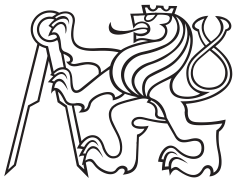
Dissertation Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

Long-term combined heat and power production and trade planning

Michal Dvořák

Supervisor: Ing. Petr Havel, Ph.D
PhD Programme: Electrical Engineering and Information Technology
Branch of Study: Control Engineering and Robotics
August 2016

Acknowledgements

First of all I would like to thank my supervisor Petr Havel for giving me an opportunity to work on such an interesting project, for his guidance, for his helpful comments on my work and for the encouragement and motivation, which helped me to complete this thesis.

Many thanks belong also to my mother for bringing me up and to my whole family for supporting me, foremost to my girlfriend Bára for her selfless patience and for motivating me to finish the work.

I would also like to express my gratitude to my colleagues Ondřej Novák and Jan Zábojník who provided me with conditions enabling smooth progress of my work. Furthermore they, with my other colleagues Ondřej Malík and Monika Hübnerová, created an enjoyable and inspiring working environment.

Declaration

This doctoral thesis is submitted in partial fulfillment of the requirements for the degree of doctor (Ph.D.). The work submitted in this dissertation is the result of my own investigation, except where otherwise stated.

I declare that I worked out this thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis. Moreover I declare that it has not already been accepted for any degree and is also not being concurrently submitted for any other degree.

Michal Dvořák
Prague, 18th August 2016

Abstract

In this thesis a comprehensive framework for solving long-term combined heat and power (CHP) operations planning problems is developed. The framework has two main parts - the first is a modelling framework which allows for modelling arbitrary CHP plants and formulation of a mixed-integer linear (MILP) optimization problem. The second is a solution algorithm which exploits the knowledge of the problem structure so that the problem is solved more efficiently.

There exist very powerful state-of-the-art general-purpose solvers for MILP problems, such as Gurobi. However, even these solvers fail to find a feasible solution within reasonable time for production planning problems of large dimensions. An idea followed in this thesis is to achieve reasonable computation times by employing the knowledge of the special problem structure.

For this purpose, a customized branch-and-bound (B&B) algorithm is proposed. The algorithm exploits the knowledge of the block-diagonal problem sub-structure, to obtain tighter bounds than a general-purpose B&B using could. Besides an enhanced horizon cutting algorithm is developed, with the purpose of providing high-quality feasible solutions for the customized B&B algorithm.

Efficiency of the proposed algorithm was evaluated based on 64 test cases using real-world data of three existing CHP plants. The performance of the proposed algorithm was compared to plain Gurobi usage. In most cases the proposed algorithm finds a certificate of near-optimality sooner than plain Gurobi does. More importantly, the proposed algorithm was able to find good feasible solutions for problems, for which Gurobi fails to find any feasible solution within the specified time limit.

Keywords: MILP, optimization, operations planning, CHP, Lagrangian relaxation, branch-and-bound, heuristics

Supervisor: Ing. Petr Havel, Ph.D

Abstrakt

V teto práci je představen komplexní framework pro modelování kogeneračních tepláren a pro plánování jejich provozu a obchodu. Framework lze rozdělit do dvou částí. Tou první je metodika modelování kogeneračních tepláren, která umožňuje postihnout libovolnou teplárnu. Model teplárny je poté využit k formulaci optimalizační úlohy pomocí smíšeného celočíselného lineárního (mixed-integer linear programming - MILP) programování. Druhou částí je algoritmus pro řešení těchto úloh, který využívá naší znalosti problému a který si dokáže poradit i s velkými instancemi problému.

Pro řešení obecných MILP úloh lze použít velice výkonných solverů. Ani ty nejvýkonnější solvery (jako je např. Gurobi) si však neporadí s velkými instancemi úlohy plánování provozu a obchodu tepláren. V této práci je navržen způsob, jak pomocí znalosti úlohy (struktury optimalizačního problému) zajistit řešitelnost těchto velkých úloh.

Je navržen algorithmus založen na metodě větví a mezí, který dokáže využít znalosti struktury úlohy k získání těsnějších mezí. Kromě toho je navržen heuristický algoritmus, který dokáže relativně rychle generovat dobrá řešení. Tyto algoritmy spolupracují a dohromady tvoří algoritmus schopný řešit velké instance úlohy.

Účinnost navrženého algoritmu byla vyhodnocena na základě 64 testovacích úloh, které využívají skutečná data tří existujících tepláren. Navržený algoritmus byl porovnán s prostým použitím solveru Gurobi. Ve většině případů náš algoritmus nalezl dostatečně kvalitní řešení rychleji než Gurobi. Co je důležitější, náš algoritmus dokázal nalézt dobrá platná řešení i v případech, ve kterých Gurobi v daném časovém limitu nedokázalo najít ani jediné řešení.

Klíčová slova: MILP, optimalizace, plánování provozu, kogenerace, teplárny, Lagrangeova relaxace, metoda větví a mezí, heuristika

Contents

Figures

Tables

Chapter 1

Introduction

The main task of district heating plants with CHP production is to supply districts with heat. Additionally, these plants often supply industrial processes with hot water and/or steam. For this purpose plants comprise various technological equipment, such as boilers, turbines or heat exchangers. The operation of these components must be carefully planned to ensure profitability of the production. Besides, a large portion of the plant's revenues comes from trading co-generated electricity on power markets. However, it is not possible to simply co-generate as much electricity as possible. The electricity must be traded in a form of so called power products, which have specified delivery patterns. Finding an optimal combination of products to trade along with a plan of technological operations is a challenging problem.

Due to the complexity of this planning problem a decision support tool is necessary. The purpose of such a tool is to provide a user with a comprehensive plan of plant's operations including instructions on how to operate the technological equipment (boilers, turbines, heat exchanges etc.) and which electrical power products to buy and which to sell. The objective of this planning is to maximize the profit of a plant. An outline of the functionality of such a decision support tool is in Figure 1.1.

This thesis aims at providing a framework for such a decision support tool, comprising

- efficient modelling technique for CHP plant processes and trading on energy markets,
- a methodology for formulating an optimization problem based on the model,
- a solution algorithm for long-term planning problems which would provide good solutions within reasonable time.

The main features of the framework are the following:

- generality of the modelling framework enabling rapid prototyping of CHP plant models,
- trading electricity in the form of arbitrarily definable power products (such as daily base-load, or weekly peak load),
- usability of the solution technique for short-term as well as long-term horizons.

The more precise specifications of the goals of this thesis will follow after a brief introduction to the problem of CHP production and trade planning.

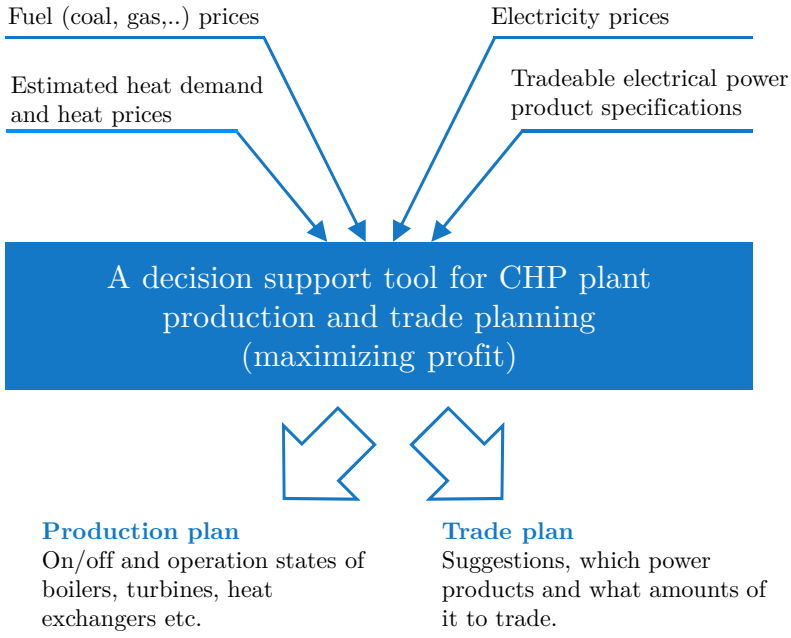


Figure 1.1: Basic scheme of CHP production and trade planning.

1.1 Operations planning of a CHP plant

This work is aimed at medium-sized or large CHP district heating plants employing a thermodynamic cycle with nominal combined heat and electricity power output of tens of megawatts and more - see Figure 1.2 for an example

of a medium-size steam cycle. The thermodynamic cycle consists of a number of components, e.g. turbines (TG1, TG2), boilers (B1, B2), heat exchangers (HE1-5) etc. An optimization problem dealing with operations planning of the components over a given time period is formulated. For the purpose of the optimization problem formulation the optimization horizon is discretized into a number of time samples. The usual sampling period is one hour.

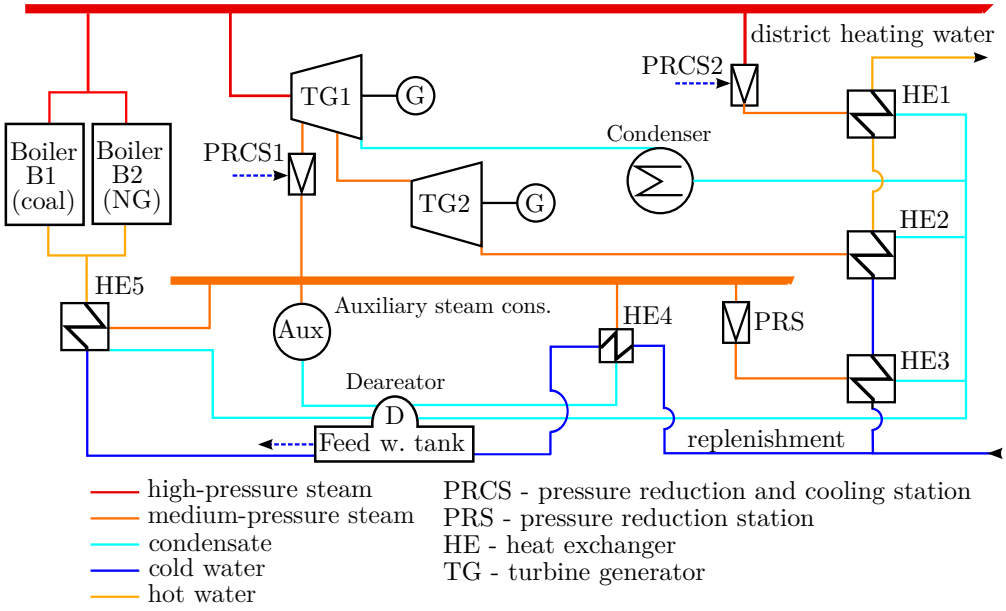


Figure 1.2: Example of a combined heat and power thermodynamic cycle.

The sought plan of operations must include:

- on/off states $u(t)$ of components in each time sample t,
- operation states $x(t)$ of the components, i.e. the actual physical quantities in the thermodynamic cycle, such as mass flows,
- contracted volume $p^{PRODi}(t)$ of each power product $PRODi$ (see section 1.1.1 for details on power products),
- fuel consumption $q_{IN}(t)$,
- start-ups $s_U(t)$ and shut-downs $s_D(t)$ of components,
- positive deviation $d_P(t)$ and negative deviation $d_N(t)$ of the supplied electrical power from the contracted power.

This plan is to be derived respecting the technological constraints:

- operation limits of components - lower and upper bounds,
- minimal number of time periods of continuous operation and a minimal number of consecutive time periods of being shut-down, i.e. minimum up and down times (MUDT),
- limits on rates of transitions from one operation state to another, i.e. ramping limits (RL).

Another set of constraints arise from power product definitions. Introduction to this topic follows in the next section. There can also exist plant-specific constraints, such as those representing rules for obtaining CHP subsidies, or those modelling the provision of ancillary services.

1.1.1 Power products

A power product is basically a delivery of electrical power over a specified period of time - a delivery period. Besides, each power product has defined

- how the power being delivered may change during the delivery period - usually it should be of a constant value or zero,
- in which hours of the delivery period the power should be delivered, i.e. in which hours the delivered power is not zero.

In other words, in selected hours of the delivery period a constant power should be delivered and zero power in the other hours.

The very basic power product is the daily base load. Within this product a constant power should be delivered during a whole day. If a plant sells a daily base product in the volume of 10MW, it is obliged to be delivering exactly 10MW in all the hours of the day, i.e. 240MWh of energy is to be delivered in total. Any deviation of the power delivery from the contracted 10MW is penalized. See Figure 1.3 for an example of a daily-base product. The figures $p^{PRODi}(t)$, $p^{PRODi}_{MIN}(t)$ and $p^{PRODi}_{MAX}(t)$ represent an example of a contracted value in hour t, a lower bound to the contracted value and an upper bound to the contracted value in a respective order.

The required delivery of power may also not be continuous. The typical examples of such products are daily peak-load (its delivery pattern is in Figure 1.4) and daily off-peak (Figure 1.5). By contracting these products in the volume of 10MW the plant makes a commitment that exactly 10MW will be delivered

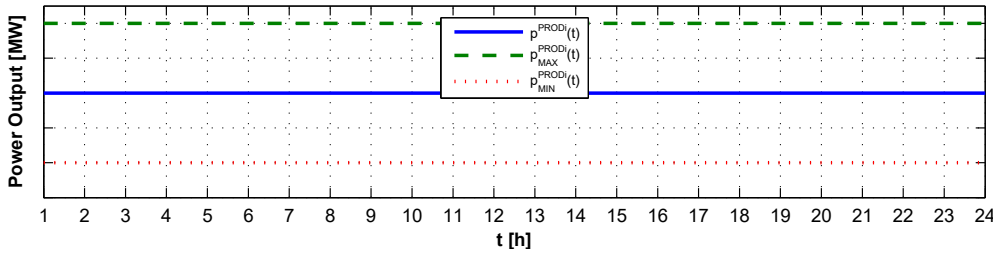


Figure 1.3: Definition of a generic daily base load product (an example of a feasible value of contracted volume in blue)

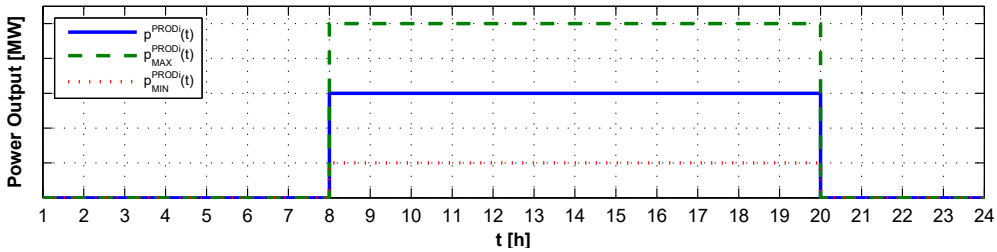


Figure 1.4: Definition of a generic daily peak load product (an example of a feasible value of contracted volume in blue)

during the delivery hours, i.e hours $8, 9, ..., 19$ in the case of daily peak-load and $1, 2..., 7, 20, ..24$ in the case of off-peak. In the other hours no power is to be delivered[1].

Analogously, the products with delivery lengths of a week, month, quarter or even a year may be defined with continuous or not continuous delivery. The proposed framework allows user to define power products with different delivery lengths (days, weeks etc.) and delivery patterns (defining in which hours non-zero power delivery is required). In all cases, the power being delivered within a product is not allowed to change in its delivery hours (in the other hours the power delivered in the context of the product has to be zero).

A district heating plant can provide a constant (and possibly not continuous) electrical power supply when operated in a condensation regime (i.e. some of the produced heat is wasted in a cooling tower). However, in this regime the utilization of cogeneration principles is limited and the overall efficiency of the plant drops. It is more efficient to trade more products at the same time and form a more convenient pattern of electrical power delivery this way. In a

[1] In non-working days, all the hours are off-peak.

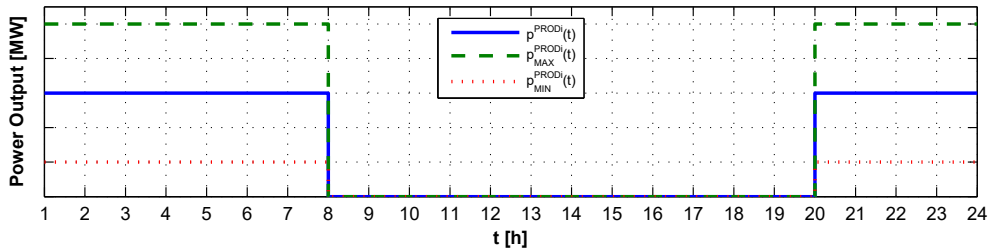


Figure 1.5: Definition of a generic daily peak off-peak product (an example of a feasible value of contracted volume in blue)

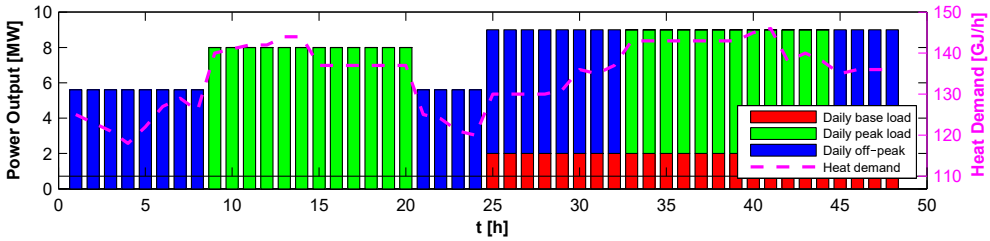


Figure 1.6: Example of a combination of traded power products.

simple case three products may be considered for trading - daily base-load, daily peak-load and daily off-peak. In Figure 1.6 an example on how a combination of these products may be traded is shown. Besides, it is also usually possible to trade a product with the delivery period of one hour on a spot market. It is therefore possible to adjust medium- or long-term plans with short-term planning considering prices on spot markets.

1.1.2 Objective of operations planning

The operators of CHP plants aim their decisions at profit maximization, i.e. the objective of the operations planning optimization problem is the maximization of revenues from contracted power products lowered by expenses on fuel, CO2 allowances, start-up and shut-down costs of components and penalty for deviation from demanded power output (1.1).

$$\begin{aligned} \text{profit} = {} & \text{revenues from heat} + \text{revenues from electricity} \\ & - \text{fuel costs} - \text{allowances cost} \\ & - \text{deviation cost} - \text{other variable costs} \end{aligned} \quad (1.1)$$

1.1.3 Operations planning problem representation

In this thesis CHP production and trade planning is formulated as an optimization problem (MILP), where $\mathbf{u}$ are binary variables. The reasons for choosing a mixed-integer linear programming (MILP) representation include the

- size of the problem of long-term optimization which requires a trade-off between model accuracy and computational performance,
- availability of very powerful general-purpose MILP solvers which evolve rapidly and are able to take advantage of current trends in parallelism,
- required versatility, i.e. plant-specific constraints must be easily representable,
- suitability for automated model generation and for rapid model prototyping,
- conclusions of the survey of methods used in literature which are mostly based on MILP.

$$\begin{aligned} \max_{\mathbf{x},\mathbf{u}} \; & \mathbf{c}^T\mathbf{x} + \mathbf{f}^T\mathbf{u} \\ & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b} \\ & \mathbf{u} \in \{0,1\} \end{aligned} \tag{MILP}$$

The drawback of the MILP approach is the requirement for linearity of constraints and the objective function. This drawback can be mitigated by approximating non-linearities by using the binary variables. I.e. non-linear functions may be approximated by piece-wise linear functions.

1.1.4 Remarks on problem complexity

Mixed integer linear programming problems are NP-hard, which follows from their combinatorial nature. For instance, having 100 binary variables, the number of all possible combinations is 2^{100}. If all the possibilities were to be evaluated, with one evaluation per nanosecond it would take about 40 trillion years. The CHP operations planning problems have about 100 of binary variables per hour of the optimization horizon. This corresponds to more than 74 thousands binary variables in the case of monthly optimization. The number of binary variables is one of the most important parameters affecting the time

required to solve the problem. It would therefore be beneficial to reduce the number of binary variables if possible.

One possibility to reduce the number of binary variables is to use a longer sampling period, e.g. to 8 hours. However, with such a coarse sampling, it becomes harder to handle faster plant dynamics such as minimum-up-and-down times. Also, this way peak values of heat demand and other input data are cut off. Hence, it the resulting plan may not be applicable in reality.

