Lehrstuhl für Technische Thermodynamik

Operations Research

Masterarbeit

zur Erlangung des Grades Master of Science RWTH Aachen University (M.Sc. RWTH)

Optimization of Distributed Energy Supply Systems by Branch-and-Price

Optimierung von dezentralen Energieversorgungssystemen mit Branch-and-Price

1. Gutachtender Prüfer:	Univ.-Prof. Dr. rer. nat. Marco E. Lübbecke[1]
2. Gutachtender Prüfer:	Univ.-Prof. Dr.-Ing. André Bardow[2]
Beratungsassistenten:	Jonas T. Witt[1], M.Sc., Dr.-Ing. Philip Voll[2], Dinah E. Majewski[2], M.Sc.

vorgelegt an der Rheinisch-Westfälischen Technischen Hochschule Aachen, Lehrstuhl für Operations Research

von: Georg Ferdinand Schneider
Seilgraben 40
52062 Aachen
Matr.-Nr.: 282470

Aachen, Oktober 2014

[1] RWTH Aachen University
Lehrstuhl für Operations Research
Kackertstraße 7
52072 Aachen

[2] RWTH Aachen University
Lehrstuhl für Technische Thermodynamik
Schinkelstraße 8
52062 Aachen

Vorwort

Bei der Erstellung dieser Arbeit bin ich von vielen Personen unterstützt worden, denen ich gerne danken möchte. Ein besonderer Dank gilt allen Mitarbeitern des Lehrstuhls für Operations Research für die schöne Zeit und die freundliche Aufnahme. Allen voran möchte ich Prof. Dr. Marco E. Lübbecke danken dafür, dass er mich mit seinen Ausführungen in der Vorlesung für die Thematik und die Methode begeistern konnte und meine Idee für die Masterarbeit sofort aufgenommen hat. Ein großes Dankeschön geht an meine beiden Betreuer Jonas T. Witt und Dinah E. Majewski für die Betreuung meiner Arbeit und die vielen Vorschläge, Hilfestellungen und Ratschläge die ich erhalten habe. Ein besonderer Dank geht an Jonas für die Weiterentwicklung und Anpassung des `GCG`-Codes, ohne die einige Rechenergebnisse nicht möglich gewesen wären. Dinah danke ich für die vielen guten Ideen und Anregungen zu meiner Arbeit.
Christian Puchert danke ich für seine Ratschläge und Anleitung zur Nutzung von `GCG` und die kreativen Herausforderungen am white board (Kai Pirinha,...). Judith Behrooz, Beatrix Lindner, Annika Thome, Sarah Kirchner, Florian Dahms, Michael Bastubbe und Sebastian Goderbauer danke ich für das freundliche Miteinander und die spannenden Ideen bei den gemeinschaftlichen Mittagspausen. Meinen studentischen Kollegen Fritz Arnold, Markus Kruber, Marc Peiter, Asya Mitseva, Jens Dodenhoff danke ich für die vielen tollen Stunden im HiWi-Raum.
Den Kollegen vom Lehrstuhl für Technische Thermodynamik danke ich für die freundliche Aufnahme und die tolle Zeit. Prof. Dr. André Bardow danke ich für die Übernahme der Aufgabe des zweiten Prüfers. Ich danke Dr. Philip Voll dafür, dass er meinen Vorschlag für die Arbeit und die doppelte Betreuung sofort angenommen und unterstützt hat. Maike Hennen und Björn Bahl danke ich für die tatkräftige Unterstützung bei der GAMS-Modellierung.
Danken möchte ich Anne J. Wyrwoll und Joachim Schneider für ihre Korrekturen und Anmerkungen. Marion Schneider und Alexander Chatterton danke ich für ihre Korrekturen und ihr künstlerisches Gespür für die Feinheiten der englischen Sprache.
Ich möchte meinen Eltern dafür danken, dass sie mir die Möglichkeit und die Unterstützung für mein weiterführendes Studium der Wirtschatswissenschaft gegeben haben und nie am Gelingen meiner Studien gezeifelt haben. Pia danke ich für den Mut, die Zuversicht und die Sicherheit die sie mir gibt.

Aachen, im Oktober 2014 Georg F. Schneider

Abstract

This work evaluates whether optimization problems resulting from modelling the optimal synthesis, design and operation of a decentralized energy supply system have an embedded structure which can be exploited by decomposition methods in solution algorithms. The objective is to determine if the the accuracy and/or the problem size in terms of number of periods of time and number of units considered may be increased, as these are limited if the branch-and-bound method combined with the simplex method is used to solve the problems. A model of the problem is formulated as a mixed-integer linear program as proposed by Yokoyama et al. (2002) and Voll (2013). The model is analyzed and two embedded structures suitable for decomposition are identified.
The first structure emphasizes the independent operation and design of every component. The second structure emphasizes the design and operation of all components and focuses on the independence of every period of time considered.
The model is reformulated using the Dantzig-Wolfe decomposition principle for both proposed embedded structures. A numerical study is conducted where the synthesis, design and operation of a fictional energy supply system is optimized by both the branch-and-bound method combined with the simplex method and by the branch-and-price method. A set of instances is created for different degrees of complexity in terms of the number of units and the number of periods of time considered.
The results show that the dual bounds obtained by solving the rootnode LP relaxation can be improved in comparison to the conventional solution approach, if the reformulation emphasizing independent components is utilized. The results provide no evidence on improvements on the considered test set for the reformulation emphasizing design and operation.
For the case of an optimal solution computing times required to solve the considered instances of a test set are found to be reduced by utilizing the branch-and-price method and the reformulation emphasizing components, if identical components are considered in the energy supply system in comparison to the non-commercial solver `SCIP`.

Kurzfassung

In dieser Arbeit wird untersucht ob Optimierungsprobleme, die in der Strukturoptimierung von dezentralen Energieversorgungssystemen auftreten, eine eingebettete Struktur aufweisen, die mit dekompositionsbasierten Optimierungsmethoden ausgenutzt werden kann. Ziel ist es zu evaluieren, ob die Genauigkeit der Modelle oder die Größe der untersuchten Systeme in Bezug auf die Anzahl der berücksichtigten diskreten Zeitschritte und/oder der Anzahl der berücksichtigten Anlagen erhöht werden kann. Diese sind bei dem aktuell angewendeten Lösungsverfahren, das auf dem Branch-and-bound-Verfahren kombiniert mit dem Simplex-Verfahren basiert, begrenzt.
Das Problem der Strukturoptimierung wird, wie von den Autoren Yokoyama et al. (2002) und Voll (2013) vorgeschlagen, als gemischt-ganzzahliges lineares Programm formuliert. Die Analyse des gemischt-ganzzahligen Programms ergibt, dass zwei Dekompositionen des Problems möglich sind.
Charakteristisch für die erste identifizierte Struktur ist, dass alle Komponenten einzeln dimensioniert und betrieben werden können. Die zweite gefundene Struktur wird dadurch charakterisiert, dass die Auslegung und der Betrieb der Komponenten unabhängig von den betrachteten Zeitpunkten ist.
Das gemischt-ganzzahlige Modell wird nach der Methode der Dantzig-Wolfe Dekomposition für beide gefundenen Strukturen reformuliert. Zur Überprüfung der gefundenen Reformulierungen wird die Strukturoptimierung eines fiktiven Energieversorgungssystems durchgeführt. Für dieses System werden Instanzen mit unterschiedlicher Komplexität, bezogen auf die berücksichtige Anzahl an diskreten Zeitschritten und die berücksichtigte Anzahl an Komponenten, definiert. Alle Instanzen werden mit Lösern, die die Branch-and-bound-Methode kombiniert mit der Simplex-Methode verwenden gelöst. Außerdem werden die Instanzen mit einem Löser der die Branch-and-price-Methode anwendet gelöst.
Die Ergebnisse zeigen, dass die duale Schranke im Wurzelknoten mit der ersten Reformulierung, die auf der unabhängigen Dimensionierung und Betrieb jeder einzelnen Komponente basiert, im Vergleich zu der bisherigen Lösungsmethode verbessert werden kann. Eine Verbesserung durch die zweite Reformulierung, die auf unabhängigen Zeitpunkten basiert, kann nicht nachgewiesen werden.
Anhand der Ergebnisse kann gezeigt werden, dass, im Falle einer optimalen Lösung der Instanz, im Vergleich zum frei verfügbaren Löser `SCIP`, geringere Lösungszeiten mit der Reformulierung, die die unabhängige Dimensionierung und den Betrieb der einzelnen Komponenten berücksichtigt, erreicht werden.

Contents

List of Figures

List of Tables

Nomenclature

Latin Symbols

Symbol	Description	Unit
A	Constraint coefficient matrix	-
a	Column of constraint matrix **A**	-
$A \subseteq N$	Set of all absorption chillers	-
b	Vector of right hand side values of constraints	-
$B \subseteq N$	Set of all boilers	-
c	Vector of constant cost coefficients	-
c	Scalar constant cost coefficient	-
CF	Cash flow of investment	€
D	Constraint coefficient matrix	-
d	Vector of right hand side values of constraints	-
d	Number of segment of piecewise linear function (part load performance)	-
D	Set of all segments of piecewise linear function (part load performance)	-
$\dot{E}$	Energy demand	kW
E	Constraint coefficient matrix	-
F	Constraint coefficient matrix	-
g	Vector of right hand side values of constraints	-
GAP	Measure to quantify the quality of a formulation in (6.4)	
H	Constraint coefficient matrix	-
i	Number of segment of piecewise linear function (investment cost degression)	-
I	Set of all segments of piecewise linear function (investment cost degression)	-
IC	Investment Cost	€
J	Set of all extreme rays R and extreme points P	-
k	Number of discrete capacity	-
k	Number of subproblems	-

Continued on next page

Latin Symbols

Symbol	Description	Unit
K	Set of discrete capacities of a component	-
K	Set of identical subproblems	-
l	Integer bound	-
L	Lagrangean function	-
m	Number of instance	-
M	Set of all instances	-
n	Number of energy conversion unit	-
N	Set of all energy conversion units	-
NPV	Net present value of investment	€
p	Constant price for each unit of energy	€/ kWh
p	Number of extreme point or generator	-
P	Formulation of an integer set	-
P	Set of all extreme points of $conv(X)$	-
P	Set of all integer points of $conv(X)$	-
P	Probability, function of τ	-
PVF	Present value factor	-
r	Number of extreme ray	-
r	Performance ratio	-
R	Set of all extreme rays of $conv(X)$	-
R	Set of all integer rays of $conv(X)$	-
s	Number of solver	-
S	Set of all solvers	-
t	Number of period of time	-
t	Time	-
T	Set of all discrete time periods	-
$\dot{U}$	Input power	kW
u	Relative input power scaled by capacity	-
$\dot{V}$	Output power or capacity	kW
v	Relative output power scaled by capacity	-
$\mathbf{x}$	Vector of variables	-
x	Variable	-
X	Set of variables	-
$\mathbf{y}$	Vector of integer variables	-
y	Binary decision variable if unit exists	-
z	Objective function value	-
Z	Set of variables	-

Greek Symbols

Symbol	Description	Unit
α	Auxiliary variable in (5.13)	-
β	Auxiliary variable in (5.13)	-
Δt	Length of time period	-
δ	Binary decision variable representing the on-/off- status of a unit	-
η	Efficiency	-
θ	Fractional sum of all aggregated master variables	-
λ	Scalar master variable	-
$\boldsymbol{\lambda}$	Vector of master variables	-
ξ	Substitute of bilinear product	kW
$\boldsymbol{\nu}$	Vector of aggregated master variables	-
π	Dual variable	-
$\boldsymbol{\pi}$	Vector of dual variables	-
$\boldsymbol{\sigma}$	Lower bound on values in (3.32)	-
τ	Parameter for performance ratio	-
φ	Discount rate of investment	-

Subscripts

Symbol	Description
CF	Of cash flow
d	Of d-th segment of piecewise linear function (part load performance)
i	Index indicating position in vector
i	Of i-th segment of piecewise linear function (investment cost degression)
j	Index indicating position in vector
k	Number of discrete capacity of an energy conversion unit
LD	Lagrangrean dual
m	Of instance m
n	Of n-th energy conversion unit
$(\mathcal{PP})$	From problem $(\mathcal{PP})$
P	of problem defined in (3.29)
p	Extreme point or generator
r	Extreme ray

Continued on next page

Subscripts

Symbol	Description
$root$	Of rootnode
s	Of solver
t	Of t-th period of time
$\underline{()}$	Lower bound or geometric mean
$+$	Only values ≥ 0
0	Of convexity constraint
$-$	Only value ≤ 0

Superscripts

Symbol	Description
$cool$	Cooling
DB	Dual bound
$heat$	Heating
k	Subproblem
m	Number of rows in a matrix
M	For maintenance
n	Number of variables
N	Nominal
q	Number of discrete variables
S	For energy sold
T	Transpose of a vector
U	For energy bought
0	Intercept
(0)	Constant cost coefficient in (4.21)
(1)	Constant cost coefficient in (4.21)
(2)	Constant cost coefficient in (4.21)
(3)	Constant cost coefficient in (4.21)
$\overline{()}$	Upper bound, or $\bar{c}$ reduced cost
$*$	Optimal
$*$	Branching candidate

Abbreviations

Symbol	**Description**
AC	Absorption chiller
CPU	Central processing unit
CHP	Combined heat and power
COP	Coefficient of performance
`CPLEX`	Branch-and-cut solver (IBM, 2011)
DFO	Derivative free optimization
e.g.	for example
Eq.	Equation
Fig.	Figure
GA	Genetic Algorithm
GB	Gigabyte
`GCG`	Generic column generation, a branch-and-price solver (Gamrath and Lübbecke, 2010)
GO	Global optimization
i.e.	that is
IP	Integer program
LP	Linear program
max	maximize/ maximum
MILP	Mixed-integer linear program
min	minimize/ minimum
MINLP	Mixed-integer nonlinear program
NLP	Nonlinear program
NPV	Net present value
PES	Primary energy source
PC	Personal computer
QP	Quadratic program
SA	Simulated annealing
`SCIP`	Solving constraint integer programs, a branch-and-cut solver (Achterberg, 2009)
s.t.	such that
Tab.	Table

Miscellaneous Symbols

Symbol	Description
$\mathbb{Q}$	Set of fractional numbers
$\mathbb{R}$	Set of real numbers
$\mathbb{Z}$	Set of integer numbers
$\lvert\,\rvert$	Cardinality of a set
$\prec$	precedes

1 Introduction

1.1 Present Situation

To satisfy the ever-increasing world's energy demand is one of the fundamental questions which have to be answered in future. A solution must be found in order to ensure the prosperity of the international human society.
Today, global economic growth is mainly fuelled by exploiting fossil energy resources accumulated during the past millennia. Future scenarios estimate the world's energy demand to increase further, driven by population growth and industrialization of emerging nations (BP, 2013; IEA, 2013).
However, there are limitations to a fossil fuelled world economy. Emissions released by the combustion of fossil fuels change the composition of the atmosphere significantly. Scholars found evidence that mankind already has an effect on world´s climate (Crutzen, 2002). A more important fact is that fossil resources are limited so that future generations will face energy scarcity if mankind continues its current rate of consumption of finite energy resources (Hubbert, 1956).
One possibility to overcome the disadvantages of an economy based on fossil resources is to use energy which is converted from renewable resources. Today the fastest growing primary energy source is energy converted from renewable resources (Schiffer, 2013). Some characteristics of renewable energy converters are challenging for engineers, such as their intermittent supply and geographically distributed structure due to their low energy density in comparison to fossil-fuel power plants.
A future transition from the current mainly fossil energy system into a mainly renewable energy system could present many potential difficulties, and as yet, many questions remain unanswered. Several trade-offs exist between the cost of energy, energy security and emissions.
The degree of complexity resulting from the multitude of options available to compose an energy supply system is impossible to manage manually. Therefore there is a need for powerful analytical tools to evaluate different energy conversion options and identify the most promising ones. The following question needs to be answered:

- What is the optimal structure, design and operational status of a mixed fossil-renewable energy system with respect to diverging objectives, e.g. total capital expenditure and/or greenhouse gas emissions?

1.2 Method and Objective

A possibility to answer the prior stated research question is to formulate a mathematical model of an energy supply system. This model can be optimized globally to evaluate possible energy conversion options and obtain conclusions to determine the optimal structure, design and operational status of an energy supply system i.e. amongst others reduce total capital expenditure and/or greenhouse gas emissions. One particular model formulation yields a mixed-integer linear program (MILP) (Yokoyama et al., 2002; Voll, 2013).
Typically models for energy supply systems are based on equations for steady-state energy and mass conservation. Restrictions occur from a demand, which has to be fulfilled, as well as by technical restrictions. Binary decision variables occur for decisions such as inclusion of an energy converter or its on/off status. Due to the complexity and combinatorial possibilities in an energy supply system the resulting MILP may be large in terms of the number of variables and equations.
Branch-and-price is a method to solve large-scale (mixed-) integer programs with a considerable size (some billion variables) in a sufficient time, especially if the model formulation has a block-diagonal structure with identical subproblems (Desrosiers and Lübbecke, 2010). Branch-and-price is successfully applied to typical problems from mathematical programming theory and practical applications (Barnhart et al., 1998; Lübbecke and Desrosiers, 2005).
Usually the described MILPs are solved by applying the branch-and-bound method combined with the simplex method (see description in Nocedal and Wright, 2006), however accuracy of the models and the size of the energy supply system in terms of the number of periods of time and number of units considered is limited to achieve a solution within an acceptable time.
The objective of this work is to evaluate, whether the underlying problem has a beneficial structure which may be exploited by decomposition. If suitable, the branch-and-price method is applied to solve MILPs resulting from the modelling of energy supply systems, so that the complexity or accuracy of the models may be increased in comparison to the conventional solution algorithm.

1.3 Structure of the Thesis

In chapter 2 a literature review is performed to evaluate possible decomposition methods and their application in energy engineering. An outline of the identified decomposition methods Dantzig-Wolfe reformulation, Lagrangrean relaxation and Benders' decomposition is presented in chapter 3. A mixed-integer linear program formulation as formulated by Yokoyama et al. (2002) and Voll (2013) is presented in chapter 4. The model is analyzed to determine possible decomposition structures in chapter 5. Numerical results of the identified structures are presented in chapter 6.

2 Optimization of Energy Supply Systems

Mathematical optimization has developed from a mere field of academic interest into a thriving technology, pushing forward the frontiers of science in an increasingly wide variety of different scientific domains. Mathematical optimization techniques are widely applied and accepted in energy engineering (Biegler and Grossmann, 2004). In particular process system engineering is a vital source of new applications and solution methods (Biegler and Grossmann, 2004).
Within the field of energy engineering a large range of different application areas of mathematical optimization exist. One example is the synthesis, design and operation of energy and process systems (Subrahmanyam et al., 1996; Yokoyama et al., 2002; Voll, 2013; Rieder, 2013).
The unit commitment problem is an example where the optimal operation of an existing energy system is considered (Sheble and Fahd, 1994; Padhy, 2004). In the unit commitment problem electric power producing utilities minimize the operation costs of their power producing units, while meeting the time-varying energy demands. Constraints occur from technical restrictions, for example minimum up-time and down-time of each unit and regulatory limitations to ensure reserve requirements related to energy security issues.
Often the task of optimal synthesis and design is conducted simultaneously with optimal operation (Frangopoulos et al., 2002; Yokoyama and Ito, 2006; Voll et al., 2012; Voll, 2013; Voll et al., 2013; Wakui and Yokoyama, 2014).
In the case of real-time applications, for example supply chain optimization and scheduling problems, many tools and knowledge from general optimization theory can be applied (Biegler and Grossmann, 2004). Optimal control tasks to operate plants or process systems is another area where a wide range of different applications occur (Biegler and Grossmann, 2004). In particular, linear and non-linear model predictive control is a well known field of application (e.g. Hölemann, 2011).
Optimization objectives may vary widely in energy engineering. Common objectives are economic criteria such as net present value (NPV) (Voll, 2013) or annual total cost (Yokoyama and Ose, 2012). Also physical entities are of interest such as total entropy generation or carbon dioxide emissions (le Roux et al., 2012; Figliozzi, 2010). These objectives may be considered simultaneously resulting in a multi-objective approach where trade-offs occur (Bouvy and Lucas, 2007; Rieder et al., 2014).
Problems resulting from this wide range of applications are linked to all classes of optimization (see section 3.1, p. 7).

Decomposition is a well known technique in optimization of energy systems. Modelling of multi-period design, planning and scheduling problems leads to mixed-integer (non-)-linear programs which often have a specific structure. This embedded structure may be exploited by decomposition.
According to Grossmann and Biegler (2004) several decomposition schemes have been proposed in the literature including Benders' decomposition (Benders, 1962), Lagrangean relaxation (Wolsey, 1998; Guignard, 2003) and Bilevel decomposition (e.g. Iyer and Grossmann, 1998). Additionally reported are cutting plane methods, augmented Lagrangean decomposition, splitting methods and nested decomposition. Biegler and Grossmann (2004) do not explicitly mention Dantzig-Wolfe type decomposition (Dantzig and Wolfe, 1961).
Sagastizábal (2012) applies Benders' decomposition and Lagrangean relaxation to optimize the behaviour of an independent system operator. Langrangean relaxation is successfully applied by Rong et al. (2008) to optimize a trigeneration system (electric power, heat and cooling) with storage and by Virmani et al. (1989) to optimize the unit commitment problem. Finardi and Luiz da Silva (2006) apply Lagrangean relaxation in a non-linear problem solving instances of the hydro unit commitment problem.
Dantzig-Wolfe type reformulation is applied by Sanghvi and Shavel (1986) to the investment planning for hydro-thermal power system expansion, solving problem sizes of up to 30,000 rows and 54,000 columns.
Several authors exploit hierarchical structures between the levels of interest, i.e. synthesis, design and operation (Frangopoulos et al., 2002). Subrahmanyam et al. (1996) present a decomposition approach for batch plant design and operation. The decomposition is based on the hierarchical relationship between the different time scales of design and scheduling.
Yokoyama and Ito (2000) and Yokoyama et al. (2002) present a decomposition method for mixed-integer linear programs with a block diagonal structure to conduct the operational planning of energy supply systems with storage. They apply Dantzig-Wolfe reformulation to obtain a master problem and several identical subproblems which may be solved independently. However, the problem is solved by a self developed algorithm which cannot guarantee finding the optimal solution. They solve subproblems associated to minimize the reduced costs of the master problem, however, not always to optimality as they apply a heuristic optimality criterion, i.e. values of binary variables of a feasible solution of the subproblem with negative reduced costs are likely to give a better feasible solution of the original problem (p. 778, Yokoyama et al., 2002). This algorithm, although calculating solutions efficiently, does not guarantee optimality.
Yokoyama and Ose (2012) and Yokoyama et al. (2014) present another approach to decompose the aforementioned problem by utilizing a hierarchical relationship between the design and operation variables. In this particular case they formulate a model taking discrete capacities into account. They solve an upper level design problem by the conventional branch-and-bound method combined with the simplex one. A lower level operation problem is solved by giving the values for the design

variables tentatively. They conduct a series of bounding operations to optimally solve the mentioned problem.
Until now no author reported the application of branch-and-price (see section 3.2, p. 9) to the problem of simultaneously optimizing synthesis, design and operation of an distributed energy supply system. In particular optimality of a solution obtained may not be guaranteed with the approaches reported by others (Yokoyama and Ito, 2000; Yokoyama et al., 2002).

