1

Optimization of the Design and Partial-Load Operation of Power Plants Using Mixed-Integer Nonlinear Programming

Marc Jüdes[1], Stefan Vigerske[2], and George Tsatsaronis[1]

[1] Institute for Energy Engineering, Technische Universität Berlin, Marchstr. 18, 10587 Berlin, juedes@iet.tu-berlin.de, tsatsaronis@iet.tu-berlin.de
[2] Department of Mathematics, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, stefan@math.hu-berlin.de

Summary. This paper focuses on the optimization of the design and operation of combined heat and power plants (cogeneration plants). Due to the complexity of such an optimization task, conventional optimization methods consider only one operation point that is usually the full-load case. However, the frequent changes in demand lead to operation in several partial-load conditions. To guarantee a technically feasible and economically sound operation, we present a mathematical programming formulation of a model that considers the partial-load operation already in the design phase of the plant. This leads to a nonconvex mixed-integer nonlinear program (MINLP) due to discrete decisions in the design phase and discrete variables and nonlinear equations describing the thermodynamic status and behavior of the plant. The model is solved using an extended Branch and Cut algorithm that is implemented in the solver LaGO. We describe conventional optimization approaches and show that without consideration of different operation points, a flexible operation of the plant may be impossible. Further, we address the problem associated with the uncertain cost functions for plant components.

Keywords. cogeneration plant, partial load performance, design optimization, cost minimization, nonconvex mixed-integer nonlinear programming, branch and cut

1.1 Introduction

In deregulated energy markets the optimization of the design and operation of energy conversion plants becomes increasingly important. To reduce the product cost during the entire operation time of a plant, both selection of an optimal plant structure and selection of optimal operating parameters in different load situations are necessary. Several design optimization methods were developed and applied to energy conversion systems in the past, e.g.,

exergoeconomic methods [8,16,36,47,54–57], evolutionary algorithms [3,7,12, 13,49], and mathematical programming methods [3–5,14,50].

All these approaches are based on deterministic models. Thus, the effect of data uncertainties is not considered. In this case, the optimization could lead to a solution that is not feasible when some variations in the data apply. Often, heuristics are used to adapt the solution to a new situation. However, guarantees on the quality of the so obtained solution with respect to optimality are in general not available.

We consider two sources of uncertainty in this work: The first one is caused by frequently changing operating conditions while the second one is associated with the cost model. Due to the volatility of demand (see Figure 1.1 for a typical load curve of a power plant), the plant operators are forced to operate a plant at operation points away from the usual design point, the full-load case. This effect is further reinforced by discontinuous and unsteady energy supplies from renewable energy sources such as wind energy (e.g., [17,19,34]).
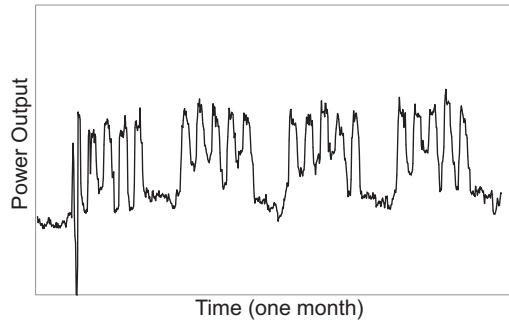

Fig. 1.1. Typical load diagram of a cogeneration plant.

An important observation is, that the consideration of partial-load operation points already in the design phase is not only meaningful for economical reasons, but also necessary to actually ensure a feasible plant operation under different partial-load conditions. This becomes even more important for cogeneration power plants, since here different amounts of each product can be requested at each time. So far only few approaches exist that could consider the partial-load operation within the design optimization, e.g., [23,36,37]. These approaches require in practical applications strong simplifications, such as a high linearity of the resulting problem. To handle the discrete decisions that are necessary to model different structures of a plant, often heuristic approaches such as genetic algorithms are applied.

Other approaches deal with the application of MINLP-techniques for the optimization of small-scale combined heat and power plants with fixed pressures and thus simplified working fluid properties within the cycle considering again only one operation point [50]. For the optimization of the operation of an

existing plant with strong simplifications in the turbine models and the performance of heat exchangers also MINLP-techniques were applied [46]. First steps towards the design optimization of power plants with consideration of their partial-load performance are discussed in [30].

The cost model is another important cause of uncertainty. A detailed knowledge of the required investment costs and of the development of interest rates and fuel prices in the future is necessary to calculate the objective function of the optimization problem, here the levelized total revenue requirement (TRR_{lev}) [8]. However, none of them is known with the required precision at the time when the optimization must be conducted. For estimating investment costs, several cost models can be found in the literature (e.g., [3, 26, 58, 60]), each of them leading after optimization to a different structure and different operation parameters.

The formulation of a model for plant-design optimization leads to a mixed-integer nonlinear program due to (a) required discrete decisions (existence, connection, and operation states of plant components) in the design part, and (b) nonlinear equations that describe, e.g., the thermodynamic properties of the working fluids and the off-design performance of components. While the number of discrete variables is still moderate, main challenges are posed by the nonconvexity of some equations. Both, discrete variables and nonconvex equations can lead to a feasible region that is disconnected and possesses many local minimal points. Thus, standard local search methods or an "easy" transformation into a mixed-integer linear model by linearization is prohibited and efficient mathematical algorithms are needed that can deal with the inherent nonconvexity of the search region to find global or good local optimal points.

