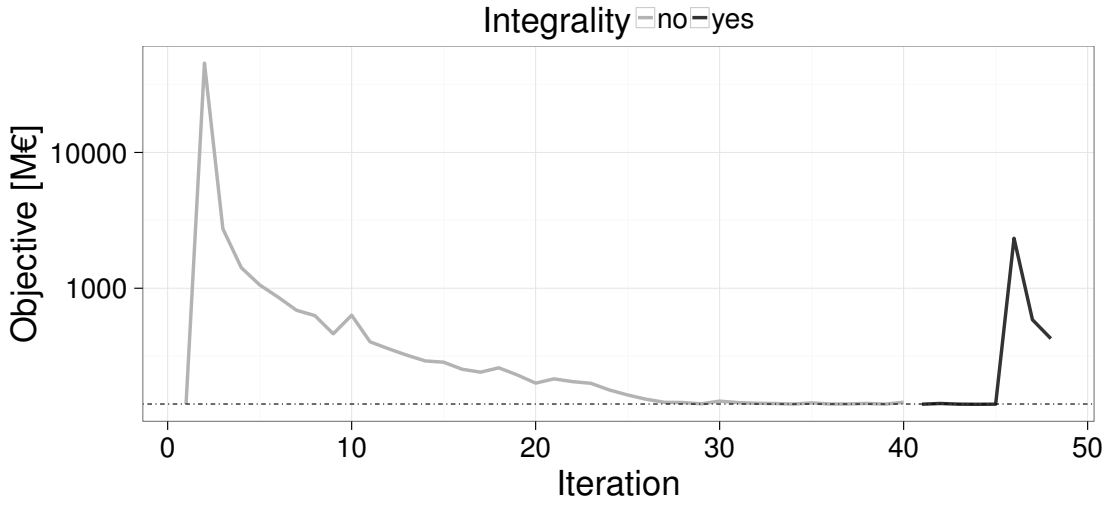


(a) Slave

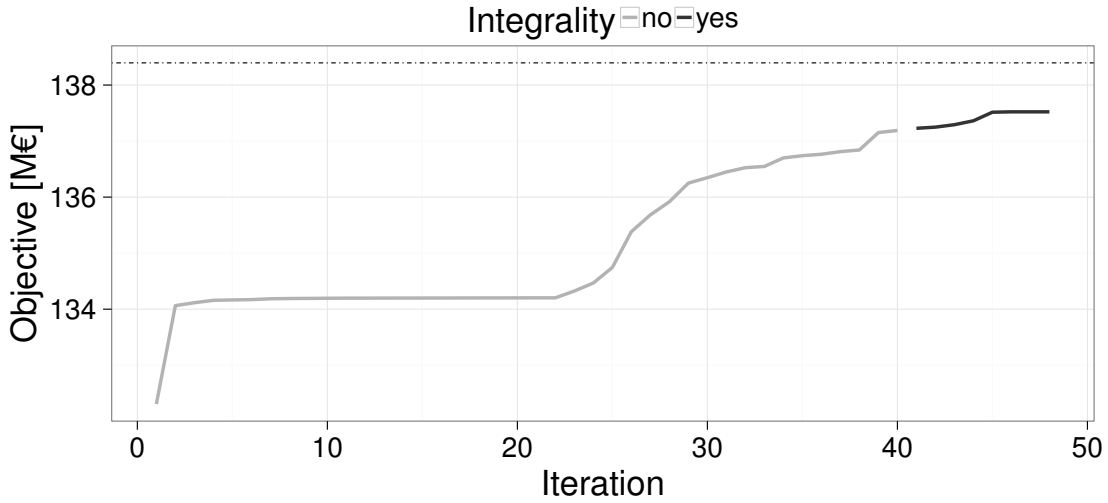


(b) Master

Figure 6.7: Bounds for the BD algorithm on week-10 with configuration $(k, c, m) = (40, 8, N)$.



(a) Slave



(b) Master

Figure 6.8: Bounds for the BD algorithm on week-10 with configuration $(k, c, m) = (40, 8, Y)$.

are those employing more iterations for the first phase. The phenomenon can be explained considering that for this type of UCP model the integrality gap can be quite small, especially for instances with longer horizons. Hence optimality cuts for the continuous relaxation can be quite effective for the original problem.

The presence of degenerate solutions and flatness in the objective function imply the dual information obtained from decoupling commitments of power plants and their dispatching, either through Benders Decomposition or Column Generation, can be insufficient to efficiently determine solutions with increasing accuracy. On the other hand, table 6.2 shows MW cuts, by exploiting geometric characteristics of the problem's polytope, can be quite effective, especially once the initial plateau is overcome, to improve the algorithm's convergence, despite the fact that the separating problem can fail 20% to 50% of the times.

Chapter 7

Analysing heuristic solutions and complexity

Our experiments showed that solving the medium-term large-scale UCP to great accuracy is challenging. Even just the solution of the continuous relaxation of the original model, or the continuous relaxation of the restricted master problem in (5.1)-(5.6) are time consuming and commercial solvers can easily encounter numerical problems when solving them. In this chapter we further investigate two subproblems: solving large-scale LPs to estimates production levels for thermal power plants (section 7.1) and finding good values for the integer variables representing the commitments of thermal units (section 7.2). Finally, we propose an alternative modelling approach (section 7.3) to circumvent these problem while still providing a reliable simulation of the power system.