3 Theory of Large Scale Optimization

3.1 Mathematical Programming

For the class of deterministic optimization problems this section provides an overview. The classification approach is adapted from Biegler and Grossmann (2004).
Deterministic optimization can be differentiated into optimization with continuous or discrete variables (see Fig. 3.1). Optimization problems with both types of variables are possible. In case of continuous optimization linear programs (LP) and non-linear programs (NLP) are important problem classes. Non-linearities occur for example if part-load behaviour of energy conversion units is considered. Quadratic programming (QP) is an important subclass of NLP with special solution methods. An important question considering NLPs is, whether the problems are convex or non-convex. The latter gives rise to multiple local optima. A special class of problems with non-differentiable objective function is derivative free optimization (DFO). These problems may be solved by simulated annealing (SA) or genetic algorithms (GA). However, SA and GA do not guarantee optimality.

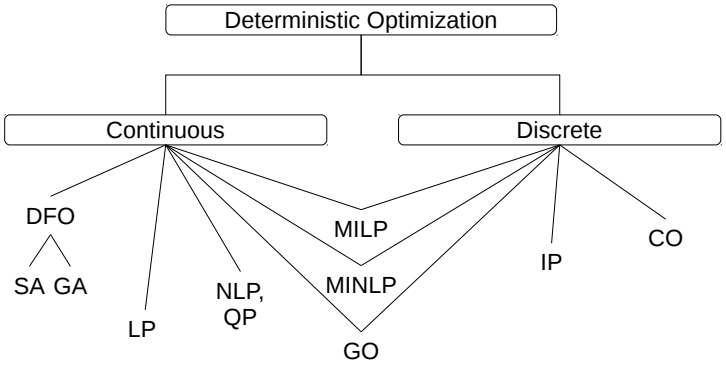

Figure 3.1: Classification scheme of problem classes in deterministic optimization (Biegler and Grossmann, 2004, adapted).

The combination of discrete and continuous optimization gives way to mixed-integer linear programs (MILPs) and mixed-integer non-linear programs (MINLPs). In case of non-convex optimization with local optima problem classes in this field are classified as global optimization problems (GO). When variables are purely integer

this is termed integer program (IP). A special case is combinatorial optimization (CO). Note that there is a wide range of different optimization problems. An attempt to provide a complete overview is presented by (NEOS, 2014).

3.1.1 Linear Programming

We define the problem

$$\begin{array}{rrcl} \min & \mathbf{c}^T\mathbf{x} & & \\ \text{s.t.} & \mathbf{A}\mathbf{x} & \leq & \mathbf{b} \\ & \mathbf{x} & \geq & 0 \ , \end{array} \tag{3.1}$$

where $\mathbf{c} \in \mathbb{R}^n_+$ is a vector of all constant cost coefficients to their respective variable and $()^T$ denote the transpose of a vector, $\mathbf{x} \in \mathbb{R}^n_+$ is a vector of continuous variables of the problem. $\mathbf{A} \in \mathbb{R}^{m\times n}$ is the constraint matrix and $\mathbf{b} \in \mathbb{R}^m$ are the right hand sight values of the constraints. Problems of the same type as (3.1) are called *linear program* (LP).
The most common method to solve LPs is the simplex method. See the textbook by Nocedal and Wright (2006) for a detailed description of the method.

3.1.2 Mixed-Integer Linear Programming

We define the problem

$$\begin{array}{rrcl} \min & \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} & & \\ \text{s.t.} & \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} & \leq & \mathbf{b} \\ & \mathbf{x} & \geq & 0 \\ & \mathbf{y} & \in & \mathbb{Z}^q_+ \ , \end{array} \tag{3.2}$$

where $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^q$ are vectors of all constant cost coefficients of their respective variables, $\mathbf{x} \in \mathbb{R}^n_+$ is a vector of continuous variables of the problem, $\mathbf{y} \in \mathbb{Z}^q_+$ is a vector of integer variables of the problem, $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{D} \in \mathbb{R}^{m\times q}$ are the constraint matrices of their respective variables and $\mathbf{b} \in \mathbb{R}^m$ is the right hand sight values of the constraints. Problems of the same type as problem (3.2) are called *mixed-integer linear program* (MILP).
A possibility to solve the above formulated MILP is the branch-and-bound method. A detailed description may be found in the textbook by Nocedal and Wright (2006). Here, the method is briefly reviewed for the case of dichotomic branching.
The first step in the branch-and-bound method is to solve the linear programming relaxation of (3.2), that is substituting the integrality constraint by $\mathbf{y} \geq 0$. The solution of the linear relaxation is a lower bound on the optimal objective function value of the integer problem. An upper bound on the objective function value is obtained from every feasible integer solution. If an integer variable remains

fractional after solving the LP relaxation one branches and creates two subproblems. At every branching decision two subproblems, called *nodes*, are added. This results in a *tree* of nodes. In each subproblem a constraint is introduced which excludes the fractional value of the variable and forces it to at least the next upper (up-branch) or lower (down-branch) integer value. If the lower bound of a branch is higher than the current valid upper bound of all active branches, this branch is terminated as no node of this subtree will improve the current objective function value.

3.2 Branch-and-Price

The aim of many decomposition and reformulation approaches in integer programming is to reduce symmetry and obtain stronger relaxations, to achieve faster convergence of the solution algorithms. Solution algorithms often rely on the dynamic addition of variables (columns) and/or constraints (cutting planes) to the problem (Desrosiers and Lübbecke, 2010). In particular, one aims to exploit embedded structures of the problem.
When solving the linear relaxation in each node of a branch-and-bound tree by column generation, this is called *branch-and-price* (Desrosiers and Lübbecke, 2010). *Branch-and-price* is successfully applied to many different applications in industry and science (Lübbecke and Desrosiers, 2005). Examples are vehicle routing, crew and machine scheduling problems (Desrosiers et al., 1984; van den Akker et al., 1999), the general assignment problem (Savelsbergh, 1997; Barnhart et al., 1998), bin packing and cutting stock problems (Vanderbeck, 1999), and graph colouring (Mehrotra and Trick, 1996).
Additional applications and an extensive body of literature are covered in the reviews by Barnhart et al. (1998) and more recently by Lübbecke and Desrosiers (2005). A comprehensive description of *branch-and-price* is presented by Desrosiers and Lübbecke (2010).
In the subsequent sections the main underlying principles of *branch-and-price* as used in this thesis are presented, following the description by Desrosiers and Lübbecke (2005, 2010) and Lübbecke (2011).

3.2.1 Definitions

Definitions used in the subsequent sections on polyhedra and reformulation theory are summarized briefly in this chapter (As presented in: Vanderbeck, 2010, 13.2.2). Let $X \subseteq \mathbb{Z}^n$ be the solution space of an integer linear program.

Definition 3.2.1. A *polyhedron* $P \subseteq \mathbb{R}^n$ is the intersection of a finite number of half spaces. In other words, one can formulate the set $P = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} \geq \mathbf{b}\}$, with $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$.

Definition 3.2.2. A polyhedron P is a formulation for X if $X = P \cap \mathbb{Z}^n$.

Note that sets such as X have many different formulations. Let P^1, P^2 be two formulations for X with $P^1 \subset P^2$. One calls P^1 a *stronger* formulation than P^2 because

$$z = \min\{\mathbf{c}^T\mathbf{x}|\mathbf{x} \in X\} \geq \min\{\mathbf{c}^T\mathbf{x}|\mathbf{x} \in P^1\} \geq \min\{\mathbf{c}^T\mathbf{x}|\mathbf{x} \in P^2\}, \tag{3.3}$$

with $\mathbf{c} \in \mathbb{R}^n$. The result is that the lower bound on z obtained from the LP relaxation with formulation P^1 is always greater or equal to that obtained from the LP relaxation of P^2.

Definition 3.2.3. Given $X \subseteq \mathbb{R}^n$, the *convex hull* of X, denoted by $conv(X)$, is the tightest, i.e. the smallest closed convex set containing X.

Note that the convex hull of an integer set X is a polyhedron. The convex hull of X is also the strongest possible formulation of X as every corner point of $conv(X)$ is an integer feasible solution.

3.2.2 Column Generation

Column generation is a method to solve linear programs by iteratively adding variables to the model. In particular column generation is an applicable method to solve linear programs with some trillions of variables. It utilizes the fact, that often only a relatively small fraction of the total number of variables is needed to prove optimality (Lübbecke, 2011). Column generation is often mentioned simultaneously with Dantzig-Wolfe reformulation (see section 3.2.3). It is particularly effective if the solved problem has a special structure, e.g. bordered block-diagonal with identical subproblems (see section 3.2.4).
Let

$$\begin{aligned} \min \quad & \sum_{j \in J} c_j \lambda_j & & \\ \text{s.t.} \quad & \sum_{j \in J} \mathbf{a}_j \lambda_j & \leq \quad & \mathbf{b} \\ & \boldsymbol{\lambda} & \in \quad & \mathbb{Z}_+^{|J|} \ , \end{aligned} \tag{3.4}$$

be an integer program and called *integer master problem*, where $c_j \in \mathbb{R}$ are the scalar coefficients of the objective function, $\lambda_j \in \mathbb{Z}_+$ are the integer variables, $\mathbf{a}_j \in \mathbb{R}^m$ is a coefficient column j of the coefficient matrix and $\mathbf{b} \in \mathbb{R}^m$ is a vector of the right hand side coefficients of the constraints.
The linear relaxation of (3.4), is called *master problem* (MP) and is defined as

$$\begin{aligned} \min \quad & \sum_{j \in J} c_j \lambda_j & & & \\ \text{s.t.} \quad & \sum_{j \in J} \mathbf{a}_j \lambda_j & \leq \quad & \mathbf{b} & \\ & \lambda_j & \geq \quad & 0 \qquad & \forall\, j \in J \ , \end{aligned} \tag{3.5}$$

where vectors have appropriate dimensions. Solving (3.5) is computationally expensive as $|J|$ is large. Instead it is solved by column generation as follows.
Consider (3.6) which is called *restricted master problem* (RMP) where only a subset $J' \subseteq J$ of variables is considered.

$$\begin{array}{lllll} \min & \sum_{j \in J'} c_j \lambda_j & & & \\ \text{s.t.} & \sum_{j \in J'} \mathbf{a}_j \lambda_j & \leq & \mathbf{b} & [\boldsymbol{\pi}] \\ & \lambda_j & \geq & 0 & \forall\, j \in J' \end{array} \tag{3.6}$$

In each iteration (3.6) is solved to optimality by applying the simplex method (see Nocedal and Wright, 2006) and a primal as well as a dual optimal solution is obtained, $\boldsymbol{\lambda}^* \in \mathbb{Q}_+^{|J'|}$ and $\boldsymbol{\pi}^* \in \mathbb{Q}_-^m$, respectively.
From linear programming theory it is known that a linear problem is solved to optimality, when there are no non-basic variables with negative reduced costs. As it is computationally expensive to calculate the reduced costs of the $|J \setminus J'|$ variables, in column generation this problem is solved as an optimization problem. Let X define a generic set over which one can optimize

$$z_{\mathcal{PP}} = \min_{\mathbf{x} \in X} \{ c(\mathbf{x}) - (\boldsymbol{\pi}^*)^T a(\mathbf{x}) \} \ . \tag{3.7}$$

This is a subproblem, called *pricing problem*, where the reduced costs of (3.5) are minimized. The functions $c(\mathbf{x}_j) = c_j$ and $a(\mathbf{x}_j) = \mathbf{a}_j$ denote that each column $j \in J$ is associated with an element $\mathbf{x}_j \in X$. Often this association may be interpreted as a set of combinatorial objects such as paths or colour patterns, thus each $\mathbf{x}_j$ bears much more information than just its corresponding column $a(\mathbf{x}_j)$ (Desrosiers and Lübbecke, 2010).
If $z_{\mathcal{PP}} < 0$, there still is a vector $\mathbf{x}_j$ with negative reduced costs. A corresponding variable λ_j and coefficient column $(c(\mathbf{x}_j), a(\mathbf{x}_j))$ is added to the RMP and another iteration starts. Else, $z_{\mathcal{PP}} \geq 0$ proves that there is no variable with negative reduced costs and the current $\boldsymbol{\lambda}^*$ is an optimal solution to both the RMP and MP.

3.2.3 Dantzig-Wolfe Decomposition

The Dantzig-Wolfe decomposition principle (Dantzig and Wolfe, 1961) was developed to exploit special embedded structures within the constraint matrix of linear programs. Later it was adapted to integer programs where reveals its 'full strength' (Desrosiers and Lübbecke, 2010). Assume a problem, where the constraint matrix is very sparse and all constraints may be divided into *easy* constraints $\mathbf{Dx} \leq \mathbf{d}$ and *complicating* constraints $\mathbf{Ax} \leq \mathbf{b}$. Consider

$$\begin{array}{lllll} \min & \mathbf{cx} & & & \\ \text{s.t.} & \mathbf{Ax} & \leq & \mathbf{b} & \\ & \mathbf{x} & \in & X \ , & \end{array} \tag{3.8}$$

which is called *original problem* and is optimized with respect to the mixed-integer set $X = \{\mathbf{x} = \binom{\mathbf{x}^{\mathbb{Z}}}{\mathbf{x}^{\mathbb{Q}}}, \mathbf{x}^{\mathbb{Z}} \in \mathbb{Z}_+^{n-q}, \mathbf{x}^{\mathbb{Q}} \in \mathbb{Q}_+^q \,|\, \mathbf{Dx} \leq \mathbf{d}\}$. Matrices and vectors have appropriate dimensions. The original problem is rather hard to solve when the embedded structure of X is not considered. The variables $\mathbf{x}$ are called *original variables*.
Note that the set X contains mixed-integer variables. In the following descriptions assume X to be a pure integer set, where differences are stated when necessary (Desrosiers and Lübbecke, 2010).

3.2.3.1 Convexification

Classically Dantzig-Wolfe decomposition is based on the representation theorems by Minkowski and Weyl (Schrijver, 1986). The approach described in this section is termed convexification because X is *convexified*.

Theorem 3.2.1. Each $\mathbf{x} \in X$ can be expressed as a *convex combination* of finitely many *extreme points* $\{\mathbf{x}_p\}_{p\in P}$ plus a non-negative conic combination of finitely many *extreme rays* $\{\mathbf{x}_r\}_{r\in R}$ of $conv(X)$, i.e.

$$\mathbf{x} = \sum_{p\in P} \mathbf{x}_p\lambda_p + \sum_{r\in R} \mathbf{x}_r\lambda_r, \quad \sum_{p\in P} \lambda_p = 1, \quad \boldsymbol{\lambda} \in \mathbb{Q}_+^{|P|+|R|} \ . \tag{3.9}$$

A so called *extended formulation*

$$\begin{array}{lrcll} \min & \sum_{p\in P} \mathrm{c}_p\lambda_p + \sum_{r\in R} \mathrm{c}_r\lambda_r & & & \\ \text{s.t.} & \sum_{p\in P} \mathbf{a}_p\lambda_p + \sum_{r\in R} \mathbf{a}_r\lambda_r & \leq & \mathbf{b} & [\boldsymbol{\pi}] \\ & \sum_{p\in P} \lambda_p & = & 1 & [\pi_0] \\ & \boldsymbol{\lambda} & \geq & 0 & \\ & \mathbf{x} & = & \sum_{p\in P} \mathbf{x}_p\lambda_p + \sum_{r\in R} \mathbf{x}_r\lambda_r & \\ & \mathbf{x} & \in & \mathbb{Z}_+^n \ , & \end{array} \tag{3.10}$$

is obtained by substituting $\mathbf{x}$ in (3.8) by (3.9) and applying the linear transformations $c_j = \mathbf{c}^T\mathbf{x}_j$ and $a_j = A\mathbf{x}_j$, $j \in P \cup R$. Vectors have appropriate dimensions. Note that the optimal objective function values of (3.8) and (3.10) are equal. The extended formulation is an *integer master problem* as (3.4).
The first constraints in (3.10) are called *coupling constraints* with respective dual variables $\boldsymbol{\pi}$ and the second constraint is called *convexity constraint* with respective dual variable π_0. Note that by the last constraint, integrality is still imposed on the original variables $\mathbf{x} \in X$.

Let

$$\begin{array}{lrcll} \min & \sum\limits_{p\in P} \mathrm{c}_p\lambda_p + \sum\limits_{r\in R} \mathrm{c}_r\lambda_r & & & \\ \text{s.t.} & \sum\limits_{p\in P} \mathbf{a}_p\lambda_p + \sum\limits_{r\in R} \mathbf{a}_r\lambda_r & \leq & \mathbf{b} & [\boldsymbol{\pi}] \\ & \sum\limits_{p\in P} \lambda_p & = & 1 & [\pi_0] \\ & \boldsymbol{\lambda} & \geq & 0 & \end{array} \tag{3.11}$$

be the linear relaxation of (3.10). (3.11) is called *master problem* (MP). The number of variables of (3.11) is high due to the cardinality of $P \cup R$, therefore an applicable solution method is column generation. The restricted master problem (RMP) of (3.11) is defined by only taking a subset of variables $p \in P' \subseteq P$, $r \in R' \subseteq R$ into account. With respective dual variables $\boldsymbol{\pi}$ and π_0, the pricing problem

$$z_{PP} = \min\{\mathbf{cx} - \boldsymbol{\pi}\mathbf{Ax} - \pi_0 \,|\, \mathbf{x} \in X\} \tag{3.12}$$

is defined, which needs to be solved at each iteration of the column generation algorithm. Ideally this pricing problem is solvable by a tailored combinatorial algorithm which yields integer solutions to (3.12). If not, it needs to be solved by some standard IP solver which may be very time consuming.
When solving (3.12) to optimality there are three possible outcomes (Lübbecke, 2011). First, if $z_{PP} < 0$ and z_{PP} finite, an optimal solution to the pricing problem is an extreme point $\mathbf{x}_p$ of X, and a variable with coefficient column $[\mathbf{cx}_p, (\mathbf{Ax}_p), 1]$ is added to the RMP. Second, if $z_{PP} = -\infty$, one obtains an extreme ray $\mathbf{x}_r$ of X, and the column $[\mathbf{cx}_r, (\mathbf{Ax}_r), 1]$ is added to the RMP. And finally, if $z_{PP} \geq 0$ there are no negative reduced costs columns anymore. This proves the optimal solution of both RMP and MP.

3.2.3.2 Discretization

While in convexification $conv(X)$ is reformulated, in *discretization* X itself is reformulated. The concept was introduced by Vanderbeck since 'it allows for the development of a unifying and complete theoretical framework to deal with all relevant issues that arise in the implementation of a branch-and-price algorithm' Vanderbeck (2000).

Theorem 3.2.2. *(Nemhauser and Wolsey, 1988)* Each $\mathbf{x} \in X$ can be expressed as an *integral combination* of a finite set of *integer points* $\{\mathbf{x}_p\}_{p\in P} \subseteq X$ plus a finite set of *integer rays* $\{\mathbf{x}_r\}_{r\in R}$ of X, i.e.

$$\mathbf{x} = \sum_{p\in P} \mathbf{x}_p\lambda_p + \sum_{r\in R} \mathbf{x}_r\lambda_r, \quad \sum_{p\in P} \lambda_p = 1, \quad \boldsymbol{\lambda} \in \mathbb{Z}_+^{|P|+|R|} \ . \tag{3.13}$$

Note that here the notation is kept as presented in (Desrosiers and Lübbecke, 2010), despite the authors slightly abused notation to ensure a better understanding. In (3.13) the set P contains integer points which usually are not identical to the corresponding extreme points of the convexification approach. Rays in R are scaled to be integer.
Again, one obtains a so called *extended formulation* (3.14) by substituting $\mathbf{x}$ in (3.8) by (3.13) and applying the linear transformations $c_j = \mathbf{c}^T\mathbf{x}_j$ and $a_j = A\mathbf{x}_j$, $j \in P \cup R$. This yields the *integer master problem*

$$\begin{aligned} \min \quad & \sum_{p\in P} \mathrm{c}_p\lambda_p + \sum_{r\in R} \mathrm{c}_r\lambda_r & & \\ \text{s.t.} \quad & \sum_{p\in P} \mathbf{a}_p\lambda_p + \sum_{r\in R} \mathbf{a}_r\lambda_r & \leq & \mathbf{b} \\ & \sum_{p\in P} \lambda_p & = & 1 \\ & \boldsymbol{\lambda} & \in & \mathbb{Z}_+^{|P|+|R|} \quad . \end{aligned} \tag{3.14}$$

Note in this case integrality is required on $\boldsymbol{\lambda}$ variables.
There are some certain characteristics of the integer master problem (3.14) obtained by applying the discretization approach (Vanderbeck, 2000):

- The LP relaxation of (3.14) gives the same dual bound as (3.10).
- If solved by column generation, the pricing problem of the linear relaxation of (3.14) is the same as in (3.12). It is necessary to ensure that it can generate integer solutions in the interior of X.
- Let X be bounded and (3.8) has a nonlinear objective function, then (3.14) still is a linear integer program.

In case of a mixed-integer set X the discretization approach is generalized such that integer variables are discretized and continuous variables are convexified (Vanderbeck and Savelsbergh, 2006). In case of $X \subseteq \{0,1\}^n$, i.e. binary programming, convexification and discretization coincide.

3.2.4 Bordered Block-Diagonal Matrices

As mentioned in the sections before Dantzig-Wolfe decomposition works particularly well if the constraint matrix has a special, a *bordered block-diagonal*, form. Mathematically this means that the set X decomposes into $X^k = \{\mathbf{x}^k \in \mathbb{Z}_+^{n^k} \mid \mathbf{D}^k\mathbf{x}^k \leq \mathbf{d}^k\}$, $k \in K$, with all matrices and vectors of appropriate dimensions and $\sum_k n^k = n, \sum_k q^k = q$. This means that the original problem (3.8) may be rewritten as

$$\begin{aligned} \min \quad & \sum_{k\in K} \mathbf{c}^k\mathbf{x}^k & & & \\ \text{s.t.} \quad & \sum_{k\in K} \mathbf{A}^k\mathbf{x}^k & \leq & \mathbf{b} & \\ & \mathbf{x}^k & \in & X^k \qquad & k \in K, \end{aligned} \tag{3.15}$$

with all matrices and vectors of appropriate dimensions. Applying Dantzig-Wolfe decomposition to the problem above, i.e. substitute each $\mathbf{x}^k$ by (3.9) or (3.13) and introduce λ_j^k variables, $j \in P^k \cup R^k$, where $P = \bigcup_{k=1}^K P^k$ and $R = \bigcup_{k=1}^K R^k$. An important special case is when some or all $(\mathbf{D}^k, \mathbf{d}^k)$ are identical. Examples are bin packing or using identical units in an energy supply system. This symmetry may cause inefficiency in a branch-and-bound algorithm, as *same* solutions can be expressed in many different ways by permuting K.
To cope with this symmetry one *aggregates* the master variables λ_p^k, that is $\nu_p := \sum_{k \in K} \lambda_p^k$ and sums up the $|K|$ convexity constraints. There is no need to aggregate extreme rays (Desrosiers and Lübbecke, 2010). For a representative P^1 one obtains the *aggregated extended formulation*

$$\begin{array}{lrcll} \min & \displaystyle\sum_{p \in P^1} \mathrm{c}_p \nu_p + \sum_{r \in R} \mathrm{c}_r \lambda_r & & & \\ \text{s.t.} & \displaystyle\sum_{p \in P^1} \mathbf{a}_p \nu_p + \sum_{r \in R} \mathbf{a}_r \lambda_r & \leq & \mathbf{b} & \\ & \displaystyle\sum_{p \in P^1} \nu_p & = & K & \\ & \boldsymbol{\nu} & \in & \mathbb{Z}_+^{|P^1|} & \\ & \boldsymbol{\lambda} & \in & \mathbb{Z}_+^{|R|} \ . & \end{array} \tag{3.16}$$

Analogously to (3.12) the corresponding k pricing problems are defined as

$$z_{PP}^k = \min\{\mathbf{c}^k \mathbf{x}^k - \boldsymbol{\pi}^k \mathbf{A}^k \mathbf{x}^k - \pi_0^k \,|\, \mathbf{x}^k \in X^k\}, \forall k \in K \ . \tag{3.17}$$

If subproblems k are identical, it is sufficient to solve only one of the k subproblems as each of them will generate exchangeable columns.