Next to the already mentioned stochastic methods [7, 10], several approaches exist for the deterministic global optimization of a (nonconvex) MINLP problem [22, 27, 40, 42]. In successive outer-approximation algorithms [11, 18, 21, 62], an initial relaxation of the MINLP problem is iteratively solved and improved until a feasible point of the MINLP problem is found. If the problem is convex, a linear relaxation can be generated by linearizing nonlinear equations. However, working with linearizations of nonconvex equations can easily cut off global optimal points or lead to an infeasible relaxation. For such problems much effort is spend on finding good convex underestimators of nonconvex functions [2, 53] since they allow us to generate a convex relaxation of the problem that can be solved efficiently. To further achieve convergence to a global optimum, convex relaxation-based methods are often embedded in a Branch and Bound framework [1]. Such methods subdivide the feasible set into smaller subregions (branching) to allow for tighter convex underestimators on the corresponding subproblems. Comparing lower bounds given by evaluating the relaxation of a subregion with upper bounds calculated from feasible points of the original problem then allows coordinating the search for a global optimum [52]. The open source software package LaGO (**La**grangian **G**lobal **O**ptimizer) [42–45] is an implementation of such a method and is used for the plant design optimization discussed in this paper (cf. Section 1.3).

1.2 Model of a cogeneration power plant

We consider a simplified gas-fired combined cycle plant with steam extraction for a subsequent desalination unit. Different publications (e.g., [28]) discuss the relatively low importance of obtaining high electric efficiencies at these plants due to some specific local conditions, e.g., low gas prices. Therefore the optimization method is applied to a simplified single-pressure combined cycle plant with a supplementary firing for each heat recovery steam generator (HRSG). Figure 1.2 shows the superstructure of such a plant. The superstructure is based on realized power plant designs (e.g., [28, 48]).
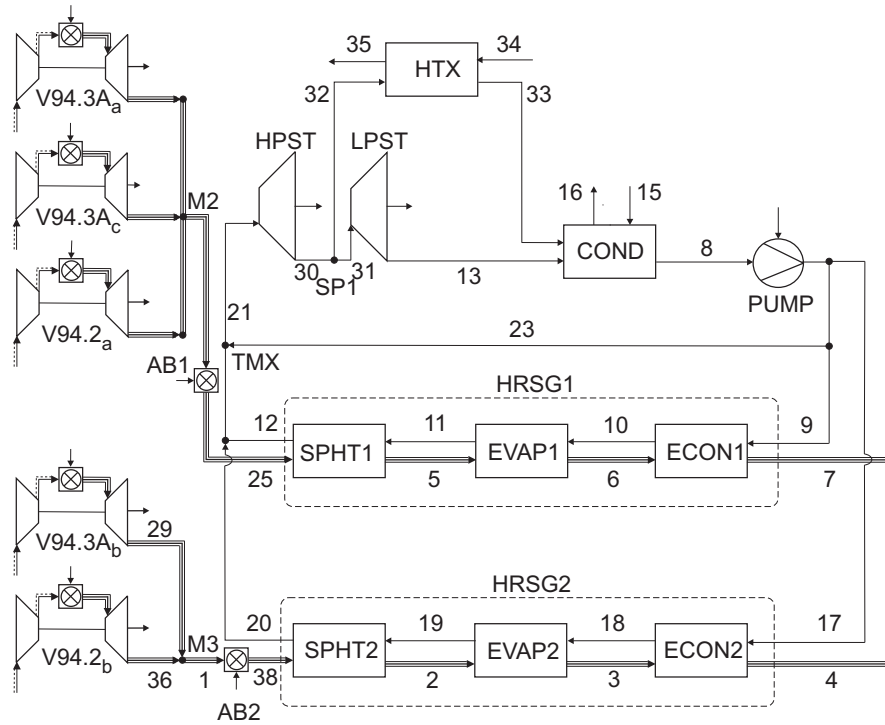

Fig. 1.2. Superstructure of the cogeneration power plant.

In accordance with [25] and industrial information, the model considers two different types of gas turbines: Three Siemens V94.3A gas turbines and two Siemens V94.2 gas turbines. The V94.3A$_a$, V94.3A$_c$ and V94.2$_a$ gas turbines can feed the first heat-recovery steam generator HRSG1. Due to earlier studies, e.g. [31], the second heat-recovery steam generator HRSG2 is fed by only two gas turbines: The V94.3A$_b$ and the V94.2$_b$ turbines. Each heat-recovery steam generator is operated independently and can be fed by a free

combination of these gas turbines. The optional additional burners AB1 and AB2 can increase the exhaust gas temperature.

The two heat-recovery steam generators consist of an economizer ECON, an evaporator EVAP and a superheater SPHT. To simplify the model of the plant, only one subsequent water injector TMX is optionally used for both heat-recovery steam generators to regulate the steam temperature. The steam is supplied to the high, the intermediate (both indicated with HPST) and the low-pressure (LPST) sections of the steam turbine. After the intermediate-pressure section of the steam turbine, the steam for the subsequent desalination unit is extracted at SP1. The condensate returning from the desalination plant is mixed with the outlet stream of the low-pressure steam turbine in the condenser COND. The feedwater pump compresses the feedwater to the required sliding pressure. Only steady-state operation points are considered.