7.1 Optimising continuous variables

During the solution of large-scale LPs state-of-the-art solvers encountered difficulties that slowed or prevented their convergence.

These problems occur at each CG iteration during the execution of the LP solver. After each new iteration of CG, the solution time required by the LP solver to solve the master problem increases significantly. These numerical instabilities are not a symptom of the well-known "instability" of column generation algorithms (Ben Amor et al., 2009; Briant et al., 2008; Gondzio et al., 2013). Indeed, the problem here is not stabilising the dual bound or the value of dual variables. As an example, we report here the results for the first 6 iterations of the CG algorithm on

the 6^{th} monthly instance with the barrier solver with crossover. The implementation used is the same as in the experimental section 5.2.1 and required 20 minutes to run on a 2 core PC with 4GB of RAM. In fig.7.1 we report the corresponding values for the master and dual bounds in each of the six iterations. It can be seen that the dual bound does not show signs of instability. To show the lack of significant instability in the dual variables we defined the following descriptors for multipliers $M = (\mu_{tzg})$ and $\Lambda = (\lambda_{tzg})$ $\forall t \in T, z \in Z, g \in G_z$.Without loss of generality, we refer to these descriptors for multipliers λ_{tzg}. Let $I = 1 \dots 6$ be the six iterations of the test and λ^i_{tzg} be the corresponding multiplier at the i-th iteration. We exclude from the computation of the descriptors the first iteration, in which multipliers are expected to assume large values and would easily skew the results. We define the *average jump* for λ_{tzg} as

$$\delta^{\lambda}_{tzg} = \mathbb{E}[|\lambda^i_{tzg} - \lambda^{i-1}_{tzg}| \, i \in I : i > 2]$$

and, given the average multiplier for $\bar{\lambda}_{tzg} = \mathbb{E}[\{\lambda^i_{tzg} \, i \in I : i \geq 2\}]$, the relative jump as

$$\hat{\delta}^{\lambda}_{tzg} = \frac{\delta^{\lambda}_{tzg}}{\bar{\lambda}_{tzg}}$$

The former gives an absolute measure of the instability of the multiplier, the latter compares it to the magnitude of the multiplier.

In fig.7.2 for each set of multipliers we report the density of the bivariate distribution $(\delta_{tzg}, \hat{\delta}_{tzg})$, of the absolute and relative jumps. We can infer that the multipliers are, on average, significantly small (magnitude of 1×10^{-6}) compared to the objective value (magnitude of 1×10^{2}). Less than 2.5% of the multipliers for both sets have significant values for both average jump ($\delta \geq 1 \times 10^{-4}$) and average relative jump ($\delta \geq 0.5$).

These tests and the way the instability occurs in the LP solver indicate this instability problem affects the LP solver and differs from the "instability" of bounds and dual variables often referred in CG literature and, thus, it is not solvable with the corresponding stabilisation methods.

Probable causes of these numerical difficulties are the flatness of the objective function and the presence of degenerate solutions. The former can be observed considering that the coefficients of the objective function in each period are quite similar. In particular production costs are supposed to remain stable within a week or more. Furthermore, we can verify a posteriori that annual solutions have zonal

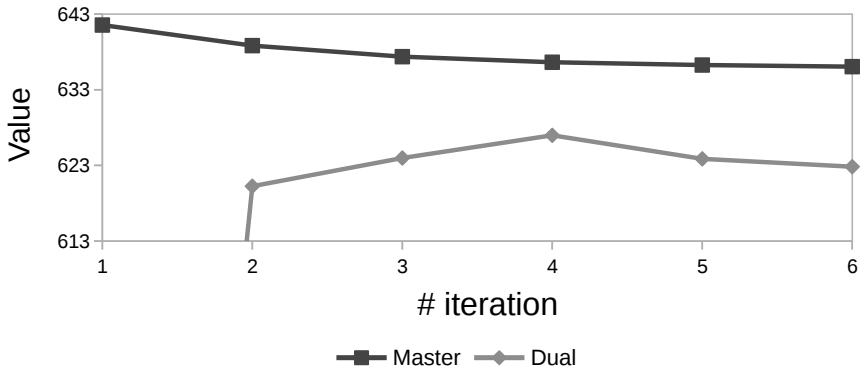


Figure 7.1: Master and dual bounds for monthly instance.

energy prices eight to nine orders of magnitude smaller than the objective value (see figure 5.2). These factors imply that the difference between the choice of producing one additional power unit at a given time and the one of producing it at another can be quite small. In general, the impact of such choice becomes smaller and smaller as the length of the planning horizon grows.

The latter has been verified empirically in two ways First, logs for experiments in 5.2.1 using CPLEX primal and dual simplex implementations showed signs of stalling which, according to CPLEX manual (IBM, 2011), is caused by degeneracy. Finally, we analysed a IACR basic solution of the 12 month instance of Italy 2011 section 3.3 counting two types of variables: basic variables at bounds and non-basic variables with zero reduced cost. We solved the model with the CPLEX 12.6 dual simplex algorithm with optimality tolerance 1×10^{-9} and reduced cost tolerance 1×10^{-7} Results are reported in table 7.1.

Variables	3,827,057
Basic variables at bounds (relative tolerance for bounds: 1×10^{-6})	10,727
Non-basic variables with zero reduced cost (threshold for reduced cost: 1×10^{-7})	674,353

Table 7.1: Measured degeneracy in IACR basic solution. A basic variable is considered at a bound if the relative difference between its value and the bound is below the tolerance. A reduced cost is considered zero if its value is below the threshold.