3.2.5 Linking Variables

Problems may occur which have independent structures in the constraint matrix that are linked by variables. As mentioned before, the branch-and-price method benefits from a bordered block-diagonal structure of the constraint matrix, in particular when subproblems are identical. Different approaches exist to split variables or constraints. See Guignard (2003) and Vanderbeck (2010) for examples. Here the case that linking variables exist in the subproblems is discussed as presented by Guignard (2003).
Consider the following integer master problem

$$\begin{array}{lrcll} \min & \displaystyle\sum_{k \in K} \mathbf{c}^k \mathbf{x}^k + \mathbf{d}\mathbf{y} & & & \\ & \text{s.t.} \quad \displaystyle\sum_{k \in K} \mathbf{A}^k \mathbf{x}^k & \leq & \mathbf{b} & \\ & \mathbf{D}^k \mathbf{x}^k & \leq & \mathbf{d}^k & \forall\, k \in K \\ & \mathbf{E}\mathbf{y} + \displaystyle\sum_{k \in K} \mathbf{F}^k \mathbf{x}^k & \leq & \mathbf{g} & \\ & \mathbf{x}^k & \in & \mathbb{Z}_+^{n^k} & \forall\, k \in K \\ & \mathbf{y} & \in & \mathbb{Z}_+^q \ , & \end{array} \tag{3.18}$$

where matrices and vectors have appropriate dimensions. Not that variable y is not dependent on k, therefore this deters the easy constraints $\mathbf{Ey} + \sum_{k\in K} \mathbf{F}^k\mathbf{x}^k \leq \mathbf{g}$ to fully decompose in k subproblems.
A circumvention of this problem is to introduce aggregated copy constraints (Guignard, 2003). Let $\mathbf{y} = \mathbf{y}'^k$, $k \in K$ be $|K|$ aggregate copy constraints for problem (3.18) and $Z^k = \{(\mathbf{x}^k, \mathbf{y}'^k) \in \mathbb{Z}_+^{n^k} \times \mathbb{Z}_+^{q^k} | \, \mathbf{D}^k\mathbf{x}^k \leq \mathbf{d}^k, \, \mathbf{E}'^k\mathbf{y}'^k + \mathbf{F}^k\mathbf{x}^k \leq \mathbf{g}'^k, \, k \in K\}$, then one obtains

$$\begin{array}{lrcll} \min & \sum\limits_{k\in K} \mathbf{c}^k\mathbf{x}^k + \mathbf{dy} & & & \\ & \text{s.t.} \quad \sum\limits_{k\in K} \mathbf{A}^k\mathbf{x}^k & \leq & \mathbf{b} & \\ & \mathbf{y}'^k & = & \mathbf{y} & \forall k \in K \\ & \mathbf{y} & \in & \mathbb{Z}_+^q & \\ & (\mathbf{x}^k, \mathbf{y}'^k) & \in & Z^k & \forall k \in K \ , \end{array} \tag{3.19}$$

where matrices and vectors have appropriate dimensions. Now the *easy* constraints in Z^k decompose. Note that $|K|$ additional linking constraints $\mathbf{y} = \mathbf{y}'^k$ need to be considered in the inter master problem. Another possibility is to introduce $\mathbf{Hy} = \mathbf{H}'^k\mathbf{y}'^k, k \in K$, which in general yields a worse bound than the aforementioned possibility (Guignard, 2003).

3.2.6 Branching

In general branch-and-price is a method to solve IPs and MILPs. When linear relaxations of the master problem are solved by column generation fractional values of the integer variables may be obtained. Therefore it is necessary to apply a branch-and-bound scheme to obtain an integer solution.
Intuitively one would consider branching on the master variables. In the case of convexification where there are no integer master variables this would be not feasible and in the case of discretization this leads to an unbalanced branch-and-bound search tree. Setting the down-branch to zero has no effect on the dual bound, whereas in the up-branch it has a significant impact on the solution. In particular, in the down-branch the regeneration of already generated solutions needs to be forbidden in the pricing problem.
A characteristic of branching rules in the context of column generation is *compatibility*. This means that branching should not complicate the pricing problem. For additional reading on branching in this context the reader may be referred to Vanderbeck (2010) and Vanderbeck and Wolsey (2010). Note that there exists tailored branching rules for specific problems, e.g. in case of a set partitioning master problem (Ryan and Foster, 1981).
In this section for the case of convexification branching on original variables and in case of discretization branching as proposed by Vanderbeck (2010) and Vanderbeck and Wolsey (2010) is presented.

3.2.6.1 Convexification - Branching on Original Variables

The preferred approach in problems with only one pricing problem is convexification. Recall that in convexification, integrality is required on the original variables $\mathbf{x}$. Considering (3.10) all x_i variables with non integer column entries are branching candidates, i.e. $\mathrm{x}_i^* = \sum_{j \in P \cup R} \mathrm{x}_{ji} \lambda_j^* \notin \mathbb{Z}_+$, with x_{ji} describing the i-th component of $\mathbf{x}_j$, $j \in P \cup R$. After deciding to pick one branching candidate, two new problems are created. In the down-branch $\mathrm{x}_i \leq \lfloor \mathrm{x}_i^* \rfloor$ and in the up-branch $\mathrm{x}_i \geq \lceil \mathrm{x}_i^* \rceil$ is imposed. In general there are two options, either add the branching decision to the master problem or to the pricing problem. As the up-branch may be handled analogously (Desrosiers and Lübbecke, 2010), here only the implementation of the down-branch is presented.

Master problem: The Reformulation of the branching constraint $\mathrm{x}_i \leq \lfloor \mathrm{x}_i^* \rfloor$ by convexification

$$\sum_{p \in P} \mathbf{x}_{pi} \lambda_p + \sum_{r \in R} \mathbf{x}_{ri} \lambda_r \leq \lfloor \mathrm{x}_i^* \rfloor \tag{3.20}$$

is added to the master problem (3.10). Note that now a modification of the objective function of the pricing problem is necessary and an additional dual variable is included. A disadvantage of this approach is, that no integer points in the interior of $conv(X)$ can be generated from the pricing problem. This may result in missing the optimal solution in the case of general integer variables $\mathbf{x}$ (see Desrosiers and Lübbecke, 2010, p. 11).

Pricing Problem: If one introduces the constraint $\mathrm{x}_i \leq \lfloor \mathrm{x}_i^* \rfloor$ in the pricing, this results into prohibiting the generation of extreme points and rays that violate the branching decision. Additionally it has to be ensured that master variables already present are eliminated. Constraints

$$\sum_{\substack{j \in P \cup R: \\ \mathrm{x}_{ji}=1}} \lambda_j = 0 \quad \text{or equivalently,} \quad \sum_{\substack{j \in P \cup R: \\ \mathrm{x}_{ji}=0}} \lambda_j = 1 \tag{3.21}$$

are added to the master problem (3.10), which is, to some extent, modifying the convexity constraint. The pricing problem is obtained by

$$\min\{\mathbf{cx} - \boldsymbol{\pi}\mathbf{Ax} - \pi_0 \,|\, \mathbf{x} \in X \cap \{\mathbf{x} \,|\, \mathrm{x}_i \leq \lfloor \mathrm{x}_i^* \rfloor\}\} \ . \tag{3.22}$$

Despite that this may complicate the pricing problem, this option is to be preferred. It has the advantage of generating a potentially stronger dual bound from the master problem relaxation, because the bound change is also convexified, i.e.

$$\begin{array}{ll} & \min\{\mathbf{cx} \,|\, \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \in conv(X), \mathrm{x}_i \leq \lfloor \mathrm{x}_i^* \rfloor\} \\ \leq & \min\{\mathbf{cx} \,|\, \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \in conv(X \cap \{\mathbf{x} \,|\, \mathrm{x}_i \leq \lfloor \mathrm{x}_i^* \rfloor\})\} \ . \end{array} \tag{3.23}$$

Another advantage of changing the pricing problem is, that now its domain is partitioned and it is possible to generate points in the interior of $conv(X)$ after the branching.
The presented methodology applies to any node in the tree and may as well be generalized to mixed-integer programs (Desrosiers and Lübbecke, 2010).

3.2.6.2 Discretization - 'Vanderbeck' Branching

As mentioned previously symmetry is a cause for major inefficiencies and therefore should be avoided in mathematical programs. In case of identical subproblems a possibility to avoid this, is to aggregate original variables $\sum_{k\in K} \mathbf{x}^k = \mathbf{z}$. This results in a single pricing problem. However, when branching decisions on the original variables are introduced, the variables again are disaggregated and one obtains distinct pricing problems (Villeneuve et al., 2005). Clearly this does not eliminate symmetry from original variables $\mathbf{x}_k$ (Vanderbeck, 2010).
Let

$$\mathbf{x}^k = \sum_{p\in P^k} \mathbf{x}_p \lambda_p^k \ , \tag{3.24}$$

be a unique projection from the master variables $\boldsymbol{\lambda}$ into the space of original variables $\mathbf{x}$. There are some shortcomings in disaggregation and projection of master variables into original variable space. For example in the aggregated master problem (3.16) disaggregating variables $\nu_p = \sum_{k\in K} \lambda_p^k$ by using (3.24) to calculate an original solution is not advantageous, a this projection is not unique and integrality of $\boldsymbol{\nu}$ does not lead necessarily to integrality of $\mathbf{x}$.
To avoid symmetry and shortcomings at the same time, one needs a trick that includes a unique projection from $\boldsymbol{\lambda}$ in $\mathbf{x}$ variables without the direct correspondence (3.24).
An attempt by Vanderbeck (2010) is to obtain values $\mathbf{x}_1^*, ..., \mathbf{x}_K^*$ by summing up the master variables in a lexicographic order of the corresponding $\mathbf{x}_p$. Let

$$\lambda_{kp}^* = \min\left\{1,\ \nu_p - \sum_{\kappa=1}^{k-1} \lambda_{\kappa p}^*,\ \max\{0, k - \sum_{q:\mathbf{x}_q \prec \mathbf{x}_p} \nu_q^*\}\right\} \ , \tag{3.25}$$

be for all $k \in K$ and $p \in P$, where $\mathbf{x}_q \prec \mathbf{x}_p$ denotes that $\mathbf{x}_q$ precedes $\mathbf{x}_p$ in that ordering. The original variables are calculated by applying (3.24). A characteristic of the lexicographic sorting is, that one always works with a unique representative solution $\mathbf{x}$ out of the many symmetric possibilities. This is used to break symmetry in integer programs (Margot, 2010).
A branching decision on fractional aggregated original variables $y_i^* = \sum_{p\in P} \mathrm{x}_{pi} \nu_p^* \notin \mathbb{Z}_+$ may be introduced by imposing

$$y_i = \sum_{p\in P} \mathrm{x}_{pi} \nu_p^* \leq \lfloor y_i^* \rfloor \quad \text{, or} \quad y_i = \sum_{p\in P} \mathrm{x}_{pi} \nu_p^* \leq \lceil y_i^* \rceil \ . \tag{3.26}$$

However, Vanderbeck (2010) states that this simple rule may only give a little improvement on the dual bound.
There are some more advanced approaches which introduce auxiliary original variables in the original formulation (Vanderbeck, 2010).
The most general branching rule is to split the master variables via modifying the convexity constraint (Vanderbeck and Wolsey, 2010). For a fractional sum of all aggregated master variables

$$\sum_{p \in P : \mathrm{x}_{pi} \geq l_i} \nu_p \; = \; \theta \; \notin \mathbb{Z}_+ \;\; , \tag{3.27}$$

where index i stands for for each corresponding original variable x_i and an integer bound l_i, one creates two branches with

$$\sum_{p \in P : \mathrm{x}_{pi} \geq l_i} \nu_p \; \geq \; \lceil \theta \rceil \quad \text{, or} \quad \sum_{p \in P : \mathrm{x}_{pi} \leq l_i - 1} \nu_p \; \geq \; K \; - \; \lfloor \theta \rfloor \;\; . \tag{3.28}$$

This essentially is a modification of the master problem. In the pricing problem $\mathrm{x}_i \geq l_i$ and $\mathrm{x}_i \leq l_1 - 1$ need to be specified. A fractional θ in (3.27) is not guaranteed and one may need to impose additional bounds. This yields a generalized partition in a *nested* way (Vanderbeck and Wolsey, 2010; Desrosiers and Lübbecke, 2010). A more detailed description and several technical details may be found in (Vanderbeck and Wolsey, 2010).
Some important facts to note are that the last rule provides the strongest dual bound and implies only a small impact like bound changes on the pricing problem. Points in the interior of $conv(X)$ may be generated and the depth of the search tree is polynomially bounded (Desrosiers and Lübbecke, 2010).

3.2.7 Implementations and Frameworks

Several frameworks exist to implement branch-and-price algorithms, e.g. `ABACUS` (Jünger and Thienel, 2000), `BCP` (Ralphs and Ladányi, 2001), `MINTO` (Nemhauser et al., 1994), `SCIP` (Achterberg, 2009) and `SYMPHONY` (Ralphs et al., 2013). A code which performs Dantzig-Wolfe decomposition of a general MIP and solves the subproblems by column generation is `BaPCod` (Vanderbeck, 2014).
The software framework `DIP` (Galati, 2010) is an open-source software framework to implement a variety of decomposition-based bounding algorithms, e.g. Dantzig-Wolfe decomposition, Lagrangean relaxation and different cutting plane methods.
A branch-and-cut-and-price solver for MIPs is `GCG` (Gamrath and Lübbecke, 2010), which is used for the computational study in this thesis. `GCG` is based on the `SCIP`-framework and performs Dantzig-Wolfe decomposition on a generic problem. The structure for the decomposition is either provided by the user or detected by some plug-in detectors implemented in `GCG`. Some features already implemented are

branching rules for every generic problem and stabilization techniques (du Merle et al., 1999). The code is open source and is jointly developed by the Chair of Operations Research, RWTH Aachen University and Zuse-Institut Berlin.

3.3 Alternative Decomposition Techniques

Multiple decomposition techniques exist in mathematical programming theory, however this thesis focuses on branch-and-price. To give an overview common decomposition techniques *Lagrangean relaxation* and *Benders´ decomposition* are presented here, as they may be alternatives to Dantzig-Wolfe decomposition. A comprehensive overview and a detailed description of the aforementioned methods is presented in detail in the literature (Guignard, 2003; Vanderbeck, 2010; Benders, 1962; Geoffrion, 1972).

3.3.1 Lagrangean Relaxation

Lagrangean relaxation is a well known decomposition technique in integer and mixed-integer programming. Different authors report a successful application to energy engineering problems (Finardi and Luiz da Silva, 2006; Rong et al., 2008; Sagastizábal, 2012).
In this section a short introduction to Lagrangean relaxation is presented, following the detailed descriptions of Guignard (2003) and Vanderbeck (2010).
Assume the optimization problem

$$z_P = \min_{\mathbf{x}}\{\mathbf{cx} \,|\, \mathbf{Ax} \geq \mathbf{b}, \mathbf{Dx} \geq \mathbf{d}, \mathbf{x} \in \mathbb{Z}_+^n\} \ , \tag{3.29}$$

with matrices and vectors of appropriate dimensions. Similar to Dantzig-Wolfe reformulation it consists out of *difficult* constraints $\mathbf{Ax} \geq \mathbf{b}$ and relatively *easy* constraints $\mathbf{Dx} \geq \mathbf{d}$. The first mentioned set of constraints are considered *difficult*, in the sense that removing $\mathbf{Ax} \geq \mathbf{b}$ from the problem and introducing them in the objective function penalized by cost $\boldsymbol{\pi}$ results in a problem which is more tractable. This leads to

$$L(\boldsymbol{\pi}) := \min_{\mathbf{x}}\{\mathbf{cx} + \boldsymbol{\pi}(\mathbf{b} - \mathbf{Ax}) \,|\, \mathbf{Dx} \geq \mathbf{d}, \mathbf{x} \in \mathbb{Z}_+^n\} \ , \tag{3.30}$$

which is called *Lagrangean relaxation*. Note that for any non-negative dual variable $\boldsymbol{\pi} \geq 0$ the dual function $L(\boldsymbol{\pi})$ provides a lower bound on the solution of the original problem (3.29). Let $\mathbf{x}^*$ denote an optimal solution of (3.29) then the following is valid $\mathbf{cx}^* \geq \mathbf{cx}^* + \boldsymbol{\pi}(\mathbf{b} - \mathbf{Ax}^*) \geq L(\boldsymbol{\pi})$).
The task now is to maximize this bound over a set of admissible dual vectors, that is finding the tightest Lagrangean lower bound. Problem

$$z_{LD} = \max_{\boldsymbol{\pi}} L(\boldsymbol{\pi}) = \max_{\boldsymbol{\pi} \geq 0} \min_{\mathbf{x} \in X}\{\mathbf{cx} + \boldsymbol{\pi}(\mathbf{b} - \mathbf{Ax})\} \tag{3.31}$$

is called the *Lagrangean dual*, where $X = \{\mathbf{x} \in \mathbb{Z}_+^n \,|\, \mathbf{Dx} \geq \mathbf{d}\}$. The Lagrangean dual is reformulated as a linear program by assuming that the set X is non-empty and bounded. Note that the Lagrangean subproblem achieves its optimum at an extreme point $\mathbf{x}_p$ of $conv(X)$, this leads to

$$z_{LD} = \max_{\boldsymbol{\pi} \geq 0} \min_{p \in P} \{\mathbf{c}\mathbf{x}_p + \boldsymbol{\pi}(\mathbf{b} - \mathbf{A}\mathbf{x}_p)\} \ . \tag{3.32}$$

If an additional variable $\boldsymbol{\sigma}$ is introduced, which represents a lower bound on the values $(\mathbf{c} - \boldsymbol{\pi}\mathbf{A})\mathrm{x}_p$, one can rewrite (3.31) as

$$\begin{aligned} z_{LD} = \max\, & (\boldsymbol{\pi}\mathbf{b} + \boldsymbol{\sigma}) \\ s.t. \quad & \boldsymbol{\pi}\mathbf{A}\mathrm{x}_p \leq c\mathrm{x}_p \qquad \forall p \in P \\ & \boldsymbol{\pi} \geq 0 \\ & \boldsymbol{\sigma} \in \mathbb{R}^1 \ . \end{aligned} \tag{3.33}$$

Taking the dual of LP (3.33) gives

$$\begin{aligned} z_{LD} = \min \quad & \sum_{p \in P} (c\mathrm{x}_p)\lambda_p \\ \text{s.t.} \quad & \sum_{p \in P} (\mathbf{A}\mathrm{x}_p)\lambda_p \geq \mathbf{b} \\ & \sum_{p \in P} \lambda_p = 1 \\ & \boldsymbol{\lambda} \in \mathbb{R}_+^{|P|} \ , \end{aligned} \tag{3.34}$$

which is an master problem as (3.11) and may be solved by column generation.
Let $\mathbf{x}(\boldsymbol{\pi})$ denote an optimal solution of (3.30) for some $\boldsymbol{\pi} \geq 0$, then $\mathbf{x}(\boldsymbol{\pi})$ is called a *Lagrangean solution* (Guignard, 2003). The following theorem states the optimality criterion for Lagrangean relaxation.

Theorem 3.3.1. *(Guignard, 2003)*

1. If $\mathbf{x}(\boldsymbol{\pi})$ is an optimal solution of (3.30) for some $\boldsymbol{\pi} \geq 0$, then $\mathbf{c}\mathbf{x}(\boldsymbol{\pi}) + \boldsymbol{\pi}(\mathbf{A}\mathbf{x}(\boldsymbol{\pi}) - \mathbf{b}) \leq z_{LD}$
2. If in addition $\mathbf{x}(\boldsymbol{\pi})$ is feasible for (3.29), then $\mathbf{c}\mathbf{x}(\boldsymbol{\pi}) + \boldsymbol{\pi}(\mathbf{A}\mathbf{x}(\boldsymbol{\pi}) - \mathbf{b}) \leq z_P \leq z_{LD}$
3. If in addition $\boldsymbol{\pi}(\mathbf{A}\mathbf{x}(\boldsymbol{\pi}) - \mathbf{b}) = 0$, then $\mathbf{x}(\boldsymbol{\pi})$ is an optimal solution of (3.29), and $z_P = \mathbf{c}\mathbf{x}(\boldsymbol{\pi})$.

3.3.2 Benders´ Decomposition

A different approach to Dantzig-Wolfe reformulation and Lagrangean relaxation where usually *columns* are generated and problems are decomposed by their *constraints*, is the decomposition approach presented by Benders (1962). The approach

is based on decomposition by *variables*, and the solution involves generation of *rows*. Commonly the decomposition by variables is referred to as *Benders´ decomposition*. Noonan and Giglio (1977) and Sagastizábal (2012) report on successful examples of applying Benders´ decomposition to problems in energy engineering. In this section the basic principle of Benders´ decomposition is outlined as presented by Vanderbeck (2010).
Classically variable decomposition deals with MILPs as defined in (3.2), where one assumes that the integer variables are *important* decision variables. The optimization is decomposed in two stages, where on the first stage the integer variables are fixed and associated continuous variables are calculated. Benders´ decomposition differs from simple hierarchical decomposition as information from pricing the continuous variables is taken into account (Vanderbeck, 2010). Note that if the problem has a bordered block-diagonal structure, the subproblem obtained when fixing the integer variables decomposes.
Benders´ decomposition is based on a reformulation of an initial MILP as a linear integer program (Vanderbeck, 2010). The resulting reformulation has typically an exponential number of constraints. This is solved by branch-and-cut. Note that specific cuts are obtained during the solution process. For a detailed description the reader may be referred to the literature (Benders, 1962; Geoffrion, 1972).