Another way of reducing the number of variables is to select typical days of the planning horizon and compute only these days. Transitions between these days must also be present in a model to encompass at least some of the plant's dynamics. This approach has several drawbacks. The first is that it is again hard to transform a solution obtained for typical days into a solution applicable in reality. This is for the same reasons as in the case of a prolonged sampling period. Also the reduced problem has less degrees of freedom than a full problem and hence a solution quality tends to be low. Finally, considering that heat demand and other input data change rapidly within a planning horizon, rather many typical days must be used. For instance, in the case of weekly planning, at least 3 days must be included - two working days and one non-working day - in order to correctly model the transitions between working and non-working day. Hence, the reduction of variables is not very extensive and the drawbacks of this method overcome its merit.

Computing all the hours or not, the number of combinations is huge, so an algorithm is required, which would be able to identify a subset of the combinations, without their explicit evaluation, which could not contain the optimal solution. These combinations may then be discarded. Usually techniques based on B&B algorithm are used for solving MILP problems. Solvers - software packages such as Gurobi, CPLEX or SCIP - are actually extremely efficient implementations of an algorithm based on B&B. However, not even these solvers are powerful enough for very large MILP problems.

The number of variables and constraints is not the only parameter affecting the time required to solve the problem. It is actually very problem-specific, i.e. dependent on the problem structure. If we had a knowledge of the problem structure, and this structure happened to be somehow convenient, we could utilize this knowledge in creating even more efficient algorithms than a general-purpose B&B is.

The complexity of CHP operations planning problems (the number of variables and constraints exceeds 1 million in some cases) is often beyond capabilities of even the most powerful solvers. In some cases, solvers are unable to find a good solution (or any) in a reasonable time.

A solution time can be considered as reasonable, if it is not prohibitive for an efficient usage of such a tool. This clearly depends on a user and the task being solved. However, users typically wish to compute several variants of a planning task. Hence, the time of two hours can be viewed as an upper bound to a non-prohibitive solution time.

In this thesis, we propose an algorithm based on B&B, which is able to exploit our knowledge of the problem structure, i.e. its block-diagonal sub-structure. Performance requirements for the the proposed algorithm are discussed in the next section.

1.2 Goals of this thesis

In this thesis a comprehensive framework for solving long-term CHP operations planning problems is developed. The framework has two main parts - the first is a modelling framework which allows for modelling arbitrary CHP plants and formulation of a MILP optimization problem. The second is a solution algorithm which exploits the knowledge of the problem structure so that the problem is solved more efficiently.

The goal of this thesis to is propose a framework, which would meet the following requirements:

- The framework must be usable for various CHP plants, i.e. it must handle all the peculiarities of different plants.
- The algorithm must provide an estimation on proximity of the current best known solution to the optimal one, i.e. its optimality gap.
- The solution algorithm must be capable of finding good solutions (within 1% gap) of long-term planning problems in reasonable time (under two hours).

The algorithm must also be tested on various real-world scenarios.

1.3 Means used to achieve the goals

This section contains a brief summary of methodology used to achieve the stated goals.

Some problem instances are simply too big for current solvers. Hence a decomposition technique solving many smaller sub-problems is required. The

solutions of these sub-problems are then used for creating a solution valid with respect to the original problem. The decomposition technique we use within our algorithm is Lagrangian relaxation (LR). The block-diagonal sub-structure is exploited so that after relaxing a small portion of constraints, the remaining problem is separable into independent sub-problems.

LR is used within a B&B algorithm. It provides tight bounds at nodes. The bounds obtained by solving LR are usually much tighter than the bounds from linear relaxations. This way the number of combinations that must be evaluated is reduced. Parallelization is used extensively in solving the relaxations and branching.

The solutions obtained from LR are rarely valid with respect to the original problem. A heuristics based on problem decomposition providing feasible solutions was therefore developed. In most cases, it provides solutions which are no more than 1% worse than the optimum. However, the heuristics does not provide any certificate of (near-)optimality. This certificate is therefore provided by the B&B algorithm.

The algorithm was implemented in Matlab and solver Gurobi was used for solving the relaxations

1.4 Thesis organization

The rest of this thesis is organized as follows. State-of-the-art analysis describing approaches commonly used for CHP planning problems (and similar) is presented first in Chapter 2. In Chapter 3 the general modelling framework is explained on an example of a medium-size district heating plant (in Figure 1.2). The solution algorithm is introduced next, in Chapter 4. Then in Chapter 5 the performance of the algorithm is evaluated on a series of test cases. This thesis then concludes with final summary.

Chapter 2

State-of-the-Art

We started the literature review by searching for an algorithm readily applicable to CHP operations planning problem. We anticipated to find a number of different approaches that we could compare and choose one according to its versatility, robustness, speed and memory requirements. However, our main findings were:

- Mostly only short-term planning problems were dealt with, using a plain general MILP solver.
- In the cases, in which medium-term or long-term planning was addressed, a relaxation of the original problem was solved, mostly by dividing it into smaller independent sub-problems. A result obtained by this method is generally not valid with respect to the original problem.
- Also a usage of LR was reported. However, there are issues with LR that will be discussed further in the text.
- In literature the algorithms tend to be tailored to a specific problem instance (e.g. a specific CHP plant). We require better versability of the algorithm that would allow us to use it for different plants.

It should be noted that we did not restrict the literature review solely to the works dealing with CHP production planning. There are other relevant problem classes, such as unit commitment problem. Methods developed for these classes could represent at least a good source of inspiration.

In the following sections we will go through the works that we believe are the most important contributions.

The surveyed works can be divided into two groups according to their topic:

- Modelling techniques and frameworks for CHP plants,
- Optimization techniques that can be used for CHP operations planning problem.

A model is a prerequisite for a formulation of an optimization problem. However, different optimization techniques require different modelling approaches and pose different restrictions upon these approaches. E.g. in order to be able to use MILP solvers a MILP representation of the problem is required. So the choice of a modelling paradigm is tightly connected to the choice of an optimization technique. The task of this review can then be distilled into:

- Finding an optimization technique capable of solving long-term CHP planning problems and
- allowing for such a modelling technique that would facilitate sufficiently precise modelling of CHP plants.

The first part of this review answers the question what is a sufficiently precise model. The second part then presents results of our search for an optimization technique able to solve the long-term CHP planning problems.

2.1 Modelling techniques and frameworks

In order to formulate the operations planning optimization problem a model of processes that take place in a CHP plant is required. In literature two basic approaches for CHP plant modelling were used

- black-box approaches using data interpolation [Fer+04] or defining operating regions of whole CHP plants [RL07b] or of individual components [TBW05; MSG13a; Bis+14a],
- first-principles approaches employing balance equations [YDOJK08; CTG05a; Agu+07; SA10; Cho+10; VG+11].

The main advantage of the first-principles approach is its generalization capability. This means that the model can capture even the operational states of a CHP plant that are not available in historical data. An optimization tool using this model can therefore offer solutions that may have not been considered

before. On the other hand a black-box model can credibly propose only solutions that are included in the historical data it was identified from.

This drawback of black box models can be handled (e.g. in the case of incomplete historical data) by employing a first principle model. This model is used to fill the gaps in data needed for regression analysis by performing simulations [Tou+11].

In literature the first-principles modelling of CHP plants was usually addressed in a different scope than district heating - most frequently in the context of process integration problems of utility systems[1] [Agu+07; VG+11] emphasizing comprehensiveness and complexity of thermodynamic cycle models more than is necessary in our case. These complicated non-linear models are used for formulation of rather small optimization problems - single-period or multi-period optimization problems without time-coupling constraints. That is why computation times remained acceptable in these cases.

First-principle models of small industrial cogeneration plants for operations planning were developed in [YDOJK08; CTG05a; TBW05], where non-linear (higher-order polynomial) descriptions of condensing turbines were used. In these cases, no complicating constraints on components operation, such as ramping limits or minimum up add down times, were modelled. These papers also address electricity trading. The produced electricity is sold regardless on its volumes (within a bilateral agreement) and no special delivery pattern is required. Yet, many CHP plants are either forced to participate on the trading with power products (have no favourable bilateral agreements) or they recognize a good market opportunity in it. As far as we know a model of electricity trading of CHP plants in the form of standardized power products was first described in our paper [DH12].

2.2 Optimization techniques

The long-term CHP planning represents a computationally very intensive task due to the combinatorial nature of mathematical programming problems containing integer variables. In current practice long-term CHP planning problems are handled by heuristic approaches, such as horizon cutting [DH12], [TBW05]. The optimization horizon is divided into convenient time periods which are then solved in a sequence with the last sample of each period representing an initial condition for the next period. It should be noted that the divison into

[1] Process integration problems of utility systems is a problem of designing the systems, considering various criteria (efficiency, environmental issues etc.) .

subproblems is done by relaxing some of the constraints. Hence, the solution to the divided problem is not generally valid with respect to the original one. Another issue with this approach is the lack of foresight when solving the time periods sequentially. This may result in poor solution quality. Also, handling long-term power products, such as yearly base-load, is problematic, not to mention that this method can not provide any information on proximity of the provided solution to the optimal one.

Available heuristic methods include relax-and-fix heuristics by [KC12] formalizing the horizon cutting of [DH12] and [TBW05] and extending it with a simple backtracking for the case, in which the procedure leads to an infeasible subproblem. Also, LP-fix heuristics deciding the values of integer variables based on the solution of linear programming relaxation is proposed by the same authors [KC12]. Metaheuristics were also applied to CHP operations planning, e.g. enhanced immune algorithm [CTG05b], harmony search algorithm [VFB07] or tabu search [KC12].

Heuristic and metaheuristic methods do not provide any information on proximity to the optimal solution, i.e. no certificate of (near-) optimality is available. The information on quality of provided solution is, however, one of the goals of this thesis. Nevertheless, heuristic methods may be very useful for generating feasible solutions for other algorithms, such as B&B.

Among deterministic methods used for handling large-scale CHP planning problems LR represents definitely the most frequently used approach in literature, e.g. [TBW05; RLL08; EGU16]. Two main issues with this method have been reported. Most importantly it is the convergence issue reported for instance by [RL07a] and [Wan+95]. To improve the convergence property an augmented version of LR is often used instead. However, augmentation prohibits decomposition of the relaxed problem into sub-problems which cancels out the main advantage of LR for our cause. Moreover, LR used for non-convex problems inherently produces an infeasible solution. Solution of a MILP obtained by LR thus require post-processing, which is not a straightforward procedure and requires a dedicated heuristic algorithm.

These issues with LR are, however, of lesser importance when LR is used within B&B for providing bounds instead of linear programming relaxation (LPR). Yet, another pitfalls emerge. Solving LR is usually much more expensive than solving LPR. Moreover, the simplex method for solving LPR is very efficient at reoptimizing after branching or generation of valid inequalities. Such techniques for LR are not available [Fra05].

An interesting approach combining metaheuristics with a deterministic method is presented by [RVP14]. The authors propose using heuristic or metaheuristic

approaches to speed up convergence of a unit commitment problem formulated as MILP. A unit commitment problem is an optimization problem similar to CHP production planning problem. It deals with deciding which power generator units must be committed/decommited in order to satisfy demand over a planning horizon. The production of generator units is constrained in a similar way as the operation of plant equipment within CHP production planning problem (minimum up and down time constraints, ramping limits etc.). The proposed cooperation of a heuristics with a deterministic MILP solver represents a viable and interesting way of providing estimation on proximity of a heuristic solution to the optimum. A similar methodology combining heuristic and deterministic algorithms is employed also in this thesis.

2.3 Summary

Considering the reviewed literature, we prefer a first-principle model as there may not be enough data available to perform a comprehensive regression analysis. Of course the data available should be used for tuning up the first principles model. Also, in literature the models of CHP plants usually include non-linear terms. Solution techniques for solving generally non-linear problems exist, but are far less powerful than the techniques for (mixed-integer) linear problems. Nevertheless, the non-linear terms can be sufficiently approximated by piece-wise linear (PWL) functions resulting in a mixed-binary linear model.

The literature survey suggests that there is no solution technique readily available, that would handle unrelaxed long-term CHP planning. Long-term problem instances are simply too big even for state-of-the-art general purpose MILP solvers. Some kind of problem decomposition must take place. The most common decomposition paradigm - LR - suffers from convergence issues and invalidity of the resulting solution. But it can be used as a bound generator for a customized B&B algorithm. Besides an heuristic may be employed for generating feasible solutions. Hence, the algorithm proposed in this thesis will be based on a customized B&B algorithm cooperating with a heuristic for obtaining feasible solutions.

Chapter 3

Framework for modelling CHP plant operations

In this chapter a framework for the modelling of CHP plants operations is introduced. This framework was developed with the emphasis on generality, i.e. the framework must be usable for various CHP plants. The framework provides means of mathematical description of typical technologies used in CHP plants, such as turbines, boilers, heat exchangers etc. Besides it allows to model the trade with electrical power products. If a CHP plant employs an uncommon technology or perhaps modelling of some plant-specific constraints is required, the framework has to be extended. However, the framework is designed so that only minor or no extensions should be required, when modelling a new CHP plant.

The purpose of the model is its usage within a formulation of a MILP optimization problem. Therefore all the terms in equalities and inequalities are linear with continuous or binary variables. These equalities and inequalities then form the constraints of the optimization problem. In the end of this chapter an objective function is defined, which completes the definition of the optimization problem.

The bottom-up and first-principle is used for the modelling of CHP plants. As mass flow and energy transfer rates are the values of interest, mass rate and energy rate balance equations represent a convenient way of modelling the thermodynamic cycles of CHP plants. This way the system can also be broken up into a set of connected components which conforms the modelling approach of an object-oriented modelling language.

The model is defined in discrete time with sampling period of 1 hour. This sampling period has been chosen according to the needs of customers as the electricity is typically traded with the granularity of 1 hour. Hence in order to

work with power product definitions properly, the sampling period cannot be longer than 1 hour. Also, for the purpose of planning the granularity of 1 hour is sufficient as the faster dynamics is not of interest in the view of operations plans.

In the following paragraphs the modelling framework is presented on the example of a medium-size district heating plant in Figure 1.2. The topology is taken from a real plant in the Czech Republic with rated power outputs of 50MW in heat and 40MW in electricity. The plant also provides steam of defined parameters for an industrial consumer.

3.1 Generalized component of the thermodynamic cycle

In accordance to the need of generality of the modelling framework a generalized component is defined. Its definition conforms the utilization of the component within MILP optimization problem formulation. The generalized component can then be used to model any component of a thermodynamic cycle. Such model will then be implicitly suitable for MILP problem formulation.

A component of the thermodynamic cycle is generally a system with several inlets and several exits through which, in time t, fluid (steam or water) flows with mass flow rates $m_i(t)$ and $m_e(t)$ respectively. Generally the parameters of fluid are given by enthalpies $h_i(t)$ and $h_e(t)$. The energy transfer rates accompanying mass flows are then $q_i(t) = m_i(t)h_i(t)$ and $q_e(t) = m_e(t)h_e(t)$. According to the conservation principles, mass rate balance (3.1a) and energy rate balance (3.1b) of a general component (Figure 3.1) must hold.

$$\sum_i m_i(t) = \sum_e m_e(t), \ \forall t \tag{3.1a}$$

$$q_{IN}(t) + \sum_i q_i(t) = q_{OUT}(t) + \sum_e q_e(t) \tag{3.1b}$$

The terms $q_{IN}(t)$ and $q_{OUT}(t)$ represent rates at which energy is being transferred in and out of the component not accompanying mass flow.

In most cases the enthalpies $h(t)$ in energy rate balance equations can be considered as constants H as they are either given by the construction of components (boiler outputs, extractions of turbines, condensers, pressure reduction stations etc.) or their dependency on mass flow rates is negligible. The energy rate balance equations can then be defined as (3.2).

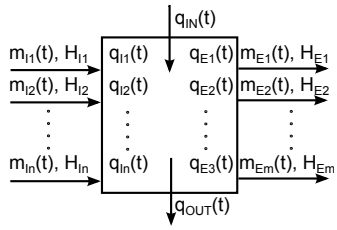


Figure 3.1: Generalized component.

$$q_{IN}(t) + \sum_i m_i(t)H_i = q_{OUT}(t) + \sum_e m_e(t)H_e \tag{3.2}$$

If the dependency of steam (condensate) parameters on mass flow rates through a component cannot be neglected (e.g. in the case of condensing turbine), it is modelled using PWL functions. With one of the enthalpies being considerably dependent on mass flow through the component it is possible to model the non-linear dependence $q_e(t) = m_e(t)h_e(t)$ as (3.3) - see Figure 3.2 for an example with four characteristic points (i.e. three linear segments). Naturally, it is possible to define any dependence in the model this way.

$$q_e(t) = \sum_k Q_{e,k}\lambda_k, \;\; m_e(t) = \sum_k M_{e,k}\lambda_k, \tag{3.3a}$$

$$0 \leq \lambda_k \leq 1, \;\; \forall k \tag{3.3b}$$

$$\{\lambda_0, \lambda_1, \lambda_2, ...\} \text{ is SOS2} \tag{3.3c}$$

Note that labelling some set of variables as special ordered set of type 2 (SOS2) is actually only another formulation of an ordered set of variables in which only two adjacent can be non-zero, see for example [Tom88]. This set can alternatively be defined manually using binary variables. However, some solvers can treat SOS2 sets more efficiently than the definitions using binary variables. The number of SOS2 and binary variables determines the combinatorial space which has to be searched through by the used solver. It is therefore always the trade-off between model accuracy and computational complexity.

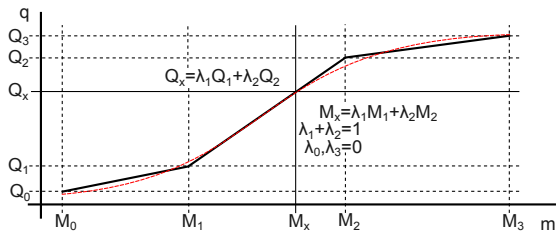


Figure 3.2: Example of a PWL function.

3.2 Modelling components of thermodynamic cycle

In the following paragraphs the generalized component (3.1) will be utilized in the modelling of the thermodynamic cycle in Figure 1.2 by first modelling individual components and then interconnecting them into the complete cycle. By using the generalized component, suitability of all the models within MILP optimization problem will be guaranteed. The values of parameters of models, such as steam enthalpies, are obtained from measurements (of temperatures, pressures etc.) or from specifications of components. If necessary, regression analysis of historical data is used to obtain the values.

3.2.1 Extraction-condensing turbine

The extraction-condensing turbine TG1 has inlet flow I, exit flow E and two extractions $X1$ and $X2$. The parameters of inlet steam are assumed to be constant. The pressure at the exit of the turbine is considered to be dependent on mass flow. The rate of useful energy $q_{OUT}^{TG1}(t)$ that can be converted to electrical power is therefore non-linearly dependent on the mass flow rate through the last stage of the turbine (the thermal efficiency of the turbine is not constant) which is modelled by a PWL approximation. The model of turbine TG1 can then be defined as (3.4) using (3.1), (3.2) and (3.3).

$$m_I^{TG1}(t) = m_{X1}^{TG1}(t) + m_{X2}^{TG1}(t) + m_E^{TG1}(t),\ \forall t \tag{3.4a}$$
$$q_I^{TG1}(t) = q_{OUT}^{TG1}(t) + q_{X1}^{TG1}(t) + q_{X2}^{TG1}(t) + q_E^{TG1}(t),\ \forall t \tag{3.4b}$$
$$q_I^{TG1}(t) = m_I^{TG1}(t) H_I^{TG1},\ \forall t \tag{3.4c}$$
$$q_{X1}^{TG1}(t) = m_{X1}^{TG1}(t) H_{X1}^{TG1},\ \forall t \tag{3.4d}$$
$$q_{X2}^{TG1}(t) = m_{X2}^{TG1}(t) H_{X2}^{TG1},\ \forall t \tag{3.4e}$$
$$q_E^{TG1}(t) = \lambda_0^{TG1}(t) Q_{E,0}^{TG1} + \lambda_1^{TG1}(t) Q_{E,1}^{TG1} + \lambda_2^{TG1}(t) Q_{E,2}^{TG1},\ \forall t \tag{3.4f}$$
$$m_E^{TG1}(t) = \lambda_0^{TG1}(t) M_{E,0}^{TG1} + \lambda_1^{TG1}(t) M_{E,1}^{TG1} + \lambda_2^{TG1}(t) M_{E,2}^{TG1},\ \forall t \tag{3.4g}$$

Note that the PWL function describing energy transfer rate accompanying the outlet mass flow rate E has three characteristic points $\left\{M_{E,0}^{TG1}, M_{E,1}^{TG1}, M_{E,2}^{TG1}\right\}$ and $\left\{Q_{E,0}^{TG1}, Q_{E,1}^{TG1}, Q_{E,2}^{TG1}\right\}$ (i.e. two linear segments).