Among the basic variables at bounds and the non-basic variables with zero reduced costs there are those representing hydroelectric and thermal power production. This implies, as the formulation suggests, that given an optimal solution it is possible to compute several other ones with the same value, or basis, by exchanging

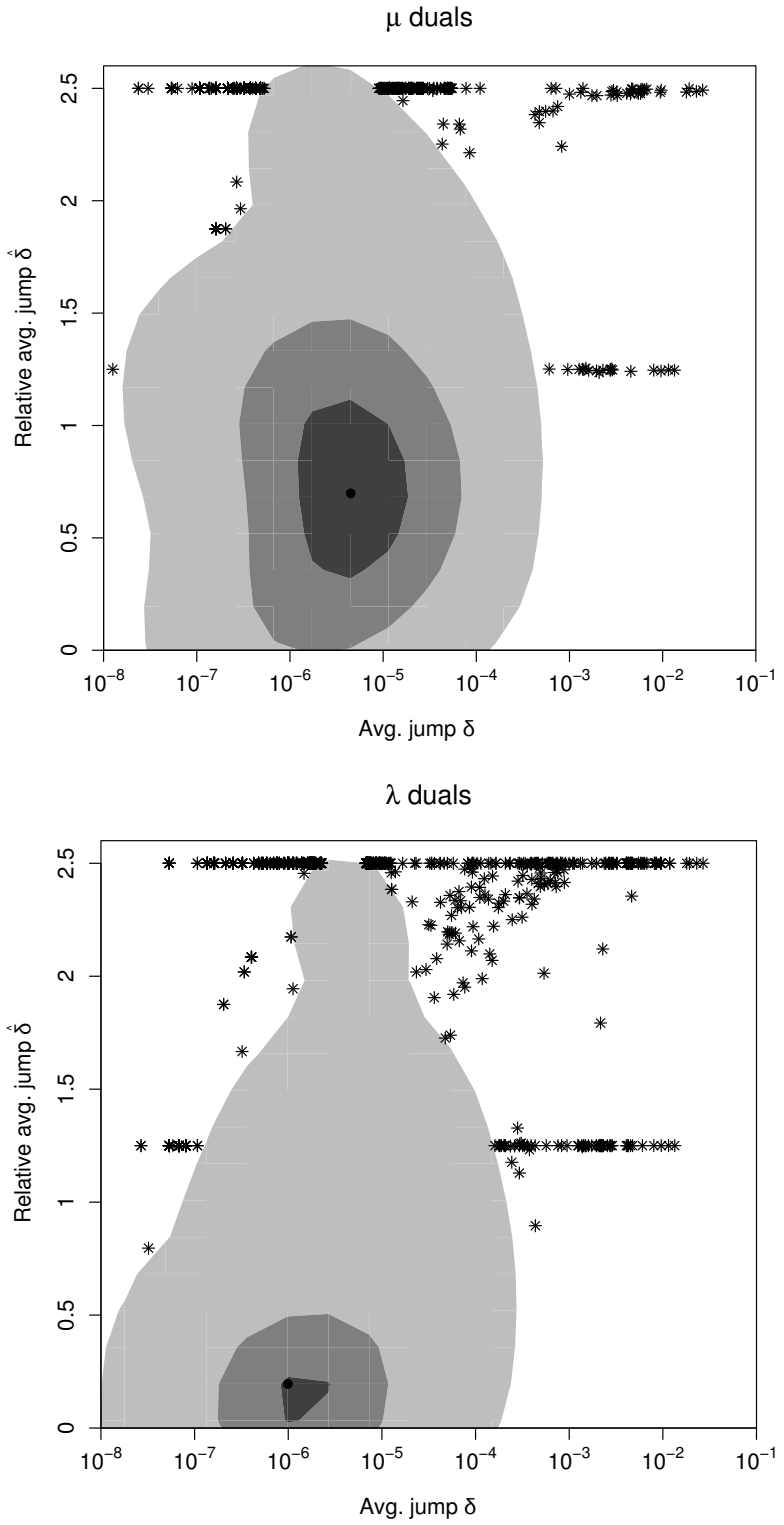


Figure 7.2: Density plot for average jumps.
Black dot represents the mean. Shaded areas from darker to lighter correspond to 25%, 75%, and 97.5% of closest points around the mean. Crosses represent outliers farther from the mean than the remaining 97.5% of points.
Points with $\delta < 1 \times 10^{-8}$ were omitted to improve readability.

hydroelectric and thermal production among different periods and zones.

In the literature there are several attempts at devising better LP solvers, based either on Interior Point Methods (IPM) or simplex algorithms, to solve large-scale problems more efficiently by exploiting their structure or directly tackling instability causes, like poor numerical conditioning and degeneracy. We remark that these solvers often do not provide a ready, generic implementation that can be easily deployed in industrial settings like in RSE S.p.A.. We considered the following two proposals of new LP solver designs for our problem:

BlockIP(Castro, 2014): an IPM designed to efficiently solve sparse, large-scale block-angular convex separable problems. Indeed, our large-scale UCP exhibits such an ideal structure. The algorithm exploits sparsity to solve a specific subset of the equations in the problem. The approach relies on some properties of the corresponding sub-matrices, in particular their spectral radius, hence the effectiveness of the method is, in general, problem-dependent. The authors report the algorithm to be significantly more efficient than CPLEX 12.5 barrier solver for large problems obtained from the literature. The authors provide a public C++ library which allows to input models and invoke the solver. We applied the algorithm to the raw formulation of our model but it showed numerical instabilities and slow convergence, likely due to formulation artefacts such as big-M constants, which prompt for an ad-hoc preprocessing phase to apply the algorithm effectively.