4 A MILP for Distributed Energy Supply Systems

In the following sections a mixed-integer linear model formulation, as presented by Yokoyama et al. (2002) and Voll (2013), for the simultaneous synthesis, design and operation of a distributed energy supply system is derived.

4.1 Problem Definition

Energy supply systems have diversified in recent years in terms of conversion technologies and primary energy sources (PES). This is partly caused by the shift to unconventional fossil PES and also by increasing usage of renewable PES. Due to the low energy densities of renewable PES in comparison to fossil ones, renewable energy conversion units tend to possess lower capacities than fossil ones. Consequently the number of units in a renewable energy system is much larger than in fossil ones.
Also the growing usage of components for combined heat and power (CHP), such as micro gas turbines, stationary gas engines and fuel cells, increases the total number of conversion units in an energy supply system. This is also caused by lower capacities per unit, as the capacities of these units typically range from 10 kW to 1 MW. Hence, if the demand stays on a constant level, in energy systems with a large share of CHP and renewable energy converters the total number of units in these systems increases.
Frangopoulos et al. (2002) define the task of synthesizing an energy supply system as a hierarchical problem, which can be differentiated into the three levels: synthesis, design and operation. The cost-efficient synthesis, design and operation of a diverse energy system leads to a problem which is impossible to solve manually in a sufficient time. A possibility to cope with this difficulty is to define the problem as an optimization problem. As there are interdependencies between the different levels of interest, it is advantageous to optimize the structure, design and operation of an energy system simultaneously. For example, energy conversion units usually have their highest efficiency at a certain load. Thus, design decisions have a significant impact on a cost-effective operation.

It is assumed that, in the beginning, an initial *superstructure* of the energy supply system is given, which contains all candidates for selection. In Fig. 4.1 an exemplary superstructure for an energy supply system is illustrated. The initial superstructure

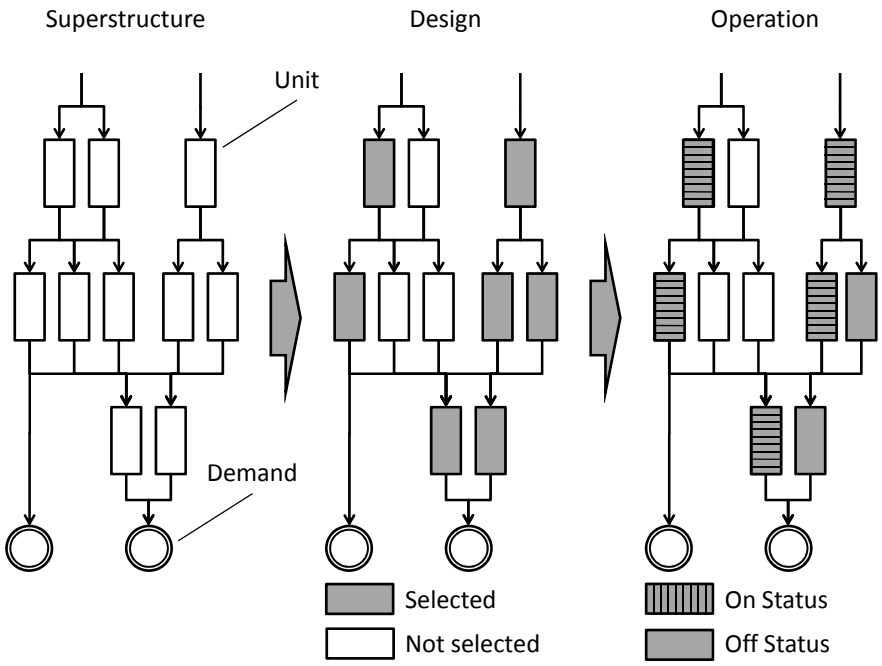

Figure 4.1: Unit selection and operation from an initial superstructure (Yokoyama et al., 2002, adapted).

is designed so that it is possible to fulfil at least the required energy demand. In the problem there are two decisions which have to be made. First, if a conversion unit is selected from the superstructure and second, if the unit is on or off in a specific period of time. The capacity of each unit may vary in a specified range and the load allocated to each unit in a certain period may vary within its capacity and a lower limit.

To assess the economic performance of the energy supply system, the net present value (NPV) of the investment of a *grassroots* energy supply system is adopted, i.e. the system is planned from scratch. Note that the specific case of grassroots synthesis can easily be expanded to the more general case of retrofit of existing energy supply systems. The generally concave investment cost degression function is linearized for every energy conversion unit by a piecewise linear function.

In this problem aggregated energy demands for multiple periods are defined. Constraints in this problem occur from quasi-stationary energy balances for each period and technical restrictions to model the part load performance of every unit. To cope with the nonlinear part load performance, again, a piecewise linear function for each component is introduced.

In this formulation the quality levels, i.e. temperatures and pressures, of all energy forms delivered are assumed to be constant (Voll, 2013).

4.2 Model Formulation

For the problem defined above, a model formulation is derived as presented by Yokoyama et al. (2002) and Voll (2013). The model is formulated for a simple generic energy system as depicted in Fig. 4.2, p. 27.

4.2.1 Notation

The following indices, sets, variables and parameters are defined.

1. Indices
    - n - Unit number
    - t - Period of time
    - d - Number of segment of piecewise linear function (part load performance)
    - i - Number of segment of piecewise linear function (investment cost degression)
2. Sets
    - N - Set of all energy conversion units
    - T - Set of all discrete periods of time
    - D - Set of all segments of piecewise linear function (part load performance)
    - I - Set of all segments of piecewise linear function (investment costs degression)
3. Continuous variables
    - $\dot{V}_{ndt}$ - Output power of unit n using segment d in period of time t
    - $\dot{V}_{nt}$ - Nonlinear output power of unit n in period of time t
    - $\dot{U}_{ndt}$ - Input power of unit n using segment d in period of time t
    - $\dot{U}_{nt}$ - Nonlinear input power of unit n in period of time t
    - ξ_{ndt} - Auxiliary variable to substitute bilinear term $\delta_{ndt} \cdot \sum_{i \in I} \dot{V}_{ni}^{N}$
    - $\dot{V}_{ni}^{N}$ - Capacity of unit n using segment i
    - $\dot{V}_{n}^{N}$ - Nonlinear capacity of unit n
    - IC_n - Investment cost of unit n
    - CF - Net cash flow of investment
    - NPV - Net present value of investment

4. Binary variables
    - y_{ni} - (Non)-existence of unit n using segment i
    - y_n - (Non)-existence of unit n
    - δ_{ndt} - On-/off- status of unit n using segment d in period of time t
    - δ_{nt} - On-/off-status of unit n in period of time t
5. Parameters
    - p^U - Constant price for each unit of energy purchased
    - p^S - Constant price for each unit of energy sold
    - φ - Discount rate of investment
    - t_{CF} - Time horizon of investment
    - $PVF(\varphi, t_{CF})$ - Present value factor of investment
    - p_n^M - Constant maintenance cost for each unit n
    - Δt_t - Length of every period of time t
    - $\dot{E}_t$ - Energy demand in every period of time t
    - η_n^N - Constant efficiency of unit n
    - $\underline{v}_{nd}$, $\overline{v}_{nd}$ - Lower and upper bound on $\dot{V}_{ndt}$ for every unit n and segment d
    - $\underline{\dot{V}}_{ni}^N$, $\overline{\dot{V}}_{ni}^N$ - Lower and upper bound on $\dot{V}_{ni}^N$ for every unit n and segment i
    - $\left(\frac{\mathrm{d}IC}{\mathrm{d}\dot{V}^N}\right)_{ni}$ - Slope of segment i of unit n
    - IC_{ni}^0 - Intercept segment i of unit n
    - $\left(\frac{\mathrm{d}u}{\mathrm{d}v}\right)_{nd}$ - Slope of segment d of unit n
    - u_{nd}^0 - Slope of segment d of unit n

4.2.2 Objective Function and Balance Equations

In the following section the objective function of the problem and the balance equations are defined. In this section nonlinearities in the objective function are kept and linearized in the subsequent sections.
From economic theory, multiple objective functions exist to assess the economic performance of an energy system, such as: total annual cost, profit, net present value, shareholder value and many more. In this thesis, following the description by

Voll (2013), the economic performance of the investigated energy supply system is assessed by the net present value[1]. The NPV is defined by

$$NPV := PVF \cdot CF - IC \ , \tag{4.1}$$

with variables and parameters as defined in the previous section and $PVF = ((\varphi + 1)^{t_{CF}} - 1)/(\varphi \cdot (\varphi + 1)^{t_{CF}})$. The net cash flow CF is the sum of all annual revenues from feed-in electricity minus annual energy delivery and maintenance costs. Note that the net cash flow is assumed to be constant for every year.

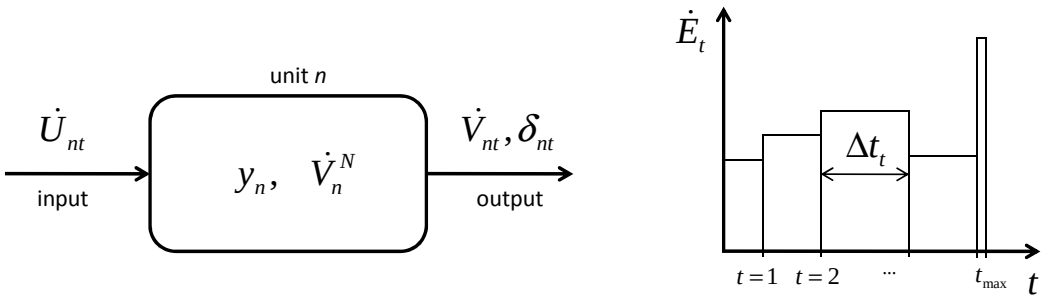

Figure 4.2: Flowsheet of a generic energy supply system. A generic energy conversion unit n converts input power $\dot{U}_{nt}$ into output power $\dot{V}_{nt}$ to fulfil a time varying energy demand $\dot{E}_t$. Each unit has a capacity $\dot{V}_n^N$. The (non)-existence is modelled by the binary variable y_n and the on/off-status of the unit at every period is modelled by a binary variable δ_{nt}. Each period has a length Δt_t, $t \in \{1, 2, ..., t_{max}\} = T$ (Voll, 2013, adapted).

For a generic energy system, as illustrated in Fig. 4.2, equation (4.1) can be rewritten as

$$\begin{aligned} PVF \cdot \sum_{n \in N} \Bigg(\sum_{t \in T} \Big(& - p^U \cdot \Delta t_t \cdot \dot{U}_{nt} \left(\dot{V}_{nt}, \delta_{nt}, \dot{V}_n^N \right) + p^S \cdot \Delta t_t \cdot \dot{V}_{nt} \Big) \\ & - p_n^M \cdot IC_n \left(y_n, \dot{V}_n^N \right) \Bigg) - \sum_{n \in N} IC_n \left(y_n, \dot{V}_n^N \right) \ . \end{aligned} \tag{4.2}$$

Equation (4.2) is the objective function of the problem and is maximized with respect to the continuous capacity $\dot{V}_n^N$, the continuous output power $\dot{V}_{nt}$ and binary variables y_n and δ_{nt} that model the decisions if the unit is selected ($y_n = 1$) or not ($y_n = 0$) and if the unit is on ($\delta_{nt} = 1$) or off ($\delta_{nt} = 0$) in period t, respectively. The net cash flow is determined by three terms: Firstly the costs of annual energy delivery are calculated by multiplying a constant purchase price for energy p^U with the input power $\dot{U}_{nt}$ and the length of a period Δt_t; secondly the maintenance costs are calculated by multiplying a constant price for maintenance p_n^M which is assumed to be a fixed percentage of the investment costs IC_n; thirdly the annual revenues

[1] Kasas et al. (2011) identified the NPV as the best suited economic criterion to asses single-objective flowsheet optimization.

from electricity feed-in are calculated by multiplying a constant price for energy sold p^S with the length of a period Δt_t and the output power $\dot{V}_{nt}$. Note that in (4.2) the terms for the input power $\dot{U}_{nt}$ and investment costs IC_n are not further specified here as this is presented in the subsequent sections.
The quasi-stationary energy balance of the system for every period is given by

$$\sum_{n \in N} \dot{V}_{nt} \;=\; \dot{E}_t \quad \forall\, (t \in T) \quad , \tag{4.3}$$

where the output power of every unit at the current period is summed up and the sum has to equal the correspondent energy demand.

4.2.3 Investment Cost

In general the investment costs per output power of a unit tend to decrease when its capacity increases (see Fig. 4.3, $IC_n(\dot{V}_n^N)$). To develop a linear formulation of the problem the investment cost degression curve $IC_n\left(y_n, \dot{V}_n^N\right)$ in (4.2) is linearized by a piecewise linear function, where $i \in I$ denotes the i-th segment. Thus in (4.2), $\sum_{n \in N} IC_n(y_n, \dot{V}_n^N)$ is replaced by $\sum_{n \in N} \sum_{i \in I} IC_n(y_{ni}, \dot{V}_{ni}^N)$.

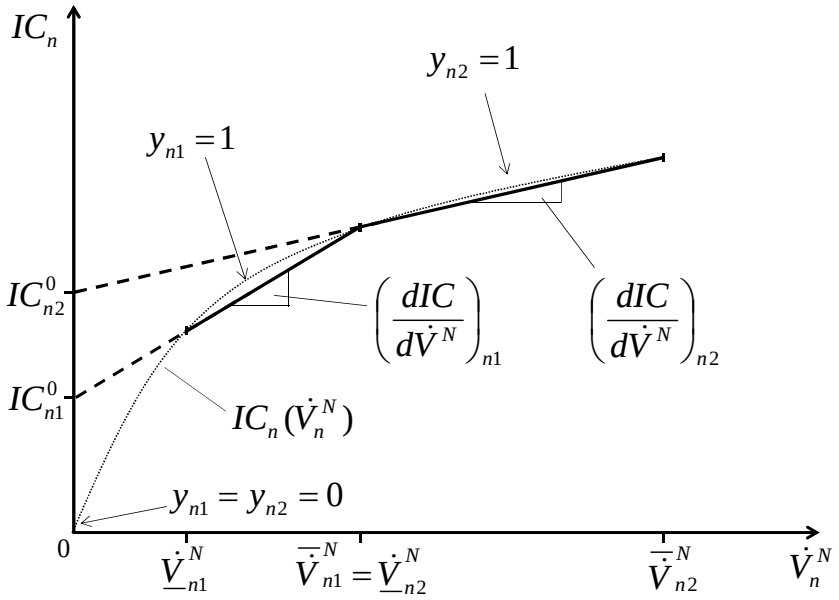

Figure 4.3: Piecewise linear function to linearize nonlinear investment cost degression with $I \in \{1, 2\}$ (Voll, 2013, adapted).

In Fig. 4.3 the linearization approach by a piecewise linear function is illustrated. The constraints

$$IC_n\left(y_{ni}, \dot{V}_{ni}^N\right) \;=\; \sum_{i \in I} \left(IC_{ni}^0 \cdot y_{ni} \;+\; \left(\frac{\mathrm{d}IC}{\mathrm{d}\dot{V}^N}\right)_{ni} \cdot \dot{V}_{ni}^N \right) \quad \forall\, (n \in N) \tag{4.4}$$

are added to the optimization problem, where IC^0_{ni} and $\left(\frac{\mathrm{d}IC}{\mathrm{d}\dot{V}^N}\right)_{ni}$ denote the intercept and slope of segment i for every component n. The following constraints ensure that if a component exists and the segment i is used, the continuous variable for the capacity will stay within its upper and lower bound $\overline{\dot{V}}^N_{ni}$ and $\underline{\dot{V}}^N_{ni}$, respectively.

$$y_{ni} \cdot \underline{\dot{V}}^N_{ni} \leq \dot{V}^N_{ni} \leq y_{ni} \cdot \overline{\dot{V}}^N_{ni} \quad \forall (n \in N, i \in I) \tag{4.5}$$

It is necessary to ensure that only one of the segments of the piecewise linear function to model the investment cost degression is active. This is ensured by introducing constraints

$$\sum_{i \in I} y_{ni} \leq 1 \quad \forall (n \in N) \quad . \tag{4.6}$$

4.2.4 Part Load Performance

For every unit n the relationship between input power $\dot{U}_{nt}$ and output power $\dot{V}_{nt}$, that is the part load performance characteristic, is

$$\dot{U}_{nt}\left(\dot{V}_{nt}, \dot{V}^N_{ni}\right) = \frac{\dot{V}_{nt}}{\eta_{nt}\left(\dot{V}_{nt}\ , \dot{V}^N_{ni}\right)} \quad \forall (n \in N, t \in T) \quad , \tag{4.7}$$

where $\eta_{nt}\left(\dot{V}_{nt}, \dot{V}^N_{ni}\right)$ is the efficiency of a unit n which depends nonlinearily on the current output power and capacity of the unit. For the case of a chiller the relationship of input to output power is referred to as coefficient of performance ($\mathrm{COP}_{nt}(\dot{V}_{nt}, \dot{V}^N_{ni})$).
To derive a linear formulation of the problem the relationship between output power and input power is approximated by a piecewise linear function for every unit n, where $d \in D$ denotes the d-th segment. Thus, in (4.2) and (4.3) all occurrences of δ_{nt} and $\dot{V}_{nt}$ are replaced by $\sum_{d \in D} \delta_{ndt}$ and $\sum_{d \in D} \dot{V}_{ndt}$, respectively.
In this thesis it is assumed that the part load performance of a unit is valid for a certain type of technology regardless of their respective capacities (Voll, 2013). This leads to equation

$$\dot{U}_{ndt}\left(\dot{V}_{ndt}, \delta_{ndt}, \dot{V}^N_{ni}\right) = \frac{1}{\eta^N_n}\left(u^0_{nd} \cdot \overbrace{\delta_{ndt} \cdot \left(\sum_i \dot{V}^N_{ni}\right)}^{nonlinear} + \left(\frac{\mathrm{d}u}{\mathrm{d}v}\right)_{nd} \cdot \dot{V}_{ndt}\right) \quad , \tag{4.8}$$

to express the performance characteristic, defined for all $n \in N, d \in D, t \in T$. In (4.8) η^N_n is the efficiency of each component at its respective nominal capacity and u^0_{nd}, $\left(\frac{\mathrm{d}u}{\mathrm{d}v}\right)_{nd}$ are intercept and slope of unit n and equation d, respectively, scaled by

the current capacity of the unit, i.e. $v_{nd} = \dot{V}_{ndt} / \sum_i \dot{V}_{ni}, u_{nd} = \dot{U}_{ndt} / \sum_i \dot{V}_{ni}$.
Constraint

$$\underbrace{\delta_{ndt} \cdot \left(\sum_i \dot{V}_{ni}^N \right)}_{nonlinear} \cdot \underline{v}_{nd} \leq \dot{V}_{ndt} \leq \underbrace{\delta_{ndt} \cdot \left(\sum_i \dot{V}_{ni}^N \right)}_{nonlinear} \cdot \overline{v}_{nd} \ , \tag{4.9}$$

defined for all $n \in N, d \in D, t \in T$, is adopted to ensure that $\dot{V}_{ndt}$ stays within its lower and upper limits, $\underline{v}_{nd}$ and $\overline{v}_{nd}$ respectively, when in in operation ($\delta_{ndt} = 1$), or zero if not ($\delta_{ndt} = 0$).
For the case of an absorption chiller the scaled performance characteristic expressed by a piecewise linear function with $d \in \{1, 2\}$ is illustrated in Fig. 4.4. Note that u_{nd} and v_{nd} as relative numbers do not depend on time anymore.

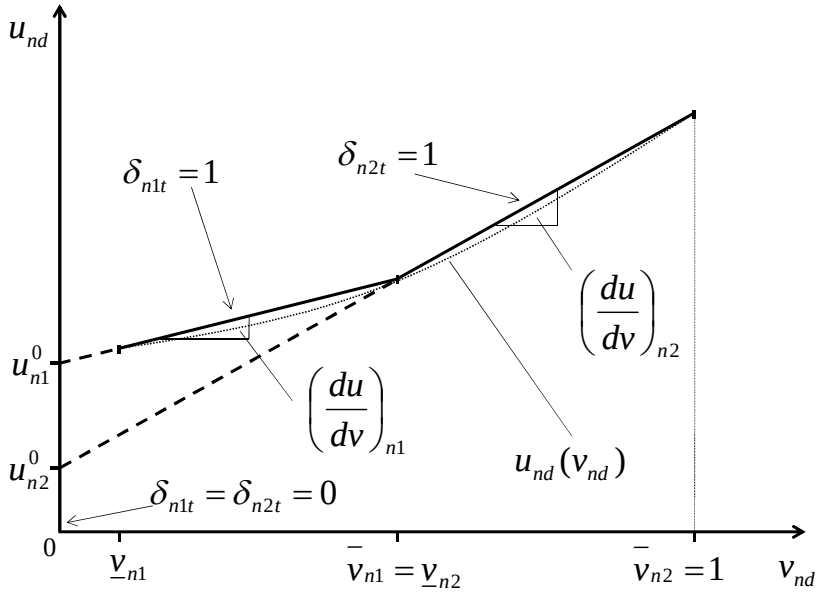

Figure 4.4: Piecewise linear function to linearize part load performance of an absorption chiller ($d \in \{1, 2\}$). Input power and output power scaled by their capacity (Voll, 2013, adapted).

4.2.5 Linearization of Part Load Performance Characteristic

In (4.8) and (4.9) the bilinear term $\delta_{ndt} \cdot \left(\sum_i \dot{V}_{ni}^N \right)$ remains a nonlinearity in the problem. The term is linearized by utilizing the method presented by Glover (1975), which allows the formulation of the problem linearly and to reproduce the behaviour of the nonlinearity perfectly.

The bilinear term $\delta_{ndt} \cdot \left(\sum_i \dot{V}_{ni}^N\right)$ is substituted by an auxiliary continuous variable ξ_{ndt} and the constraints

$$\sum_{i \in I} \dot{V}_{ni}^N + (\delta_{ndt} - 1) \cdot \max_i \left\{\overline{\dot{V}}_{ni}^N\right\} \leq \xi_{ndt} \leq \sum_{i \in I} \dot{V}_{ni}^N \ , \tag{4.10}$$

$$\delta_{ndt} \cdot \min_i \left\{\underline{\dot{V}}_{ni}^N\right\} \leq \xi_{ndt} \leq \delta_{ndt} \cdot \max_i \left\{\overline{\dot{V}}_{ni}^N\right\} \ , \tag{4.11}$$

are introduced, defined for all $n \in N, d \in D, t \in T$. Note that (4.8) and (4.9) change by the substitution, defined for all $n \in N, d \in D, t \in T$, to:

$$\dot{U}_{ndt}\left(\dot{V}_{ndt}, \delta_{ndt}, \xi_{ndt}\right) = \frac{u_{nd}^0}{\eta_n^N} \cdot \xi_{ndt} + \left(\frac{\mathrm{d}u}{\mathrm{d}v}\right)_{nd} \frac{1}{\eta_n^N} \cdot \dot{V}_{ndt} \tag{4.12}$$

$$\xi_{ndt} \cdot \underline{v}_{nd} \leq \dot{V}_{ndt} \leq \xi_{ndt} \cdot \overline{v}_{nd} \tag{4.13}$$

Constraints (4.10, 4.11) guarantee the perfectly correct reproduction of the bilinear term $\delta_{ndt} \cdot \left(\sum_i \dot{V}_{ni}^N\right)$:

- If $\delta_{ndt} = 1$, then by (4.10), $\xi_{ndt} = 1 \cdot \left(\sum_i \dot{V}_{ni}^N\right) = \left(\sum_i \dot{V}_{ni}^N\right)$ and stays within its bounds by (4.11);
- If $\delta_{ndt} = 0$, then by (4.11), $\xi_{ndt} = 0 \cdot \left(\sum_i \dot{V}_{ni}^N\right) = 0$, while (4.10) stays valid.