Next, the heat rate q_{OUT}^{TG1} is being converted to mechanical and then electrical power p_{OUT}^{TG1} by the turbine with constant efficiency coefficient η^{TG1} (generator losses). Note that the coefficient η^{TG1} may also be modelled as PWL function of mass flow rates if necessary.

$$p_{OUT}^{TG1}(t) = \eta^{TG1} q_{OUT}^{TG1}(t),\ \forall t \tag{3.5}$$

3.2.2 Backpressure turbine

In the case of backpressure turbine TG2 the approximation using constant enthalpies is sufficient (with the maximal error compared to measurements of 1.5% in the case of the plant considered in Figure 1.2).

$$m_I^{TG2}(t) = m_E^{TG2}(t),\ \forall t \tag{3.6a}$$
$$m_I^{TG2}(t) H_I^{TG2} = q_{OUT}^{TG2}(t) + m_E^{TG2}(t) H_E^{TG2},\ \forall t \tag{3.6b}$$
$$p_{OUT}^{TG2}(t) = \eta^{TG2} q_{OUT}^{TG2}(t),\ \forall t \tag{3.6c}$$

3.2.3 Boiler

The boiler B1 with feed-water flow m_I^{B1} and exit flow of superheated steam m_E^{B1} is modelled with (3.7). It is assumed that the parameters of feed-water

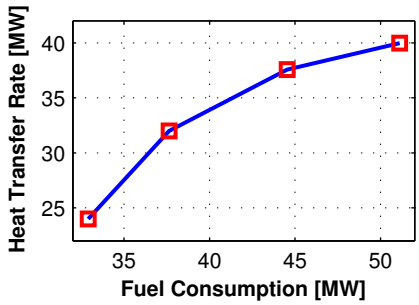


Figure 3.3: Fuel consumption characteristics of the boiler B1

and produced steam are constant. Based on historical data, it has been found necessary to model the dependence of heat rate being transferred to water q_{IN}^{B1} on the fuel (coal) consumption rate q_F^{B1} (fuel consumption characteristics) using PWL function with four characteristic points $\left\{Q_{F,0}^{B1}, Q_{F,1}^{B1}, Q_{F,2}^{B1}, Q_{F,3}^{B1}\right\}$ and $\left\{Q_{IN,0}^{B1}, Q_{IN,1}^{B1}, Q_{IN,2}^{B1}, Q_{IN,3}^{B1}\right\}$, see Figure 3.3.

$$m_I^{B1}(t) = m_E^{B1}(t) \tag{3.7a}$$

$$m_I^{B1}(t)H_I^{B1} + q_{IN}^{B1} = m_E^{B1}(t)H_E^{B1},\ \forall t \tag{3.7b}$$

$$q_F^{B1}(t) = \sum_i \lambda_i^{B1}(t)Q_{F,i}^{B1} \tag{3.7c}$$

$$q_{IN}^{B1}(t) = \sum_i \lambda_i^{B1}(t)Q_{IN,i}^{B1} \tag{3.7d}$$

The other boilers are modelled analogously.

3.2.4 Heat exchanger

In the heat exchangers in Figure 1.2 energy is transferred into district heating water or boiler feed-water by condensation of the steam passing through. The condensate exits the heat exchanger with approximately the temperature of saturated steam and the enthalpy of condensate can therefore be considered constant. According to [Kap+09], a steam-heated heat exchanger can be modelled with (3.8).

$$q_{OUT}^{HE}(t) = m_S^{HE}(t)\left(H_I^{HE} - H_E^{HE}\right),\ \forall t \tag{3.8a}$$

$$q_{OUT}^{HE}(t) = m_W^{HE}(t)C_W\left(T_H - T_C\right),\ \forall t \tag{3.8b}$$

In practice, an independent control system is used to control the values of return water temperature T_C, hot water temperature T_H and mass flow rate of heated water m_W^{HE}. The control system secures the supply of heat into the district heating network, i.e. it ensures that the equation (3.8b) holds. The operations planning therefore needs only to satisfy the heat demand in terms of energy supplied in steam from the turbine (3.8a).

It should be noted that the proposed model of heat exchanger describes a special case of heat exchanger - the steam-water heat exchanger. Water-to-water heat exchangers may also be used in district heating plants for some technological purposes. However, their operation usually does not directly affect steam production and operation of turbines. Their operation is therefore transparent for the optimization problem. Besides, the parameters of passing fluids are typically controlled by an independent control system. If necessary (e.g. special constraints apply), they may be modelled using linear or PWL balance equations (3.1).

3.2.5 Pressure reduction and cooling station

In a pressure reduction and cooling station (PRCS) the parameters of its inlet steam are changed by injecting cooling water with flow m_{CW}^{PRCS} into the steam flow m_I^{PRCS} and by steam expansion. A PRCS can then be modelled by (3.9).

$$m_I^{PRCS}(t) + m_{CW}^{PRCS}(t) = m_E^{PRCS}(t), \ \forall t \tag{3.9a}$$

$$m_I^{PRCS}(t) H_I^{PRCS} + m_{CW}^{PRCS}(t) H_{CW}^{PRCS} = m_E^{PRCS}(t) H_E^{PRCS}, \ \forall t \tag{3.9b}$$

3.2.6 Pressure reduction station

In a pressure reduction station (PRS) the steam is not cooled by water injection and only expansion occurs, therefore the whole process is isoenthalpic and it is sufficient to define (3.10).

$$m_I^{PRS}(t) = m_E^{PRS}(t), \ \forall t \tag{3.10}$$

Although the model (3.10) is trivial it is usually necessary to formulate, as there may be upper and lower bounds on flows or other constraints defined.

3.2.7 Condenser

The model of the condenser C is analogous to the model of heat exchanger (3.8).

3.2.8 Deareator

A deareator may be viewed as a heat exchanger with mixed streams. For the purpose of the optimization problem only mass and energy rate balances are of interest and therefore a deareator is modelled by (3.11). The indices CW and B denote cooling water for PRCSs and feed-water for boilers respectively.

$$\sum_i m_i^D(t) = m_B^D(t) + m_{CW}^D(t),\ \forall t \tag{3.11a}$$

$$\sum_i m_i^D(t) H_i^D = m_B^D(t) H_B^D + m_{CW}^D(t) H_{CW}^D,\ \forall t \tag{3.11b}$$

3.3 Interconnection of components into closed steam cycle

The interconnection of the individual components in Fig. 1.2 into the closed thermodynamic cycle can be implemented using a nodal component (3.12) which is a special case of the generalized component (3.1) with no energy transferred in or out of the component. If the nodal component is employed to model a pipe junction or a steam header all the steam enthalpies are of the same values.

$$\sum_i m_i^J(t) = \sum_e m_e^J(t),\ \forall t \tag{3.12}$$

3.4 Modelling constraints on operation of components

Steam cycle components have lower and upper bounds (LB and UB respectively) on mass flow and energy rates. Also the case when a component is shut-down needs to be modelled. For this purpose a binary variable $u^c(t)$ is defined representing on/off state of a component c in time sample t. The mass flow rate of a component can then be constrained using (3.13).

$$M_{p,MIN}^c u^c(t) \leq m_p^c(t) \leq M_{p,MAX}^c u^c(t),\ \forall t \tag{3.13}$$

To avoid its unnecessary start-ups and shutdowns, the operation of a component may also be restricted in a minimal number of time periods of continuous

operation and a minimal number of consecutive time periods of being shutdown, i.e. minimum up and down times (MUDT) T_U and T_D formulated by equations (3.14), this formulation is adopted from [HOO09].

$$\sum_{i=t-T_U^c+1}^{t} s_U^c(i) \leq u^c(t),\ \forall t > T_U^c \tag{3.14a}$$

$$\sum_{i=t-T_D^c+1}^{t} s_D^c(i) \leq 1 - u^c(t),\ \forall t > T_D^c \tag{3.14b}$$

The variables $s_U^c(t)$ and $s_D^c(t)$ used in (3.14) represent start-ups and shutdowns of a component. The variable $s_U^c(t)$ takes on value of 1 if the component c is started-up in time sample t and 0 otherwise. The variable $s_D^c(t)$ takes on value of 1 if the component is shutdown in sample t and 0 otherwise. These values are defined by equations and inequalities (3.15). Since $u^c(t)$ is a binary variable, the inequalities (3.15b) guarantee that either $s_U^c(t)$ is non-zero or $s_D^c(t)$ is, but not both. Then if in the current sample the component is offline ($u^c(t) = 0$) it follows from (3.15a) and (3.15c) that the value of $s_D^c(t)$ is 1 only if the component was online in the previous sample ($s_D^c(t) = u^c(t-1)$). Similarly if $u^c(t) = 1$ then $s_U^c(t) = 1 - u^c(t-1)$ and $s_U^c(t) = 1$ if and only if $u^c(t) = 1$ and $u^c(t-1) = 0$. If both $u^c(t)$ and $u^c(t-1)$ are zero then also both $s_U^c(t)$ and $s_D^c(t)$ have to be zero. The advantage of this formulation is that variables $s_U^c(t)$ and $s_D^c(t)$ implicitly take binary values even if defined as continuous variables. These variables are used for formulation of minimum up and down times, ramping limits and for computation of start-up and shutdown costs (by multiplying their values with costs per start-up/shutdown) in the objective function.

$$s_U^c(t) - s_D^c(t) = u^c(t) - u^c(t-1),\ \forall t > 0 \tag{3.15a}$$

$$s_U^c(t) \leq u^c(t),\ s_D^c(t) \leq 1 - u^c(t),\ \forall t \tag{3.15b}$$

$$0 \leq s_U^c(t) \leq 1,\ 0 \leq s_D^c(t) \leq 1,\ \forall t > 0 \tag{3.15c}$$

Another technical limitations of many components, especially of boilers and turbines, which have to be taken into consideration, are the ramping limits (RL) reflecting the dynamics which cannot be neglected. E.g. if maximal change in heat transfer rate of a boiler q_{IN}^B between two adjacent time samples is R_U^B in up-direction and R_D^B in down-direction, then these restrictions may be modelled using equations (3.16), where R_{SU}^B and R_{SD}^B represent maximal allowable increase of q_{IN}^B from off-state and decrease to off-state respectively.

This description was adopted from [CA06].

$$q^B_{IN}(t) - q^B_{IN}(t-1) \leq R^B_U u^B(t-1) + R^B_{SU} s^B_U(t) + \\ + Q^B_{IN,MAX}(1 - u^B(t)),\ \forall t > 0 \tag{3.16a}$$

$$q^B_{IN}(t-1) - q^B_{IN}(t) \leq R^B_D u^B(t) + R^B_{SD} s^B_D(t) + \\ + Q^B_{IN,MAX}(1 - u^B(t-1)),\ \forall t > 0 \tag{3.16b}$$

3.5 Model of trading on power markets

One of the important features of the presented optimization problem is its ability to consider the trading on power markets with standardized power products. The topic of power products was introduced in Section 1.1.1. The proposed framework supports power products with different delivery lengths (days, weeks etc.) and delivery patterns (defining in which hours non-zero power delivery is required). The power products can be modelled with (3.17).

$$p^{PRODi}(t) = \begin{cases} p^{PRODi}_{VOL} & t \in \text{delivery hours} \\ 0 & \text{otherwise} \end{cases} \tag{3.17a}$$

$$P^{PRODi}_{MIN}(t) u^{PRODi} \leq p^{PRODi}(t) \leq P^{PRODi}_{MAX}(t) u^{PRODi},\ \forall i \tag{3.17b}$$

In (3.17) the variable p^{PRODi}_{VOL} is the contracted volume of the product $PRODi$ binding together values of $p^{PRODi}(t)$ in the time samples corresponding to the delivery hours (hours in which the required power to be delivered within the product $PRODi$ is non-zero). The binary variable u^{PRODi} then defines whether the product is contracted or not.

3.6 Modelling constraints on production

Finally, constraints resulting from heat demand and electricity trading have to be formulated. The electricity production of turbines $p^{TGi}(t)$ in the thermodynamic cycle should match the electricity demand defined by contracted power products $p^{PRODj}(t)$ in each time sample t. This is expressed by Eq. (3.18), where the terms d_N and d_P stand for shortage and surplus of power generation respectively (both are non-negative variables). The term p^{AC} defines auxiliary power consumption of the plant which is usually a function of on/off states of components, such as boilers.

$$\sum_i p^{TGi}(t) + d_N(t) = \sum_j p^{PRODj}(t) + p^{AC}(t) + d_P(t), \ \forall t \tag{3.18}$$

In contrast to the electricity supply, the heat demand $Q_{REQ}(t)$ has to be satisfied without deviation, the constraint is therefore defined as (3.19), where $\sum_i q_{OUT}^{HEi}(t)$ is heat transferred from the thermodynamic cycle into district network by heat exchangers.

$$\sum_i q_{OUT}^{HEi}(t) = Q_{REQ}(t), \ \forall t \tag{3.19}$$

3.7 Using the model to formulate optimization problem

An optimization problem consists of an objective function which is minimized or maximized subject to a set of constraints. The constraints of the CHP operations planning optimization problem have been formulated in the sections above. What remains to be defined in order to complete the formulation is an objective function.

The operators of CHP plants aim their decisions at profit maximization, i.e. the maximization of revenues from contracted power products lowered by expenses on fuel, CO_2 allowances, start-up and shutdown costs of components and penalty for deviation of actual power output from the contracted power output. With the introduced models, variables and constraints, the objective of the optimization problem can be defined as (3.20), where Q_{REQ} is the thermal energy supplied to the district with revenues R_Q per unit of energy, p^{PRODi} is supplied electrical energy in the form of product i with revenues R_P^{PRODi} per unit, q_F^b is fuel consumption of boiler b with costs C_F per unit of fuel and costs C_{ALW} per CO_2 allowance, C_{SU}^c and C_{SD}^c are costs per start-up and shutdown of a component c. Finally, C_{DEVN} and C_{DEVP} are prices for negative and positive deviation.

$$\max_{p^{PRODi},\, q^b_{FUEL},\, s^c_U,\, s^c_D,\, d_N,\, d_P} J =$$

$$\sum_t \Bigg(\overbrace{R_Q(t)Q_{REQ}(t)}^{\text{revenues from heat}} + \overbrace{\sum_i R_P^{PRODi}(t)p^{PRODi}(t)}^{\text{revenues from el.products}} - \overbrace{\sum_b (C_F(t) + C_{ALW}(t))q_F^b(t)}^{\text{fuel costs}} - \overbrace{\sum_c C^c_{SU}(t)s^c_U(t) + C^c_{SD}(t)s^c_D(t)}^{\text{start-up and shutdown costs}} - \underbrace{(C_{DEVN}(t)d_N(t) + C_{DEVP}(t)d_P(t))}_{\text{deviation penalty}} \Bigg) \quad (3.20)$$

Although, the revenues from heat are given by the heat demand, which has to be satisfied, the constant term is included in the objective so that the objective value then directly represents the profit of the plant.

Chapter 4

Algorithm for solving the optimization problem

The algorithm proposed in this thesis is based on the knowledge that most of the constraints are specific for a single hour, i.e. all the variables within these constraints describe the operation of a CHP plant in the same hour. Mass balance equations (3.1) represent an example of these single-hour constrains.

If all the constraints had this property, we would simply decompose the optimization problem into hourly sub-problems and solve these simple problems individually. Unfortunately, this is not the case. There is another group of constraints which makes the solving process more complicated, hence this group is called the complicating (or coupling) constraints. These constraints comprise variables describing operation of a plant in different hours. The typical example of complicating constraints are the minimum up and down time constraints (3.14).

The presence of the complicating constraints prevents us to decompose the problem and solve it easily as a number of relatively small independent sub-problems. We can omit the complicating constraints. However, the solution obtained this way would be invalid with respect to the original problem and would have a better value of the objective function than has the real optimum, i.e. it would represent a bound (an upper bound for maximization problems) on the objective value.

In this chapter we will explain how to utilize the structure of the optimization problem for getting good feasible solutions and proving that these solutions are really close to the optimum. First, we will introduce the matrix representation of the problem, which is more convenient for the description of algorithms and will then be utilized by the following sections. Then in Chapter 4.2 the general B&B algorithm will be described. After that the proposed algorithm is outlined and its main components are described. The next section will deal with an algorithm

with the purpose of providing good feasible solutions. Section 4.5 will present an algorithm providing certificates of (near-)optimality for the found solutions. Finally, in Section 4.6 implementation details are provided.

4.1 Matrix representation of the optimization problem

The optimization problem consisting of the objective function (3.20) and the set of constraints (equalities and inequalities) defined in Chapter 3 can be written in a compact way as (MILP).

$$\begin{aligned} \max_{\mathbf{x},\mathbf{u}} \; & \mathbf{c}^T\mathbf{x} + \mathbf{f}^T\mathbf{u} \\ & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b} \\ & \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \leq \mathbf{d} \\ & \mathbf{u} \in \{0,1\} \end{aligned} \tag{MILP}$$

In this formulation $\mathbf{x}$ represents a vector of continuous variables, e.g. mass flows, energy flows etc., and $\mathbf{u}$ is a vector of binary variables (on/off states). The constraints $\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b}$ are the complicating constraints (coupling sub-problems together) and $\mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \leq \mathbf{d}$ are sub-problem constraints, i.e. constraints which can be rewritten in the following form:

$$\begin{aligned} \mathbf{C}_1\mathbf{x}_1 + \mathbf{D}_1\mathbf{u}_1 & & & \leq \mathbf{d}_1 \\ \mathbf{C}_2\mathbf{x}_2 + \mathbf{D}_2\mathbf{u}_2 & & & \leq \mathbf{d}_2 \\ & \ddots & & \\ & & \mathbf{C}_N\mathbf{x}_N + \mathbf{D}_N\mathbf{u}_N & \leq \mathbf{d}_N, \end{aligned}$$

where N is the number of sub-problem constraints. The block diagonal structure of the sub-problem constraints can be exploited by formulating a relaxation (simplification) of the original problem (LR($\boldsymbol{\lambda}$)). In this relaxation the coupling constraints are replaced by terms in the objective function penalizing their violation.

$$\begin{aligned} \max_{\mathbf{x},\mathbf{u}} \mathbf{c}^T\mathbf{x} + \mathbf{f}^T\mathbf{u} + \boldsymbol{\lambda}^T(\mathbf{b} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u}) \\ \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \leq \mathbf{d} \\ \mathbf{u} \in \{0,1\} \end{aligned} \qquad (\mathrm{LR}(\boldsymbol{\lambda}))$$

The most important property of (LR($\boldsymbol{\lambda}$)) is that its (optimal) objective value for any positive value of the $\boldsymbol{\lambda}$ multipliers is always greater or equal to the (optimal) objective value of (MILP). This property of (LR($\boldsymbol{\lambda}$)) directly follows from the duality theory. Every solution to (LR($\boldsymbol{\lambda}$)) hence represents an upper bound to the optimal value of (MILP).

Another way of obtaining bounds to the optimal solution of (MILP) is using its linear relaxation (LPR). This relaxation consists in replacing the integrality constraints $\mathbf{u} \in \{0,1\}$ with $\mathbf{u} \in \langle 0,1 \rangle$.

$$\begin{aligned} \max_{\mathbf{x},\mathbf{u}} \mathbf{c}^T\mathbf{x} + \mathbf{f}^T\mathbf{u} \\ \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b} \\ \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \leq \mathbf{d} \\ \mathbf{u} \in \langle 0,1 \rangle \end{aligned} \qquad (\mathrm{LPR})$$

It should be noted that the purpose of these relaxations lies solely in providing good bounds to the optimal objective value. The other important property of these relaxations is that they are relatively easily solvable, comparing to the original problem.