Space Vector Decomposition(Desrosiers et al., 2015): an algorithmic framework that unifies and generalise well-known simplex algorithms like the Improved Primal Simplex (IPS). The authors are currently trying to obtain new insights on degeneracy in simplex algorithms. Among those, the IPS could be a general approach suitable for our problem. Currently there are no generic public-available libraries for it, therefore a problem-dependent implementation is needed.

7.2 Optimising integer variables

Most of the complexity of the UCP is due to the presence of integer variables. Once they are fixed the problem reduces to a much simpler economic dispatch problem, as model (4.13)-(4.16), and can be solved efficiently by commercial solvers even for large-scale instances, as we verified in chapter 4.1.2.

It can be noticed, however, that the value of quasi-optimal solutions (gap $\leq 1\%$)

computed by the rounding heuristic in section 5.2.1 is close to the value of the continuous relaxation provided by the IACR model (4.1)-(4.9). This could then suggest that fractional and integer solutions are similar, in particular that fractional solutions have few violations of the constraints with integer variables 3.2–3.12. It could therefore be possible to design better algorithms to compute high-quality primal solutions by selectively fix those violations.

In the following we report an analysis on the solutions obtained by relaxing the constraints on integer variables in the original problem (3.1)-(3.21). We considered two relaxations: (i) the relaxation obtained by removing minimum up-down constraints 3.5–3.12 and (ii) the IACR relaxation.

7.2.1 Flexibility of power plants

We compared the results of our previous experiments in 5.2.1(*Standard Scenario*) with the ones that can be obtained by relaxing the flexibility constraints (*Flexible Scenario*). In the latter case we ran both IACR+CD and RH as tested in subsection 5.2.1 but setting $\tau^{\mathrm{ON}} = \tau^{\mathrm{OFF}} = 1$ for all subgroups. In Table 7.2 we report the main comparison results for the Flexible Scenario with the standard one in terms of relative difference between the following values: the IACR+CD primal bound, its computation time and the RH[5] primal bound. Numerical stability problems were encountered during the column generation process while solving instances 1.1, 1.6, 4.1 and 6.2, even with the use of the BF strategy; they are marked with a '*' as before.

The Flexible Scenario turns out to be much easier to optimise than the standard one, as the IACR+CD heuristic manages to save more than 40% of computation time on average. However, the improvement in the best IACR+CD and RH solution values is limited: about 2.5% on average, and never more than 3.2%. Hence, it would be tempting to conclude that the flexibility constraints are automatically respected, even if a relaxed formulation is optimised. However, while we observed that only 10% of constraints (3.9)–(3.12) are violated on average in the Flexible Scenario solutions, the number of violations in some zones actually reaches 80% of the constraints across the whole optimisation period.

Indeed, the number of flexibility constraint violations depends on the structural properties of the system. In our test cases, for example, the 80% violations figure comes from the region of Sardinia, characterised by relatively low demand and several plants with low flexibility, $\tau^{\mathrm{ON}} = 60$ and $\tau^{\mathrm{OFF}} = 20$, and low marginal costs. In this

Id	Size	Difference with Standard Scenario		
		IACR+CD	Time	RH
*1.1	1 month	–	–	–
1.2		2.6%	43.0%	0.5%
1.3		3.0%	26.9%	0.5%
1.4		2.9%	54.1%	0.6%
1.5		2.2%	9.4%	0.5%
*1.6		–	–	–
1.7		1.8%	46.2%	0.2%
1.8		2.8%	51.1%	0.3%
1.9		1.9%	48.0%	0.2%
1.1		3.2%	24.9%	0.5%
1.11		3.2%	13.3%	0.4%
1.12		2.9%	42.5%	0.5%
2.1	2 months	2.5%	30.2%	0.4%
2.2		3.1%	39.5%	0.5%
2.3		2.6%	4.6%	0.4%
2.4		2.2%	31.4%	0.3%
2.5		2.6%	55.5%	0.4%
2.6		2.7%	23.6%	0.5%
3.1	3 months	2.8%	41.5%	0.5%
3.2		2.7%	60.1%	0.5%
3.3		2.1%	71.8%	0.3%
3.4		2.9%	6.5%	0.5%
*4.1	4 months	–	–	–
4.2		2.3%	44.1%	0.5%
4.3		2.5%	46.0%	0.5%
6.1	6 months	2.8%	215.4%	0.3%
6.2		–	–	–

Table 7.2: Results for the Flexible Scenario and comparison with the Standard Scenario

region the Flexible Scenario solution turns off plants whenever the demand is too low, thus, considering the inherent presence of peaks in the demand curve, it can hardly respect minimum up/down constraints. Other regions, instead, such as the Northern zones, are characterised by higher demand and more modern and flexible plants. For these zones, we observed fewer violations of the flexibility constraints in the Flexible Scenario. Explicitly handling flexibility is therefore necessary to obtain an effective simulation that highlights potential limits of the system.