Next to the operation at full load (operation point OP1), we considered three more characteristic load conditions, each with a different demand for electricity and extracted steam, cf. Table 1.1. This, in reality uncertain, load information can be obtained from statistical methods, expert-knowledge, or existing data sets by using data mining techniques. We achieved good results using so-called self organizing maps (SOM, see also [33]), a special kind of artificial neural networks for the classification of data sets [20, 29], e.g., the load profile shown in Figure 1.1.

Table 1.1. Operation points of the cogeneration power plant shown in Figure 1.2.

Name	Operating hours h/a	Electric output MW	$\dot{m}_{32}$ t/h
OP1	1972	750	133.1
OP2	1972	600	86.7
OP3	1972	500	78.7
OP4	1972	400	86.7

In the following sections we give some insight into the thermodynamic and economic parts of the model and its formulation as a mathematical program.

1.2.1 Thermodynamic model

The thermodynamic part of the model describes the physical behavior of the plant. We start with a discussion of the design phase of the plant, the calculation of the thermodynamic properties of the working fluids and the component sizing. After completing the plant design we can calculate the investment costs (cf. Section 1.2.2). In the subsequent phase the off-design performance of single components and of the overall plant can be computed. Considering partial-load operation points in the design optimization allows us

to evaluate the operation costs also for these operation points. In this way, the optimization model becomes more realistic, but it requires an integration into the model of the off-design behavior for every considered operation point.

A possible result of the design optimization is, for example, the suggestion to install two gas turbines for each heat-recovery steam generator. In this case and when the power plant is operated under partial-load conditions, one of these gas turbines could be switched off while the other one would be running under full-load conditions. If so, the overall efficiency increases (lower fuel cost), but the investment costs increase too.

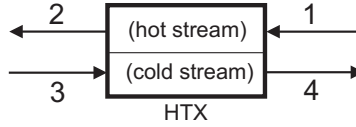Exemplary, we illustrate the design and off-design model of a heat exchanger, see Figure 1.3.

Fig. 1.3. Schematic illustration of a heat exchanger (HTX)

Design modelling using a heat exchanger

As an example for the design modelling of components, an adiabatic heat exchanger is discussed. To build a model of a heat exchanger, its independent variables have to be known. Equations $(1.1) - (1.6)$ are necessary for specifying the heat exchanger performance. The subscripts indicate the stream numbers shown in Fig. 1.3:

$$\dot{Q} = \dot{m}_1 \, (h_1 - h_2) \tag{1.1}$$

$$\dot{Q} = \dot{m}_3 \, (h_4 - h_3) \tag{1.2}$$

$$\dot{Q} = k \, A \, \Delta T_{\log} \tag{1.3}$$

$$\Delta T_{\log} = \frac{(T_2 - T_3) - (T_1 - T_4)}{\ln(T_2 - T_3) - \ln(T_1 - T_4)} \tag{1.4}$$

$$h_i = f(T_i, p_i \text{ or } x_i), \qquad i = 1, \ldots, 4 \tag{1.5}$$

$$p_{\text{exit},j} = f(p_{\text{inlet},j}), \qquad j = \text{cold}, \text{hot} \tag{1.6}$$

where $\dot{Q}$ denotes the rate or heat transfer heat rate, k is the overall heat transfer coefficient, A represents the heat exchanger surface area, h denotes the enthalpy, T is the temperature, p is the pressure, and x represents the steam quality.

Eight of these 18 ($\dot{m}_1$, $\dot{m}_3$, h_1, h_2, h_3, h_4, T_1, T_2, T_3, T_4, p_1, p_2, p_3, p_4, $\dot{Q}$, $\Delta T_{\log}$, k, A) variables can be selected more or less independently. In the design optimization, the values of these independent decision variables have

to be determined to maximize the overall plant efficiency or to minimize the overall product cost using the respective objective function. Here, the sizing and costing of the components refers to the design case where the maximal values of pressure, temperature, and mass flow rate are used. The information obtained from the design case is used to calculate the off-design performance, where additional equations have to be considered.

Off-Design modelling

The description of the off-design performance of a heat exchanger requires additional equations, some of which are taken from the commercial software EBSILONProfessional 7.00 [51].

$$\dot{Q} = k \, A \, \Delta T_{\log} \tag{1.7}$$

$$\frac{1}{k} = \frac{1}{\alpha_{c,N} \, FK_1} + \frac{1}{\alpha_{h,N} \, FK_2} \tag{1.8}$$

$$FK_1 = \left(\frac{\dot{m}_c}{\dot{m}_{c,N}} \right)^{\mathrm{Exp}_1} \tag{1.9}$$

$$FK_2 = \left[1 - 0.0005 \left(\frac{(T_{h,i,N} + T_{h,e,N}) - (T_{h,i} + T_{h,e})}{2} \right) \right] \left(\frac{\dot{m}_h}{\dot{m}_{h,N}} \right)^{\mathrm{Exp}_2} \tag{1.10}$$

Here, α represents the heat transfer coefficient, c and h stand for cold and hot and Exp_1 and Exp_2 denote some component specific exponents.