Except for some rare cases, solutions of these relaxations are never valid with respect to the original problem. However, these relaxations will be used in the algorithm for obtaining good feasible solutions and also within the customized B&B algorithm developed for providing information on how these solutions are good in reality.

4.2 General branch-and-bound algorithm

The general branch and bound algorithm is based on splitting the solution space (4.1) of a MILP[1] problem into two disjoint subspaces $S_0 = S \cap \{u_i = 0\}$

[1]Please note that in this case the MILP problem contains only binary variables and no general integer variables - these problems are also denoted as mixed-binary linear programming (MBLP).

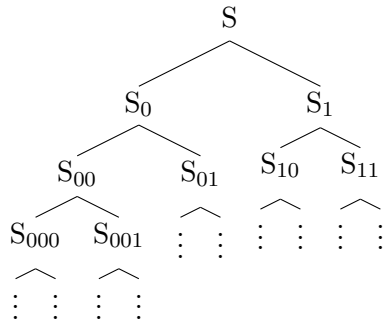
and $S_1 = S \cap \{u_i = 1\}$, where i is an index of a binary variable selected by a heuristics. This heuristics tries to find a sequence of variables u_i promising the fastest convergence towards as tight bound as possible.

$$S = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{Z}^p : \ \mathbf{Ax} + \mathbf{Bu} \leq \mathbf{b}, \mathbf{Cx} + \mathbf{Du} \leq \mathbf{d}\} \tag{4.1}$$

The space S represents a solution set of the root node of a B&B tree. The subspaces S_0 and S_1 then represent solution sets of the first level nodes. These subspaces are again split to $S_{00} = S \cap \{u_i = 0\} \cap \{u_j = 0\}$, $S_{01} = S \cap \{u_i = 0\} \cap \{u_j = 1\}$, $S_{10} = S \cap \{u_i = 1\} \cap \{u_j = 0\}$ and $S_{11} = S \cap \{u_i = 1\} \cap \{u_j = 1\}$, creating descendants to nodes S_0 and S_1. This way the splitting continues until:

- all the binary variables are fixed,
- the solution of a relaxation satisfies $\mathbf{u} \in \mathbb{Z}^p$,
- a node and its corresponding branch (all the descendant nodes) can be pruned, i.e. the bound provided by relaxation for this node is worse than the best known solution.

The resulting B&B tree looks like the following example.



The idea behind the B&B algorithm is that by using the bounds obtained from relaxations, an otherwise huge tree is reduced and only a small fraction of nodes is worth evaluation. Usually a linear relaxation with the solution space (4.2) is solved at each node. As soon as a feasible solution is known the pruning may begin.

$$P = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^p : \ \mathbf{Ax} + \mathbf{Bu} \leq \mathbf{b}, \mathbf{Cx} + \mathbf{Du} \leq \mathbf{d}\} \tag{4.2}$$

Solutions are generally obtained by

- heuristic algorithms which are usually able to find a very good feasible solution much sooner than by searching the tree,
- reaching a leaf of the tree, i.e. after all the binary variables are fixed or the solution of a relaxation satisfies integrality constraints.

Clearly it is crucial to have a good feasible solution as soon as possible, in order to quickly reduce the solution space worth evaluation.

4.3 Outline of the proposed algorithm

The algorithm proposed in this thesis consists of two subalgorithms. The purpose of the first algorithm is to provide good feasible solutions. This algorithm is a heuristic and does not provide any estimation on how good the provided solutions actually are. However, to meet the second goal of this thesis (see Section 1.2) the estimation on proximity of solutions to optimum is required. For this purpose the second algorithm is introduced, which provides the certificate of near-optimality for known solutions. This second subalgorithm is a customized B&B algorithm using the proposed heuristics for obtaining good feasible solutions. Figure. 4.1 shows an outline of the overall algorithm.

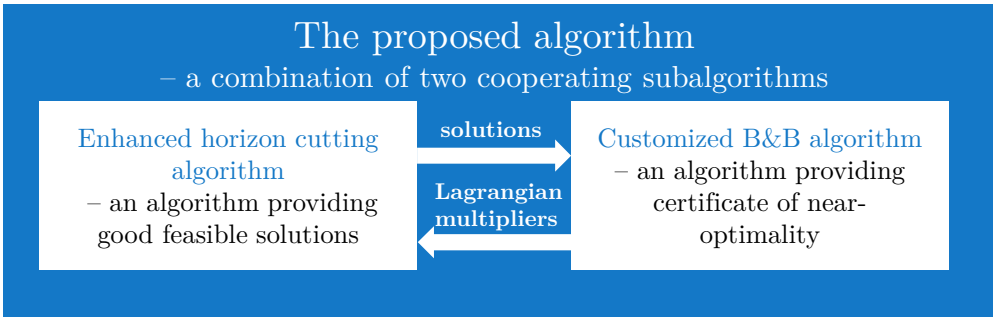


Figure 4.1: Outline of the algorithm proposed for CHP production and trade planning problem.

The two sub-algorithms cooperate and form the following algorithm:

1. Start EHC algorithm with an initial set of Lagrangian multipliers $\boldsymbol{\lambda}$ - see sections 4.1 and 4.5 for details.
2. Start the customized B&B algorithm to run in parallel.
3. Perform one iteration of the customized B&B (i.e. branch a B&B node).
4. Retrieve new feasible solutions provided by EHC algorithm and hand these over to the customized B&B algorithm.
5. Check whether EHC algorithm is finished. If it is, restart it with the current values of Lagrangian multipliers to run in parallel.
6. If there are any solutions and a certificate of (near-)optimality for the best known solution is found, end algorithms and return the solution.
7. Continue with step 3.

The algorithms will be explained in detail in Sections 4.4 and 4.5.

4.4 Algorithm providing good feasible solutions

Commonly used approach to handle very large CHP operations planning problem instances is horizon cutting [DH12], [TBW05] and [KC12]. The optimization horizon is divided into convenient time periods which are then solved in a sequence with the last sample of each period representing an initial condition for the next period. In the following text, this approach will be denoted as a basic horizon cutting (BHC) algorithm.

Within the BHC algorithm, the optimization problem is not decomposed in the fashion of (LR($\boldsymbol{\lambda}$)). Instead, it is formulated as a sequence of smaller optimization problems - segments - representing overlapping time periods of the optimization horizon. See Figure 4.2 fo illustration. These segments are then connected by fixing the overlapping part to the values obtained by solving a previous segment.

The main difference to the (LR($\boldsymbol{\lambda}$)) decomposition approach is in the way how the linking constraints are handled. In the case of (LR($\boldsymbol{\lambda}$)) these are omitted and their violation is penalized in the objective function. With the BHC approach, the linking constraints are not omitted, but formulated in a different way. For instance consider a problem with planning horizon of 24h being divided into 8h

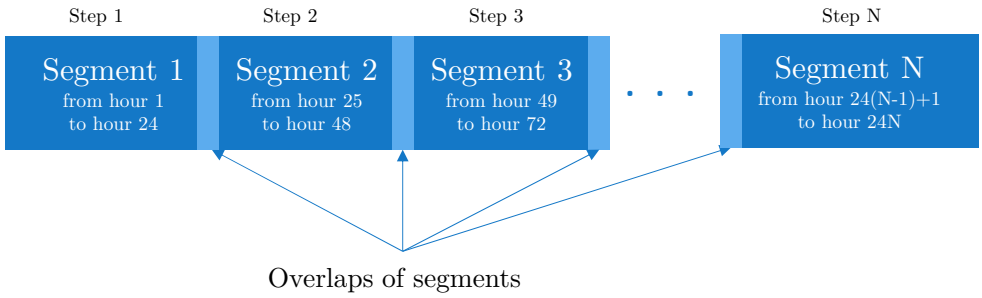


Figure 4.2: Outline of the basic horizon cutting algorithm with the segment length of 24h.

segments and a turbine having 24h minimum up time. In (LR($\boldsymbol{\lambda}$)) the minimum up time constraint is omitted and its violation is penalized. In the case of BHC this constraint is divided into three 8h minimum up constraints, one for each segment. Properly using the overlapping parts of segments we may even enforce the 24h minimum up time constraint.

Hence, the main advantage of the BHC approach is that when a solution is found, it is feasible with respect to the original problem (if the connection of segments is done carefully).

There are, however, also some issues:

- With power products spanning more than one segment, the BHC may easily fail as the power production decided in a former segment may not be feasible in the latter.
- The solution quality is greatly affected by the decisions made in the segment computed first as these are made without considerations of any future time periods. The solution may therefore sub-optimal.
- If a segment is infeasible (with respect to its initial condition) it is hard or impossible to track what was the cause of infeasibility. Hence if the infeasibility occurs the algorithm fails.

4.4.1 Enhancement of horizon cutting

We propose a novel EHC algorithm with the purpose to overcome the aforementioned issues of BHC. In the following sections the enhancements will be presented and explained.

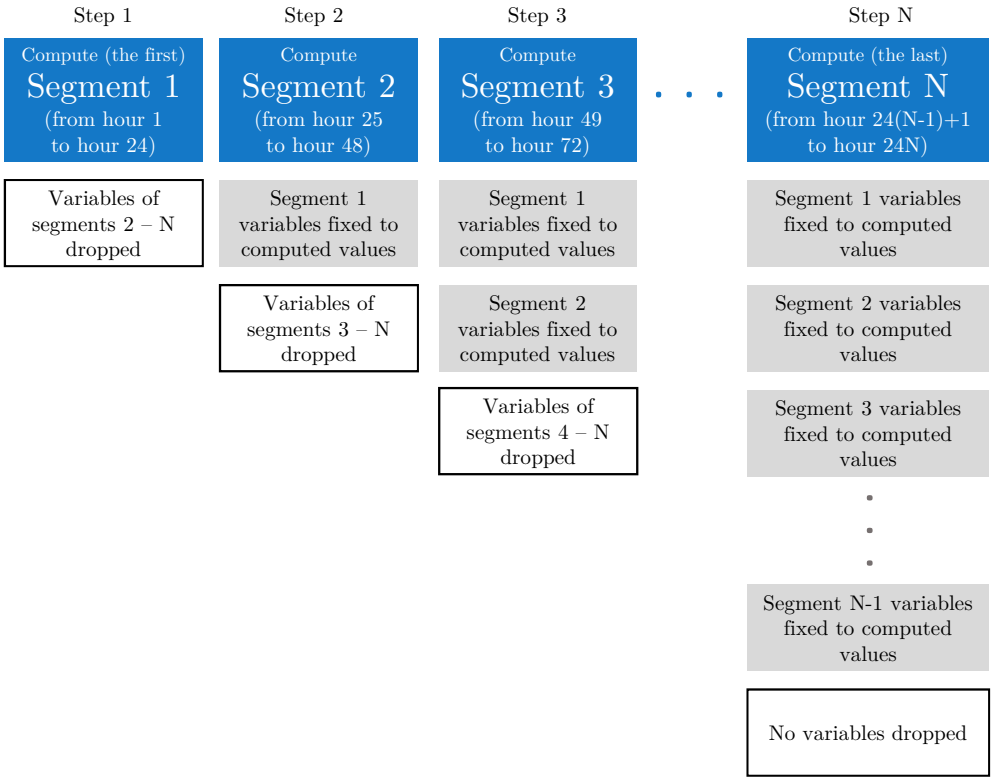


Figure 4.3: Outline of EHC algorithm with the segment length of 24h and the segment sequence starting at segment 1 and ending at the last segment N.

At each step of EHC algorithm, a sub-problem based on problem (MILP). In the sub-problem some of the variables of (MILP) are fixed and some are omitted, using a concept of variables dropping which will be explained further in the text. These sub-problems are then solved in a specified sequence. The outline of the algorithm is in Figure 4.3. In this figure a sequence starting at segment 1 and ending at the last segment N is chosen. However, the order of segments in the sequence can be chosen arbitrarily, which will be utilized in the next section, dealing with a parallel computation of several different sequences. See Figure 4.4 for example.

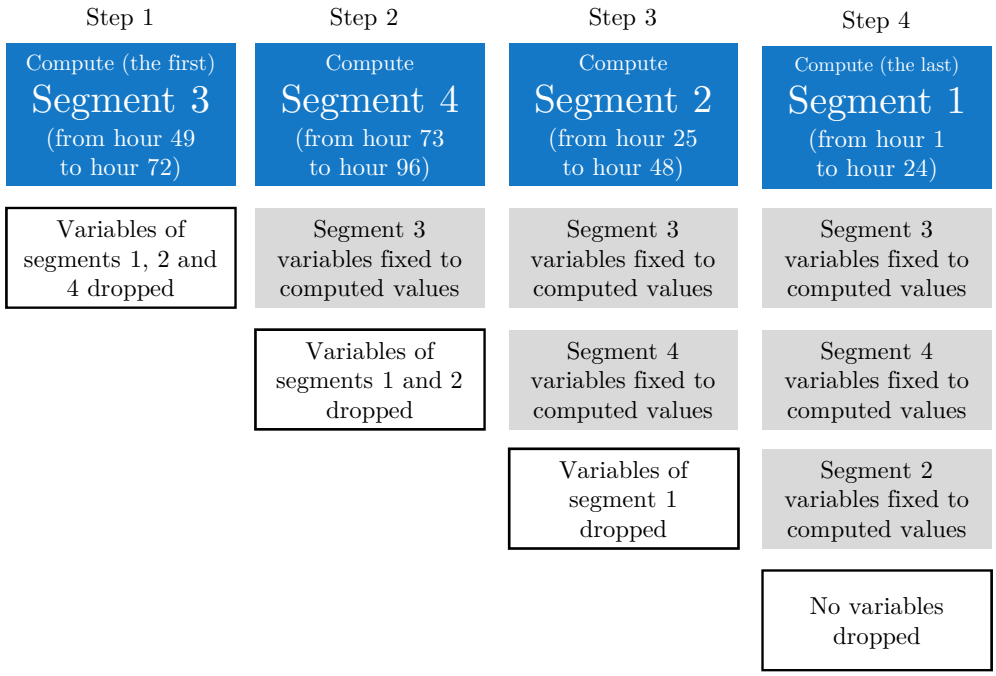


Figure 4.4: Outline of EHC algorithm with the segment length of 24h and the segment sequence $\{3, 4, 2, 1\}$.

Dropping variables

In order to handle power products spanning more than one segment another approach for dividing the problem into segments must be employed. We introduce the concept of variables dropping. Instead of formulating smaller overlapping problems representing the segments, a currently computed segment is formulated first by fixing all the variables of already computed segments and dropping all the variables of following segments. E.g. the first segment is formulated by dropping all the variables belonging to other segments (no variables are fixed as no segments were computed yet).

When a variable is dropped from the optimization problem, it is removed from all the constraints while preserving validity of the constraints[2]. To preserve the validity of a constraint the variable being dropped is fixed to its lower or upper bound, depending on the sign of its coefficient and sense of the constraint.

[2]The validity means, that all the possible solutions to the original problem satisfy the constraint.

It should be noted that to drop a variable from an equality, this equality must be transformed into two inequalities.

The advantage of dropping variables over simply omitting constraints can be illustrated using the minimum up time constraint (4.3a). When the variable $s_U(t-2)$ is dropped the resulting constraint is (4.3b), where $s_U^{LB}(t-2)$ represents a lower bound to $s_U(t-2)$ and its value is 0. The resulting constraint is definitely tighter that no constraint at all.

$$s_U(t-2) + s_U(t-1) + s_U(t) \leq u(t) \tag{4.3a}$$

$$s_U(t-1) + s_U(t) \leq u(t) - s_U^{LB}(t-2) \longrightarrow s_U(t-1) + s_U(t) \leq u(t) \tag{4.3b}$$

$$\begin{aligned} &\max_{\mathbf{y}} \mathbf{g}^T\mathbf{y} \\ &\mathbf{F}\mathbf{y} \leq \mathbf{h} \\ &\mathbf{y}_i \in \{0,1\}\, \forall i \in I \end{aligned} \tag{MILP(y))}$$

Let's consider having the optimization problem (MILP) written in a simpler way as (MILP(y)), not distinguishing between binary and continuous variables and between coupling constraints and the decomposable part of the problem. After fixing all the variables belonging to the already computed segments a and dropping all the variables of not yet computed segments n except the variables of the segment being currently computed c, the resulting optimization problem can be described with (4.4), where the variables $\mathbf{y}_c$ represent the variables of the currently computed segment, matrix $\mathbf{F}_c$ is the corresponding part of the matrix $\mathbf{F}$ and vector $\mathbf{g}_c$ is the corresponding part of the objective function. The values $\bar{\mathbf{y}}_a$ are the solution to already computed segments. The matrices $\mathbf{F}_n^+$ and $\mathbf{F}_n^-$ are the positive and negative parts of matrix $\mathbf{F}_n$ (i.e. $\mathbf{F}_n^+ + \mathbf{F}_n^- = \mathbf{F}_n$) representing the part of $\mathbf{F}$ belonging to the not yet computed segments. Finally the values $\mathbf{y}_n^{LB}$ and $\mathbf{y}_n^{UB}$ are the lower and upper bounds to the variables of the not yet computed segments.

$$\max_{\mathbf{y}_c} \mathbf{g}_c^T\mathbf{y}_c \tag{4.4a}$$

$$\mathbf{F}_c\mathbf{y}_c \leq \mathbf{h} - \mathbf{F}_a\bar{\mathbf{y}}_a - \mathbf{F}_n^+\mathbf{y}_n^{LB} - \mathbf{F}_n^-\mathbf{y}_n^{UB} \tag{4.4b}$$

$$\mathbf{y}_i \in \{0,1\}\, \forall i \in I \tag{4.4c}$$

There are several features of (4.4) that should be noted:

- All the sub-problem constraints of the segment c are contained within (4.4) in their unchanged form (these constraints do not contain any variables of segments other than c).
- The sub-problem constraints belonging to other segments than c are simply discarded as these do not contain the variables $\mathbf{y_c}$. The same is true for coupling constraints which do not contain any of $\mathbf{y_c}$ variables.
- The number of variables is relatively small comparing to (MILP(y)) (depending on the number of segments). Also the number of constraints is small, depending on the number of coupling constraints.
- The order in which the segments (4.4) are solved can be chosen arbitrarily.
- It may be infeasible due to the fixation of $\mathbf{y}_a$. In such a case a procedure called redispatch is employed.

Solving multiple segment sequences in parallel

As was already mentioned, when using BHC, the solution quality is greatly affected by the decisions made in the segment computed first. To handle this issue, we exploit the fact that, utilizing the concept of dropping variables, the segments can be computed in an arbitrary order. The order corresponding to the best possible solution quality is, however, not known. Hence we propose to perform several computations with different segment sequences. The advantage of performing several computations (with different segment sequences) is that the possibility of failure of finding a feasible solution is smaller than in the case of a single computation (say with the the first segment being the starting segment). Hence, we employ a parallel computation of several sequences, each starting at different segments. With the parallelization the computation time required to finish all the sequences remains reasonable (no more than several times the time required by BHC, depending on the number of processors available).

Redispatch

An infeasibility of a segment may occur, i.e. the particular values of variables of already computed segments $\mathbf{y}_a$ may yield the segment infeasible. To cope with this event a redispatch of already computed continuous variables $\mathbf{y_a^c}$ is performed, while the values of already computed binary $\mathbf{y_a^b}$ variables remain

fixed. The resulting redispatch optimization problem (4.5) thus contains several times more variables than (4.4), but the number of binary variables remains the same. Hence solving the redispatch is computationally expensive but still tractable as it does not contain more binary variables than (4.4). Preserving the fixations of binary variables helps keeping the tractability of the optimization problem, however, also brings a possibility (according to our experience small) of redispatch failure.

$$\max_{\mathbf{y}_c} \mathbf{g}_a^{cT}\mathbf{y}_a^c + \mathbf{g}_c^T\mathbf{y}_c \tag{4.5a}$$

$$\mathbf{F}_a^c\mathbf{y}_a^c + \mathbf{F}_c\mathbf{y}_c \leq \mathbf{h} - \mathbf{F}_a^b\bar{\mathbf{y}}_a^b - \mathbf{F}_n^+\mathbf{y}_n^{LB} - \mathbf{F}_n^-\mathbf{y}_n^{UB} \tag{4.5b}$$

$$\mathbf{y}_i \in \{0,1\}\,\forall i \in I \tag{4.5c}$$

In other words, when infeasibility occurs, all the computed values of continuous variables are discarded and recomputed along with the current segment. Because (4.5) may be quite a big problem with respect to continuous variables, the redispatch should be performed as seldom as possible. As tight bounds to variables $\mathbf{y}^{LB}$, $\mathbf{y}^{UB}$ as possible can reduce the number of redispatches.