4.2.6 Linking of Design and Operation

As mentioned previously it is necessary to link design and operation. The constraints

$$\sum_{d \in D} \delta_{ndt} \leq \sum_{i \in I} y_{ni} \quad \forall\, (n \in N, t \in T) \tag{4.14}$$

are introduced to ensure that every unit n is operated at a period of time t, only if it exists. Additionally this constraints ensures that only one segment d of the piecewise linear function to linearize the part load performance of a unit is utilized.

The aforementioned formulation leads to the following optimization problem:

Find	$\dot{V}_{ni}^N$, $\dot{V}_{ndt}$, ξ_{ndt}, δ_{ndt}, y_{ni}	$\forall\, (n \in N, d \in D, i \in I, t \in T)$,
which maximize	(4.2)	
subject to	(4.3) - (4.6), (4.10) - (4.14)	
	$\dot{V}_{ni}^N$, $\dot{V}_{ndt}$, $\xi_{ndt} \geq 0$	$\forall\, (n \in N, d \in D, i \in I, t \in T)$,
	δ_{ndt}, $y_{ni} \in \{0, 1\}$	$\forall\, (n \in N, d \in D, i \in I, t \in T)$.

To enable better reading compact notation

$$c_{nd}^{(0)} = \mathrm{PVF} \cdot \Delta t_t \cdot (-p^U) \cdot \frac{u_{nd}^0}{\eta_n^N} \quad , \tag{4.15}$$

$$c_{nd}^{(1)} = \mathrm{PVF} \cdot \Delta t_t \cdot \left((-p^U) \cdot \left(\frac{\mathrm{d}u}{\mathrm{d}v} \right)_{nd} \frac{1}{\eta_n^N} + p^S \right) \quad , \tag{4.16}$$

$$c_{ni}^{(2)} = IC_{ni}^0 \cdot \left(1 + \mathrm{PVF} \cdot p_n^M \right) \quad , \tag{4.17}$$

$$c_{ni}^{(3)} = \left(\frac{\mathrm{d}IC}{\mathrm{d}\dot{V}^N} \right)_{ni} \cdot \left(1 + \mathrm{PVF} \cdot p_n^M \right) \quad , \tag{4.18}$$

defined for all $(n \in N, d \in D, t \in T, i \in I)$ and

$$\max_{i \in I} \left\{ \dot{V}_{ni}^N \right\} = \overline{\dot{V}}_{ni_{max}}^N \quad \forall \, (n \in N) \quad , \tag{4.19}$$

$$\min_{i \in I} \left\{ \dot{V}_{ni}^N \right\} = \underline{\dot{V}}_{ni_{min}}^N \quad \forall \, (n \in N) \quad , \tag{4.20}$$

are introduced. Considering (4.15) - (4.18) in (4.2) yields the following compact formulation of the objective function:

$$\max \sum_{n \in N} \sum_{t \in T} \sum_{d \in D} \left(c_{nd}^{(0)} \cdot \xi_{ndt} + c_{nd}^{(1)} \cdot \dot{V}_{ndt} \right) - \sum_{n \in N} \sum_{i \in I} \left(c_{ni}^{(2)} \cdot y_{ni} + c_{ni}^{(3)} \cdot \dot{V}_{ni}^N \right) . \tag{4.21}$$

5 Model Analysis and Dantzig-Wolfe Reformulation

In the following chapter the structure of the constraints of the model presented in chapter 4 is analyzed. The analysis focuses on exploring symmetries in the model, which may be exploited by Dantzig-Wolfe reformulation. Two structures suitable for Dantzig-Wolfe decomposition are presented and the respective master and pricing problems are formulated.

5.1 Model Analysis

As presented in the chapter 4 the model formulation yields the MILP

$$(\mathcal{OP}) := \max \sum_{n\in N}\sum_{t\in T}\sum_{d\in D}\left(c^{(0)}_{nd}\cdot\xi_{ndt} + c^{(1)}_{nd}\cdot\dot{V}_{ndt}\right) - \sum_{n\in N}\sum_{i\in I}\left(c^{(2)}_{ni}\cdot y_{ni} + c^{(3)}_{ni}\cdot\dot{V}^N_{ni}\right) \quad (5.1)$$

$$s.t.$$

$$\sum_{n\,\in\, N}\sum_{d\in D}\dot{V}_{ndt} = \dot{E}_t \quad \forall\, t\in T \quad (5.2)$$

$$y_{ni}\cdot\underline{\dot{V}}^N_{ni} \le \dot{V}^N_{ni} \le y_{ni}\cdot\overline{\dot{V}}^N_{ni} \quad \forall\,(n\in N, i\in I) \quad (5.3)$$

$$\sum_{i\in I} y_{ni} \le 1 \quad \forall\,(n\in N) \quad (5.4)$$

$$\sum_{d\in D}\delta_{ndt} \le \sum_{i\in I} y_{ni} \quad \forall\,(n\in N, t\in T) \quad (5.5)$$

$$\sum_{i\in I}\dot{V}^N_{ni} + (\delta_{ndt} - 1)\cdot\overline{\dot{V}}^N_{ni_{max}} \le \xi_{ndt} \le \sum_{i\in I}\dot{V}^N_{ni} \quad \forall(n\in N, d\in D, t\in T) \quad (5.6)$$

$$\xi_{ndt}\cdot\underline{\dot{V}}_{nd} \le \dot{V}_{ndt} \le \xi_{ndt}\cdot\overline{\dot{V}}_{nd} \quad \forall\,(n\in N, d\in D, t\in T) \quad (5.7)$$

$$\delta_{ndt}\cdot\underline{\dot{V}}^N_{ni_{min}} \le \xi_{ndt} \le \delta_{ndt}\cdot\overline{\dot{V}}^N_{ni_{max}} \quad \forall\,(n\in N, d\in D, t\in T) \quad (5.8)$$

$$\delta_{ndt},\, y_{ni} \in \{0,1\} \quad \forall\,(n\in N, d\in D, \ldots \quad (5.9)$$

$$i\in I, t\in T)\ ,$$

with notation (4.19, 4.20), which is termed *original problem* of the optimization of synthesis, design and operation of a decentralized energy supply system and called $(\mathcal{OP})$.

5.1.1 Decomposition by Components

Consider constraints (5.2) as coupling constraints and constraints (5.3) - (5.9) as one part which may be decomposed for each component n, then problem ($\mathcal{OP}$) has a bordered block-diagonal structure. An interpretation for this decomposition is that the design and operation of a component n is independent from other components. All components are linked by the energy balance and, together, have to fulfil the energy demand at every period of time. Hence, a subproblem for each component may be specified and solved independently. Each subproblem is much smaller than the overall problem and may be solved efficiently.
To describe the embedded structures, problem ($\mathcal{OP}$) is expressed using vectors and matrices for variables and constraints. All variables are expressed as vector

$$\mathbf{x} = ((\mathbf{x}^1)^T, (\mathbf{x}^2)^T, (\mathbf{x}^3)^T, ..., (\mathbf{x}^{|N|})^T)^T \tag{5.10}$$

with,

$$\begin{aligned}\mathbf{x}^n = (&\xi_{n11}, ..., \xi_{n|D||T|},\\ &\dot{V}_{n11}, ..., \dot{V}_{n|D||T|},\\ &\dot{V}^N_{n1}, ..., \dot{V}^N_{n|I|},\\ &\delta_{n11}, ..., \delta_{n|D||T|},\\ &y_{n1}, ..., y_{n|I|})^T \qquad \forall\, n \in N\ .\end{aligned} \tag{5.11}$$

Then ($\mathcal{OP}$) is rewritten as

$$\begin{aligned} \max \quad & \mathbf{c}^T\mathbf{x} & & \\ & \text{s.t.} & & \\ & \mathbf{A}\mathbf{x} & = \quad & \mathbf{b} \\ \sum_{n\in N} & \mathbf{D}^n\mathbf{x}^n & \leq \quad & \mathbf{d} \\ & \mathbf{x} & \in \quad & X = X^1 \times \cdots \times X^N \qquad \forall\, n \in N\ , \end{aligned} \tag{5.12}$$

with matrices and vectors of appropriate dimensions and

$$\begin{aligned} X^n = \Bigg\{ \mathbf{x}^n = \begin{pmatrix} \mathbf{x}^{\mathbb{Q}} \\ \mathbf{x}^{\mathbb{Z}} \end{pmatrix} | \, \mathbf{x}^{\mathbb{Q}} \in \mathbb{Q}^{\alpha}_{+}, \, \mathbf{x}^{\mathbb{Z}} \in \mathbb{Z}^{\beta}_{+}, ... \\ \alpha = (2 \cdot |D| \cdot |T| + |I|), \, \beta = (|D| \cdot |T| + |I|) \Bigg\}, \forall\, n \in N\ . \end{aligned} \tag{5.13}$$

The problem has a bordered block diagonal structure. The border $\mathbf{Ax} = \mathbf{b}$ results from the energy balances (5.2) which link all components at every period of time. Constraints $\mathbf{D}^n\mathbf{x}^n \leq \mathbf{d}$ correspond to (5.3) - (5.9) which have a block diagonal form, where one block corresponds to one energy conversion unit. These blocks are identical in the case of identical units considered and then may be aggregated (see section 3.2.4, p. 14).
Fig. 5.1 illustrates the identified structure of the problem. The figure displays the

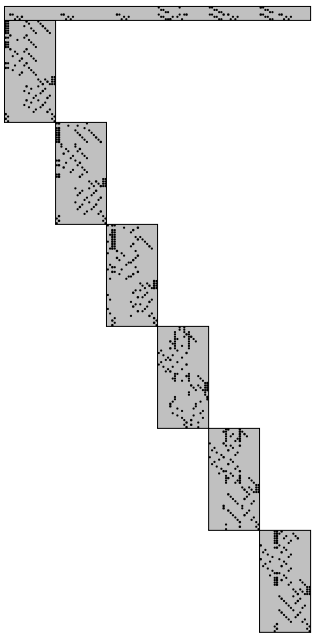

Figure 5.1: Structure of the constraint matrix of problem $\mathcal{OP}$. Constraints sorted to illustrate the decomposition emphasizing components. Exemplary instance with three boilers and three absorption chillers in the initial superstructure. Three periods of time are considered. Generated with `GCG`. Note that block sizes do not correspond to the notation in (5.12).

incidence matrix for all constraints and variables in ($\mathcal{OP}$). Note that block sizes do not correspond to the notation in (5.12). The horizontal border displayed at the top of the constraint matrix represents the energy balances which link all components for every period of time. For every component a subproblem may identified.

5.1.2 Decomposition by Design and Operation

The constraint matrix of problem ($\mathcal{OP}$) has another embedded structure which may be exploited by Dantzig-Wolfe reformulation. To examine this structure, vectors

$$\mathbf{x} = ((\mathbf{x}^1)^T, (\mathbf{x}^2)^T, (\mathbf{x}^3)^T, ..., (\mathbf{x}^{|T|})^T)^T \ , \tag{5.14}$$

with

$$\begin{aligned} \mathbf{x}^t = (&\xi_{11t}, ..., \xi_{|N||D|t}, \\ &\dot{V}_{11t}, ..., \dot{V}_{|N||D|t}, \\ &\delta_{11t}, ..., \delta_{|N||D|t})^T \qquad \forall\, t \in T \ , \end{aligned} \tag{5.15}$$

and

$$\begin{aligned} \mathbf{y} = &\dot{V}^N_{11}, ..., \dot{V}^N_{|N||I|}, \\ &y_{11}, ..., y_{|N||I|})^T \ , \end{aligned} \tag{5.16}$$

are introduced. Considering vectors $\mathbf{x}$ and $\mathbf{y}$ problem ($\mathcal{OP}$) may be rewritten as

$$\begin{aligned} \max \quad & \mathbf{c}_1^T\mathbf{x} + \mathbf{c}_2^T\mathbf{y} & & \\ \text{s.t.} \quad & \mathbf{By} + \mathbf{0x} & \leq \quad & \mathbf{b} \\ & \mathbf{Fy} + \sum_{t\in T} \mathbf{D}^t\mathbf{x}^t & \leq \quad & \mathbf{d} \\ & \mathbf{x} & \in \quad & X = X^1 \times \cdots \times X^T \\ & \mathbf{y} & \in \quad & Y \end{aligned} \tag{5.17}$$

with matrices and vectors of appropriate dimensions and $\mathbf{0}$ as a matrix with only zeros. The sets X^t and Y are defined by

$$X^t = \left\{ \mathbf{x}^t = \begin{pmatrix} \mathbf{x}^{\mathbb{Q}} \\ \mathbf{x}^{\mathbb{Z}} \end{pmatrix} | \, \mathbf{x}^{\mathbb{Q}} \in \mathbb{Q}_+^{\gamma}, \mathbf{x}^{\mathbb{Z}} \in \mathbb{Z}_+^{\theta}, \ldots \right.$$
$$\left. \gamma = 2 \cdot |N| \cdot |D|, \theta = |N| \cdot |D| \right\} \forall\, t \in T \ , \tag{5.18}$$

$$Y = \left\{ \mathbf{y} \in \mathbb{Z}_+^{2\cdot|N|\cdot|I|} \right\} \ . \tag{5.19}$$

Rewriting problem ($\mathcal{OP}$) in (5.17) shows that the constraint matrix of the problem has an arrowhead structure, i.e. it has a horizontal and a vertical border.

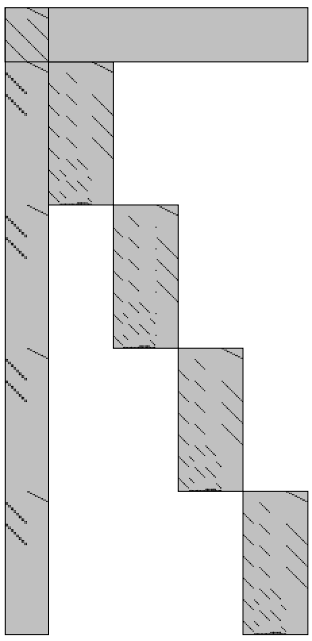

Figure 5.2: Structure of the constraint matrix of problem ($\mathcal{OP}$). Constraints sorted to illustrate the decomposition emphasizing design and operation. Exemplary instance with four boilers and four absorption chillers considered in the initial superstructure. Four periods of time considered. Generated with `GCG`. Note that block sizes do not correspond to notation in (5.17).

The constraint matrices are displayed in Fig. 5.2, where block sizes do not correspond to the notation in (5.17). The horizontal border is caused by coupling constraints (5.3) and (5.4) which do not depend on the period of time. These constraints limit the existence and capacity of every unit n. If capacity and existence are fixed the units may be operated at every period of time independently. The matrix expression $\mathbf{Fy} + \mathbf{D}^t\mathbf{x}^t \leq \mathbf{d}$ corresponds to constraints (5.2) and (5.5) - (5.9). They do not fully decompose at every period of time as they are linked by variables $\mathbf{y}$, hence the vertical border in Fig. 5.2. These are exactly the variables to describe the existence of a unit and its capacity. However, this drawback may be eliminated by introducing copy constraints (see section 3.2.5).

5.2 Dantzig-Wolfe Reformulation Emphasizing Components

Problem ($\mathcal{OP}$) is reformulated in the discretization version for mixed-integer linear programs as, in the case of identical units, the subproblems may be aggregated. The reformulation of a mixed-integer linear program in the discretization version is presented by Vanderbeck and Savelsbergh (2006) and is applied to problem ($\mathcal{OP}$) in this thesis. In this approach continuous variables are convexified whereas integer variables are discretized.
A finite set P^n for every subproblem n in (5.3) - (5.9) is defined and called *generating set* containing p *generators*. A generator is a solution of every subproblem n, where an integer point is obtained for every integer variable and at least one or more extreme points for every continuous variable. In contrast to discretization on a pure integer set, performing discretization on mixed-integer sets requires to impose integrality on a sum of master variables. In particular on master variables which have the same integer values in the coefficient columns. A unifying framework for both convexification, discretization of integer and mixed-integer problems is presented by Vanderbeck and Savelsbergh (2006).
To keep notation tractable, the LP relaxation of the aggregated master problem is presented. Reformulating ($\mathcal{OP}$) by introducing

$$\dot{V}_{ndt} = \sum_{p \in P^n} \dot{V}_{pndt} \lambda_p^n \ , \tag{5.20}$$

$$\dot{V}_{ni}^N = \sum_{p \in P^n} \dot{V}_{pni}^N \lambda_p^n \ , \tag{5.21}$$

$$\xi_{ndt} = \sum_{p \in P^n} \xi_{pndt} \lambda_p^n \ , \tag{5.22}$$

$$\delta_{ndt} = \sum_{p \in P^n} \delta_{pndt} \lambda_p^n \ , \tag{5.23}$$

$$y_{ni} = \sum_{p \in P^n} y_{pni} \lambda_p^n \ , \tag{5.24}$$

defined for all $n \in N, d \in D, t \in T, i \in I$, the LP relaxation of the aggregated integer master problem of the reformulation emphasizing components for an arbitrary set P^1 takes the form

$$\max \sum_{p \in P^1} \left(\sum_{t \in T} \sum_{d \in D} \left(c_{1d}^{(0)} \cdot \xi_{p1dt} + c_{1d}^{(1)} \cdot \dot{V}_{p1dt} \right) - \sum_{i \in I} \left(c_{1i}^{(2)} \cdot y_{p1i} + c_{1i}^{(3)} \cdot \dot{V}_{p1i}^{N} \right) \right) \cdot \nu_p \tag{5.25}$$

s.t.

$$\sum_{p \in P^1} \sum_{d \in D} \dot{V}_{p1dt} \cdot \nu_p \;=\; \dot{E}_t \quad \forall\, t \in T \qquad [\pi_t] \tag{5.26}$$

$$\sum_{p \in P^1} \nu_p \;=\; |N| \qquad [\pi_0] \tag{5.27}$$

$$\boldsymbol{\nu} \;\geq\; 0 \tag{5.28}$$

with the aggregated master variable $\nu_p = \sum_{n \in N} \lambda_p^n$. No extreme rays occur as all variables are bounded. The corresponding pricing problem is defined by

$$\max \sum_{t \in T} \sum_{d \in D} \left(c_{1d}^{(0)} \cdot \xi_{1dt} + (c_{1d}^{(1)} \;-\; \pi_t) \cdot \dot{V}_{1dt} \right) - \sum_{i \in I} \left(c_{1i}^{(2)} \cdot y_{1i} + c_{1i}^{(3)} \cdot \dot{V}_{1i}^{N} \right) - \pi_0 \tag{5.29}$$

s.t.

$$y_{1i} \cdot \underline{\dot{V}}_{1i}^{N} \;\leq\; \dot{V}_{1i}^{N} \;\leq\; y_{1i} \cdot \overline{\dot{V}}_{1i}^{N} \quad \forall\, (i \in I) \tag{5.30}$$

$$\sum_{i \in I} y_{1i} \;\leq\; 1 \tag{5.31}$$

$$\sum_{d \in D} \delta_{1dt} \;\leq\; \sum_{i \in I} y_{1i} \quad \forall\, (t \in T) \tag{5.32}$$

$$\sum_{i \in I} \dot{V}_{1i}^{N} + (\delta_{1dt} \;-\; 1) \cdot \overline{\dot{V}}_{1i_{max}}^{N} \;\leq\; \xi_{1dt} \;\leq\; \sum_{i \in I} \dot{V}_{1i}^{N} \quad \forall (d \in D, t \in T) \tag{5.33}$$

$$\xi_{1dt} \cdot \underline{\dot{V}}_{1d} \;\leq\; \dot{V}_{1dt} \;\leq\; \xi_{1dt} \cdot \overline{\dot{V}}_{1d} \quad \forall\, (d \in D, t \in T) \tag{5.34}$$

$$\delta_{1dt} \cdot \underline{\dot{V}}_{1i_{min}}^{N} \;\leq\; \xi_{1dt} \;\leq\; \delta_{1dt} \cdot \overline{\dot{V}}_{1i_{max}}^{N} \quad \forall\, (d \in D, t \in T) \tag{5.35}$$

$$\delta_{1dt},\, y_{1i} \;\in\; \{0,1\} \quad \forall\, (d \in D, i \in I, t \in T)\ , \tag{5.36}$$

Where π_t, $\forall\, t \in T$ are the corresponding dual variables of constraints (5.26) and π_0 is the dual variable of the aggregated convexity constraint (5.27). By constraints (5.36) the above stated pricing problem is still a mixed-integer linear program, which has to be solved at every iteration of the column generation algorithm. Depending on the size of the problem considered this may cause longer solution times compared to problems where the pricing problem can be solved with a tailored combinatorial algorithm.