For a steam turbine operating at off-design conditions, the characteristic curve describing its partial-load performance is modelled in accordance with Equation (1.11) which describes the so-called Stodola law. It correlates the inlet and outlet pressures p_i and p_e, the inlet temperature T_i and the mass flow rate $\dot{m}$ at the actual (partial load) and nominal (design point) conditions:

$$\frac{\dot{m}}{\dot{m}_N} = \frac{p_e}{p_{e,N}} \sqrt{\frac{T_{i,N}}{T_i}} \sqrt{\frac{1 - (p_i/p_e)^2}{1 - (p_{i,N}/p_{e,N})^2}} \tag{1.11}$$

The isentropic efficiency η_s of the turbine, that compares the real and ideal expansion in the turbine, is a function of the mass flow rate:

$$\frac{\eta_s}{\eta_{s,N}} = -1.0176 \left(\frac{\dot{m}}{\dot{m}_N} \right)^4 + 2.4443 \left(\frac{\dot{m}}{\dot{m}_N} \right)^3 - 2.1812 \left(\frac{\dot{m}}{\dot{m}_N} \right)^2$$
$$+ 1.0535 \left(\frac{\dot{m}}{\dot{m}_N} \right) + 0.701. \tag{1.12}$$

At partial-load operation, the efficiency η_s must be adjusted with respect to changes in the outlet steam quality Δx_e. When the exiting steam quality x_e is lower than 1, this adjustment is carried out using the following approximation

$$\eta_{s,\mathrm{corr}} = \eta_s - \frac{1}{2} \Delta x_e, \tag{1.13}$$

where η_s denotes the isentropic efficiency in accordance with (1.12) and $\eta_{s,\mathrm{corr}}$ denotes the resulting isentropic efficiency after the correction for steam quality.

Working fluid properties

The thermodynamic properties of the exhaust gases are calculated using equations from Knacke, Kubaschewski, and Hesselmann [32]. For example, the molar enthalpy of a pure ideal gas stream i is calculated with the following equation:

$$h_i = 10^3 \left(h_i^{\text{ref}} + a_i y + \frac{b_i}{2} y^2 - c_i y^{-1} + \frac{d_i}{3} y^3 \right) , \qquad (1.14)$$

where $y = T/1000$, h^{ref} is associated with the reference value calculating the enthalpy and a_i, b_i, c_i, and d_i are constants depending on the substance being considered. All gas streams in the process are treated as mixtures of ideal gases. Therefore the molar enthalpy h of these gas streams is calculated with the aid of the respective mole fractions x_j of the j components:

$$h = \sum_j x_j \, \dot{n} \, h_j \qquad (1.15)$$

Here is $\dot{n}$ the molar flow rate that can be calculated using the mass flow rate $\dot{m}$ and the respective molar mass M

$$\dot{n} = \frac{\dot{m}}{M} . \qquad (1.16)$$

To enable the software to find a good solution, the high degree polynomials from the original water steam properties (IAPWS IF97 [61]) are simplified here to polynomials of a degree at most four. A detailed discussion of these polynomials is presented in [3].

1.2.2 Economic model and uncertainty in investment costs

In addition to the thermodynamic model of the power plant, an economic analysis is needed to calculate its objective function, the annual levelized **T**otal **R**evenue **R**equirement (TRR_{lev} [8]). TRR includes the fuel costs, the operating and maintenance expenses as well as the carrying charges (which consider the capital recovery, interest, dividends, taxes, and insurances). The levelized TRR_{lev} is a function of the annual values TRR_{n}

$$TRR_{\text{lev}} = i_{\text{eff}} \frac{(1 + i_{\text{eff}})^n}{(1 + i_{\text{eff}})^n - 1} \sum_n \frac{TRR_n}{(1 + i_{\text{eff}})^n} , \qquad (1.17)$$

where i_{eff} denotes the effective interest rate and n the number of years considered in the analysis.

Purchased equipment cost (PEC in the following), fuel cost ($\dot{C}_{\text{f}}$ in the following) and operating hours have a strong influence on a cost-effective design of the plant. PEC are calculated for the full-load case, since here the largest

pressures, temperatures and mass flow rates occur (see also Section 1.2.1), whereas $\dot{C}_f$ and other operating cost are calculated at every operation point.

For calculating the PEC, cost functions for each component are used. These functions depend on the characteristic variables of a component, e.g., the surface area of a heat exchanger A or the power output of a turbine $\dot{W}$. However, in most cases the "real" cost function, if there is one, is not known. Instead, different cost models are discussed in the literature (e.g., [3, 26, 58, 60]). Two very different cost models for a steam turbine [3, 26] are shown in Figure 1.4.
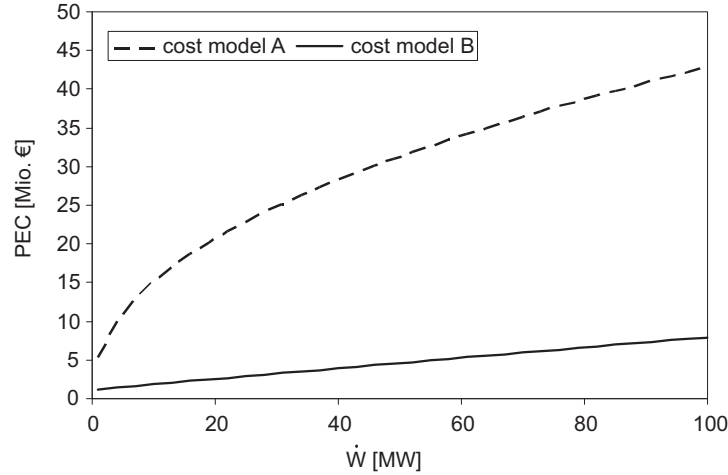

Fig. 1.4. PEC calculated with two extremely different cost models for a steam turbine. Here, PEC is a function of the power output $\dot{W}$.