7.2.2 Infeasibility in fractional solutions

We also looked at the fractional solution obtained by the IACR relaxation on a more complex, more realistic model provided by RSE on a scenario for Europe 2020, which includes the same countries as the one used in 2.2. This model is implemented in the latest version of sMTSIM (Siface et al., 2014) and differs from the original model (3.1)-(3.21) for having ramp-up and reserve constraints. Due to these constraints, that couple different power plants, and other requirements on the output produced by the simulation, the model does not group power plants as in 3.2.

We measured the number and extent of violations to the minimum up down constraints in the fractional solution. Given a thermal plant with technical minimum $p_{\min}$ and fractional production levels $(x_t)_{t\in T}^{\top}$ in the IACR solution, we define an approximated commitment $\mathbf{u} \in \{0,1\}^T$ s.t. $u_t = 1 \iff x_t \geq p_{\min}$.

Then, consider the longest intervals in the horizon where the commitment does not change, i.e. intervals

$$\delta = [t_a, t_b] \in T \text{ s.t. } (u_t = u_{t_a}\ \forall t \in \delta) \ \wedge\ (u_{t_a} \neq u_{t_a-1}) \ \wedge (u_{t_b} \neq u_{t_b-1})$$

and let $\Delta = \{\delta_j \subset T : \max(\delta_j) = \min(\delta_{j+1})+1\}$ be the ordered set of the intervals and $J = \{1 \dots |\Delta|\}$ be the set of interval indices in Δ. Then let $\Sigma = \{(s_j, l_j)\}_{j\in J}$ be the corresponding schedule as a list of couples (s_j, l_j) where $s_j = \min\{u_t : t \in \delta_j\}$ is the state of the plant and $l_j = |\delta_j|$ the interval length. Let

$$\hat{\tau} : \{0,1\} \mapsto \mathbb{N} : \hat{\tau}(s) = \begin{cases} \tau^{\text{OFF}} & s = 0 \\ \tau^{\text{ON}} & s = 1 \end{cases}$$

Thus an interval δ_j is feasible if and only if $l_j \geq \hat{\tau}(s_j)$. Let $\lambda_j = \frac{l_j}{\hat{\tau}(s_j)}$ be the length of an interval δ_j relative to the corresponding minimum up or down time.

In the model there are three classes of flexibility for thermal power plants: low ($\tau^{ON} = 50$,$\tau^{OFF} = 20$), medium ($\tau^{ON} = 10$,$\tau^{OFF} = 4$) and high ($\tau^{ON} = 1$,$\tau^{OFF} = 1$), i.e. complete flexibility, for which equations 3.5–3.12 are not binding.

In table 7.3 we report for each zone and flexibility class of thermal plants the following statistics on infeasible intervals: the median relative length λ, in percentage, the number of infeasible intervals and the number of plants having at least one infeasible intervals, along with the total number of thermal plants of the corresponding flexibility class. Plants without infeasible intervals or with high flexibility are not included in the statistics.

As an example, in figure 7.3 we plot the schedule of a French plant as follows: for each interval $\delta_j \in \Delta$ a vertical line of length l_j is plotted at the j-th position on the x-axis; the colour and the line type depends on the state s_j: a dashed light grey line for the "on" state and a solid dark one for the "off" state. Two horizontal lines of the respective type mark the minimum up and down times τ_{ON} and τ_{OFF}.

The fractional solution has a significant number of violated flexibility constraints. As with the Flexible Scenario 7.2.1, the amount of constraints violations depends on the structural relationship between demand and supply in each zone. However, contrary to the smaller and simpler system used in previous tests, the violations are spread over the system; in particular, their location cannot be circumscribed to a specific region as in Sardinia for the Flexible Scenario.

Zone	Flexibility class	Median % λ_j	Nr. of infeasible intervals	Plants with infeasible intervals	Total plants
AT	low	25.0%	814	1	1
CH	medium	50.0%	50	1	2
CS	low	5.0%	40	3	7
CS	medium	40.0%	130	2	7
DE	low	30.0%	1551	4	5
DE	medium	22.5%	6	1	2
FR	low	20.0%	1665	5	6
FR	medium	25.0%	562	14	14
NO	low	20.0%	2368	9	13
NO	medium	50.0%	256	5	36
SA	low	18.0%	1733	4	7
SL	low	14.0%	1082	1	2
SU	low	18.0%	2593	6	10
SU	medium	50.0%	909	6	14