5.3 Dantzig-Wolfe Reformulation Emphasizing Design and Operation

For this reformulation the original variables are expressed as convex combination of $p \in P$ extreme points of $conv(X)$

$$\dot{V}_{ndt} = \sum_{p \in P} \dot{V}_{pndt} \lambda_p \ , \tag{5.37}$$

$$\dot{V}^N_{ni} = \sum_{p \in P} \dot{V}^N_{pni} \lambda_p \ , \tag{5.38}$$

$$\xi_{ndt} = \sum_{p \in P} \xi_{pndt} \lambda_p \ , \tag{5.39}$$

$$\delta_{ndt} = \sum_{p \in P} \delta_{pndt} \lambda_p \ , \tag{5.40}$$

$$y_{ni} = \sum_{p \in P} y_{pni} \lambda_p \ , \tag{5.41}$$

$$\sum_{p \in P} \lambda_p = 1 \ , \tag{5.42}$$

$$\lambda_p \geq 0, \quad \forall\, p \in P \ , \tag{5.43}$$

defined for all $n \in N, d \in D, t \in T, i \in I$. And the set X defined as

$$X = \left\{ \dot{V}_{ndt},\, \dot{V}^N_{ni},\, \xi_{ndt} \in \mathbb{R}_+,\, \delta_{ndt},\, y_{ni} \in \mathbb{Z}_+ \,|\, (5.2),\, (5.5) - (5.9) \right\} \ , \tag{5.44}$$

defined for all ($n \in N$, $i \in I, d \in D, t \in T$). Introducing (5.37) - (5.43) in ($\mathcal{OP}$) a extended formulation

$$\max \sum_{p \in P} \sum_{n \in N} \left(\sum_{t \in T} \sum_{d \in D} \left(c^{(0)}_{nd} \cdot \xi_{pndt} + c^{(1)}_{nd} \cdot \dot{V}_{pndt} \right) - \sum_{i \in I} \left(c^{(2)}_{ni} \cdot y_{pni} + c^{(3)}_{ni} \cdot \dot{V}^N_{pni} \right) \right) \cdot \lambda_p \tag{5.45}$$

$$s.t.$$

$$\sum_{p\in P}\sum_{i\in I} y_{pni}\cdot\lambda_p \leq 1 \quad \forall n\in N \qquad [\pi_n^{(1)}] \tag{5.46}$$

$$\sum_{p\in P}\left(y_{pni}\cdot\underline{\dot{V}}_{ni}^{N} - \dot{V}_{pni}^{N}\right)\cdot\lambda_p \leq 0 \quad \forall\,(n\in N, i\in I) \qquad [\pi_{ni}^{(2)}] \tag{5.47}$$

$$\sum_{p\in P}\left(-y_{pni}\cdot\overline{\dot{V}}_{ni}^{N} + \dot{V}_{pni}^{N}\right)\cdot\lambda_p \leq 0 \quad \forall\,(n\in N, i\in I) \qquad [\pi_{ni}^{(3)}] \tag{5.48}$$

$$\sum_{p\in P}\lambda_p = 1 \qquad [\pi_0] \tag{5.49}$$

$$\lambda_p \geq 0 \quad \forall p\in P \tag{5.50}$$

$$\sum_{p\in P}\dot{V}_{pndt}\lambda_p = \dot{V}_{ndt}\ , \tag{5.51}$$

$$\sum_{p\in P}\dot{V}_{pni}^{N}\lambda_p = \dot{V}_{ni}^{N}\ , \tag{5.52}$$

$$\sum_{p\in P}\xi_{pndt}\lambda_p = \xi_{ndt}\ , \tag{5.53}$$

$$\sum_{p\in P}\delta_{pndt}\lambda_p = \delta_{ndt}\ , \tag{5.54}$$

$$\sum_{p\in P} y_{pni}\lambda_p = y_{ni}\ , \tag{5.55}$$

$$y_{ni},\ \delta_{ndt} \in \{0,1\} \tag{5.56}$$

is obtained, where $\pi_n^{(1)}$, $\pi_{ni}^{(2)}, \pi_{ni}^{(3)}$ and π_0 are the dual variables of their respective constraints defined for all $n \in N, i \in I$. The corresponding pricing problem is defined by

$$\max \sum_{n\in N}\sum_{t\in T}\sum_{d\in D}\left(c_{nd}^{(0)}\cdot\xi_{ndt} + c_{nd}^{(1)}\cdot\dot{V}_{ndt}\right) - \sum_{n\in N}\sum_{i\in I}\left(\overline{c}_{ni}^{(2)}\cdot y_{ni} + \overline{c}_{ni}^{(3)}\cdot\dot{V}_{ni}^{N}\right) - \pi_0 \tag{5.57}$$

$$\overline{c}_{ni}^{(2)} = c_{ni}^{(2)} - \pi_n^{(1)}\cdot(1) - \pi_{ni}^{(2)}\cdot(\underline{\dot{V}}_{ni}^{N}) - \pi_{ni}^{(3)}\cdot(-\overline{\dot{V}}_{ni}^{N}) \tag{5.58}$$

$$\overline{c}_{ni}^{(3)} = c_{ni}^{(3)} - \pi_{ni}^{(2)}\cdot(-1) - \pi_{ni}^{(3)}\cdot(1) \tag{5.59}$$

s.t.

$$\sum_{n\in N}\sum_{d\in D}\dot{V}_{ndt} = \dot{E}_t \quad \forall\, t\in T \tag{5.60}$$

$$\sum_{d\in D}\delta_{ndt} \leq \sum_{i\in I} y_{ni} \quad \forall\,(n\in N, t\in T) \tag{5.61}$$

$$\sum_{i\in I}\dot{V}_{ni}^{N} + (\delta_{ndt} - 1)\cdot\overline{\dot{V}}_{ni_{max}}^{N} \leq \xi_{ndt} \leq \sum_{i\in I}\dot{V}_{ni}^{N} \quad \forall(n\in N, d\in D, t\in T) \tag{5.62}$$

$$\xi_{ndt}\cdot\underline{\dot{V}}_{nd} \leq \dot{V}_{ndt} \leq \xi_{ndt}\cdot\overline{\dot{V}}_{nd} \quad \forall\,(n\in N, d\in D, t\in T) \tag{5.63}$$

$$\delta_{ndt}\cdot\underline{\dot{V}}_{ni_{min}}^{N} \leq \xi_{ndt} \leq \delta_{ndt}\cdot\overline{\dot{V}}_{ni_{max}}^{N} \quad \forall\,(n\in N, d\in D, t\in T) \tag{5.64}$$

$$\delta_{ndt},\ y_{ni} \in \{0,1\} \quad \forall\,(n\in N, d\in D, \ldots \\ i\in I, t\in T)\ . \tag{5.65}$$

Note that variables y_{ni} and $\dot{V}_{ni}^{N}$ may be considered as *linking variables*, as they do not depend on t. An interesting fact is that problem (5.57) - (5.65) is a mixed-integer linear program. This may be a disadvantage as it has to be solved at every iteration of the column generation algorithm.

6 Numerical Study

A numerical study is conducted to evaluate whether the proposed reformulations are beneficial in solving the described optimization problem. The evaluation of the results aims to elucidate if the accuracy of the model and/or the size of the investigated energy supply system may be increased in terms of the number of units in the superstructure and/or the number of periods of time considered. To achieve this a set of instances is defined with varying degrees of complexity, in terms of the number of units in the superstructure and/or the number of periods of time considered.
The instances are solved by the branch-and-cut solvers `SCIP` 3.1.0.1 (Achterberg, 2009) and `CPLEX` 12.5.0 (IBM, 2011) and by the branch-and-price solver `GCG` 2.0.0 (Gamrath and Lübbecke, 2010). For each instance a model is implemented using the GAMS modeling language (Rosenthal, 2014). These models are exported as CPLEX LP[1] files and each problem is solved with the different solvers. As `GCG` undertakes the Dantzig-Wolfe reformulation automatically, the problem file and a corresponding text file containing the decomposition are passed to `GCG`.
The numerical experiments are conducted on a DELL™ Optiplex 990 desktop PC with an Intel™ i7-2600 CPU with 3400 Mhz and 16 GB memory. The operating system is the Linux based openSUSE 13.1 64-bit. All calculations are obtained with the solver settings at default values and by using only one thread of the multicore CPU. The aggregation of subproblems in MILPs as original problems is still under development in `GCG`. First results presented in this chapter utilize a prerelease version of `GCG` if subproblems are aggregated.
The following chapter presents firstly the definition and characterization of the considered instances; followed by the results of solving the LP relaxation in the rootnode of all instances, in order to evaluate the strength of the initial problem formulation; and lastly results of the optimal solution of the instances.

6.1 Instance Definition

To conduct a numerical study a fictional energy supply system is assumed. The superstructure of the system is depicted in Fig. 6.1 and contains $n \in N = B \cup A$ energy conversion units. To leave solution times at a reasonable level only two types

[1] CPLEX LP format encodes LPs and MILPs. It was developed by CPLEX Optimization Inc., see description for example in http://plato.asu.edu/cplex_lp.pdf

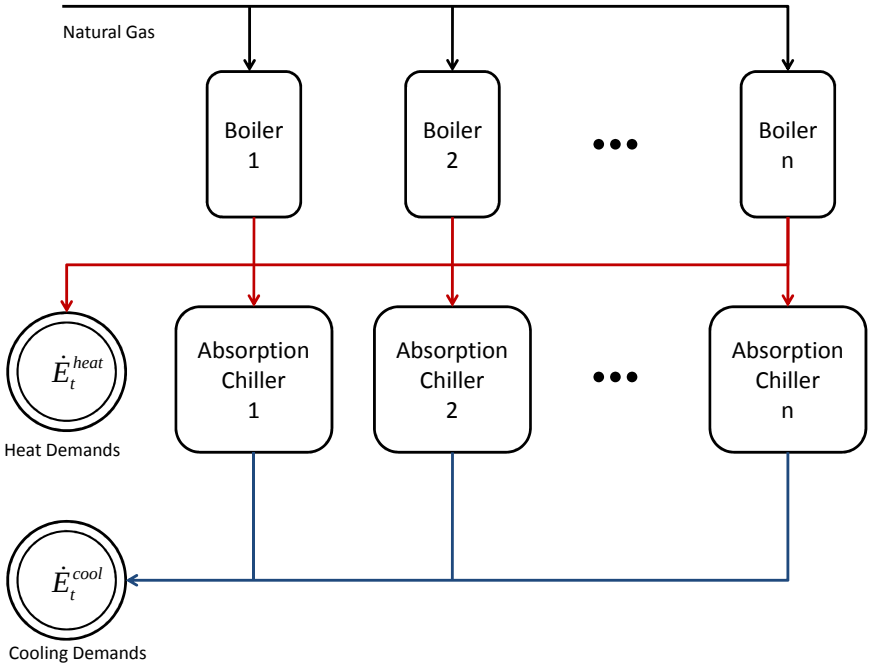

Figure 6.1: Scheme of the superstructure of the considered fictional energy supply system. A set of $n \in B$ identical boilers supply heat to the absorption chillers (ACs) and to fulfil the heating demand $\dot{E}_t^{heat}$. A set of $n \in A$ ACs converts heat into cooling and fulfils the cooling demand $\dot{E}_t^{cool}$.

of energy conversion units are considered: $n \in B$ boilers to provide heat and $n \in A$ absorption chillers (ACs) to provide cooling. The boilers convert natural gas into heat to fulfil the heating demand and to power the ACs. ACs convert heat into cooling and fulfil the cooling demand. The part load behaviour of each component is taken into account by a piecewise linear function. The general model parameters and the parameters of the part load behaviour are taken from Voll (2013) and listed in the appendix on p. 65. Some characteristic data of the considered energy conversion units is presented in Tab. 6.1.
The heating and cooling demands are calculated from hourly data provided by the chair of Technical Thermodynamics. The data is derived from real world measurements in an existing energy supply system. The data set contains a spatially aggregated heating and cooling demand for every hour of a representative year. The energy demands are discretized by the number of periods of time considered, i.e. if t periods are considered demands will be calculated from the hourly data with a constant value during each period. The length of each period is calculated in a way that the total amount of energy required remains constant if the discretized and the hourly demands are summed up over the regarded year. The hourly data is presented in Fig. B.1 and Fig. B.2, p. 68.
For the specific instances the general model formulation in chapter 4, p. 23 needs to be adapted. The generic energy balance (4.3) is removed from the problem and an

energy balance (6.1) for heating and (6.2) for cooling is adopted instead.

$$\sum_{n \in B} \sum_{d \in D} \left(\dot{V}_{ndt} \right) - \sum_{n \in A} \sum_{d \in D} \left(c_{nd}^{(0)} \cdot \xi_{ndt} + c_{nd}^{(1)} \cdot \dot{V}_{ndt} \right) = \dot{E}_t^{heat} \quad \forall\, t \in T \tag{6.1}$$

$$- \sum_{n \in A} \sum_{d \in D} \left(\dot{V}_{ndt} \right) = \dot{E}_t^{cool} \quad \forall\, t \in T \tag{6.2}$$

Also the objective function needs a modification as only boilers consume primary energy. The objective function (4.21) is changed to:

$$\max \sum_{n \in B} \sum_{t \in T} \sum_{d \in D} \left(c_{nd}^{(0)} \cdot \xi_{ndt} + c_{nd}^{(1)} \cdot \dot{V}_{ndt} \right) - \sum_{n \in N} \sum_{i \in I} \left(c_{ni}^{(2)} \cdot y_{ni} + c_{ni}^{(3)} \cdot \dot{V}_{ni}^{N} \right). \tag{6.3}$$

The investigations are undertaken for three, four and five identical boilers and absorption chillers each and $t = \{1, ..., 7\}$ periods of time. The number of equations and variables of the instances are presented in Tab. 6.2 - 6.4. All parameters may be found in the appendix (see section B.1, p. 67). The following naming convention is introduced for all instances considered. All instances are named `n<y>t<x>`, where `<y>` refers to the total number of energy conversion units and `<x>` refers to the number of periods of time considered in the problem.

Table 6.1: Characteristic data of components boiler and absorption chiller (AC) (Voll, 2013), IC_n - Investment cost of component n, η_n^N - Efficiency at nominal power of component n.

Technology	**Parameter**			
	Thermal power range in MW	Investment cost in €×10^3	Maintenance cost % of IC_n	η_n^N -
Boiler	0.10 - 14.0	34 - 380	15	0.90
AC	0.05 - 6.50	75 - 520	1	0.67

Table 6.2: Number of constraints, binary variables and continuous variables and the total number of variables of instances `n6t<x>`, `x` $\in \{1, ..., 7\}$.

Instance	**Constraints**	**Binary**	**Continuous**	**Total**
`n6t1`	110	24	36	60
`n6t2`	190	36	60	96
`n6t3`	270	48	84	132
`n6t4`	350	60	108	168
`n6t5`	430	72	132	204
`n6t6`	510	84	156	240
`n6t7`	590	96	180	276

Table 6.3: Number of constraints, binary variables and continuous variables and the total number of variables of instances n8t<x>, x ∈ {1, ..., 7}.

Instance	Constraints	Binary	Continuous	Total
n8t1	146	32	48	80
n8t2	252	48	80	128
n8t3	358	64	112	176
n8t4	464	80	144	224
n8t5	570	96	176	272
n8t6	676	112	208	320
n8t7	782	128	240	368

Table 6.4: Number of constraints, binary variables and continuous variables and the total number of variables of instances n10t<x>, x ∈ {1, ..., 7}.

Instance	Constraints	Binary	Continuous	Total
n10t1	182	40	60	100
n10t2	314	60	100	160
n10t3	446	80	140	220
n10t4	578	100	180	280
n10t5	710	120	220	340
n10t6	842	140	260	400
n10t7	974	160	300	460

6.2 Results and Discussion

In the following two sections the results of the conducted numerical study are presented and discussed. These results allow the evaluation of the quality of the reformulations and optimal solution times. To solve the defined instances the following solvers are considered and abbreviations used are introduced:

- The branch-and-cut solver CPLEX (IBM, 2011).
- The branch-and-cut solver SCIP (Achterberg, 2009).
- The branch-and-price solver GCG with the reformulation emphasizing components (GCG n) (see section 5.2, p. 37).
- The branch-and-price solver GCG with the reformulation emphasizing components and aggregation of identical pricing problems (GCG n AGG) (see section 5.2, p. 37).
- The branch-and-price solver GCG with the reformulation emphasizing design and operation (GCG t) (see section 5.3, p. 39).

6.2.1 Evaluation of Formulation Strength

To evaluate the strength of the model formulation, the results of the solution of the instances in the rootnode are compared for the different solvers utilized. As the dual bound of `GCG n` and `GCG n AGG` are equal in the rootnode only the results of `GCG n` are presented. To measure the quality of a formulation

$$\text{GAP} = \frac{z^* - z_{root}^{DB}}{z_{root}^{DB}} \tag{6.4}$$

is defined, with z^* as the optimal objective value of the instance and z_{root}^{DB} the value of the dual bound of the rootnode solution, i.e. the solution value of the LP relaxation in the rootnode. The difference of both, scaled by the dual bound in the rootnode, gives a measure to quantify the *relative distance* to the optimal value. If this distance is low for one solver compared to another solver on the same instance, then it is likely that this solver converges faster to the optimal solution.
In Fig. 6.2 - 6.4 the results for the measure GAP for the rootnode solutions of all instances utilizing the different solvers are presented.

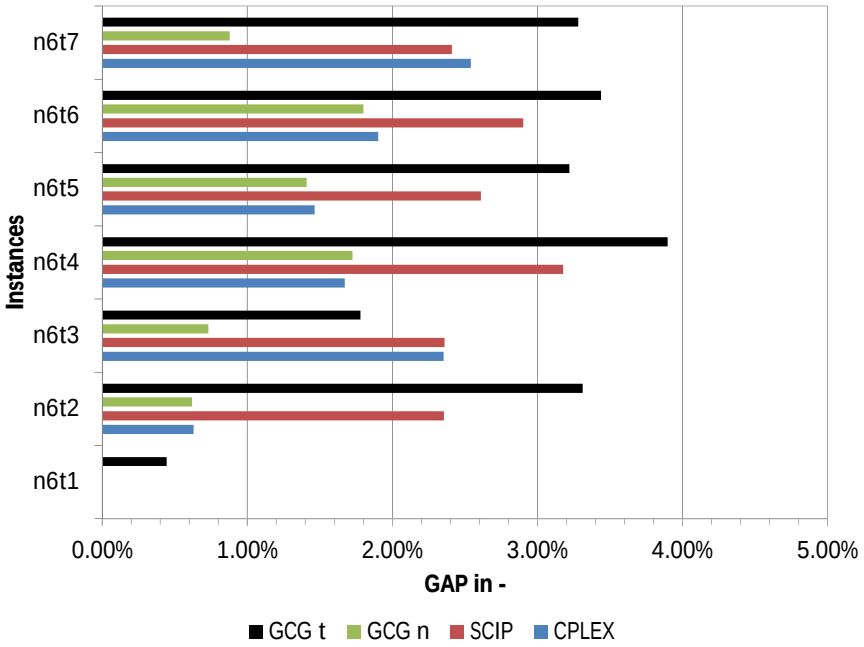

Figure 6.2: Results for the measure GAP of the rootnode solution as defined in (6.4) for instances `n6tx`, `x` $\in \{1, ..., 7\}$. Results for different solvers and decompositions. 'GCG n' - Solver `GCG` with decomposition by emphasizing components, 'GCG t' - Solver `GCG` with decomposition by emphasizing design and operation.

The minimum and maximum value, the value of the median and the 0.75- and 0.25-quantile values for the results of the different solvers on the set of instances are calculated and presented in Tab. 6.5.

The results show that in most cases the lowest value for GAP is obtained by utilizing GCG and the decomposition by components (GCG n). The value of the median of solver GCG n, 0.77 %, calculated from the values for all instances is the lowest of all solvers. Thus the reformulation is successful and a tighter formulation of the problem is obtained.

Table 6.5: Maximum, 0.75-quantile, median, 0.25-quantile and minimum values of GAP for all solvers and instances.

Quantile	CPLEX in %	SCIP in %	GCG n in %	GCG t in %
Maximum	3.91	4.51	1.8	7.51
0.75-Quantile	2.72	3.96	1.2	3.61
Median	1.95	3.08	0.77	3.28
0.25-Quantile	0.63	2.35	0.62	2.85
Minimum	0.00	0.00	0.00	0.44

The second best values for GAP are obtained by the branch-and-cut solver CPLEX. For some instances values for GAP are obtained which are in the same range as the values provided by GCG n or even better. The values obtained for the solver SCIP are higher in comparison to GCG n and CPLEX with a median of 3.08 %.
By solving the rootnode of all instances by utilizing the solver GCG with the decomposition by emphasizing design and operation (GCG t) the highest values for GAP in comparison to the other solvers are obtained. This results in a value of 3.28 % of the median, which is slightly higher than the one obtained with the solver SCIP.
As a result for this set of instances the reformulation by emphasizing design and operation has no improving effect on the dual bound obtained by solving the LP relaxation in the rootnode. There is the possibility that this may be different if instances with an increasing number of periods of time are considered, as the decomposition GCG t decomposes partly for every period of time.
An interesting result is that the difference between the values of GAP calculated by utilizing solvers GCG n and CPLEX is higher for instances n10tx compared to differences of instances n6tx (compare in Fig. 6.2 and 6.4). This is a plausible result as symmetry increases when changing from instances n6tx to n10tx which is exploited by GCG n.

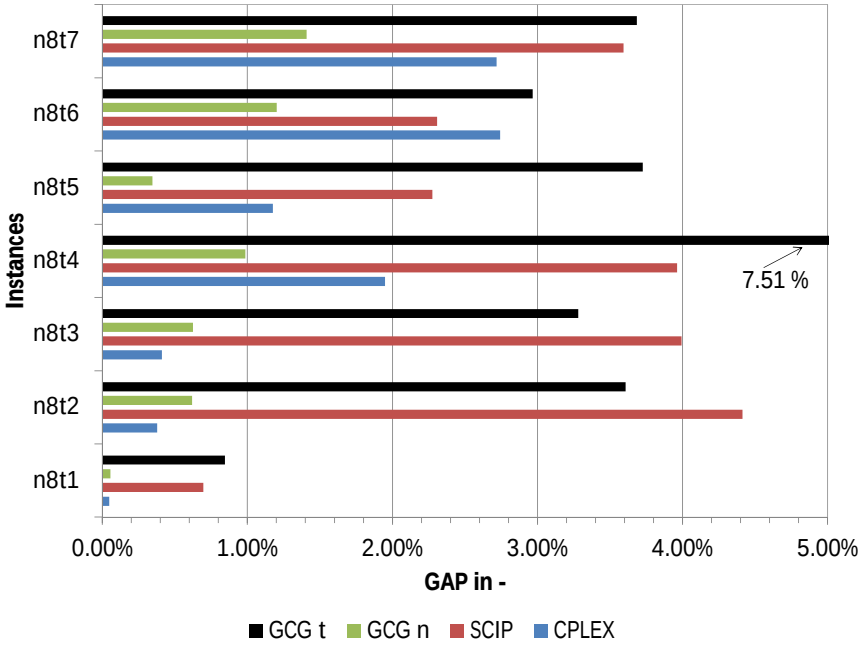

Figure 6.3: Results for the measure GAP of the rootnode solution as defined in (6.4) for instances `n8tx`, `x` $\in \{1, ..., 7\}$. Results for different solvers and decompositions. 'GCG n' - Solver `GCG` with decomposition by emphasizing components, 'GCG t' - Solver `GCG` with decomposition by emphasizing design and operation.

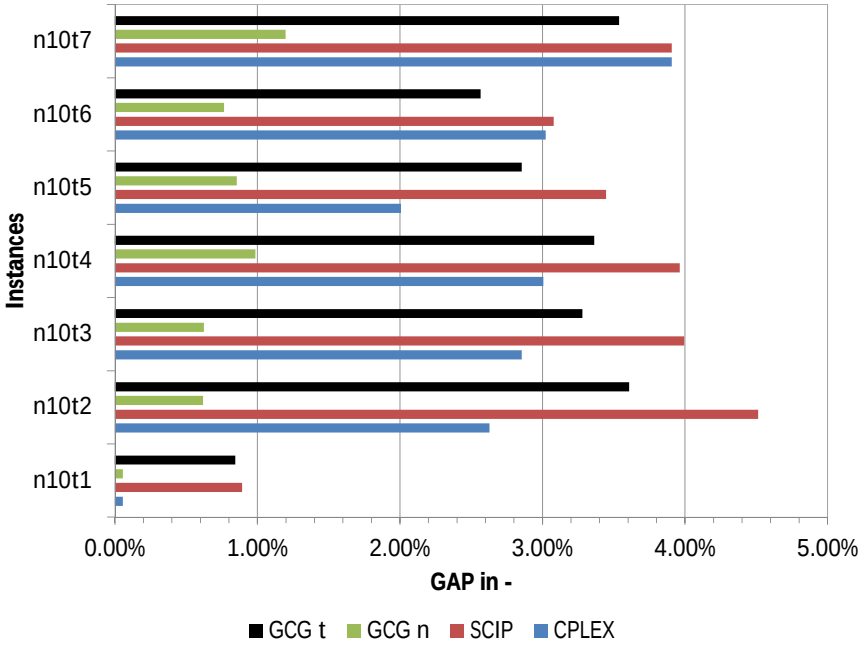

Figure 6.4: Results for the measure GAP of the rootnode solution as defined in (6.4) for instances `n10tx`, `x` $\in \{1, ..., 7\}$. Results for different solvers and decompositions. 'GCG n' - Solver `GCG` with decomposition by emphasizing components, 'GCG t' - Solver `GCG` with decomposition by emphasizing design and operation.