Although the differences in the PEC shown in Figure 1.4 seem to be too large from the engineer's point of view, we consider these two very different models as an academic example for the uncertainties associated with cost functions. It must be emphasized that such differences in a "real world" engineering problem are unrealistic. For considering the effects of the different cost approaches (Section 1.4.2) we employ in the objective function the assumed reliability of each function as a weighting factor.

1.2.3 Formulation as mathematical program

Two different types of variables are used to formulate the optimization problem as a MINLP: Binary and continuous variables. Binary variables are needed to decide, (a) which of two different types of gas turbines are used for each heat-recovery steam generator, (b) which of two possible heat-recovery steam generators are used, and (c) whether there is a need for the additional burners

AB1 and AB2. Additionally we use binary variables to determine the actual state of some streams. Here, these variables indicate whether the working fluid is superheated or its thermodynamic state is within the liquid–vapor region and thus the steam quality lies strictly between 0 and 1.

Isentropic efficiencies, heat exchanger surfaces, and the thermodynamic properties of each working fluid represent continuous variables. The latter are calculated by nonlinear equations such as (1.14) and (1.15). Table 1.2 shows the decision variables for the design and the off-design models.

Table 1.2. Binary (y_i) and continuous decision variables in the model with full-load operation conditions OP1 and partial-load operation conditions OP2 to OP4 (see Table 1.1) indicated by superscripts. GT1-3 indicate the electric power output of the gas turbines V94.3A$_a$–V94.3A$_c$ and GT4-5 indicate the electric power output of the gas turbines V94.2$_a$ and V94.2$_b$, respectively.

Component	Variable	design conditions (OP1)	off design conditions (OP2-OP4)	cost part
GT1-5	operation	y_{GT1}^1 - y_{GT5}^1	$y_{GT1}^{2,3,4}$ - $y_{GT5}^{2,3,4}$	–
GT1-5	existence	–	–	y_{GT1}^c - y_{GT5}^c
AB1,2	operation	y_{AB1}^1, y_{AB2}^1	$y_{AB1}^{2,3,4}, y_{AB2}^{2,3,4}$	–
AB1,2	existence	–	–	y_{AB1}^c, y_{AB2}^c
HRSG1,2	operation	y_{HRSG1}^1, y_{HRSG2}^1	$y_{HRSG1}^{2,3,4}, y_{HRSG2}^{2,3,4}$	–
number of binary variables		9	27	7
GT1-5	power output	$\dot{W}_{GT1}^1$ - $\dot{W}_{GT5}^1$	$\dot{W}_{GT1}^{2,3,4}$ - $\dot{W}_{GT5}^{2,3,4}$	–
AB1,2	fuel flow rate	$\dot{m}_{f,AB1}^1, \dot{m}_{f,AB2}^1$	$\dot{m}_{f,AB1}^{2,3,4}, \dot{m}_{f,AB2}^{2,3,4}$	–
TMX	mass flow rate	$\dot{m}_{23}^1$	$\dot{m}_{23}^{2,3,4}$	–
	pressure	p_{14}^1	–	–
	pressure	p_8^1	–	–
SPHT	ΔT	$\Delta T_{SPHT,N}$	–	–
EVAP	ΔT	$\Delta T_{EVAP,N}$	–	–
ECON	ΔT	$\Delta T_{ECON,N}$	–	–
COND	ΔT	$\Delta T_{COND,N}$	–	–
number of cont. variables		17	24	0

The entire model with its 41 independent continuous and 43 independent binary variables is formulated in GAMS [24] as one system of equations in form of a large mixed-integer nonlinear program. Here, the following equations are used: Mass, energy, and impulse balances, equations for calculating the working fluid properties, equations for calculating the components full- and partial-load performance, and cost equations. Additional constraints result from the limitation of temperatures, mass flow rates, and pressures in the partial-load cases compared to the design (full-load) case.