Table 7.3: Infeasible intervals by zone and flexibility class

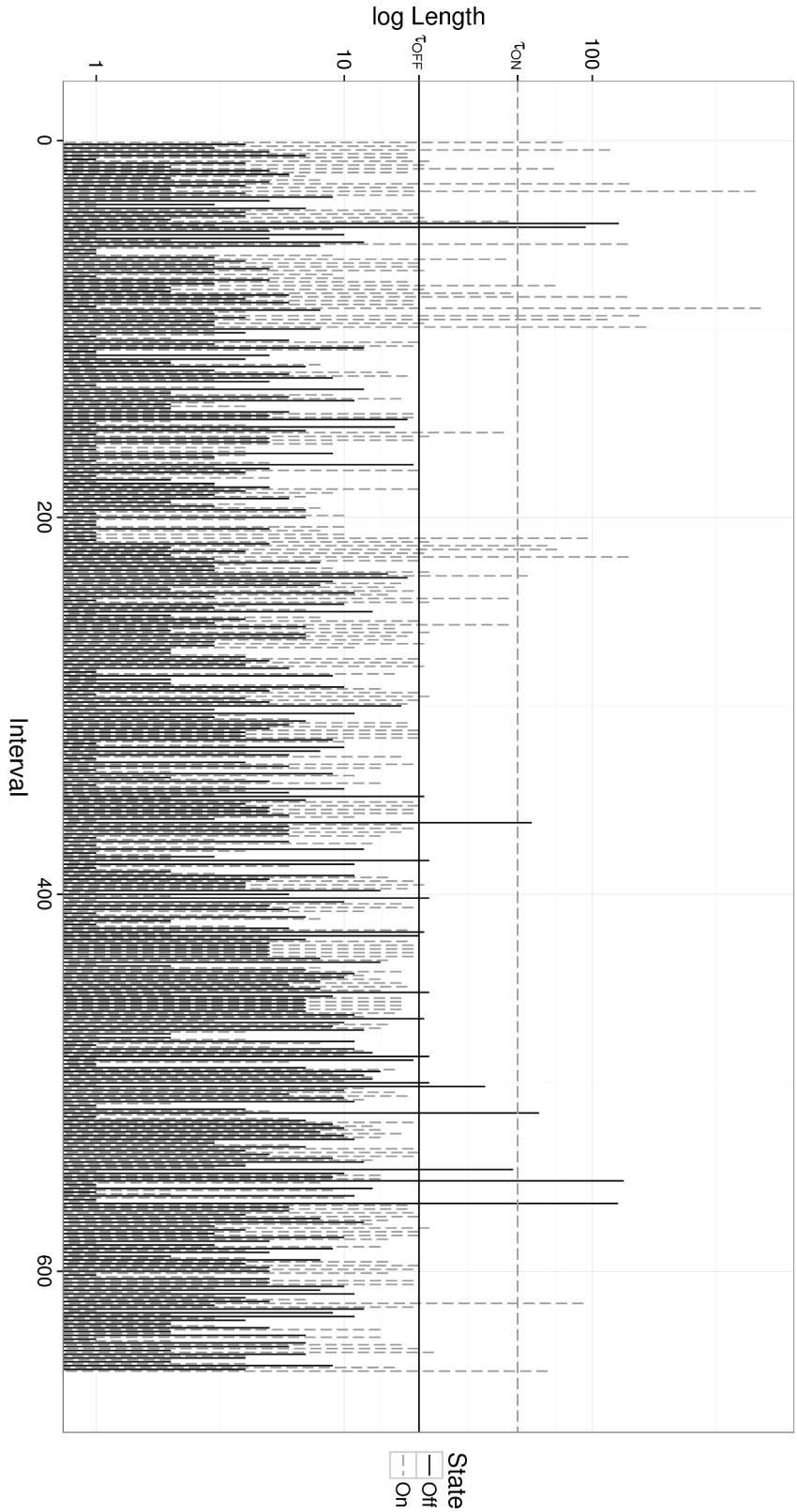


Figure 7.3: Fractional scheduling for a French plant with $\tau_{ON} = 50$ and $\tau_{OFF} = 20$.

7.2.3 Final remarks

We explained the similarity in the solution values between flexible and inflexible scenarios and between integral and fractional solutions as follows: our heuristic algorithms consistently produce solutions that are either optimal or very close to optimality (as shown in subsection 5.2.1). Many different solutions exist with similar near-optimal objective function values: some of them respect the flexibility constraints and some others do not. However, when relaxing the flexibility constraints, it is unlikely to find a solution that respects them. It seems even unlikely to find a heuristic that is able to repair relaxed solutions by fixing the violated constraints, because the problem of finding a feasible solution starting from a relaxed one that violates some flexibility constraints is $\mathcal{NP}$-hard in the general case. This stresses the need of modelling the flexibility constraints explicitly.

7.3 Alternative modelling options

To overcome the aforementioned issues alternative modelling approaches could be investigated. As an example, we suggest sampling the simulation horizon in shorter representative intervals, such as a few weeks for each weather season, and simulate each one independently from the others with stochastic programming, in which uncertainty in the inputs, especially demand and non-dispatchable production from wind and PV power plants, is represented explicitly in the model. Indeed, in the literature (Burstedde, 2012) there are models that simulate a limited set of representative days or weeks, from which the ideal behaviour of the system across the whole year can be inferred. This method yields smaller and sparser models that are therefore easier to optimise. The stochastic modelling in each interval would instead allow to consider a significant set of realisations of the uncertain parameters, in particular those that lead to extreme events in the system, e.g. a power discharge, which could otherwise be neglected by a deterministic simulation on short intervals, and that are crucial to obtain a reliable simulation.

Chapter 8

Conclusions

In this thesis, we have studied medium-term UCP in the context of fundamental energy market simulations, performed by our industrial partner RSE S.p.A.. The problem differs from the one found in the literature as it requires computing accurate solutions for large-scale UCP problems in short time and conventional computing power. We proposed a heuristic based on spatial decomposition and two exact bounding algorithms which extends from the ones proposed in the literature for MILP and UCP.

To the best of our knowledge, other studies on algorithms for the UCP do not address the specific characteristics of our problem and they cannot thus be employed, especially because they would be too inefficient to deliver solutions in affordable time with conventional computing power.