Tightening bounds of variables

Tightening bounds represents one of the core presolving techniques for MILP optimization problems [Mah10]. MILP solvers perform presolve before the B&B algorithm is started (less extensive presolve is also performed at nodes of the B&B algorithm). The purpose of the presolve is twofold:

- To reduce the size of the optimization problem with respect to the number of variables and constraints. It can considerably reduce the amounts of data to be processed, allowing to speed-up the algebraic operations.
- To tighten the difference between spaces of feasible continuous and of feasible integer solutions. This is crucial for efficiency of the B&B algorithm as it is greatly dependent on the tightness of the linear relaxation.

Tightening a bound $l_j \leq y_j \leq u_j$ means that a lower bound $\hat{l}_j$ higher than l_j is sought and an upper bound $\hat{u}_j$ lower than u_j is sought. This is performed observing that constraints (4.6) of the optimization problem (MILP(y)) with bounds L_i and B_i defined as (4.7) yield another bound for a variable k with the coefficient $a_{ik} \neq 0$. If the coefficient a_{ik} is positive, then the constraint yields an upper bound $\hat{u}_{ik}$ on variable y_k using (4.9). The negative coefficient a_{ik} then yields a lower bound $\hat{l}_{ik}$ using (4.8).

$$U_i \leq \sum a_{ij} y_{ij} \leq b_i \leq L_i \tag{4.6}$$

$$L_i = \sum_{j:a_{ij}>0} a_{ij} l_j + \sum_{j:a_{ij}<0} a_{ij} u_j \tag{4.7a}$$

$$U_i = \sum_{j:a_{ij}>0} a_{ij} u_j + \sum_{j:a_{ij}<0} a_{ij} l_j \tag{4.7b}$$

$$\hat{l}_{ik} = \frac{b_i - L_i + a_{ik} u_k}{a_{ik}} \tag{4.8}$$

$$\hat{u}_{ik} = \frac{b_i - L_i + a_{ik} l_k}{a_{ik}} \tag{4.9}$$

If any of the inferred bounds $\hat{l}_{ik}$ ($\hat{u}_{ik}$) is better than the current bound l_k (u_k) then the bound may be improved. The new bounds $\hat{l}_k$ and $\hat{u}_k$ are therefore obtained from (4.10).

$$\hat{l}_k = \begin{cases} \max \hat{l}_{ik} & \text{if } \max \hat{l}_{ik} > l_k \\ l_k & \text{otherwise} \end{cases} \tag{4.10a}$$

$$\hat{u}_k = \begin{cases} \min \hat{u}_{ik} & \text{if } \min \hat{u}_{ik} < u_k \\ u_k & \text{otherwise} \end{cases} \tag{4.10b}$$

Clearly, the tightening can be (and is) performed iteratively. If any of the variable bounds improve, the constraint bounds (4.7) can be updated accordingly and may yield another variables bound improvements.

In the case of solving the optimization problem using a plain general-purpose solver, it is not necessary to attempt to improve variables bounds as this is what the solver does at the presolve phase (among other problem improvements and reductions). However, when decomposing the problem into sub-problems, it is important to do this tightening before the decomposition as after the decomposition some important information is lost (sub-problems do not contain all the constraints).

Within EHC it is important to perform the tightening before the variables are dropped (4.4). It is, however, important to do the tightening considering fixation of binaries only as these fixations are final, the fixations of continuous variables may be recomputed by solving redispatch (4.5).

Generally, tightening is very important in the case of solving a decomposed problem especially after some of the binary variables have already been fixed as the tightening may yield another variables fixations (among sub-problems).

It should also be noted that when an upper bound of a binary variable drops below the value of 1, this variable can be safely fixed to the value 0, and vice versa.

Window

Another way of reducing the number of necessary redispatches is to consider future segments while computing the current segment. This means including some of the otherwise dropped variables into the optimization problem (4.4). Typically we include the variables (and constraints) of several segments following the currently computed segment. The binary variables of the added segments are relaxed to continuous variables with their values restricted into interval $\langle 0; 1 \rangle$. We denote this approach as a window and it is illustrated by Figure 4.5.

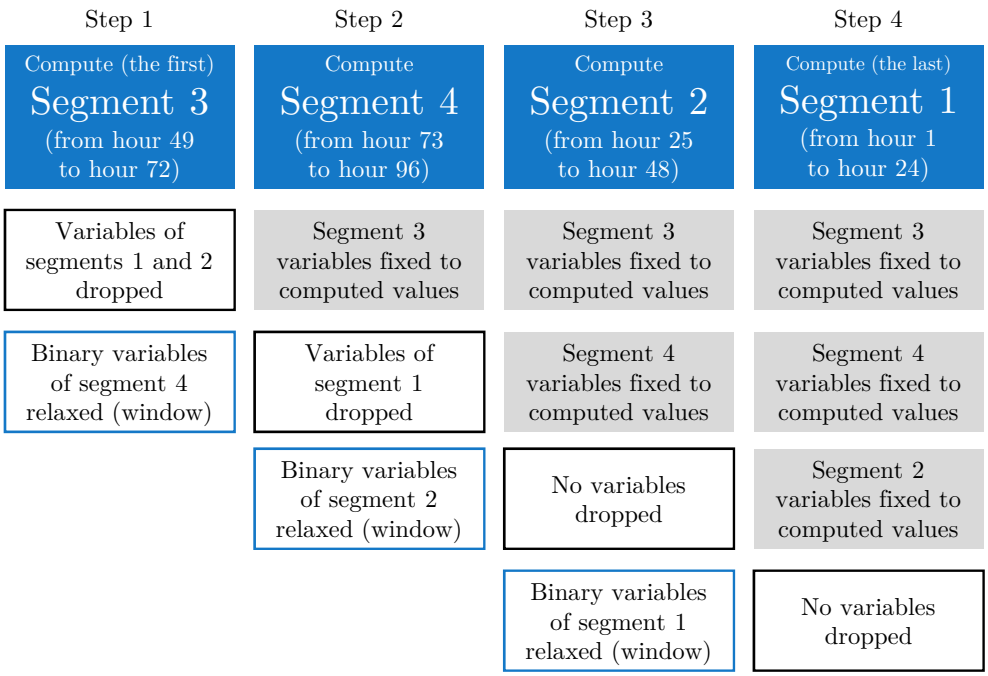


Figure 4.5: Outline of EHC algorithm with the segment length of 24h, window length 1 and the segment sequence $\{3, 4, 2, 1\}$.

The foresight that a window brings, positively affects also a solution quality. However, longer windows have a negative impact on the computation times (a window of the same size as the remaining number of segments corresponds to the redispatch).

Penalizing relaxed constraints in the objective function

When Lagrangian multipliers $\boldsymbol{\lambda}$ of (LR($\boldsymbol{\lambda}$)) providing a tight bound for (MILP) are known, these multipliers can be used to improve the efficiency of EHC. The objective function of (4.4) is modified by the penalization of the violation of constraints (in their unrelaxed form) which are relaxed in (4.4), i.e. for which $\mathbf{F}_a^b \bar{\mathbf{y}}_a^b + \mathbf{F}_n^+ \mathbf{y}_n^{LB} + \mathbf{F}_n^- \mathbf{y}_n^{UB} \neq 0$.

However, when EHC is started, no Lagrangian multipliers are known, yet. These will be known after several iterations of Lagrangean relaxation (of the B&B root node), the procedure that will be described in the following chapter. Hence, as EHC and B&B algorithms run in parallel, after the first run of EHC is finished, the EHC is restarted using the best known multipliers. In some cases this leads to better solutions and reduces chance of failing to find a feasible solution.

Parametrization

There are three parameters affecting the speed of the algorithm, the solution quality and the possibility of a failure. The optimal settings of these parameters may differ among CHP plants.

Computation sequences. The order in which the segments will be computed can be chosen arbitrarily, hence the number of all possible sequences is very large, it is $n!$. According to our experience, sequences of adjacent segments work well - beginning from a starting segment to the end of the planning horizon. When the end is reached, it is continued from the segment before the starting segment to the first segment. Example of such a sequence is in Figure 4.6. Also the starting segments of sequences are chosen so that the sequences are different to each other as possible.

Length of segments. The lowest possible segment length is the length of one hour, which corresponds to the sampling period. However, it is more efficient to compute larger segments. This has two reasons. The first is that the state-of-the-art solvers usually can easily handle the problems with the segment lengths from 8 hours to 48 hours. In some cases, it is much faster to compute a segment with

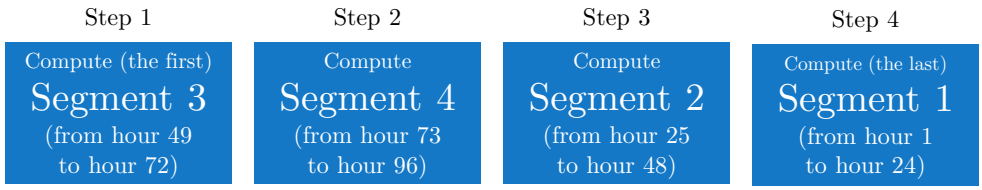


Figure 4.6: Example of a computed segment sequence, first computing segments 3 to 4 and then continuing with segments 2 to 1.

length of 24 hours than to compute 24 single hour segments. This is because of the overhead of the solver.

The second and the more important reason is that by using longer segments, the number of coupling constraints decreases (some of the constraints coupling together 2 different hours would fall into a single segment now). A typical dependences of a relative number of coupling constraints on segment length are in Figure 4.7. The figure shows this dependence for three different models (plants). The number of coupling constraints has great impact on the quality of solutions produced by the algorithm. It also decreases the probability of the algorithms's failure. Considering the three cases in Figure 4.7, the segment lengths of 24 hours seem as a viable option for all the three models. However, a segment of this length must also be easily computable, otherwise a shorter segment should be chosen.

Window length. In the cases of problems with rather smaller spaces of feasible solutions, it may be hard to find a feasible solution. Hence, if the EHC fails to find a solution with the current setting of window, the window length is increased and the algorithm is restarted.

Algorithm for providing feasible solutions

A single run of EHC, i.e. a run with a defined starting segment (and other parameters), is summarized in the following paragraphs.

1. Let s be the index of the starting segment. Create a list S of sub-problem indices ordered in the following manner:

$$S = [s, s+1, s+2, ..., N, s-1, s-2, ..., 1]\,,$$

where N is the number of sub-problems. Let i be the index of the set S pointing at the currently solved segment. Initialize i to the value 1.

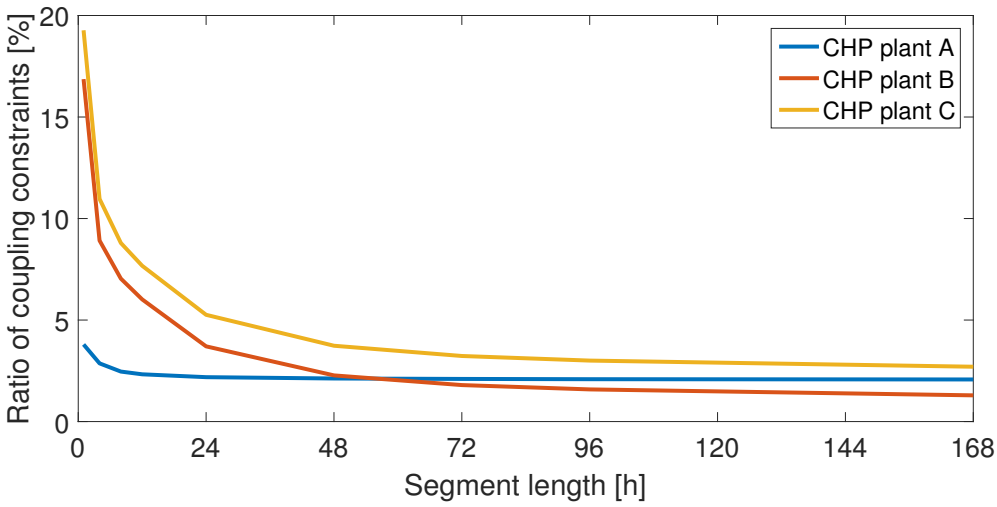


Figure 4.7: Typical dependence of a relative number of coupling constraints on segment length.

2. Fix the variables of sub-problems S_1 to S_{i-1} to the already computed values (if any). Tighten the bounds of variables with respect to the values of binary variables.

3. Penalize the violation of the constraints containing variables of sub-problems S_{i+1} to S_N in the objective function (if multipliers are known).

4. Drop variables of sub-problems S_{i+w+1} to S_N from the optimization problem, where w is window length.

5. Relax binary variables of S_{i+1} to S_{i+w} to continuous variables with bounds $\langle 0, 1 \rangle$.

6. Compute the resulting optimization problem.

7. If the problem is infeasible unfix the values of continuous variables of S_1 to S_{i-1} and recompute.

8. If the problem remains infeasible, stop the algorithm and return no solution.

9. If $i == N$, stop the algorithm and return the solution. Else increment the value of i and continue with the step 2.

4.5 Providing certificate of near-optimality

The operators using the decision support tool for CHP operations planning usually want to know how good is the solution provided by the tool. The best measure of a solution quality is to compare it against the optimal solution[3]. To do this, the optimal solution must be known (or at least its objective value) and a certificate of optimality must be provided. This can be relatively easy found in the case of linear and convex optimization problems. In the case of non-linear optimization problems (such as MILP problems) it is not simple (or even not possible) to provide the certificate.

In the case of MILP optimization problems, such as (MILP(y)), it is possible to find an upper bound to the optimal objective function σ^* by solving a relaxation of the problem. The main property of a valid relaxation is that its solution space contains the complete solution space of the original problem, i.e. all the solutions which are feasible for the original problem are also feasible for the relaxed problem. As the objective functions of the relaxation and the original problem are the same, the optimal objective value of the relaxation σ^{UB} must always be better or equal to the optimum of the original problem σ^*.

The goal is to find a relaxation for which the value σ^{UB} is as close to σ^* as possible. The value of σ^{UB} then represents an estimate to σ^* which can be used to obtain an estimate on proximity of a solution to the optimum. Having the objective value of the currently best known solution to the original problem σ^{LB}, the estimate on proximity to optimum can be described as (4.11). See Figure 4.8 for illustration.

$$\Delta\sigma = \frac{\sigma^{UB} - \sigma^{LB}}{\|\sigma^{LB}\|} \tag{4.11}$$

This value represents a so-called gap of the current incumbent (the currently best known solutions) and is usually expressed in percentage as $\Delta\sigma^{\%} = 100\Delta\sigma$. Reaching zero (or sufficiently small) gap then certifies the optimality of the incumbent.

The most simple relaxation of MILP is linear relaxation (LPR) which is used within a general B&B algorithm to facilitate pruning of the B&B tree.

We customized the general B&B for the purpose of solving CHP operations planning. This customization lies above all in the way how bounds are computed and how branching variables are selected. Besides, the customized B&B

[3] Alternatively, it can be compared to a reference solution, e.g. a solution that would be derived by the operators themselves, not having the decision support tool at hand.

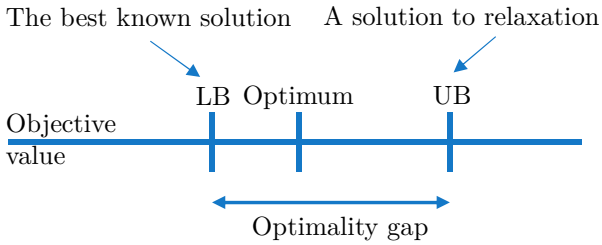


Figure 4.8: An illustration explaining the concept of optimality gap.

utilize EHC as a heuristic for obtaining feasible solutions. The following sections introduce these concepts.

4.5.1 Customized B&B algorithm for CHP operations planning

The algorithm proposed in this thesis is based on the framework of the general B&B while a bounding function and a branching rule are tailored for the purpose of CHP operations planning, using the knowledge of the problem structure, i.e. its block diagonal sub-structure. Also, the knowledge on what types of constraints usually form the set of coupling constraints is used for selecting an efficient branching variables selection. The properties of the customized B&B algorithm are described in the following paragraphs.

Bound computation

We propose to compute two types of bounds by solving two different relaxations. The first bound can be computed quickly but is generally not so tight as the second, computation of which is more time-consuming.

The first bound is based on solving a linear relaxation (LPR). Actually the full (LPR) is solved only in the root node. This is because even (LPR) can be quite hard to solve as it contains all the constraints and variables of (MILP) except for the integrality constraints. Hence, in other nodes the relaxation (LPR($\boldsymbol{\lambda}$)) is solved, with $\boldsymbol{\lambda}$ being the best (i.e. providing the tightest bound) Lagrangian multipliers inherited from the parent node, where these were obtained by solving LR. Solving (LPR($\boldsymbol{\lambda}$)) is inexpensive as it splits into small independent linear sub-problems.

$$\begin{aligned} \max_{\mathbf{x},\mathbf{u}} \mathbf{c}^T\mathbf{x} + \mathbf{f}^T\mathbf{u} + \boldsymbol{\lambda}^T(\mathbf{b} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u}) \\ \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \leq \mathbf{d} \\ \mathbf{u} \in \langle 0, 1 \rangle \end{aligned} \qquad (\text{LPR}(\boldsymbol{\lambda}))$$

The purpose of this relaxation is to check whether the node can be pruned without solving the full (LR($\boldsymbol{\lambda}$)). Also a possible infeasibility of the node can be discovered. Especially in lower levels of the B&B tree the bounds obtained by solving (LPR($\boldsymbol{\lambda}$)) are tight enough to prune most nodes on their basis.

As was already indicated, a tighter bound can be obtained by solving (LR($\boldsymbol{\lambda}$)). Ideally we would find the particular $\boldsymbol{\lambda}$ for which the optimal objective value of (LR($\boldsymbol{\lambda}$)) is the smallest, and hence the tightest, possible, i.e. solving (LR). In the root node we start with the dual values to solution of (LPR). The algorithm then continues by iteratively improving the coefficients $\boldsymbol{\lambda}$.

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \max_{\mathbf{x},\mathbf{u}} \mathbf{c}^T\mathbf{x} + \mathbf{f}^T\mathbf{u} + \boldsymbol{\lambda}^T(\mathbf{b} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u}) \\ \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \leq \mathbf{d} \\ \mathbf{u} \in \{0, 1\} \\ \boldsymbol{\lambda} \geq 0 \end{aligned} \qquad (\text{LR})$$

It can take quite long time for the LR to converge to the best $\boldsymbol{\lambda}$. Hence, we perform only several iterations of LR and when the convergence slows down it is interrupted and the lowest objective value of (LR($\boldsymbol{\lambda}$)) is used as a bound in the current node. The corresponding multipliers $\boldsymbol{\lambda}$ are then used as the starting ones in descendant nodes.

Non-negativity of $\boldsymbol{\lambda}$. Only non-negative values of $\boldsymbol{\lambda}$ can produce a bound to (MILP). This condition follows from the duality theory. The reason is that violation of constraints $\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b}$ is penalized. If a constraint i of $\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b}$ is violated, i.e. $a_i x_i + b_i u_i > b_i$, the term $b_i - a_i x_i - b_i u_i$ is negative and therefore with positive λ_i it decreases the objective value of (LR($\boldsymbol{\lambda}$)) and thus penalizes the violation. If any λ_i was negative then actually the satisfying of the constraint would be penalized and the optimal objective value of (LR($\boldsymbol{\lambda}$)) would not represent bound to (MILP).

Improving $\boldsymbol{\lambda}$. The question is how to improve the coefficients $\boldsymbol{\lambda}$, i.e. how to find new coefficients $\boldsymbol{\lambda}_{i+1}$ which would provide a better bound than the current coefficients $\boldsymbol{\lambda}_i$. There exist at least three well known approaches for finding $\boldsymbol{\lambda}_{i+1}$:

- sub-gradient method,
- cutting plane method
- and bundle method.

These three methods will be described in the following paragraphs, their advantages and drawbacks are discussed and finally one of these methods is chosen.