Due to the simultaneous solution of the optimization problem, *all* equations and constraints of the model have to be satisfied. Therefore, it is necessary that even if some gas turbines associated with one heat-recovery steam generator are not included in the actual design of the plant, the exhaust gas mass flow rate passing through the heat-recovery steam generator and its thermodynamic variables are not equal to zero. Otherwise the calculation of values like the logarithmic mean temperature difference $\Delta T_{\log}$ in the heat exchangers yield function evaluation errors in the optimization process. Hence, e.g. the mixer M3 (Figure 1.2) has to be modelled in an appropriate way:

$$y_{\mathrm{M3}} \leq y_{94.3,\mathrm{b}} + y_{94.2,\mathrm{b}} \,, \tag{1.18}$$

$$y_{\mathrm{M3}} \geq y_{94.3,\mathrm{b}} \,, \tag{1.19}$$

$$y_{\mathrm{M3}} \geq y_{94.2,\mathrm{b}} \,. \tag{1.20}$$

Here, the binary variable y_{M3} indicates the operation of the heat-recovery steam generator in the respective load case ($y_{\mathrm{HRSG},2}$ in Table 1.2). The energy balance equation is formulated using a so-called big-M formulation incorporating the upper bounds on the variables $\dot{n}_1$ and h_1 ($\overline{\dot{n}}_1$ and $\overline{h}_1$, respectively):

$$\dot{n}_1\, h_1 \leq \dot{n}_{29}\, h_{29} + \dot{n}_{36}\, h_{36} + (1 - y_{\mathrm{M3}})\,(-1)\,\overline{\dot{n}}_1\,\overline{h}_1 \,, \tag{1.21}$$

$$\dot{n}_1\, h_1 \geq \dot{n}_{29}\, h_{29} + \dot{n}_{36}\, h_{36} - (1 - y_{\mathrm{M3}})\,(-1)\,\overline{\dot{n}}_1\,\overline{h}_1 \,. \tag{1.22}$$

Finally, the equations describing the chemical composition of the gas streams are formulated as

$$x_{1,\mathrm{N}_2}\, \dot{n}_1 \leq \dot{n}_{29}\, x_{29,\mathrm{N}_2} + \dot{n}_{36}\, x_{36,\mathrm{N}_2} + (1 - y_{\mathrm{M3}})\,\overline{x}_{1,\mathrm{N}_2}\,\overline{\dot{n}}_1 \tag{1.23}$$

$$x_{1,\mathrm{N}_2}\, \dot{n}_1 \geq \dot{n}_{29}\, x_{29,\mathrm{N}_2} + \dot{n}_{36}\, x_{36,\mathrm{N}_2} - (1 - y_{\mathrm{M3}})\,\overline{x}_{1,\mathrm{N}_2}\,\overline{\dot{n}}_1 \,. \tag{1.24}$$

where $\dot{n}_i$ denote the mole flow rates and x_{N_2} the mole fraction of nitrogen. The equations for the other substances in the exhaust gas streams have to be formulated accordingly. Obviously, (1.21) – (1.24) can be satisfied for both cases, $y_{\mathrm{M3}} = 0$ and $y_{\mathrm{M3}} = 1$. For $y_{\mathrm{M3}} = 0$ the left hand side variables can be chosen according to the heat exchanger requirements.

1.3 Solution of the MINLP

As mentioned before, the presence of discrete decisions and nonlinear nonconvex equations describing the design, thermodynamic status, and behavior of the plant leads to a nonconvex MINLP whose solution requires sophisticated algorithms. Currently, there are only a few solvers available that can explicitly handle nonconvex MINLPs. To our best knowledge LaGO [45] is currently the only freely available one.

In this section we describe LaGOs enhanced Branch and Cut algorithm in more detail. At first the algorithm approximates nonconvex functions by

convex underestimators, i.e., a convex function that underestimates the respective original function. Next, the obtained convex relaxation is linearized via the construction of supporting hyperplanes. These cutting planes are used to initialize and improve a linear relaxation of the problem. By means of this relaxation it is possible to efficiently compute reliable lower bounds to the global optimum and starting points for local searches for feasible solutions. A successive branching of the search space enables an improvement of the underestimators in the progression of the algorithm and thus a tightening of the linear relaxation.

In the following sections we focus on some components of LaGOs algorithm. The preprocessing routines include the investigation of the problem structure of a given MINLP (Section 1.3.1), the initialization of relaxations that lead to a linear outer approximation (Section 1.3.2), and methods that are used for the reduction of variables bounds (Section 1.3.3). Finally, we give a short overview of LaGOs Branch and Cut algorithm (Section 1.3.4). We use a general MINLP formulation to emphasize the wide applicability of the proposed method.

1.3.1 Problem structure analysis

Problem formulation

A general MINLP can be formulated as

$$
\begin{aligned}
\min \quad & b_0^\top x && \text{(P)} \\
\text{such that} \quad & h(x) \leq 0, \\
& x \in [\underline{x}, \overline{x}], \\
& x_j \in \mathbb{Z}, \qquad j \in B,
\end{aligned}
$$

where $B \subseteq \{1,\ldots,n\}$, b_0, $\underline{x}$, $\overline{x} \in \mathbb{R}^n$, and $h : \mathbb{R}^n \to \mathbb{R}^m$ is twice-continuously differentiable. The set $[\underline{x}, \overline{x}] := \{x \in \mathbb{R}^n | \underline{x}_i \leq x_i \leq \overline{x}_i\}$ is referred as *box* and constitutes finite bounds on the variables. For the sake of simplicity we assume that the objective function is linear and equality constraints were replaced by two inequalities. Note, that to handle a nonlinear objective function $h_0(x)$, one can minimize a new variable y under the additional constraint $h_0(x) \leq y$.

LaGO requires procedures for the evaluation of function values, gradients, and Hessians. This restriction to "black-box functions" has the advantage that very general functions can be handled, but also the disadvantage that without insight into the algebraic structure of the functions $h_i(x)$ advanced reformulation and box reduction techniques (as in [1, 40, 53]) cannot be used and we are forced to use sampling methods in some components of LaGO.