We performed extensive computational experiments to asses the most effective method to solve the problem to high accuracy and to characterise the properties that make the problem hard to solve by state-of-the-art algorithms for MILP problems. We could, indeed, improve the quality of real-world simulations for RSE S.p.A.in affordable time. However, the study has shown how the inherent structure and size of the model proves to be difficult to optimise to the desired accuracy, even just for the continuous part of the model. These numerical difficulties are caused by the inability of state-of-the-art linear solvers to distinguish between different commitments, and combinations thereof, when the size of either the system or the horizon becomes considerably large.

In the literature new approaches for solving highly-degenerate linear problems are being studied (Castro, 2014; Desrosiers et al., 2015), and they might improve the convergence and efficiency of linear solvers on problems like ours.

A better approach to obtain efficient and robust algorithms for the problem would be to adopt a more efficient formulation, which affords equivalent if not higher reliability by solving smaller or more robust problems. We propose solving stochastic models on representative samples of the simulation horizon. Further research is needed to assess the difference between the solutions obtained with the proposed approach and the one studied in this work, and to find methods that could reduce such difference, increasing the value of simulations for their intended purposes, i.e. provide reliable support to the analysis of domain experts.

Bibliography

ACER (2011). Framework guidelines on capacity allocation and congestion management for electricity. Technical report, Agency for the Cooperation of Energy Regulators (ACER).

Aissi, H., Bazgan, C., and Vanderpooten, D. (2005). Approximation complexity of min-max (regret) versions of shortest path, spanning tree, and knapsack. In *Algorithms – ESA 2005: 13th Annual European Symposium, Palma de Mallorca, Spain, October 3-6, 2005. Proceedings*, pages 862–873, Berlin, Heidelberg. Springer Berlin Heidelberg.

Belloni, A., Lima, A. L. D. S., Maceira, M. E. P., and Sagastizábal, C. A. (2003). Bundle relaxation and primal recovery in unit commitment problems. the brazilian case. *Annals of Operations Research*, 120(1):21–44.

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

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Computational Management Science*, 2(1):3–19.

Benini, M., Gelmini, A., L'Abbate, A., and Taverna, A. (2014). Valutazione dell'impatto al 2020 sul mercato elettrico italiano di un price coupling con i paesi vicini. In *AIET 2014*.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena scientific Belmont.

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

Briant, O., Lemaréchal, C., Meurdesoif, P., Michel, S., Perrot, N., and Vanderbeck, F. (2008). Comparison of bundle and classical column generation. *Mathematical programming*, 113(2):299–344.

Brouwer, A. S., van den Broek, M., Seebregts, A., and Faaij, A. (2014). Impacts of large-scale intermittent renewable energy sources on electricity systems, and how these can be modeled. *Renewable and Sustainable Energy Reviews*, 33:443 – 466.

Burstedde, B. (2012). The neuling model. Technical report, EWI Working Paper.

Castro, J. (2014). Interior-point solver for convex separable block-angular problems. Technical report, Università Politècnica de Catalunya.

Chang, G. W., Tsai, Y. D., Lai, C. Y., and Chung, J. S. (2004). A practical mixed integer linear programming based approach for unit commitment. In *Power Engineering Society General Meeting, 2004. IEEE*, pages 221–225 Vol.1.

Deepak, R. and Takriti, S. (2005). Minimum up/down polytopes of the unit commitment problem with start-up costs. *IBM Research Report*.

Desrosiers, J., Gauthier, J.-B., and Lübbecke, M. E. (2015). Vector space decomposition for linear programs. Technical report, GERARD HEC Montréal.

ENTSO-E (2013). 2030 visions scenarios. Technical report, European Network of Transmission System Operators (ENTSO-E).

EUROPEAN COMMISSION (2013). EU Energy, Transport and GHG emissions – trends to 2050. `http://ec.europa.eu/transport/media/publications/doc/trends-to-2050-update-2013.pdf`.

EUROPEAN COMMISSION (2015). Research challenges to Increase the Flexibility of Power Systems. `https://setis.ec.europa.eu/energy-research/sites/default/files/library/ERKC_PB_Flexibility.pdf`.

Eurostat (2015). Electricity production and supply statistics. `http://ec.europa.eu/eurostat/statistics-explained/index.php/Electricity_production_and_supply_statistics`.

Finardi, E. C. and da Silva, E. L. (2006). Solving the unit commitment problem of hydropower plants via lagrangian relaxation and sequential quadratic programming. *Power Systems, IEEE Transactions on*, 21(2):83–844.

Fischetti, M., Salvagnin, D., and Zanette, A. (2009). Minimal infeasible subsystems and benders cuts. *Mathematical Programming*.

Frangioni, A. (2006). Solving nonlinear single-unit commitment problems with ramping constraints. *Operations Research*, 54:775.

Frangioni, A., Gentile, C., and Lacalandra, F. (2007). Hybrid Lagrangian-MILP approaches for Unit Commitment Problems. Technical report, Istituto di Analisi dei Sistemi ed Informatica, CNR.

Frangioni, A., Gentile, C., and Lacalandra, F. (2009). Tighter approximated MILP formulations for unit commitment problems. *Power Systems, IEEE Transactions on*, 24(1):105–113.

G. Migliavacca (2009). Guida all'uso dell'interfaccia utente del simulatore MTSIM ed alla metodologia di creazione della base dati. Technical report, RSE s.p.a.