Sub-gradient method. This method is often used for its simplicity. In the following text, this method will be outlined. See [Lem01] for more detailed description.

The method is based on the well known gradient descent method, which uses a gradient denoting the steepest descent direction to find a local minimum of smooth functions. However, the objective function of (LR) is not smooth, and therefore not differentiable everywhere. Actually, it is not differentiable in the vertices of the polyhedron $S = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{Z}^p : \mathbf{Cx} + \mathbf{Du} \leq \mathbf{d}\}$. Since the optimal solution $(\mathbf{x}_i^*, \mathbf{u}_i^*)$ lies at a vertex of S the gradient descent method cannot be used.

Instead a sub-gradient method is used. There is no unique gradient at vertices of S, however, there is an infinite number of sub-gradients. A sub-gradient is a plane touching S at $(\mathbf{x}_i^*, \mathbf{u}_i^*)$ and everywhere else either touching or below the graph of S. One of the sub-gradients is (4.12), i.e. the violation of relaxed constraints (the more violated a constraint is the higher value of the corresponding λ_j must be).

$$\mathbf{g}(\mathbf{x}_i^*, \mathbf{u}_i^*) = \mathbf{b} - \mathbf{A}\mathbf{x}_i^* - \mathbf{B}\mathbf{u}_i^* \tag{4.12}$$

The new value of multipliers $\boldsymbol{\lambda}_{i+1}$ are then computed with (4.13),

$$\boldsymbol{\lambda}_{i+1} = \max\left(0, \boldsymbol{\lambda}_i + s_i g(\mathbf{x}_i^*, \mathbf{u}_i^*)\right) \tag{4.13}$$

where s_i is the step with computed with

$$s_i = \frac{a_i \left(J_D(\boldsymbol{\lambda}_i) - J^{LB}\right)}{\|\mathbf{g}_i\|^2} \tag{4.14}$$

where $J_D(\boldsymbol{\lambda}_i)$ is the optimal objective value of (LR) for $\boldsymbol{\lambda}_i$, J^{LB} is the objective value of the currently best known solution and a_i is a scalar within $\langle 0, 2\rangle$, the value of which usually starts at 2 and is halved every time an iteration does not result in an improvement of the bound.

The main advantages of this approach is its straightforward implementation and that the computation of $\boldsymbol{\lambda}_{i+1}$ is inexpensive. However the convergence properties of the sub-gradient algorithm are typically not good.

Cutting planes. The cutting plane method (see [Lem01]) is based on an idea of iteratively building an estimation of the Lagrangian dual (LR). With each iteration the accuracy of the estimation is improved by adding a new cutting plane based on a solution $(\mathbf{x}_i^*, \mathbf{u}_i^*)$ obtained by solving (LR($\boldsymbol{\lambda}$)) for the current value $\boldsymbol{\lambda}$. The estimation can be expressed in the form (LRe),

$$\begin{gathered} \min_{z,\boldsymbol{\lambda}} z \\ z \geq \mathbf{c}^T\mathbf{x}_i + \mathbf{f}^T\mathbf{u}_i + \boldsymbol{\lambda}(\mathbf{b} - \mathbf{A}\mathbf{x}_i - \mathbf{B}\mathbf{u}_i),\ i \in P \subset S \\ \boldsymbol{\lambda} \geq 0 \end{gathered} \tag{LRe}$$

where P is a subset of extreme points of polyhedron (4.15). The subset P is built by solving (LR($\boldsymbol{\lambda}$)) for different values of $\boldsymbol{\lambda}$.

$$S = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{Z}^p :\ \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b}\} \tag{4.15}$$

It can be shown that (LRe) is equivalent to (LR) if the set P contains all the vertices of S. This condition is sufficient but not necessary. Even if some of the vertices are omitted, the estimation (LRe) can have the same objective value as (LR). This is true if for all the not included vertices $(\mathbf{x}_k, \mathbf{u}_k)$ holds that $z > \mathbf{c}^T\mathbf{x}_k + \mathbf{f}^T\mathbf{u}_k + \boldsymbol{\lambda}^*(\mathbf{b} - \mathbf{A}\mathbf{x}_k - \mathbf{B}\mathbf{u}_k)$. Hence it is not necessary to enumerate all the vertices in order to obtain a good estimate on (LR).

The Lagrangian dual (LR) is a minimization problem, hence its relaxation (LRe) is an underestimation of (LR). Since the problem (LR($\boldsymbol{\lambda}$)) is a restricted version of (LR) (the variables $\boldsymbol{\lambda}$ are fixed to a value obtained from (LRe)), $z_k \leq \sigma_k$ must hold, where z_k and σ_k are the objective values to (LRe) and (LR($\boldsymbol{\lambda}$)) respectively. Actually if $z_k \geq \sigma_k$ then the optimal dual values $\boldsymbol{\lambda}$ have been found, otherwise $z_k < \sigma_k$, yielding the cut (4.16).

$$z \geq \mathbf{c}^T\mathbf{x}_k + \mathbf{f}^T\mathbf{u}_k + \boldsymbol{\lambda}(\mathbf{b} - \mathbf{A}\mathbf{x}_k - \mathbf{B}\mathbf{u}_k) \tag{4.16}$$

There are three main drawbacks of the cutting plane method. First, the problem (LRe) must be solved at every iteration to obtain new values of multipliers $\boldsymbol{\lambda}$. The number of variables of this problem is equal to the number of

constraints $\mathbf{Ax}+\mathbf{Bu} \leq \mathbf{b}$. The number of constraints of (LRe) is the number of currently known vertices of the polyhedron S, which corresponds to the count of iterations so far. Although the problem is not huge (at least in the first iterations), it is still much more computation than in the case of the sub-gradient method. However, according to our experience, solving (LRe) takes no more than a fraction of the time needed to compute (LR($\boldsymbol{\lambda}$)), hence this issue is not prohibitive.

Second, the cutting plain method is known to suffer from instability. According to [DA12], it may make large steps away from the optimum even (or in fact especially) when the current solution is close.

Besides, (LRe) may be unbounded. In this case an artificial constraints have to be added, bounding the problem. However, until (LRe) becomes bounded, it does not yield a good bound (the value of bound depends on the bounding constraint).

These drawbacks are overcome by the bundle method which introduces a regularization term into the objective value of (LRe).

Bundle method. The bundle method is based on the cutting plains method (see [Lem01]). To avoid the instability, a regularizing quadratic term is introduced, i.e. the optimization problem (RLRe) is solved, instead of (LRe). The purpose of the regularization term is to prevent the algorithm from making large steps in $\boldsymbol{\lambda}$. Please note that the problem (RLRe) is always bounded.

$$\begin{gathered} \min_{z,\boldsymbol{\lambda}} z + \frac{1}{2\tau}\left\|\boldsymbol{\lambda}-\hat{\boldsymbol{\lambda}}\right\|^2 \\ z \geq \mathbf{c}^T\mathbf{x}_i + \mathbf{f}^T\mathbf{u}_i + \boldsymbol{\lambda}^T(\mathbf{b}-\mathbf{A}\mathbf{x}_i-\mathbf{B}\mathbf{u}_i),\ i \in P_k \subset S \\ \boldsymbol{\lambda} \geq 0 \end{gathered} \tag{RLRe}$$

In (RLRe) the vector $\hat{\boldsymbol{\lambda}}$ is so called stability centre, which corresponds to the currently best known vector $\boldsymbol{\lambda}$, i.e. the last vector $\boldsymbol{\lambda}_k$ which provided a solution to (LR($\boldsymbol{\lambda}$)) with a smaller objective value $J_D(\boldsymbol{\lambda}_k)$ comparing to the best known $\hat{J_D}(\boldsymbol{\lambda})$ in the iteration k. The iteration in which $J_D(\boldsymbol{\lambda}_k) < \hat{J_D}(\hat{\boldsymbol{\lambda}}) - \epsilon$ for $\epsilon \geq 0$ is usually called a descent step. The current $\hat{J_D}(\hat{\boldsymbol{\lambda}})$ is then updated to the value $J_D(\boldsymbol{\lambda}_k)$ and $\hat{\boldsymbol{\lambda}}$ is updated with $\boldsymbol{\lambda}_k$. If $J_D(\boldsymbol{\lambda}_k) \geq \hat{J_D}(\hat{\boldsymbol{\lambda}}) - \epsilon$ then the iteration is called a null step in which the stability centre remains the same and the estimation (RLRe) is improved using $(\mathbf{x}_k, \mathbf{u}_k)$.

The parameter τ has a considerable impact on the convergence of the bundle method. The value of parameter τ can be derived by a line search, which would require to compute many trial problems (LR($\boldsymbol{\lambda}$)), which is not feasible in our

case. We have the best experience with formula (4.17) from [MA07], where $\tau_{k+1} = \frac{1}{\psi}$.

$$\psi = -\frac{2}{\tau_k}\left[\frac{J_D(\boldsymbol{\lambda}_k) - \hat{J}_D(\hat{\boldsymbol{\lambda}})}{z_k - \hat{J}_D(\hat{\boldsymbol{\lambda}})} - 1\right] \tag{4.17}$$

The main drawback of the bundle method is that in some cases many null steps have to be taken before the estimation (RLRe) is accurate enough for producing a descent step. According to our experience, it is usually very problem-specific whether the bundle method converges quickly or slowly. However, comparing to the sub-gradient method, the bundle method performs better in every case.

Includes a graph comparing a typical convergence of the method with the sub-gradient method

Solving sub-MILP at nodes. As branching proceeds the number of binary variables in the descendant nodes decreases. When tightening of bounds of variables takes place, it decreases quickly. Hence, when this number drops to a a defined value (say several-times the number of binaries in a sub-problem), we propose to compute the remaining (sub-MILP) instead of its linear and Lagrangian relaxations, hoping that the it would yield a good bound faster than further branching.

$$\begin{aligned} &\max_{\mathbf{x},\mathbf{u}} \mathbf{c}^T\mathbf{x} + \mathbf{f}^T\mathbf{u} \\ &\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b} \\ &\mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \leq \mathbf{d} \\ &\mathbf{u} \in \{0,1\} \\ &u_i = v_i,\ \forall i \in I \end{aligned} \tag{sub-MILP}$$

In (sub-MILP) the set I contains all the indices of the variables which were fixed due to the branching (including the variables that were fixed by the tightening). The values v_i are then the values to which the variables u_i were fixed.

Another merit of solving (sub-MILP) can be finding a new feasible solution (hopefully incumbent).

Branching variables selection

We utilize our knowledge of the problem structure (more specifically the knowledge of the typical types of constraints) to propose an efficient strategy for branching variables selection. Many of the coupling constraints $\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \leq \mathbf{b}$ are so-called clique inequalities (4.18). Note that $u_k \in \{0,1\}\, \forall k \in I \cap J$.

$$\sum_{i \in I} u_i - \sum_{j \in J} u_j \leq 1 - |J| \tag{4.18}$$

The main property of clique inequalities is that after fixing one variable u_i to 1 or u_j to 0, the constraint yields values for the remaining variables $u_{k \in I} = 0$ and $u_{s \in J} = 1$ for $k \neq i$ and $s \neq j$. This can be viewed as fixing the whole clique. It should be noted that fixing variables u_i to 0 or u_j to 1 does not result in fixing the clique.

This property can be used within a branching scheme. The main idea is that the more variables are fixed at branching the tighter the bound. Hence we propose the following branching variable selection rule.

Let C_p and C_n be the collections of cliques which will be fixed after fixing a variable u_k to 1 and 0 respectively. The number of variables N_k^p constrained by the cliques C_p can provide an estimate on how many binary variables will be fixed after fixing the variable u_k to 1. A number N_k^n can also be obtained for fixing u_k to 0. The variable u_k is selected such that u_k maximizes (4.19).

$$\max_{\forall k} \left(w_k \cdot \min \left\{ N_k^p, N_k^n \right\} \right) \tag{4.19}$$

The weights w_k are computed so that the binaries u_k which are far from integrality in the solution to (LPR) were preferred. Hence $w_k = 0.5 - |\hat{u}_k - 0.5|$, where $\hat{u}_k$ is the value of u_k in the current solution to (LPR).

Branching more than one variable at once. To utilize the strength of parallelization it is convenient to branch according to several variables at once, so that the relaxations can be computed in parallel. To fully utilize the computation capabilities of the used computer, the number of branching variables should be equal to the number of processors (or a little bit higher as some of the relaxations may be solved faster than other). Hence the usual number of branching variables in the case of eight-core processor is 3.

Tightening at nodes

The procedure of bounds tightening was already described in the section 4.4.1. Here we add an explanation, why is the tightening at branching important. The capabilities of current commercial MILP solvers in the field of problem presolving are amazing. We do not aim at beating the solvers in this area. However, within the customized B&B algorithm we never solve the whole MILP (except for (sub-MILP) problems). We solve relaxations, i.e. (LPR) and (LR($\boldsymbol{\lambda}$)). The solver never knows the original MILP problem and therefore miss valuable

information (such as the integrality of variables $\mathbf{u}$ in the case of (LPR)) that would facilitate the tightening.

When creating relaxations the problem should already be presolved, i.e. the bounds of the variables tightened. If we wanted to use the solver presolve capabilities, the solver would have to provide information on how the model was presolved, not only the presolved model itself, as some information on the presolve could not be recovered on the basis of the presolved model only.

For example, the presolve procedure usually remove many variables either by fixing them to a value or by aggregation $x_i = \sum_{j \in J} a_j x_j$. Generally the information on fixations and aggregations cannot be recovered from the presolved model. However, the solvers usually do not provide these information and solving the presolved model instead of the original one is out of question, because the aggregations usually destroy the convenient sub-structure of the problem - most importantly the portion of the coupling constraints is much higher than before and the sub-problems are thus much less self-contained.

There exists a solver allowing to retrieve the presolve information. It is the free solver SCIP [Ach09]. The presolve is almost as efficient as in the case of the commercial solver Gurobi, however, it is still quite time consuming. It is therefore not practical to use this presolve at every node. For this purpose we implemented our own presolver, that does not remove so many variables as SCIP would, but it requires less time than the SCIP presolver does.

The fast tightening using our presolver is performed at every node and also when a new lower or upper bound is available. Knowing bounds J^{LB} and J^{UB} the constraints (4.20) can be formulated.

$$J^{LB} \leq \mathbf{c}^T \mathbf{x} + \mathbf{f}^T \mathbf{u} \leq J^{UB} \tag{4.20}$$

Tightening can be performed based on these constraints, possibly improving the bounds of variables.

Customized B&B algorithm

The proposed custom B&B algorithm can be summarized as follows.

0. Initialize the algorithm.
   a. Create a set of nodes $N \in \{\text{root}\}$ and the set of known solutions $S \in \{\emptyset\}$.
   b. Presolve the root node.
   c. Compute full (LPR) and define the best known multipliers $\hat{\boldsymbol{\lambda}}$ with the value set to the optimal dual values to (LPR).
   d. If the root is infeasible, end the algorithm.
1. Fetch new feasible solutions from EHC, which is running in parallel.
2. Remove the nodes from N for which upper bound (UB) is less than the current lower bound (LB).
3. Select the node with the highest UB from N and remove it from N.
4. If the gap between UB of the selected node and the current LB is less than the defined target gap (e.g. 1%), end the algorithm.
5. Update UB of the selected node.
   a. If the number of binary variables is less than a defined sub-MILP boundary, compute (sub-MILP).
      (i) If the (sub-MILP) yields a feasible solution, add it to S.
   b. Compute a defined number of iterations of (LR) or until it converges.
      (i) Start the algorithm with the multipliers $\hat{\boldsymbol{\lambda}}$.
      (ii) If the problem (LR) yields a feasible solution, add it to S.
      (iii) If better multipliers $\bar{\boldsymbol{\lambda}}$ than $\hat{\boldsymbol{\lambda}}$ were found, set $\hat{\boldsymbol{\lambda}} = \bar{\boldsymbol{\lambda}}$.
6. If infeasibility is detected continue with the step 1.
7. Select k branching variables.
8. Create 2^k child nodes and add them to N.
9. For each node do:
   a. Fix the branching variables to their values.

b. Tighten bounds of variables.
c. Compute preliminary bound by solving (LPR($\boldsymbol{\lambda}$)) using $\hat{\boldsymbol{\lambda}}$.
d. If child is infeasible, remove it from N.
e. If the preliminary bound is worse than LB, remove the node from N.

10. Continue with step 1.

4.6 Remarks on implementation

A prototype implementation of the proposed algorithm was created. The main properties of the implementation are:

- The algorithm is implemented in Matlab (version 2015b).
- It uses the SCIP presolver (version 3.2.1) written in C++ through a .NET wrapper.
- The commercial solver Gurobi (version 6.5) is used for solving the relaxations.
- The implementation employs parallelization when branching, computing (LR) and computing (LPR($\boldsymbol{\lambda}$)).
- The algorithm is scalable with respect to the problem size, i.e. the size of a solvable problem is given by the number of processors and the memory available.

Chapter 5

Test cases

In this chapter the efficiency of the proposed algorithm (the customized B&B algorithm in conjunction with EHC) is evaluated by comparing it to the plain usage of the state-of-the-art general purpose solver Gurobi.

The algorithm will be tested on three sets of scenarios, each set corresponding to one real-world CHP plant. These scenarios reflect real planning tasks solved by the CHP plants. The plants are denoted as CHP A, CHP B and CHP C for the purpose of this thesis. We can not include the real names of the plants for confidentiality reasons.

The three CHP plants differ in parameters, such as the nominal output or the number of components. Also, each plant has a different power product portfolio. These properties of the plants affect the size of the resulting optimization problems, i.e the number of variables and constraints.

The main parameters of the three CHP plants are summarized in Table 5.1. Beside the technical parameters the table contains also the information on the number of variables and constraints per 1 hour of the planning horizon. These numbers illustrate the complexity of the planning tasks - the largest problems have around 1 million variables (from which more than 100 thousands are binary variables).

	CHP A	CHP B	CHP C
Heat power output [MW]	50	200	300
Electricity power output [MW]	40	220	120
Condensing steam turbines nr.	1	2	2
Back-pressure steam turbines nr.	1	0	0
Boilers nr.	2	6	6
Heat exchangers nr.	2	7	7
Binary variables per 1h nr.	75	99	124
Continuous variables per 1h nr.	454	277	1102
Constraints per 1h nr.	1360	1183	3537

Table 5.1: Properties of the plants considered in test cases.

Each set of scenarios contains several instances of the problem differing in horizon lengths and in the input data, such as the requirement for heat production or the prices of fuel and electricity. For example, the test cases with the horizon length of one month, i.e. 672, 720 or 744 hours, are based on the input data from different months of a year. Definitions of the scenario sets are in Table. 5.2. Finally, we would like to emphasize that the data used for creating the scenarios are real data of the years 2011-2013.

Scenario sets			
Set	Planning horizon length	Number of hours	Number of test cases in the set
CHP A	weekly	168	10
	monthly	672 - 744	10
	quarterly	2160 - 2208	4
CHP B	weekly	168	10
	monthly	672 - 744	10
CHP C	weekly	168	10
	monthly	672 - 744	10

Table 5.2: Definitions of scenario sets.

The tests were performed on a computer with the following hardware configuration:

- Processor Intel(R) Xeon(R) CPU E5-1650 @ 3.20GHz, 6 Cores, 12 Logical Processors,
- Installed Physical Memory (RAM) 32,0 GB.

5.1 Test case results

In this section results of the test cases will be presented and discussed. The proposed algorithm is compared against plain usage of Gurobi, which according to the benchmarks [Ben] is the most efficient MILP solver. By the proposed algorithm we mean a combination of customized B&B algorithm and EHC, working in cooperation in parallel. This cooperation was described and explained in Chapter 4. Gurobi was also used for relaxations and sub-problems within the proposed algorithm.

Both algorithms (the proposed algorithm and plain Gurobi) were given a task to find a solution with the gap 1%, i.e. a solution with the certificate that it is not more than 1% worse than the optimum. Also, algorithms should reach the solution within two hours.

The algorithms will be compared on the basis of three figures:

- Runtime required to reach the 1% gap. The runtime is limited by the value of 7200s (2 hours).
- Reached optimality gaps, i.e. worst-case estimation on the distance of the best found solution to the optimum.
- Objective values of the best found solution.