Block separability

At first, LaGO investigates the sparsity structure and block separability of the functions $h_i(x)$. A function is called *block-separable* if it can be represented as

a sum of sub-functions, each depending only on a small number of variables. Block separability is a common property of real world applications where singular complex components are coupled by linear constraints. Also the model discussed in the previous section is highly block separable, since the power plant components, which are described by nonlinear equations, are linked only by linear equations for the working fluids (molar fraction x_i, mass flow rate $\dot{m}$, temperature T, pressure p, enthalpy h, and entropy s). Using a simple sampling technique for the recognition of sparsity patterns in the Hessian of the black-box functions $h_i(x)$ [45], LaGO automatically reformulates each function into the form

$$h_i(x) = c_i + b_i^\top x + \sum_{k=1}^{q_i} x_{Q_{i,k}}^\top A_{i,k} x_{Q_{i,k}} + \sum_{k=1}^{p_i} h_{i,k}(x_{N_{i,k}}), \qquad (1.25)$$

where $c_i \in \mathbb{R}$, $b_i \in \mathbb{R}^n$, and the index sets $Q_{i,k}$ and $N_{i,k}$ denote quadratic and nonlinear nonquadratic variables, respectively. They are also referred as *blocks* of the function $h_i(x)$. The structure (1.25) allows us to distinguish between linear, quadratic, and nonquadratic parts of a function, and to treat each block separately if advantageous.

Convexity

Originating from the block separable formulation (1.25), LaGO checks for each quadratic block $x_{Q_{i,k}}^\top A_{i,k} x_{Q_{i,k}}$ and each nonquadratic block $h_{i,k}(x_{N_{i,k}})$ whether it represents a convex function or not. Therefore, for a quadratic function $x_{Q_{i,k}}^\top A_{i,k} x_{Q_{i,k}}$, it is sufficient to check whether the minimal eigenvalue of $A_{i,k}$ is nonnegative. For a nonquadratic function $h_{i,k}(x_{N_{i,k}})$, the minimal eigenvalue of the Hessian $\nabla^2 h_{i,k}(x_{N_{i,k}})$ is evaluated at sample points from the box $[\underline{x}_{N_{i,k}}, \overline{x}_{N_{i,k}}]$. Observe that only the sign of the eigenvalue is of interest, so that even for curvaceous functions a sufficiently rich set of sampling points yields correct results.

1.3.2 Relaxations

As a result of the structure analysis, LaGO knows for each block in the formulation (1.25) whether it is convex or not. For the computation of a convex underestimator $\breve{h}_i(x)$ of each nonconvex function $h_i(x)$, first nonconvex nonquadratic terms in $h_i(x)$ are underestimated by possibly nonconvex quadratic terms. Afterwards, each nonconvex quadratic term is convexified. Finally, a linear relaxation is generated by linearizing the convexified functions in reference points and dropping of integrality restrictions on the variables x_j, $j \in B$.

Quadratic underestimators

Let $g : \mathbb{R}^r \to \mathbb{R}$ be a nonconvex function $h_{i,k}$ from (1.25), $r = |N_{i,k}|$. A quadratic underestimator $q(x) = x^\top A x + b^\top x + c$ of $g(x)$ is computed by

using a powerful sampling algorithm [41, 45]. This algorithm allows LaGO to determine quadratic underestimators of nonconvex functions for which only function and derivative evaluation methods are available. Starting with an initial sample set S consisting, among others, of vertices of the box $[\underline{x}, \overline{x}]$ and a distinguished sample point $\hat{x} \in S$ (often a local minimizer of $g(x)$), the following two steps are iterated:

1. Determine coefficients A, b, and c of $q(x)$ by solving the linear program

$$\min_{A,b,c} \quad \sum_{x \in S} g(x) - q(x) \qquad\qquad (\text{U})$$

$$\text{such that} \quad q(x) \leq g(x), \qquad x \in S,$$

$$q(\hat{x}) = g(\hat{x}).$$

2. For points $\tilde{x} \in S$ with $q(\tilde{x}) = g(\tilde{x})$ maximize locally the error $q(x) - g(x)$ over the box $[\underline{x}, \overline{x}]$ by solving a nonlinear program starting from $\tilde{x}$. If this yields a point $\check{x}$ with $q(\check{x}) - g(\check{x}) > \delta_{\text{tol}}$, add the inequality $q(\check{x}) \leq g(\check{x})$ to (U) and go to step 1. Otherwise, i.e., the maximal error $\delta_{\max}$ is below the tolerance δ_{tol}, lower $q(x)$ by $\delta_{\max}$ (i.e., subtract $\delta_{\max}$ from c) and stop.

Figure 1.5 illustrates a quadratic underestimator for the logarithmic mean temperature difference obtained by this method.
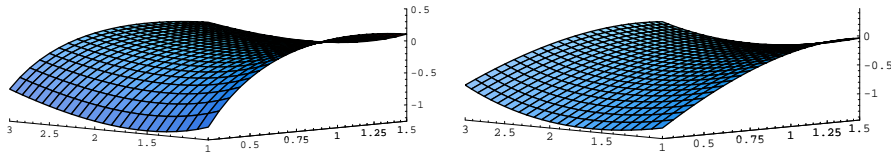

Fig. 1.5. The function $\Delta T_{\log} = (\Delta T_1 - \Delta T_2)/(\ln \Delta T_1 - \ln \Delta T_2)$ (logarithmic mean temperature difference) as used in the modelling of the partial-load performance (left, cf. (1.4)) and a corresponding quadratic underestimator (right). $\Delta T_{\log}$ has been fixed for visualization.