6.2.2 Performance of Solvers

6.2.2.1 Geometric Mean of Solution Times

To compare the performance of the different solution alternatives all solvers are utilized to solve the instances defined in section 6.1 to optimality.
Let $s \in S$ be the set of solvers available for optimization and $m \in M$ the set of instances to be solved to optimality. t^*_{ms} is defined as the computing time required by solver s to solve instance m to optimality. Let

$$\underline{t}^*_s = \left(\prod_{m \in M} t^*_{ms} \right)^{\frac{1}{|M|}} \quad \forall s \in S \tag{6.5}$$

be the geometric mean of solution times for each solver $s \in S$ calculated from the product of solution times t^*_{ms} achieved on instances $m \in M$. The different solvers are compared with respect to their geometric mean $\underline{t}^*_s$. The results are presented in Fig. 6.5 for every solver. The number of solved instances are displayed and the relative speed of a solver compared to the speed of solver `SCIP` are reported. Unsolved or failed instances are accounted for with the time limit of 3 h.

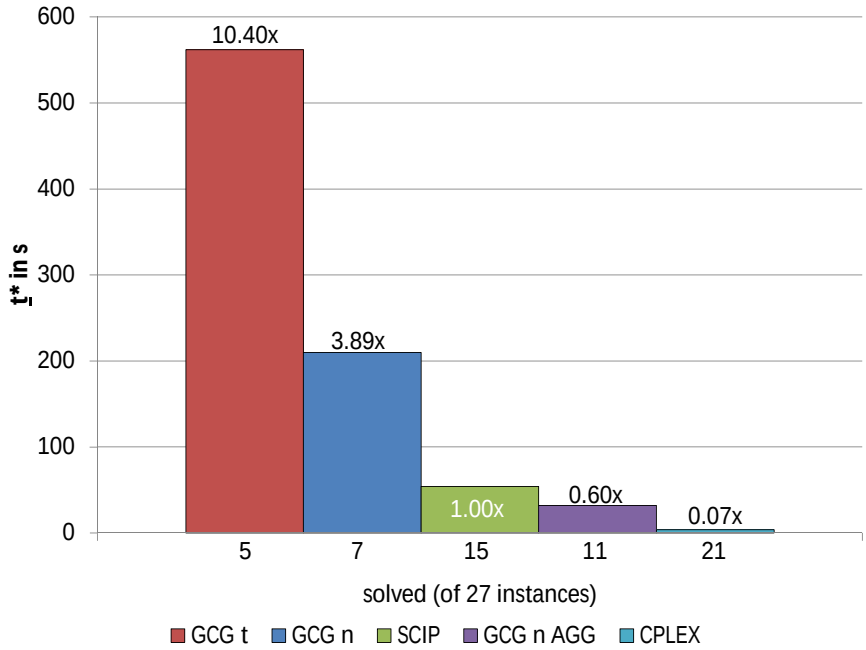

Figure 6.5: Comparison of the geometric mean $\underline{t}^*$ of all solution times for the different solvers and decompositions. Unsolved or failed instances are accounted for with the time limit of 3 h. 'GCG n' - Solver `GCG` with decomposition by emphasizing components, 'GCG n AGG' - Solver `GCG` with decomposition by emphasizing components and aggregation of the pricing problems, 'GCG t' - Solver `GCG` with decomposition by emphasizing design and operation.

The only solver able to solve all problems to optimality is the solver `CPLEX`. Furthermore it is relatively the fastest solver of all alternatives with its geometric mean ranging at only 0.07 times the value of the solver `SCIP`.
`GCG` with different decompositions and aggregation applied fails to solve all instances. The minimum of 5 solved instances of the total 21 instances is obtained by utilizing `GCG t`. Remarkable is the increase in solved instances and reduction of computing time by introducing aggregation of pricing problems. The geometric mean of the solutions times of `GCG n AGG` ranges at 0.60 times the value of `SCIP`, where `GCG n` without aggregation ranges at 3.89 times. This is in particular interesting as `GCG n AGG` fails to solve 4 instances less than `SCIP` and each fail triggers a penalty of 3 h. In total the results show that algorithmically and efficiently the commercial solver `CPLEX` is superior to the other alternatives. It solves all instances of the test set and its geometric mean is one magnitude lower than the next competitor. However, decomposition in particular by emphasizing components may improve solution times if identical subproblems are available. `GCG n AGG` with aggregated pricing problems is able to compete and underbid the non-commercial branch-and-cut solver `SCIP`.

6.2.2.2 Performance Profile

A possibility to compare the performance of different optimization solvers on solving a set of $m \in M$ instances is to calculate a performance profile (Dolan and Moré, 2002). A performance profile is defined as the 'distribution function of a performance metric (Dolan and Moré, 2002)'.
Let t^*_{ms} be the computing time required to solve instance m by solver s. The performance of solver s on instance m is compared to the best solver in instance m by defining

$$r_{ms} = \frac{t^*_{ms}}{\min_{s \in S}\{t^*_{ms}\}} \tag{6.6}$$

as the *performance ratio*. A limit r_M is introduced so that $r_M \geq r_{ms}$
$\forall\, (s \in S, m \in M)$, and $r_M = r_{ms}$ if a solver s fails to solve the instance m. Dolan and Moré (2002) show that r_M does not have an effect on the performance evaluation. To assess the overall performance of a solver s

$$P_s(\tau) = \frac{1}{|M|} \cdot \text{size}\{m \in M \,|\, log_2(r_{ms}) \leq \tau\}, \quad \forall\, s \in S \tag{6.7}$$

is defined, where $P(\tau)$ is the probability that the performance ratio of solver s is within a constant parameter $\tau \in \mathbb{R}$ of the best possible ratio, scaled by log_2. For each solver the performance profile $P_s : \mathbb{R} \mapsto [0, 1]$ is a piecewise constant function. Performance profiles allow an unbiased comparison of different solvers on a test set. The best solvers are in the upper left corner of the graph (e.g. Fig. 6.6). The performance profile calculated for all solver alternatives considered in this thesis is presented in Fig. 6.6.

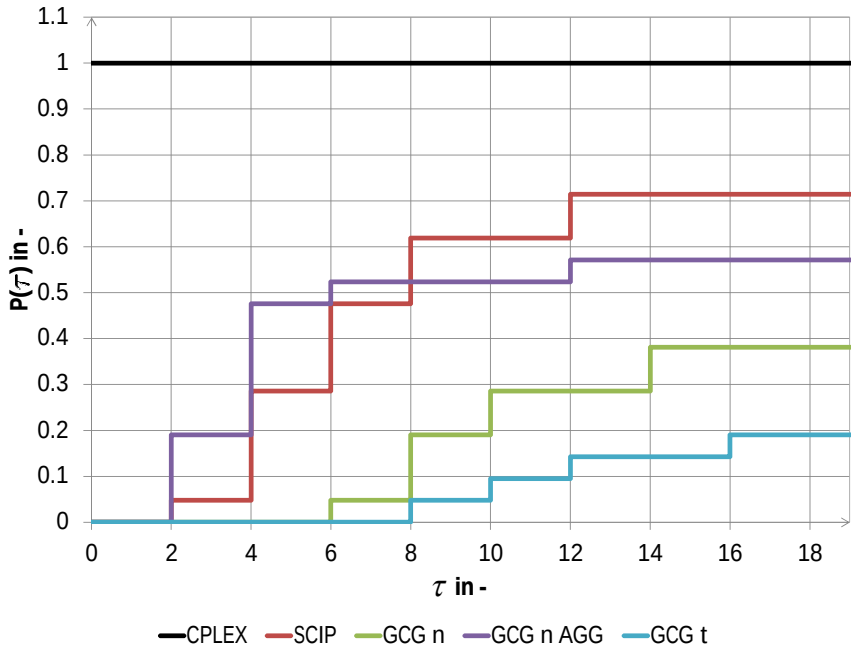

Figure 6.6: Probability $P(\tau)$ scaled by $log()_2$ for every solver that the performance ratio (6.6) is within a factor $\tau \in \mathbb{R}$ of the best possible ratio. $log_2(r_M) = 18.04$. 'GCG n' - Solver `GCG` with decomposition by emphasizing components, 'GCG n AGG' - Solver `GCG` with decomposition by emphasizing components and aggregation of the pricing problems, 'GCG t' - Solver `GCG` with decomposition by emphasizing design and operation.

The profile is generated setting $log_2(r_M) = 18.04$ for all instances which a solver fails to solve. Analyzing these results, as in the case of the geometric mean of solution times, it becomes clear that `CPLEX` outperforms all other alternatives as it is always the fastest alternative. The solver `GCG n AGG` slightly outperforms the solver `SCIP` for τ lower then 8. For τ greater then 8 `SCIP` is faster on more instances. For all values of τ `GCG n` and `GCG t` are inferior to the prior mentioned alternatives.

This section is concluded by a comment on the inferior performance of `GCG t` in comparison to the other solver alternatives. The results provide no evidence, that the reformulation is beneficial. This may also be caused by the chosen reformulation technique, i.e. Dantzig-Wolfe reformulation. From the analysis of the structure (see section 5.1.2) a decomposition utilizing the method proposed by Benders (1962) may be more suitable. If the variables describing existence and capacity are fixed the problem decomposes into t subproblems for each period of time which then may be solved independently.

7 Summary and Conclusions

The initially formulated objective of this thesis is to evaluate, whether the accuracy or the size of the model of a decentralized energy supply system may be increased in terms of number of units or number of periods of time considered, if the structure of a resulting mixed-integer linear program is exploited by a solver utilizing decomposition.
To evaluate possible decomposition approaches a literature review is performed to identify available decomposition approaches in energy engineering and in particular in the field of optimization of energy supply systems.
An outline of the identified decomposition methods based on Dantzig-Wolfe reformulation (Dantzig and Wolfe, 1961), Lagrangrean relaxation (Guignard, 2003) and Benders' decomposition (Benders, 1962) is presented, with an emphasis on the detailed description of Dantzig-Wolfe reformulation and its application in a branch-and-bound framework termed *branch-and-price*.
The formulation of the task to optimize the synthesis, design and operation of a generic decentralized energy supply system as a mixed-integer linear program is presented as formulated by Yokoyama et al. (2002) and Voll (2013). The model is analyzed to explore symmetries in the problem which are exploitable by decomposition.
The analysis identifies two embedded structures suitable for Dantzig-Wolfe decomposition. The decompositions and the respective reformulations of the presented mixed-integer linear program are introduced.
In the first identified structure the decomposition by components is emphasized. An interpretation is that each component may be sized and operated independently of the other components. The components are linked by the energy balances for every period of time, where all components together need to fulfil the respective energy demand. If the energy balances are considered in an overall master problem, a subproblem is obtained for every component considered. In case of Dantzig-Wolfe decomposition and identical components considered, theses subproblems may be aggregated and only one of these aggregated subproblems needs to be solved at every iteration of the solution process.
The second identified structure emphasizes the design and operation of all components. The selection of a unit and its capacity is not dependent on the periods of time. Hence, the constraints limiting the capacity and the existence link the periods of time. The constraint matrix does not fully decompose into an independent subproblem for every period of time, as the periods are linked by the mentioned variables for existence and capacity.

A numerical study is performed to evaluate whether the proposed reformulations are beneficial when solving the described optimization problems. A set of instances is defined to optimize a fictional energy supply system with different degrees of complexity in terms of the number of units in the superstructure and the number of periods of time considered. As solvers the branch-and-cut solvers SCIP (Achterberg, 2009), CPLEX (IBM, 2011) and the branch-and-price solver GCG (Gamrath and Lübbecke, 2010) with the specified decompositions are utilized.
To evaluate the quality of the different formulations the LP relaxation in the root-node of all instances is solved and the obtained dual bound is compared to the optimal objective function value of each instance. This is a measure to indicate how *good* the formulation of the problem is. The results indicate that the solver GCG with the decomposition by components achieves a tighter formulation of the problem and the lowest values for the quality measure comparing the dual bound to the optimal objective function value with a median of 0.77 % in comparison to CPLEX (median: 1.95 %), SCIP (median: 3.08 %) and GCG with the decomposition emphasizing design and operation (median: 3.28 %). From the results no evidence is found that for the considered set of instances the reformulation by emphasizing design and operation improves the dual bounds in the rootnode. However, the identified embedded structure is a major source for symmetry in the problem. Possibly other decomposition methods, for example Benders' decomposition, may perform better.
To evaluate the performance in solving problems to optimality the solvers are utilized to optimize all instances of the defined test set. From the results the geometric mean of the computing time required to solve all instances to optimality is calculated for each solver. The results show that CPLEX is superior to the other alternatives as it is the only solver which is able to solve all 21 instances of the test set to optimality and the geometric mean of the solution times is one magnitude lower than the next competitor. GCG with the decomposition by components and an aggregation of subproblems is able to compete and underbid the non-commercial solver SCIP, with its geometric mean of the solution times ranging at only 0.6 times of the result for solver SCIP.
The evaluation of the performance of the proposed solvers by performance profiles confirms the previous results. CPLEX is found to be the fastest solver on every instance. The results for GCG with the decomposition emphasizing components and aggregated subproblems and for SCIP show that both are able to solve about 50 % of the instances in less then 64 times of the required solution time of CPLEX.

Future perspective:

An interesting perspective is to include additionally different conversion unit types, e.g. CHP-units, wind turbines and turbo-chillers, and verify if an aggregation of subproblems still is possible and beneficial. In future further improvements of GCG for a trouble free optimization of mixed-integer linear programs with aggregation are to be expected and hence the solution performance can be estimated to further

increase. An alternative would be to implement a branch-and-price code from scratch specifically adapted to energy engineering problems.
However, the results of this thesis also show the superior performance of commercial branch-and-cut solvers such as `CPLEX` which may be tuned additionally to exploit the inherent symmetry of the problems described in this thesis.
An interesting field are models where the capacity of a component is considered as discrete values. Reformulations proposed in this thesis are applicable in the same way to these models. In particular if the discreteness is adopted to circumvent nonlinearities, the reformulations may improve overall solution performance.
A last comment focuses on the evaluation of the pricing problems. As these need to be solved at every iteration of the solution process a tailored combinatorial algorithm to solve these problems is a great advantage. Future studies could focus on this specific detail.

Bibliography

Achterberg, T. (2009). SCIP: Solving Constraint Integer Programs. *Mathematical Programming Computation*, 1(1):1–41.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research*, 46(3):316–329.

Benders, J. F. (1962). Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik*, 4(1):238–252.

Biegler, L. T. and Grossmann, I. E. (2004). Retrospective on Optimization. *Computers & Chemical Engineering*, 28(8):1169–1192.

Bouvy, C. and Lucas, K. (2007). Multicriterial Optimisation of Communal Energy Supply Concepts. *Energy Conversion and Management*, 48(11):2827–2835.

BP (2013). BP Statistical Review of Wold Energy. Technical report, London, UK.

Crutzen, P. J. (2002). Geology of Mankind. *Nature*, 415(6867):23.

Dantzig, G. B. and Wolfe, P. (1961). The Decomposition Algorithm for Linear Programs. *Economitrica*, 29(4):767–778.

Desrosiers, J. and Lübbecke, M. E. (2005). A Primer in Column Generation. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, volume 3, chapter 1, pages 1–29. Springer, New York, USA.

Desrosiers, J. and Lübbecke, M. E. (2010). Branch-Price-and-Cut Algorithms. In Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P., and Smith, J. C., editors, *Wiley Encyclopedia of Operations Research and Management Science*, pages 1–18. John Wiley & Sons, Inc., Hoboken, USA.

Desrosiers, J., Soumis, F., and Desrochers, M. (1984). Routing With Time Windows by Column Generation. *Networks*, 14(4):545–565.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking Optimization Software with Performance Profiles. *Mathematical Programming*, 91(2):201–213.

du Merle, O., Villeneuve, D., Desrosiers, J., and Hansen, P. (1999). Stabilized Column Generation. *Discrete Mathematics*, 194(1-3):229–237.

Figliozzi, M. (2010). Vehicle Routing Problem for Emissions Minimization. *Transportation Research Record: Journal of the Transportation Research Board*, 2197:1–7.

Finardi, E. C. and Luiz da Silva, E. (2006). Solving the Hydro Unit Commitment Problem via Dual Decomposition and Sequential Quadratic Programming. *IEEE Transactions on Power Systems*, 21(2):835–844.

Frangopoulos, C. A., von Spakovsky, M. R., and Sciubba, E. (2002). A Brief Review of Methods for the Design and Synthesis Optimization of Energy Systems. *International Journal of Applied Thermodynamics*, 5(4):151–160.

Galati, M. (2010). *Decomposition Methods for Integer Linear Programming*. Phd, Lehigh University, Bethlehem, USA.

Gamrath, G. and Lübbecke, M. E. (2010). Experiments with a Generic Dantzig-Wolfe Decomposition for Integer Programs. In Fiesta, P., editor, *Symposium on Experimental Algorithms (SEA 2010)*, pages 239–252, Berlin, Germany. Springer.

Geoffrion, A. M. (1972). Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260.

Glover, F. (1975). Improved Linear Integer Programming Formulations of Nonlinear Integer Problems. *Management Science*, 22(4):455–460.

Grossmann, I. E. and Biegler, L. T. (2004). Part II. Future Perspective on Optimization. *Computers & Chemical Engineering*, 28(8):1193–1218.

Guignard, M. (2003). Lagrangean Relaxation. *Top*, 11(2):151–200.

Hölemann, S. (2011). *Nichtlineare modellgestützte prädiktive Regelungsstrategie für die Blockregelung eines membranbasierten Oxyfuel-Kraftwerksprozesses*. Dissertation, RWTH Aachen University.

Hubbert, M. K. (1956). Nuclear Energy and the Fossil Fuels. Technical report, Exploration and Production Research Division, Shell Development Company, Houston, USA.

IBM (2011). IBM ILOG CPLEX Optimization Studio CPLEX User ' s Manual, version 12.4. Technical report, IBM Corporation.

IEA (2013). World Energy Outlook. Technical report, International Energy Agency, Paris, France.

Iyer, R. R. and Grossmann, I. E. (1998). A Bilevel Decomposition Algorithm for Long-Range Planning of Process Networks. *Industrial & Engineering Chemistry Research*, 37(2):474–481.

Jünger, M. and Thienel, S. (2000). The ABACUS System for Branch-and-Cut-and-Price Algorithms in Integer Programming and Combinatorial Optimization. *Software - Practice and Experience*, 30(11):1325–1352.

Kasas, M., Kravanja, Z., and Novak Pintaric, Z. (2011). Suitable Modeling for Process Flow Sheet Optimization Using the Correct Economic Criterion. *Industrial & Engineering Chemistry Research*, 50(6):3356–3370.

le Roux, W. G., Bello-Ochende, T., and Meyer, J. P. (2012). Optimum Small-Scale Open and Direct Solar Thermal Brayton Cycle for Pretoria, South Africa. In *Proceedings of SASEC 2012*, Stellenbosch, South Africa.

Lübbecke, M. E. (2011). Column Generation. In Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P., and Smith, J. C., editors, *Wiley Encyclopedia of Operations Research and Management Science*, pages 1–19. John Wiley & Sons, Inc., Hoboken, USA.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected Topics in Column Generation. *Operations Research*, 53(6):1007–1023.

Margot, F. (2010). Symmetry in Integer Linear Programming. In Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G., and Wolsey, L. A., editors, *50 Years of Integer Programming 1958-2008*, chapter 17, pages 647–686. Springer, Berlin, Germany.

Mehrotra, A. and Trick, M. A. (1996). A Column Generation Approach for Graph Coloring. *INFORMS Journal on Computing*, 8(4):344–354.

Nemhauser, G. L., Savelsbergh, M. W. P., and Sigismondi, G. C. (1994). MINTO, a mixed INTeger optimizer. *Operations Research Letters*, 15(1):47–58.

Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., Chichester, UK.

NEOS (2014). *Optimization Taxonomy*. http://www.neos-guide.org/content/optimization-taxonomy, Last accessed: 20 Sep 2014.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, USA, 2nd edition.

Noonan, F. and Giglio, R. J. (1977). Planning Electric Power Generation: A Nonlinear Mixed Integer Model Employing Benders Decomposition. *Management Science*, 23(9):946–956.

Padhy, N. P. (2004). Unit Commitment — A Bibliographical Survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205.

Ralphs, T. K., Guzelsoy, M., and Mahajan, A. (2013). SYMPHONY Version 5. 5 User's Manual. Technical report, Cor@l Laboratory, Lehigh University.

Ralphs, T. K. and Ladányi, L. (2001). COIN / BCP User's Manual. Technical Report January, Lehigh University, Bethlehem, USA.

Rieder, A. (2013). Dynamische Entwurfsoptimierung eines KWK-Systems. *BWK - Das Energie-Fachmagazin*, 65(10):7–13.

Rieder, A., Christidis, A., and Tsatsaronis, G. (2014). Multi Criteria Dynamic Design Optimization of a Small Scale Distributed Energy System. *Energy*.

Rong, A., Lahdelma, R., and Luh, P. B. (2008). Lagrangian Relaxation Based Algorithm for Trigeneration Planning with Storages. *European Journal of Operational Research*, 188(1):240–257.

Rosenthal, R. E. (2014). GAMS — A User ' s Guide. Technical Report August, GAMS Development Corporation, Washington D.C., USA.

Ryan, D. M. and Foster, B. A. (1981). An Integer Programming Approach to Scheduling. *Computer Scheduling of Public Transport*, pages 269–280.

Sagastizábal, C. (2012). Divide to Conquer: Decomposition Methods for Energy Optimization. *Mathematical Programming*, 134(1):187–222.

Sanghvi, A. P. and Shavel, I. H. (1986). Investment Planning for Hydro-Thermal Power System Expansion: Stochastic Programming Employing the Dantzig-Wolfe Decomposition Principle. *IEEE Transactions on Power Systems*, 1(2):115–121.

Savelsbergh, M. W. P. (1997). A Branch-and-Price Algorithm for the Generalized Assignment Problem. *Operations Research*, 45(6):831–841.

Schiffer, H.-W. (2013). Weltweite Energieversorgung bis 2060. *BWK - Das Energie-Fachmagazin*, 65(11/12):29–41.

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., Chichester, UK.