Gestore Mercato Elettrico (GME) (2014). 2014 excel historical data. `http://www.mercatoelettrico.org/en/download/DatiStorici.aspx`.

Gondzio, J., González-Brevis, P., and Munari, P. (2013). New developments in the primal–dual column generation technique. *European Journal of Operational Research*, 224(1):41–51.

Guan, X., Zhai, Q., and Papalexopoulos, A. (2003). Optimization based methods for unit commitment: Lagrangian relaxation versus general mixed integer programming. In *Power Engineering Society General Meeting, 2003, IEEE*, volume 2. IEEE.

Hedman, K. W., O'Neill, R. P., and Oren, S. S. (2009). Analyzing valid inequalities of the generation unit commitment problem. In *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, pages 1–6.

Hooker, J. N. (2009). Planning and scheduling by logic-based benders decomposition. *Operations Research*, 55(3):588–602.

IBM (2011). *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual*. `https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.1/ilog.odms.studio.help/pdf/usrcplex.pdf`.

Kirby, B. J. (2003). *Spinning reserve from responsive loads*. United States. Department of Energy.

Kjeldsen, N. H. and Chiarandini, M. (2012). Heuristic solutions to the long-term unit commitment problem with cogeneration plants. *Computers & Operations Research*, 39(2):269–282.

Lanati, F., Gelmini, A., Benini, M., and Gallanti, M. (2015). A methodology to assess the impact of 2030 eu climate and energy targets on the national power systems: The italian case. In *2015 12th International Conference on the European Energy Market (EEM)*, pages 1–5.

Leuthold, F. U., Weigt, H., and Hirschhausen, C. (2010). A large-scale spatial optimization model of the european electricity market. *Networks and Spatial Economics*, 12(1):75–107.

Lienert, M. and Lochner, S. (2012). The importance of market interdependencies in modeling energy systems – the case of the european electricity generation market. *International Journal of Electrical Power & Energy Systems*, 34(1):99–113.

Livera, A. M. D., Hyndman, R. J., and Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527.

Long Zhao and Bo Zeng (2012). Robust unit commitment problem with demand response and wind energy. In *Power and Energy Society General Meeting*, pages 1–8. IEEE.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

Magnanti, T. L. and Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484.

Maher, S. J., Desaulniers, G., and Soumis, F. (2014). Recoverable robust single day aircraft maintenance routing problem. *Computers & Operations Research*, 51:130–145.

Mercier, A., Cordeau, J.-F., and Soumis, F. (2005). A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451–1476.

Morales-Espana, G., Latorre, J. M., and Ramos, A. (2013). Tight and Compact MILP Formulation of Start-Up and Shut-Down Ramping in Unit Commitment. *IEEE Transactions on Power Systems*, 28(2):1288–1296.

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

Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters*, 36(4):444–449.

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

Rei, W., Cordeau, J.-F., Gendreau, M., and Soriano, P. (2008). Accelerating Benders Decomposition by Local Branching. *INFORMS Journal on Computing*, 21(2):333–345.

S. Salam (2007). Unit commitment solution methods. In *Proceedings of World Academy of Science, Engineering and Technology*.

Saharidis, G. K. D., Minoux, M., and Ierapetritou, M. G. (2010). Accelerating Benders method using covering cut bundle generation. *International Transactions in Operational Research*, 17(2):221–237.

Saravanan, B., Das, S., Sikri, S., and Kothari, D. P. (2013). A solution to the unit commitment problem : a review. *Frontiers in Energy*, 7(2):223–236.

Siface, D., Vespucci, M. T., and Gelmini, A. (2014). Solution of the mixed integer large scale unit commitment problem by means of a continuous stochastic linear programming model. *Energy Systems*, 5(2):269–284.

Spieker, C., Teuwsen, J., Liebenau, V., Muller, S. C., and Rehtanz, C. (2015). European electricity market simulation for future scenarios with high renewable energy production. In *PowerTech, 2015 IEEE Eindhoven*, pages 1–6. IEEE.

Taverna, A. (2011). Analisi e sperimentazione di algoritmi di programmazione intera per la simulazione annuale del sistema elettrico italiano. Master's thesis, Università degli Studi di Milano, Milan, Italy.

Tingfang, Y. and Ting, T. O. (2008). Methodological priority list for unit commitment problem. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 01*, Csse 08, pages 176–179. IEEE Computer Society.

Van den Bergh, K. and Delarue, E. (2014). Dc power flow in unit commitment models. Technical report, KU Leuven Energy Institute, TME Branch.

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

Yu, G. and Yang, J. (1998). On the robust shortest path problem. *Computers & Operations Research*, 25(6):457–468.

Zani, A., Migliavacca, G., Nemcek, P., Auer, H., de Castro, D. B., and Esterl, T. (2015). The ebadge approach to an intelligent pan-european balancing mechanism. *Electric Power Systems Research*, 120:80 – 86. Smart Grids: World's Actual Implementations.

Zeng, B. and Zhao, L. (2011). Solving two-stage robust optimization problems by a constraint-and-column generation method. *University of South Florida, FL, Tech. Rep.*

Zhang, L., Capuder, T., and Mancarella, P. (2016). Unified Unit Commitment Formulation and Fast Multi-Service LP Model for Flexibility Evaluation in Sustainable Power Systems. *IEEE Transactions on Sustainable Energy*, 7(2):658–671.