The last figure is included because the gap is only a worst case estimate. Hence a solution with the gap of exactly 1% provided by the customized B&B may actually be better than another solution provided by Gurobi with gap 0.1% (and vice versa).

Test case results for CHP A, weekly planning problems (168h)						
	Objective value [-]		Runtime [s]		Reached gap [%]	
Test case	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm
1	421 332	↑ 430 911	44	→ 54	0,83%	↓0,02%
2	438 935	↑ 447 046	7	→ 61	0,87%	↓0,01%
3	481 496	↑ 492 230	8	→ 62	0,92%	↓0,00%
4	407 805	↑ 414 726	21	→ 52	0,81%	↓0,08%
5	431 092	↑ 440 328	9	→ 75	0,93%	↓0,03%
6	416 829	↑ 428 362	14	→ 59	0,98%	↓0,05%
7	425 530	↑ 436 694	9	→ 47	1,00%	↓0,02%
8	461 826	↑ 473 095	7	→ 41	0,95%	↓0,01%
9	455 414	↑ 466 132	11	→ 46	0,82%	↓0,02%
10	459 999	↑ 465 979	10	→ 60	0,70%	↓0,03%

↑↓ Green arrows denote the testcases in which for a particular figure the proposed algorithm performed **better than plain Gurobi**.

→ Yellow arrows denote the testcases in which for a particular figure the proposed algorithm performed **about equal as plain Gurobi**.

Table 5.3: Test case results of weekly planning for CHP A.

5.1.1 Test case results for CHP A

Tables 5.3, 5.4 and 5.5 contain results for the plant CHP A. Considering the parameters of plants in Table 5.1, the optimization problems associated with plant CHP A should be easier than the task of the other two plants. This was confirmed with the results. Even quarterly planning problems could be computed within a reasonable time.

In the tables, the arrows indicate whether the proposed algorithm performed better (a green arrow), worse (a red arrow) or similarly (a yellow arrow), comparing to plain Gurobi. In the case of the objective value the higher value the better (hence an arrow up is green and an arrow down is red). For the other figures - runtime and gap - lower values are the better (hence the arrow down is green).

A yellow horizontal arrow denotes a test case for which the corresponding figure has about the same value for both the algorithms. The objective values are considered about the same if their value does not differ in more than 0.1%.

Similarly the gaps with the difference of 0.1% and less are considered about the same. Finally, two runtimes differing in less than 5 minutes are denoted as about the same.

We can see that the proposed algorithm performed better in all the test cases with monthly (Table 5.4) and quarterly (Table 5.5) planning problems. The average speed-up comparing to Gurobi was 45% and 59% respectively. I.e. the proposed algorithm required about a half the time to find a solution with gap 1%.

Apart from runtimes, the proposed algorithm outperformed the plain Gurobi in the terms of reached gaps and objective values. This was especially true in the case of quarterly planning, in which the proposed algorithm provided significantly better solutions (and gaps) in much shorter time, than Gurobi did.

Test case results for CHP A, monthly planning problems (672h-744h)

Test case	Objective value [-]		Runtime [s]		Reached gap [%]	
	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm
1	2 340 038	↓ 2 330 728	529	↓ 209	0,57%	↓ 0,11%
2	2 365 659	↑ 2 373 692	404	→ 263	0,59%	↓ 0,08%
3	2 384 427	→ 2 385 550	448	→ 313	0,58%	↓ 0,14%
4	2 395 089	↑ 2 401 392	455	→ 228	0,58%	↓ 0,08%
5	2 348 486	↑ 2 399 710	524	→ 270	0,93%	↓ 0,13%
6	2 367 917	↑ 2 418 628	466	→ 264	0,93%	↓ 0,15%
7	2 441 342	→ 2 439 379	442	→ 277	0,55%	↓ 0,13%
8	2 410 763	↑ 2 417 831	594	↓ 236	0,60%	↓ 0,08%
9	2 392 232	↑ 2 405 074	490	→ 277	0,59%	↓ 0,03%
10	2 353 480	↑ 2 408 051	496	→ 307	0,96%	↓ 0,12%

↑↓ Green arrows denote the testcases in which for a particular figure the proposed algorithm performed **better than plain Gurobi**.

→ Yellow arrows denote the testcases in which for a particular figure the proposed algorithm performed **about equal as plain Gurobi**.

↓ Red arrows denote the testcases in which for a particular figure the proposed algorithm performed **worse than plain Gurobi**.

Table 5.4: Test case results of monthly planning for CHP A.

Gurobi outperforms the proposed algorithm in the case of weekly planning problems (Table 5.3). The reason is that these problems are easily solvable by plain Gurobi. On the other hand the proposed algorithm has a noticeable

overhead which is not proportional to the problem size. Hence, it is better suited for larger problem instances. This is actually what the proposed algorithm is designed for - to enable solving large problem instances.

Test case results for CHP A, quarterly planning problems (2160h-2184h)						
	Objective value [-]		Runtime [s]		Reached gap [%]	
Test case	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm
1	5 082 002	➡ 5 085 562	7200	⬇ 1324	1,03%	⬇ 0,32%
2	5 211 059	⬆ 5 275 536	2347	⬇ 931	0,98%	⬇ 0,01%
3	5 175 574	⬆ 5 275 427	2174	⬇ 1313	0,98%	⬇ 0,18%
4	5 097 858	⬆ 5 143 637	2729	⬇ 1189	0,93%	⬇ 0,32%

⬆⬇ Green arrows denote the testcases in which for a particular figure the proposed algorithm performed **better than plain Gurobi**.

➡ Yellow arrows denote the testcases in which for a particular figure the proposed algorithm performed **about equal as plain Gurobi**.

Table 5.5: Test case results of quarterly planning for CHP A.

Test case results for CHP B, weekly planning problems (168h)						
	Objective value [-]		Runtime [s]		Reached gap [%]	
Test case	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm
1	89 154	→ 89 160	128	→ 104	0,39%	↑ 0,50%
2	137 435	→ 137 421	492	↓ 143	0,39%	↓ 0,22%
3	147 485	→ 147 463	506	↓ 164	0,82%	↓ 0,46%
4	128 730	→ 128 732	1044	↓ 130	0,81%	↓ 0,54%
5	117 271	→ 117 196	199	→ 148	0,41%	↑ 0,58%
6	133 696	→ 133 768	491	↓ 141	0,37%	↑ 0,57%
7	125 457	→ 125 487	338	→ 196	0,39%	↑ 0,59%
8	119 349	→ 119 361	641	↓ 192	0,99%	↓ 0,61%
9	102 547	↓ 102 438	513	↓ 197	0,55%	↓ 0,31%
10	68 812	↓ 68 738	818	↓ 261	0,89%	→ 0,93%

↑↓ Green arrows denote the testcases in which for a particular figure the proposed algorithm performed **better than plain Gurobi**.

→ Yellow arrows denote the testcases in which for a particular figure the proposed algorithm performed **about equal as plain Gurobi**.

↑↓ Red arrows denote the testcases in which for a particular figure the proposed algorithm performed **worse than plain Gurobi**.

Table 5.6: Test case results of weekly planning for CHP B.

5.1.2 Test case results for CHP B

The tables 5.6 and 5.7 show the results for weekly and monthly planning problems of CHP B. Considering the weekly planning problems the average speed-up is 58%. There are several red arrows in Table 5.6, denoting that in some cases Gurobi found a better solution or provided a better gap. However, it should be noted, that the proposed algorithm always satisfied the requirements of 1% gap and maximal runtime of 2 hours. And it always provided solutions within the required gap faster than Gurobi.

The performance in the case of monthly problems is worse - 21% slow-down on average. However, only once Gurobi outperformed the proposed algorithm by more than 5 minutes of runtime. In the most cases both algorithms performed about the same, even in terms of objective values and gaps.

The figures in Table 5.7 divide the test cases into two groups. The first group, months 4 - 8, can be seen as easy problems as all these were solved within 12 minutes by either algorithms. This can be explained by the fact, that the CHP

operations planning in the months with lower heat production is typically easier than in the other months. The reason for this phenomenon arise from the fact that in these months the technology operates with a considerable slack - the plant is dimensioned for a higher power output. In these test cases the solution time of the proposed algorithm was actually never worse by more than 6 minutes, comparing to Gurobi.

The second group - the months 1, 2, 3, 9 and 10 - may be declared as hard problems. For these months the proposed algorithm never performed worse than Gurobi, i.e. it has always found solutions within 1% gap in better or about the same time.

Test case results for CHP B, monthly planning problems (672h-744h)

Test case	Objective value [-]		Runtime [s]		Reached gap [%]	
	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm
1	568 566	→ 568 714	2001	↓ 562	0,38%	↓ 0,20%
2	518 244	↓ 502 312	4813	→ 4576	0,42%	→ 0,37%
3	339 907	↑ 343 340	7200	→ 7200	7,25%	↓ 0,00%
4	214 520	→ 214 520	554	→ 599	0,00%	→ 0,00%
5	214 573	→ 214 573	283	→ 532	0,00%	↑ 0,49%
6	214 651	→ 214 651	369	→ 627	0,00%	↑ 0,40%
7	214 625	→ 214 625	474	→ 613	0,00%	→ 0,00%
8	214 642	→ 214 642	409	↑ 720	0,00%	↑ 0,74%
9	193 165	↑ 214 628	7200	→ 7200	30,25%	↓ 12,00%
10	479 160	↓ 473 387	7108	→ 7200	0,85%	↑ 1,00%

↑↓ Green arrows denote the testcases in which for a particular figure the proposed algorithm performed **better than plain Gurobi**.

→ Yellow arrows denote the testcases in which for a particular figure the proposed algorithm performed **about equal as plain Gurobi**.

↑↓ Red arrows denote the testcases in which for a particular figure the proposed algorithm performed **worse than plain Gurobi**.

Table 5.7: Test case results of monthly planning for CHP B.

Test case results for CHP C, weekly planning problems (168h)						
	Objective value [-]		Runtime [s]		Reached gap [%]	
Test case	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm
1	988 723	→ 988 750	2241	→ 1972	0,73%	↓ 0,55%
2	907 239	→ 907 225	4652	→ 4720	0,56%	↑ 0,73%
3	889 013	↑ 907 156	7200	→ 7200	2,96%	↓ 1,09%
4	942 884	↑ 945 721	7200	↓ 1830	1,29%	↓ 1,00%
5	976 403	↑ 1 006 601	7200	→ 7200	4,38%	↓ 1,40%
6	1 007 041	↓ 1 005 137	1804	↓ 282	0,99%	→ 1,00%
7	922 596	↑ 936 646	7200	↓ 5145	1,93%	↓ 0,04%
8	912 782	↑ 922 002	7200	↓ 225	1,64%	↓ 0,68%
9	999 746	↑ 1 004 146	2576	→ 2440	0,89%	↓ 0,46%
10	896 642	↑ 905 699	7200	↓ 2591	1,32%	↓ 0,69%

↑↓ Green arrows denote the testcases in which for a particular figure the proposed algorithm performed **better than plain Gurobi**.

→ Yellow arrows denote the testcases in which for a particular figure the proposed algorithm performed **about equal as plain Gurobi**.

↑↓ Red arrows denote the testcases in which for a particular figure the proposed algorithm performed **worse than plain Gurobi**.

Table 5.8: Test case results of weekly planning for CHP C.

5.1.3 Test case results for CHP C

Finally, the tables 5.8 and 5.9 present the results for CHP C. Both weekly and monthly problems were hard to solve by either of algorithms.

For weekly planning problems, Gurobi was able to finish within 2 hours only in 4 cases, while the proposed algorithm in 8 test cases. Both algorithms were able to find a feasible solution within two hours, while the solution quality was always considerably better in the case of the proposed algorithm.

Gurobi was unable to find any solution within the time-limit for any of the monthly planning problems (Table 5.9). On the contrary, the proposed algorithm was able to find a solution every time and in many cases even a very good one. Actually, such huge planning problems like monthly CHP C planning represent our motivation for proposing an algorithm based on the knowledge of the problem structure. The test case results in Table 5.9 show that the proposed algorithm can tackle even these huge planning problems.

Test case results for CHP C, monthly planning problems (672h-744h)						
	Objective value [-]		Runtime [s]		Reached gap [%]	
Test case	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm	Gurobi	Proposed algorithm
1	-	↑ 4 110 021	7200	→ 7200	-	↓ 1,12%
2	-	↑ 4 114 622	7200	↓ 4766	-	↓ 0,89%
3	-	↑ 3 151 892	7200	→ 7200	-	↓ 3,82%
4	-	↑ 926 544	7200	→ 7200	-	↓ 6,20%
5	-	↑ 637 336	7200	→ 7200	-	↓ 4,00%
6	-	↑ 639 232	7200	→ 7200	-	↓ 3,20%
7	-	↑ 642 789	7200	→ 7200	-	↓ 3,02%
8	-	↑ 616 840	7200	↓ 4881	-	↓ 0,93%
9	-	↑ 906 396	7200	→ 7200	-	↓ 16,70%
10	-	↑ 2 171 410	7200	→ 7200	-	↓ 6,42%

↑↓ Green arrows denote the testcases in which for a particular figure the proposed algorithm performed **better than plain Gurobi**.

→ Yellow arrows denote the testcases in which for a particular figure the proposed algorithm performed **about equal as plain Gurobi**.

Table 5.9: Test case results of monthly planning for CHP C.

5.1.4 Test case result summary

The summary of test results is in table 5.10. The planning problems can be divided into two groups. The first group contains problems that can be viewed as easy as these are always solved within 20 minutes by either of algorithms. The other problems can be denoted as hard.

The proposed algorithm outperforms or equals Gurobi with respect to runtime in 63 of 64 both easy and hard problems. In the case of hard problems, the proposed algorithm outperforms or equals Gurobi in all the 29 hard test cases with the average improvement in runtime of 45%. Table 5.10 also shows that the proposed algorithm tend to offer better solutions for both problem classes with respect to objective values as well as gaps.

The benefit of the proposed algorithm is most evident in the case of the hardest scenario set - monthly planning problems of CHP C. Although, in many cases the proposed algorithm did not find a 1% gap solution within the time-limit of two hours, it was always able to find a very good solution. On the contrary, Gurobi failed to find any solution in any of these test cases.

	Easy problems	**Hard problems**
Count	35	**29**
Mean improvement in objective value [-]	7 658	**16 637**
Mean improvement in objective value [%]	0,81%	**1,18%**
Occurances of better or about equal objective [-]	32	**26**
Occurances of worse objective [-]	3	**3**
Mean improvement in gap [%][1]	0,47%	**1,61%**
Mean runtime improvement [min]	2	**36**
Occurances of shorter or about equal runtime [-]	34	**29**
Occurances of worse runtime [-]	1	**0**

[1] It is the mean value of differences between gaps provided by Gurobi and gaps provided by the proposed algorithm.

Table 5.10: Test results summary.

Considering the easy test cases, Gurobi outperforms the proposed algorithm in the runtime, however, the runtimes never differ in more than 6 minutes, which can, from a user's perspective, be seen as negligible. And we should again emphasize that the propose algorithm is intended for tackling the hard problems.

Chapter 6

Conclusions

In this thesis a framework for modelling CHP plants and optimization problem for planning their operations is proposed together with an efficient solving algorithm. The main requirements for the framework as defined in Section 1.2 are

1. usability of the framework for various plants,
2. knowledge of the gap, i.e the estimation on proximity to the optimal solution,
3. capability of finding good solutions (within 1% gap) of long-term planning problems in reasonable time (under two hours).

Considering the first requirement, we proposed a general modelling framework, which was successfully used for three different plants. We believe that the framework is general enough to cover all the possible peculiarities of different CHP plants. The MILP formulation allows to model even non-linear terms and therefore non-linear behaviour of plant's thermodynamic cycle components can easily be approximated by piece-wise linear functions, when necessary.

We developed a solution technique combining a heuristic approach for finding feasible solutions (EHC) with a customized B&B providing estimation on proximity of a solution to the optimal one. The knowledge of the gap was the second requirement. We performed 64 tests using real-world data of three existing CHP plants and we compared the proposed algorithm to plain Gurobi usage. Both approaches were set up with the time-limit of 2 hours and the gap requirement of 1%. In most cases the proposed algorithm finds the certificate of near-optimality sooner than plain Gurobi does.

The third requirement stated that the solution algorithm must be able to provide a solution within 1% gap in reasonable time. The reasonable time was

quantified as 2 hours. The proposed algorithm was able to reach this requirement in most cases - in 51 of 64 cases. After 2 hours, the gap was still higher than 1% in 13 cases. The reason is that in 7 of the 13 cases the problem was simply too large and only several B&B nodes could be evaluated within the given time. In the remaining 6 cases neither the EHC heuristics nor the customized B&B was able to find a solution within the gap of 1%.

Plain Gurobi approach failed to provide a 1% gap solution within the time limit in 18 cases. More importantly, in 10 cases, Gurobi was not able to find any solution at all within the time limit, while the proposed algorithm always found a solution and usually a very good one (around 5% gap on average). This ability of finding good feasible solutions for problems, which are too hard for Gurobi, can also be viewed as the main merit of the proposed algorithm.

Besides, the proposed algorithm provides better solutions on average and in the case of hard problems it is also faster than Gurobi.

6.1 Contributions of the thesis

This section summarizes the main contributions of this thesis:

1. A comprehensive modelling framework aimed at CHP operations planning was introduced. The framework is general enough to cover the needs of various CHP plants.

2. The framework also supports modelling the trade on power markets.

3. A methodology describing how to formulate an optimization problem for the CHP production and trade planning was provided.

4. A heuristic algorithm able to find very good solutions of the optimization problem quickly was developed.

5. A customized B&B algorithm exploiting our knowledge of the convenient block-diagonal problem structure was provided in order to to obtain certificates of (near-)optimality.

6. The proposed algorithm is able to solve planning problems of dimensions which are not tractable for currently used approaches.

7. The modelling framework, EHC and the customized B&B algorithm were tested on a real-world, real-data case study consisting of 64 test cases. The tests confirmed the efficiency of the algorithms.

6.2 Future work

Finally we include a list of possible enhancements of the algorithm which could increase its efficiency.

- Ability to update the solutions of relaxations (LPR) and (LR($\boldsymbol{\lambda}$)) (such as state-of-the-art solvers do) after branching instead of recomputing it from scratch could considerably reduce computation times.
- Other heuristics providing feasible solutions should be incorporated.
- A heuristics able to project an infeasible solution obtained from (LR($\boldsymbol{\lambda}$)) onto the feasible solution space, could represent an efficient source of good feasible solutions (if the relaxation is tight).