Sheble, G. B. and Fahd, G. N. (1994). Unit Commitment Literature Synopsis. *IEEE Transactions on Power Systems*, 9(1):128–135.

Subrahmanyam, S., Pekny, J. F., and Reklaitis, G. V. (1996). Decomposition Approaches to Batch Plant Design and Planning. *Industrial & Engineering Chemistry Research*, 35(6):1866–1876.

van den Akker, J. M., Hoogeveen, J. A., and van de Velde, S. L. (1999). Parallel Machine Scheduling by Column Generation. *Operations Research*, 47(6):862–872.

Vanderbeck, F. (1999). Computational Study of a Column Generation Algorithm for Bin Packing and Cutting Stock Problems. *Mathematical Programming*, 86(3):565–594.

Vanderbeck, F. (2000). On Dantzig-Wolfe Decomposition in Integer Programming and Ways to Perform Branching in a Branch-and-Price Algorithm. *Operations Research*, 48(1):111–128.

Vanderbeck, F. (2010). Branching in Branch-and-Price: a Generic Scheme. *Mathematical Programming*, 130(2):249–294.

Vanderbeck, F. (2014). *BaPCOD - A Generic Branch-and-Price Code*. https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/BaPCod, Last accessed: 20 Sep 2014.

Vanderbeck, F. and Savelsbergh, M. W. P. (2006). A Generic View of Dantzig–Wolfe Decomposition in Mixed Integer Programming. *Operations Research Letters*, 34(3):296–306.

Vanderbeck, F. and Wolsey, L. A. (2010). Reformulation and Decomposition of Integer Programs. In Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G., and Wolsey, L. A., editors, *50 Years of Integer Programming 1958-2008*, pages 431–502. Springer, Berlin, Germany.

Villeneuve, D., Desrosiers, J., Lübbecke, M. E., and Soumis, F. (2005). On Compact Formulations for Integer Programs Solved by Column Generation. *Annals of Operations Research*, 139(1):375–388.

Virmani, S., Adrian, E. C., Imhof, K., and Mukherjee, S. (1989). Implementation of a Lagrangian Relaxation Based Unit Commitment Problem. *IEEE Transactions on Power Systems*, 4(4):1373–1380.

Voll, P. (2013). *Automated Optimization-Based Synthesis of Distributed Energy Supply Systems*. PhD thesis, RWTH Aachen University.

Voll, P., Hennen, M., Klaffke, C., Lampe, M., and Bardow, A. (2013). Exploring the Near-Optimal Solution Space for the Synthesis of Distributed Energy Supply Systems. *Chemical Engineering Transactions*, 35:277–282.

Voll, P., Lampe, M., Wrobel, G., and Bardow, A. (2012). Superstructure-free Synthesis and Optimization of Distributed Industrial Energy Supply Systems. *Energy*, 45(1):424–435.

Wakui, T. and Yokoyama, R. (2014). Optimal Structural Design of Residential Cogeneration Systems in Consideration of Their Operating Restrictions. *Energy*, 64:719–733.

Wolsey, L. A. (1998). *Integer Programming*. John Wiley & Sons, Inc., Hoboken, USA.

Yokoyama, R., Hasegawa, Y., and Ito, K. (2002). A MILP Decomposition Approach to Large Scale Optimization in Structural Design of Energy Supply Systems. *Energy Conversion and Management*, 43(6):771–790.

Yokoyama, R. and Ito, K. (2000). A Novel Decomposition Method for MILP and its Application to Optimal Operation of a Thermal Storage System. *Energy Conversion and Management*, 41(16):1781–1795.

Yokoyama, R. and Ito, K. (2006). Optimal Design of Gas Turbine Cogeneration Plants in Consideration of Discreteness of Equipment Capabilities. *Journal of Engineering for Gas Turbines and Power*, 128(2):336–343.

Yokoyama, R. and Ose, S. (2012). Optimization of Energy Supply Systems in Consideration of Hierarchical Relationship Between Design and Operation. In Desideri, U., Manfrida, G., and Sciubba, E., editors, *Proceedings of ECOS*, page 389, Perugia, Italy. Firenze University Press.

Yokoyama, R., Shinano, Y., Taniguchi, S., Ohkura, M., and Wakui, T. (2014). Optimization of Energy Supply Systems by MILP Branch and Bound Method in Consideration of Hierarchical Relationship Between Design and Operation. Technical Report April, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany, ISSN: 1438-0064.

Appendix

A Linear Equipment Models

A.1 Investment Cost Degression

Table A.1: Nodes of piecewise linear function to describe investment cost degression as presented by Voll (2013). AC: Absorption Chiller.

Technology	**Node 1**		**Node 2**		**Node 3**	
n	$\underline{\dot{V}}_{n1}^{N}$ in kW	$IC_n(\underline{\dot{V}}_{n1}^{N})$ in €	$\overline{\dot{V}}_{n1}^{N} = \underline{\dot{V}}_{n2}^{N}$ in kW	$IC_n(\overline{\dot{V}}_{n1}^{N})$ in €	$\overline{\dot{V}}_{n2}^{N}$ in kW	$IC_n(\overline{\dot{V}}_{n2}^{N})$ in €
Boiler	100	34343	700	49245	14000	379580
AC	50	68493	750	154012	6500	522651

A.2 Part Load Performance

Table A.2: Nodes of piecewise linear function to describe part load performance of the components as presented by Voll (2013). AC - Absorption Chiller, v - scaled output power, u - scaled input power

Technology	**Node 1**		**Node 2**		**Node 3**	
n	$\underline{v}_{n1}$ in -	$u_{n1}(\underline{v}_{n1})$ in -	$\overline{v}_{n1} = \underline{v}_{n2}$ in -	$u_{n2}(\underline{v}_{n2})$ in -	$\overline{v}_{n2}$ in -	$u_{n2}(\overline{v}_{n2})$ in -
Boiler	0.2	0.2184	0.6	0.6094	1.0	1.0004
AC	0.2	0.2722	0.6	0.4833	1.0	0.9833

Table A.3: General parameter of the components as presented by Voll (2013). η_n^N is the efficiency or coefficient of performance of component n operating at its full capacity, p_n^M are the maintenance costs for every unit n given as a fraction of its investment costs.

Technology	**Parameter**	
n	η_n^N in -	p_n^M in -
Boiler	0.9	0.15
AC	0.67	0.01

B Data of Instances

B.1 Parameter of Instances

Table B.1: General parameter equal for all instances.

Symbol	Description	Value	Unit
φ	Discount rate of investment	8	%
t_{CF}	Time horizon of investment	10	years
p^U	Price for natural gas	0.06	€per kWh

Table B.2: Characteristics of energy demand data set.

Quantile	Heating demand in kW	Cooling demand in kW
Maximum	9196.9	-1000.7
0.95-Quantile	5447.1	-1343.9
0.75-Quantile	3799.8	-1670.2
Median	3012.8	-2726.0
0.25-Quantile	2154.2	-3836.0
0.05-Quantile	1522.0	-7087.7
Minimum	1500	-13910.3

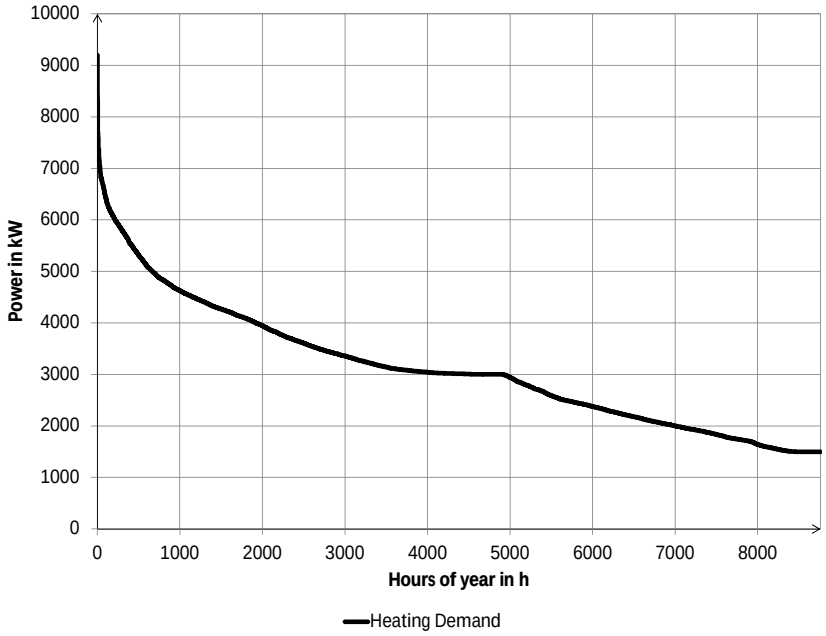

Figure B.1: Sorted heating demand in kW of energy supply system for every hour of a year.

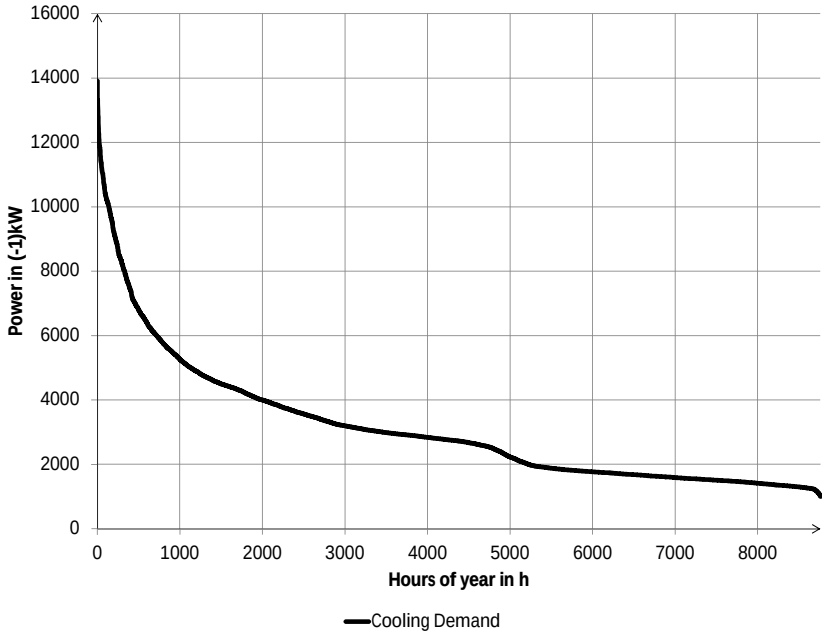

Figure B.2: Sorted cooling demand in kW of energy supply system for every hour of a year.

Table B.3: Time dependent input data for all instances.

Instances	**Period**		**Parameter**	
n<x>t1	t	E_t^{heat} in kW	E_t^{cool} in kW	Fraction in -
	1	9197	-13910	1
n<x>t2	t	E_t^{heat} in kW	E_t^{cool} in kW	Fraction in -
	1	9197	-13910	0.0001
	2	1500	-1001	2.0785
n<x>t3	t	E_t^{heat} in kW	E_t^{cool} in kW	Fraction in -
	1	9197	-13910	0.0001
	2	5348	-7455	0.1045
	3	1500	-1001	1.8111
n<x>t4	t	E_t^{heat} in kW	E_t^{cool} in kW	Fraction in -
	1	9197	-13910	0.0001
	2	6631	-9607	0.0133
	3	4066	-5304	0.3262
	4	1500	-1001	1.4988
n<x>t5	t	E_t^{heat} in kW	E_t^{cool} in kW	Fraction in -
	1	9197	-13910	0.0001
	2	7273	-10683	0.0031
	3	5348	-7455	0.0797
	4	3424	-4228	0.4617
	5	1500	-1001	1.3810
n<x>t6	t	E_t^{heat} in kW	E_t^{cool} in kW	Fraction in -
	1	9197	-13910	0.0001
	2	7658	-11328	0.0012
	3	6118	-8746	0.0266
	4	4578	-6165	0.1578
	5	3039	-3583	0.6035
	6	1500	-1001	1.2600
n<x>t7	t	E_t^{heat} in kW	E_t^{cool} in kW	Fraction in -
	1	4272	-2173	0.3333
	2	3308	-2553	0.2500
	3	2444	-3384	0.1667
	4	2333	-4014	0.0833
	5	2289	-4903	0.1667
	6	9463	-1372	0.0001
	7	1473	-13976	0.0001

B.2 Numerical Results of Instances

B.2.1 Numerical Results of the Evaluation of Formulation Strength

Table B.4: Results of the solution of the LP relaxation in the rootnode for instances `n6tx`, `x` $\in \{1, ..., 7\}$.

Instance z* in €		**Values**		
`n6t1`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
1,11E+008	`CPLEX`	0,01	1,11E+08	0,00%
	`SCIP`	0	1,11E+008	0,00%
	`GCG n`	0	1,11E+008	0,00%
	`GCG t`	0	1,10E+008	0,44%
`n6t2`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
2,50E+007	`CPLEX`	0,01	2,48E+007	0,63%
	`SCIP`	0,2	2,44E+007	2,35%
	`GCG n`	0,1	2,48E+007	0,62%
	`GCG t`	4	2,42E+007	3,31%
`n6t3`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
2,06E+007	`CPLEX`	0,03	2,02E+007	2,35%
	`SCIP`	0,2	2,02E+007	2,36%
	`GCG n`	1,3	2,05E+007	0,73%
	`GCG t`	6,1	2,03E+007	1,78%
`n6t4`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
3,36E+007	`CPLEX`	0,04	3,31E+007	1,67%
	`SCIP`	0,3	3,26E+007	3,18%
	`GCG n`	0,3	3,31E+007	1,72%
	`GCG t`	11	3,24E+007	3,90%
`n6t5`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
3,81E+007	`CPLEX`	0,05	3,76E+007	1,46%
	`SCIP`	0,3	3,72E+007	2,61%
	`GCG n`	0,4	3,76E+007	1,41%
	`GCG t`	13,5	3,69E+007	3,22%
`n6t6`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
4,37E+007	`CPLEX`	0,08	4,29E+007	1,90%
	`SCIP`	0,5	4,25E+007	2,90%
	`GCG n`	0,2	4,29E+007	1,80%
	`GCG t`	10,6	4,23E+007	3,44%
`n6t7`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
3,00E+007	`CPLEX`	0,5	2,93E+007	2,54%
	`SCIP`	0,6	2,93E+007	2,41%
	`GCG n`	0,8	2,98E+007	0,88%
	`GCG t`	19,6	2,91E+007	3,28%

Table B.5: Results of the solution of the LP relaxation in the rootnode for instances `n8tx`, `x` $\in \{1, ..., 7\}$.

Instance z* in €	**Values**			
`n8t1`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
1,05E+008	`CPLEX`	0,01	1,05E+08	0,05%
	`SCIP`	0,1	1,04E+008	0,70%
	`GCG n`	0,1	1,05E+008	0,06%
	`GCG t`	3,7	1,04E+008	0,84%
`n8t2`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
2,50E+007	`CPLEX`	0,04	2,49E+007	0,38%
	`SCIP`	0,2	2,39E+007	4,41%
	`GCG n`	0,1	2,48E+007	0,62%
	`GCG t`	6,3	2,41E+007	3,61%
`n8t3`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
2,80E+007	`CPLEX`	0,06	2,79E+007	0,41%
	`SCIP`	0,3	2,70E+007	3,99%
	`GCG n`	0,1	2,79E+007	0,62%
	`GCG t`	11,9	2,72E+007	3,28%
`n8t4`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
3,34E+007	`CPLEX`	0,07	3,27E+007	1,95%
	`SCIP`	0,3	3,21E+007	3,96%
	`GCG n`	1,1	3,31E+007	0,99%
	`GCG t`	11,8	3,10E+007	7,51%
`n8t5`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
3,80E+007	`CPLEX`	0,09	3,76E+007	1,18%
	`SCIP`	0,7	3,72E+007	2,28%
	`GCG n`	0,3	3,79E+007	0,35%
	`GCG t`	24,8	3,66E+007	3,73%
`n8t6`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
4,35E+007	`CPLEX`	0,1	4,23E+007	2,74%
	`SCIP`	0,8	4,25E+007	2,31%
	`GCG n`	0,7	4,30E+007	1,20%
	`GCG t`	31,6	4,22E+007	2,97%
`n8t7`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
3,00E+007	`CPLEX`	0,19	2,92E+007	2,72%
	`SCIP`	1,1	2,90E+007	3,59%
	`GCG n`	0,4	2,96E+007	1,41%
	`GCG t`	38,4	2,89E+007	3,69%

Table B.6: Results of the solution of the LP relaxation in the rootnode for instances `n10tx`, `x` $\in \{1, ..., 7\}$.

Instance z* in €		**Values**		
`n10t1`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
1,05E+008	`CPLEX`	0,04	1,05E+08	0,06%
	`SCIP`	0,1	1,04E+008	0,89%
	`GCG n`	0,1	1,05E+008	0,06%
	`GCG t`	3,7	1,04E+008	0,84%
`n10t2`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
2,50E+007	`CPLEX`	0,05	2,44E+007	2,63%
	`SCIP`	0,1	2,39E+007	4,51%
	`GCG n`	0,1	2,48E+007	0,62%
	`GCG t`	16,4	2,41E+007	3,61%
`n10t3`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
2,80E+007	`CPLEX`	2,86	2,73E+007	2,85%
	`SCIP`	0,5	2,70E+007	3,99%
	`GCG n`	0,15	2,79E+007	0,62%
	`GCG t`	33	2,72E+007	3,28%
`n10t4`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
3,34E+007	`CPLEX`	0,11	3,24E+007	3,01%
	`SCIP`	0,3	3,21E+007	3,96%
	`GCG n`	0,1	3,31E+007	0,99%
	`GCG t`	15,2	3,23E+007	3,36%
`n10t5`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
3,79E+007	`CPLEX`	0,13	3,72E+007	2,01%
	`SCIP`	0,2	3,67E+007	3,45%
	`GCG n`	0,5	3,76E+007	0,85%
	`GCG t`	101	3,69E+007	2,85%
`n10t6`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
4,33E+007	`CPLEX`	0,15	4,20E+007	3,02%
	`SCIP`	0,3	4,20E+007	3,08%
	`GCG n`	0,6	4,30E+007	0,77%
	`GCG t`	100	4,22E+007	2,57%
`n10t7`	**Solver**	**t* root in s**	**Dual Bound Root**	**GAP**
2,99E+007	`CPLEX`	0,33	2,88E+007	3,91%
	`SCIP`	0,9	2,88E+007	3,91%
	`GCG n`	0,8	2,96E+007	1,20%
	`GCG t`	82,4	2,89E+007	3,54%

B.2.2 Numerical Results of the Performance of Solvers

Table B.7: Numerical results for computing times required by solvers to solve instances to optimality. Entries marked with an asterisk are set to value of 3 h as solvers failed to solve this instance.

Instance	CPLEX t* in s	SCIP t* in s	GCG n t* in s	GCG n AGG t* in s	GCG t t* in s
n6t1	0.01	0.03	0.25	0.06	3.41
n6t2	0.02	0.25	2.18	0.19	5334.53
n6t3	2.37	20.94	1429	46.87	10800*
n6t4	0.09	138.3	457	126.43	10800*
n6t5	7.2	1510.3	10800*	10800*	10800*
n6t6	17.03	1005.49	10800*	10800*	10800*
n6t7	120.77	12216.36	10800*	874	10800*
n8t1	0.01	0.1	0.7	0.13	10.94
n8t2	0.11	0.8	36.2	0.3	5276.1
n8t3	0.55	17.8	10800*	4.6	10800*
n8t4	56.29	11751.5	10800*	488.8	10800*
n8t5	623.43	7019.4	10800*	10800*	10800*
n8t6	10224.83	10800*	10800*	10800*	10800*
n8t7	92.4	10800*	10800*	10800*	10800*
n10t1	0.04	0.8	4.8	0.1	4.8
n10t2	0.09	5	432.5	0.3	10800*
n10t3	0.06	200.2	10800*	0.17	10800*
n10t4	230.48	10800*	10800*	10800*	10800*
n10t5	86.99	10800*	10800*	10800*	10800*
n10t6	16200.13	10800*	10800*	10800*	10800*
n10t7	25539.92	10800*	10800*	10800*	10800*

C MILP for Discrete Capacities

If only a small range of capacities are under consideration of the different energy conversion units considered in an energy supply system it may be beneficial to introduce discrete values for the capacity of each unit n. The formulation in chapter 4.2, p. 25 can be easily adapted to this case. The following indices, sets, variables and parameters are introduced

1. Indices
    - n - Unit number
    - t - Period of time
    - d - Number of segment of piecewise linear function (part load performance)
    - k - Number of capacity
2. Sets
    - N - Set of all energy conversion units
    - T - Set of all discrete periods
    - D - Set of all segments of piecewise linear function (part load performance)
    - K - Set of all capacities of unit n
3. Continuous variables
    - $\dot{V}_{nkdt}$ - Output power of unit n with capacity k using segment d in period t
4. Binary variables
    - y_{nk} - (Non)-existence of unit n with its k-th capacity
    - δ_{nkdt} - Operational status of unit n with capacity k using segment d in period t
5. Parameters
    - $c_{nkd}^{(0)}$ - constant cost parameters
    - $c_{nkd}^{(1)}$ - constant cost parameters
    - $c_{nk}^{(2)}$ - constant investment cost for unit n with capacity k

- $\dot{E}_t$ - Energy demand in every period t
- $\underline{\dot{V}}^N_{nkd}$, $\overline{\dot{V}}^N_{nkd}$ - Lower and upper bound on $\dot{V}_{nkd}$ for every unit n and capacity k segment d

By this the model may be derived. Note decompositions as described in chapter 5 may be applied to this model.

$$\max \sum_{n\in B}\sum_{k\in K}\sum_{d\in D}\sum_{t\in T}\left(c^{(0)}_{nkd}\cdot\delta_{nkdt} + c^{(1)}_{nkd}\cdot\dot{V}_{nkdt}\right) - \sum_{n\in N}\sum_{k\in K}\left(c^{(2)}_{nk}\cdot y_{nk}\right) \tag{C.1}$$

s.t.

$$\sum_{n\,\in\,N}\sum_{k\in K}\sum_{d\in D}\left(\dot{V}_{nkdt}\right) = \dot{E}_t \quad \forall\, t\in T \tag{C.2}$$

$$\delta_{nkdt}\cdot\underline{\dot{V}}_{nkd} \le \dot{V}_{nkdt} \le \delta_{nkdt}\cdot\overline{\dot{V}}_{nkd} \quad \forall\,(n\in N, d\in D, k\in K, t\in T) \tag{C.3}$$

$$\sum_{d\in D}\delta_{nkdt} \le y_{nk} \quad \forall\,(n\in N, t\in T, k\in K) \tag{C.4}$$

$$\sum_{k\in I} y_{nk} \le 1 \quad \forall\,(n\in N) \tag{C.5}$$

$$\delta_{nkdt},\, y_{nk} \in \{0,1\} \quad \forall\,(n\in N, d\in D, \ldots \quad k\in K, t\in T) \tag{C.6}$$

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß übernommen sind, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form noch nicht als Prüfungsarbeit eingereicht worden.

Aachen, den 15. Oktober 2014

Georg Ferdinand Schneider