Appendix A

Glossary

B&B branch and bound. vi, 7, 8, 11–13, 27–29, 34, 36, 39, 40, 42, 45, 47, 49–51, 57, 58

BHC basic horizon cutting. 29, 30, 33

CHP combined heat and power. iv, vi, vii, 1–3, 5–12, 15, 23–25, 29, 37–39, 49, 53, 57, 58

EHC enhanced horizon cutting. vii, 28–32, 35–39, 47, 49, 50, 57, 58

LB lower bound. 47

LPR linear programming relaxation. 12

LR Lagrangian relaxation. 8, 9, 12, 40

MBLP mixed-binary linear programming. 27

MILP mixed-integer linear programming. 6, 7, 9, 10, 12, 15–17, 27, 34, 38, 39, 45, 50, 57

PRCS pressure reduction and cooling station. 20

PRS pressure reduction station. 20

PWL piece-wise linear. 12, 17–19

SOS2 special ordered set of type 2. 17

UB upper bound. 47

Appendix B

Nomenclature

η^c	Mechanical efficiency of component c	[-]
C^c_{SD}	Cost per shutdown of component c	[€]
C^c_{SU}	Cost per start-up of component c	[€]
C_W	Specific heat capacity of water	[Jkg^{-1}K^{-1}]
C_{ALW}	CO_2 allowances cost per unit of fuel	[€/MWh]
C_{DEVN}	Cost per unit of negative deviation from contracted power output	[€/MW]
C_{DEVP}	Cost per unit of positive deviation from contracted power output	[€/MW]
C_F	Cost per unit of fuel	[€/MWh]
$d_N(t)$	Shortage of power generation compared to the contracted value in time sample t	[MW]
$d_P(t)$	Surplus of power generation compared to the contracted value in time sample t	[MW]
H^c_p	Constant enthalpy of steam flow $m^c_p(t)$	[MWh/t]
$h^c_p(t)$	Variable enthalpy of steam flow $m^c_p(t)$	[MWh/t]
J	Objective value - overall profit during optimization horizon	[€]
$m^c_p(t)$	Mass flow rate through inlet or exit pipe p of component c in time sample t	[t/h]

$M^c_{p,k}$ Characteristic value of $m^c_p(t)$ defining a linear segment of a PWL function[t/h]

$M^c_{p,MAX}$ Upper bound for $m^c_p(t)$ [t/h]

$M^c_{p,MIN}$ Lower bound for $m^c_p(t)$ [t/h]

$p^c_{OUT}(t)$ Electrical power output from conversion of heat rate $q^c_{OUT}(t)$ [MW]

$p^{PRODi}(t)$ Delivered power in the form of power product $PRODi$ in time sample t [MW]

P^{PRODi}_{MAX} Upper bound for delivered power in the form of power product $PRODi$ in time sample t [MW]

P^{PRODi}_{MIN} Lower bound for delivered power in the form of power product $PRODi$ in time sample t [MW]

p^{PRODi}_{VOL} Contracted volume of power product $PRODi$ [MW]

$q^c_p(t)$ Energy transfer rate accompanying mass flow $m^c_p(t)$ [MW]

$Q^c_{IN,k}$ Characteristic value of $q^c_{IN}(t)$ defining a linear segment of a PWL function [MW]

$Q^c_{IN,MAX}$ Upper bound on heat transfer rate into a component not accompanying mass flow [MW]

$Q^c_{IN,MIN}$ Lower bound on heat transfer rate into a component not accompanying mass flow [MW]

$q^c_{IN}(t)$ Rate at which energy is being transferred in the component c not accompanying mass flow [MW]

$Q^c_{OUT,k}$ Characteristic value of $q^c_{OUT}(t)$ defining a linear segment of a PWL function [MW]

$q^c_{OUT}(t)$ Rate at which energy is being transferred out of the component c not accompanying mass flow [MW]

$Q^c_{p,k}$ Characteristic value of $q^c_p(t)$ defining a linear segment of a PWL function[MW]

$q^b_F(t)$ Fuel consumption rate of boiler b in time sample t [MW]

$Q_{REQ}(t)$ Heat demand in time sample t [MW]

R_D^c	Ramp-down rate limit for power output of component c per time sample	[MW]
R_U^c	Ramp-up rate limit for power output of component c per time sample	[MW]
R_{SD}^c	Ramp-down rate limit for power output of component c from on state per time sample	[MW]
R_{SU}^c	Ramp-up rate limit for power output of component c from off state per time sample	[MW]
R_P^{PRODi}	Revenues per supplied unit of electrical energy in the form of product $PRODi$	[€/MWh]
R_Q	Revenues per unit of delivered heat	[€/MWh]
$s_D^c(t)$	Shutdown of component c in time sample t	[-]
$s_U^c(t)$	Start-up of component c in time sample$t$	[-]
T_D^c	Minimal number of consecutive time samples of being shutdown	[-]
T_U^c	Minimal number of time samples of continuous operation	[-]
T_C	Temperature of water returning from district heating network	[K]
T_H	Temperature of hot water which flows to district heating network	[K]
$u^c(t)$	On/off state of component c in time sample t	[-]
u^{PRODi}	Binary variable defining whether $PRODi$ is contracted	[-]

Appendix C

Bibliography

[Ach09] Tobias Achterberg. "SCIP: Solving constraint integer programs". In: *Mathematical Programming Computation* 1.1 (2009). http://mpc.zib.de/index.php/MPC/article/view/4, pp. 1–41.

[Agu+07] O. Aguilar et al. "Design and Optimization of Flexible Utility Systems Subject to Variable Conditions: Part 1: Modelling Framework". In: *Chemical Engineering Research and Design* 85.8 (2007), pp. 1136–1148. ISSN: 0263-8762. DOI: 05/cherd06062. URL: http://www.sciencedirect.com/science/article/pii/S0263876207731532.

[Ben] *Mixed Integer Linear Programming Benchmark (MIPLIB2010)*. http://plato.asu.edu/ftp/milpc.html. Accessed: 2016-08-16.

[Bis+14a] Aldo Bischi et al. "A detailed {MILP} optimization model for combined cooling, heat and power system operation planning". In: *Energy* 74 (2014), pp. 12 –26. ISSN: 0360-5442. DOI: http://dx.doi.org/10.1016/j.energy.2014.02.042. URL: http://www.sciencedirect.com/science/article/pii/S0360544214001765.

[CA06] M. Carrion and J. M Arroyo. "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem". In: *IEEE Transactions on Power Systems* 21.3 (Aug. 2006), pp. 1371–1378. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2006.876672.

[Cho+10] Soon-Young Choi et al. "Mathematical modeling and control of thermal plant in the district heating system of Korea". In: *Applied Thermal Engineering* 30.14-15 (2010), pp. 2067–2072. ISSN: 1359-4311. DOI: 16/j.applthermaleng.2010.05.014.

URL: http://www.sciencedirect.com/science/article/pii/S1359431110002103.

[CTG05a] Sung-Ling Chen, Ming-Tong Tsay, and Hong-Jey Gow. "Scheduling of cogeneration plants considering electricity wheeling using enhanced immune algorithm". In: *International Journal of Electrical Power; Energy Systems* 27.1 (2005), pp. 31 –38. ISSN: 0142-0615.

[CTG05b] Sung-Ling Chen, Ming-Tong Tsay, and Hong-Jey Gow. "Scheduling of cogeneration plants considering electricity wheeling using enhanced immune algorithm". In: *International Journal of Electrical Power & Energy Systems* 27.1 (2005), pp. 31 –38. ISSN: 0142-0615.

[DA12] Trinh-Minh-Tri Do and Thierry Artières. "Regularized Bundle Methods for Convex and Non-convex Risks". In: *J. Mach. Learn. Res.* 13.1 (Dec. 2012), pp. 3539–3583. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2503308.2503355.

[DH12] Michal Dvořák and Petr Havel. "Combined heat and power production planning under liberalized market conditions". In: *Applied Thermal Engineering* 43.0 (2012), pp. 163 –173. ISSN: 1359-4311. DOI: 10.1016/j.applthermaleng.2011.12.016. URL: http://www.sciencedirect.com/science/article/pii/S1359431111007137.

[EGU16] Laureano F. Escudero, María Araceli Garín, and Aitziber Unzueta. "Cluster Lagrangean decomposition in multistage stochastic optimization". In: *Computers & Operations Research* 67 (2016), pp. 48 –62. ISSN: 0305-0548. DOI: http://dx.doi.org/10.1016/j.cor.2015.09.005. URL: http://www.sciencedirect.com/science/article/pii/S0305054815002154.

[Fra05] A Frangioni. "About Lagrangian methods in integer optimization". English. In: *ANNALS OF OPERATIONS RESEARCH* 139.1 (2005), 163–193. ISSN: 0254-5330. DOI: {10.1007/s10479-005-3447-9}.

[HOO09] K.W. Hedman, R.P. O'Neill, and S.S. Oren. "Analyzing valid inequalities of the generation unit commitment problem". In: *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*. 2009, pp. 1–6. DOI: 10.1109/PSCE.2009.4840214.

[Kap+09] P. Kapustenko et al. “Control of Plate Heat Exchanger Outlet Temperature Using Butterfly Valve and Parametric Model Predictive Control Technique”. In: *Chemical Engineering Transactions* 18 (2009), pp. 827–832. DOI: 10.3303/CET0918135.

[KC12] Niels Hvidberg Kjeldsen and Marco Chiarandini. “Heuristic solutions to the long-term unit commitment problem with cogeneration plants”. In: *Computers & Operations Research* 39.2 (2012), pp. 269 –282. ISSN: 0305-0548. DOI: 10.1016/j.cor.2011.03.018.

[Lem01] Claude Lemaréchal. “Lagrangian Relaxation”. In: *Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions*. Ed. by Michael Jünger and Denis Naddef. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 112–156. ISBN: 978-3-540-45586-8. DOI: 10.1007/3-540-45586-8_4. URL: http://dx.doi.org/10.1007/3-540-45586-8_4.

[MA07] Alfredo J. Mezger and Katia C. de Almeida. “Short term hydrothermal scheduling with bilateral transactions via bundle method”. In: *International Journal of Electrical Power & Energy Systems* 29.5 (2007), pp. 387 –396. ISSN: 0142-0615. DOI: http://dx.doi.org/10.1016/j.ijepes.2006.10.002. URL: http://www.sciencedirect.com/science/article/pii/S014206150600189X.

[Mah10] Ashutosh Mahajan. “Presolving Mixed–Integer Linear Programs”. In: *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc., 2010. ISBN: 9780470400531. DOI: 10.1002/9780470400531.eorms0437. URL: http://dx.doi.org/10.1002/9780470400531.eorms0437.

[MSG13a] Sumit Mitra, Lige Sun, and Ignacio E. Grossmann. “Optimal scheduling of industrial combined heat and power plants under time-sensitive electricity prices”. In: *Energy* 54 (2013), pp. 194 –211. ISSN: 0360-5442. DOI: http://dx.doi.org/10.1016/j.energy.2013.02.030. URL: http://www.sciencedirect.com/science/article/pii/S0360544213001448.

[RL07a] Aiying Rong and Risto Lahdelma. “An effective heuristic for combined heat-and-power production planning with power ramp constraints”. In: *Applied Energy* 84.3 (2007), pp. 307 –325. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2006.07.005. URL:

http://www.sciencedirect.com/science/article/pii/S0306261906001024.

[RL07b] Aiying Rong and Risto Lahdelma. "Efficient algorithms for combined heat and power production planning under the deregulated electricity market". In: *European Journal of Operational Research* 176.2 (2007), pp. 1219 –1245. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2005.09.009. URL: http://www.sciencedirect.com/science/article/pii/S037722170500706X.

[RLL08] Aiying Rong, Risto Lahdelma, and Peter B. Luh. "Lagrangian relaxation based algorithm for trigeneration planning with storages". In: *European Journal of Operational Research* 188.1 (2008), pp. 240 –257. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2007.04.008. URL: http://www.sciencedirect.com/science/article/pii/S0377221707003840.

[RVP14] Dewan Fayzur Rahman, Ana Viana, and João Pedro Pedroso. "Metaheuristic search based methods for unit commitment". In: *International Journal of Electrical Power & Energy Systems* 59.0 (2014), pp. 14 –22. ISSN: 0142-0615.

[SA10] Jarmo Söderman and Pekka Ahtila. "Optimisation model for integration of cooling and heating systems in large industrial plants". In: *Applied Thermal Engineering* 30.1 (Jan. 2010), pp. 15–22. ISSN: 1359-4311. DOI: 16/j.applthermaleng.2009.03.018. URL: http://www.sciencedirect.com/science/article/pii/S1359431109001008.

[TBW05] Eva Thorin, Heike Brand, and Christoph Weber. "Long-term optimization of cogeneration systems in a competitive market environment". In: *Applied Energy* 81.2 (June 2005), pp. 152–169. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2004.04.012.

[Tom88] J. A. Tomlin. "Special ordered sets and an application to gas supply operations planning". In: *Mathematical Programming* 42.1 (1988), pp. 69–84. ISSN: 1436-4646. DOI: 10.1007/BF01589393. URL: http://dx.doi.org/10.1007/BF01589393.

[Tou+11] Michal Touš et al. "Effective biomass integration into existing combustion plant". In: *Energy* 36.8 (2011). PRES 2010, pp. 4654 –4662. ISSN: 0360-5442. DOI: 10.1016/j.energy.2011.04.036.

[VFB07] A. Vasebi, M. Fesanghary, and S.M.T. Bathaee. "Combined heat and power economic dispatch by harmony search algorithm". In: *International Journal of Electrical Power & Energy Systems* 29.10 (2007), pp. 713 –719. ISSN: 0142-0615. DOI: 10.1016/j.ijepes.2007.06.006. URL: http://www.sciencedirect.com/science/article/pii/S0142061507000634.

[VG+11] Patricia Velasco-Garcia et al. "Utility systems operation: Optimisation-based decision making". In: *Applied Thermal Engineering* 31.16 (2011), pp. 3196 –3205. ISSN: 1359-4311.

[Wan+95] S. J. Wang et al. "Short-term generation scheduling with transmission and environmental constraints using an augmented Lagrangian relaxation". In: *Power Systems, IEEE Transactions on* 10.3 (1995), pp. 1294–1301. ISSN: 0885-8950. DOI: 10.1109/59.466524.

[YDOJK08] J.M. Yusta, P.M. De Oliveira-De Jesus, and H.M. Khodr. "Optimal energy exchange of an industrial cogeneration in a day-ahead electricity market". In: *Electric Power Systems Research* 78.10 (2008), pp. 1764–1772. ISSN: 0378-7796. DOI: 16/j.epsr.2008.03.012. URL: http://www.sciencedirect.com/science/article/pii/S0378779608000977.

[Fer+04] G. Ferrari-Trecate et al. "Modeling and control of co-generation power plants: a hybrid system approach". In: *IEEE Transactions on Control Systems Technology* 12.5 (Sept. 2004), pp. 694–705. ISSN: 1063-6536. DOI: 10.1109/TCST.2004.826958.

Appendix D
List of author's publications

D.1 Publications related to this thesis

D.1.1 Publications in journals indexed in Web of Science

Michal Dvořák and Petr Havel. "Combined heat and power production planning under liberalized market conditions". In: *Applied Thermal Engineering* 43 (2012), pp. 163– 173. Co-authorship: 80%. Cited by [SDS16; SIV15; Tac+15; SSB14; Bis+14b; HB14; ZD14a; MSG13b; AP13; KV12].

D.1.2 Conference publications indexed in Web of Science

Michal Dvořák and Petr Havel. "Combined heat and power production planning under liberalized market conditions". In: *Chemical Engineering Transactions* 25 (2011), pp. 527–532. Co-authorship: 90%.

D.1.3 Other conference publications

Michal Dvořák and Petr Havel. "Decomposition methods for operations scheduling problem in a combined heat and power plant". In: 10th International Conference on Environment and Electrical Engineering 2011. Co-authorship: 90%.

D.2 Publications not related to this thesis

D.2.1 Publications in journals indexed in Web of Science

Jan Zábojník and Michal Dvořák. "Power grid simulation model for long term operation planning". In: *Applied Thermal Engineering* (2014). Co-authorship: 10%. Cited by [Kos+15; Ost15; LVK14].

D.2.2 Conference publications indexed in Web of Science

Jan Zábojník and Michal Dvořák. "Power grid simulation model for long term operation planning". In: *Chemical Engineering Transactions* 35 (2013), pp. 1105–1110. Co-authorship: 10%. Cited by [Goe+14; ZD14b; ZD14a].

D.2.3 Other conference publications

Jan Zábojník and Michal Dvořák. "Modelling and simulation of large scale power grids". In: *Proceedings of the 14th International Conference on Environment and Electrical Engineering*, pp.182-186. Co-authorship: 10%.

Appendix E

Citing articles from Web of Science

[AP13] Michele Anatone and Valentina Panone. "Integration of CCHP and solar plants for household applications". In: *2013 INTERNATIONAL CONFERENCE ON RENEWABLE ENERGY RESEARCH AND APPLICATIONS (ICRERA)*. International Conference on Renewable Energy Research and Applications. 2013, 499–504. ISBN: 978-1-4799-1464-7.

[Bis+14b] Aldo Bischi et al. "A detailed MILP optimization model for combined cooling, heat and power system operation planning". In: *ENERGY* 74 (2014). 26th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems (ECOS), Guilin, PEOPLES R CHINA, JUL 16-19, 2013, 12–26. ISSN: 0360-5442. DOI: {10.1016/j.energy.2014.02.042}.

[Goe+14] Peter Goerbe et al. "Experimental study of the nonlinear distortion caused by domestic power plants". In: *APPLIED THERMAL ENGINEERING* 70.2, SI (2014), 1288–1293. ISSN: 1359-4311. DOI: {10.1016/j.applthermaleng.2014.05.072}.

[HB14] Patrick Hochloff and Martin Braun. "Optimizing biogas plants with excess power unit and storage capacity in electricity and control reserve markets". In: *BIOMASS & BIOENERGY* 65 (2014), 125–135. ISSN: 0961-9534. DOI: {10.1016/j.biombioe.2013.12.012}.

[Kos+15] Anja Kostevsek et al. "Locally Integrated Energy Sectors supported by renewable network management within municipalities". In: *APPLIED THERMAL ENGINEERING* 89 (2015), 1014–1022.

ISSN: 1359-4311. DOI: {10.1016/j.applthermaleng.2015.01.037}.

[KV12] Jiri Jaromir Klemes and Petar Sabev Varbanov. "Heat integration including heat exchangers, combined heat and power, heat pumps, separation processes and process control". In: *APPLIED THERMAL ENGINEERING* 43.SI (2012), 1–6. ISSN: 1359-4311. DOI: {10.1016/j.applthermaleng.2012.03.044}.

[LVK14] Hon Loong Lam, Petar Sabev Varbanov, and Jiri Jaromir Klemes. "Applied Thermal Engineering towards sustainable development". In: *APPLIED THERMAL ENGINEERING* 70.2, SI (2014), 1051–1055. ISSN: 1359-4311. DOI: {10.1016/j.applthermaleng.2014.05.092}.

[MSG13b] Sumit Mitra, Lige Sun, and Ignacio E. Grossmann. "Optimal scheduling of industrial combined heat and power plants under time-sensitive electricity prices". In: *ENERGY* 54 (2013), 194–211. ISSN: 0360-5442. DOI: {10.1016/j.energy.2013.02.030}.

[Ost15] Poul Alberg Ostergaard. "Reviewing EnergyPLAN simulations and performance indicator applications in EnergyPLAN simulations". In: *APPLIED ENERGY* 154 (2015), 921–933. ISSN: 0306-2619. DOI: {10.1016/j.apenergy.2015.05.086}.

[SDS16] Li Sun, Stephen Doyle, and Robin Smith. "Understanding steam costs for energy conservation projects". In: *APPLIED ENERGY* 161 (2016), 647–655. ISSN: 0306-2619. DOI: {10.1016/j.apenergy.2015.09.046}.

[SIV15] Mirko M. Stojiljkovic, Marko G. Ignjatovic, and Goran D. Vuckovic. "Greenhouse gases emission assessment in residential sector through buildings simulations and operation optimization". In: *ENERGY* 92.3, SI (2015), 420–434. ISSN: 0360-5442. DOI: {10.1016/j.energy.2015.05.021}.

[SSB14] Mirko M. Stojiljkovic, Mladen M. Stojiljkovic, and Bratislav D. Blagojevic. "Multi-Objective Combinatorial Optimization of Trigeneration Plants Based on Metaheuristics". In: *ENERGIES* 7.12 (2014), 8554–8581. ISSN: 1996-1073. DOI: {10.3390/en7128554}.

[Tac+15] Leonardo Taccari et al. "Short-Term Planning of Cogeneration Power Plants: a Comparison Between MINLP and Piecewise-Linear MILP Formulations". In: *12TH INTERNATIONAL SYMPOSIUM ON PROCESS SYSTEMS ENGINEERING AND 25TH*

EUROPEAN SYMPOSIUM ON COMPUTER AIDED PROCESS ENGINEERING, PT C. Ed. by Gernaey, KV and Huusom, JK and Gani, R. Vol. 37. Computer Aided Chemical Engineering C. 2015, 2429–2434. ISBN: 978-0-444-63576-1.

[ZD14a] Jan Zábojník and Michal Dvořák. "Modelling and Simulation of Large Scale Power Grids". In: *2014 14TH INTERNATIONAL CONFERENCE ON ENVIRONMENT AND ELECTRICAL ENGINEERING (EEEIC)*. 14th International Conference on Environment and Electrical Engineering (EEEIC), Krakow, POLAND, MAY 10-12, 2014. 2014, 182–186. ISBN: 978-1-4799-4660-0.

[ZD14b] Jan Zábojník and Michal Dvořák. "Power grid simulation model for long term operation planning". In: *APPLIED THERMAL ENGINEERING* 70.2, SI (2014), 1294–1305. ISSN: 1359-4311. DOI: {10.1016/j.applthermaleng.2014.05.064}.