Convexification of quadratic terms

For the convexification of nonconvex quadratic terms $f(x) = x^\top A x$, LaGO uses α-underestimators as introduced by Adjiman and Floudas [2]. An α-underestimator of $f(x)$ over the box $[\underline{x}, \overline{x}]$ is the function

$$\check{f}(x) = f(x) + \sum_{i=1}^{r} \frac{\max\{0, -\lambda_1(WAW)\}}{(\overline{x}_i - \underline{x}_i)^2}(x_i - \underline{x}_i)(x_i - \overline{x}_i), \qquad (1.26)$$

where $\lambda_1(D)$ denotes the minimal eigenvalue of a matrix D and the diagonal matrix W has the box-width $\overline{x} - \underline{x}$ on its diagonal and has been introduced for scaling reasons. It is clear that $\check{f}$ is convex and $\check{f}(x) \leq f(x)$ for all $x \in [\underline{x}, \overline{x}]$ [2]. Figure 1.6 illustrates a convexified quadratic underestimator.
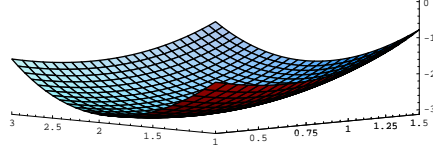
Fig. 1.6. Convex α-underestimator of the nonconvex quadratic underestimator in Figure 1.5.

Linear relaxation

The linear relaxation (R) of (P) is generated by linearizing each nonlinear function $h_i(x)$ (if convex) or its convexification $\breve{h}_i(x)$ at a reference point $\hat{x}$. For initialization, the point $\hat{x}$ is chosen to be an optimal point of the nonlinear convex relaxation. During the Branch and Cut algorithm (Section 1.3.4), (R) is augmented by further linearizations at candidates for optimal points of (P) and optimal points of (R).

Further, to address the integrality restrictions on x_j, $j \in B$, in (R), mixed-integer-rounding cuts, which have their origin in mixed-integer linear programming [35, 38], are added to (R). These linear inequalities allow us to cut off non-integral solutions from (R) [45].

As a third method for the computation of valid inequalities, LaGO can generate so called interval gradient cuts [9, 42], which are based on interval arithmetic calculations [39]. Assume that for a function $h_i(x)$ it is possible to compute an enclosure $[\underline{d}, \overline{d}] \subset \mathbb{R}^n$ of the gradient $\nabla h_i(x)$ over $[\underline{x}, \overline{x}]$, i.e., $\nabla h_i(x) \in [\underline{d}, \overline{d}]$ for all $x \in [\underline{x}, \overline{x}]$. Then, given a reference point $\hat{x} \in [\underline{x}, \overline{x}]$,

$$h_i(\hat{x}) + \nabla h_i(\hat{x})(x - \hat{x}) + \min_{d \in [\underline{d}, \overline{d}]} (d - \nabla h_i(\hat{x}))^\top (x - \hat{x}) \leq h_i(x) \quad \forall x \in [\underline{x}, \overline{x}].$$

Introducing new positive variables y^+ and y^- and writing $x - \hat{x} = y^+ - y^-$ with $y_j^+ = \max\{0, (x - \hat{x})_j\}$ and $y_j^- = \max\{0, -(x - \hat{x})_j\}$, $j = 1, \ldots, n$, we obtain (due to the inequality $h_i(x) \leq 0$ in (P)) the interval gradient cut

$$h_i(\hat{x}) + \nabla h_i(\hat{x})(x - \hat{x}) + (\underline{d} - \nabla h_i(\hat{x}))^\top y^+ - (\overline{d} - \nabla h_i(\hat{x}))^\top y^- \leq 0.$$

These cuts have the advantage that they can be derived directly from a nonlinear and nonconvex function and thus do not rely on a prior convexification step. On the other hand, the modelling of the conditions $y_j^+ = \max\{0, (x - \hat{x})_j\}$ and $y_j^- = \max\{0, -(x - \hat{x})_j\}$ would require additional discrete variables. Therefore, in order to fit into the linear relaxation (R), we instead add only the equations $x - \hat{x} = y^+ - y^-$, $y^+ \leq \overline{x} - \hat{x}$, $y^- \leq \hat{x} - \underline{x}$, and $y_i^+ + y_i^- \leq \max\{\overline{x}_i - \hat{x}_i, \hat{x}_i - \underline{x}_i\}$, $i = 1, \ldots, n$, to (R).

1.3.3 Methods to tighten the bounding box

Since the quality of the underestimators and cuts depends strongly on the bounding box $[\underline{x}, \overline{x}]$, it can be advantageous to apply boxreduction procedures in the preprocessing. Also during the Branch and Cut algorithm, a branching operation might facilitate possible reductions of variable bounds, and even detect infeasibility for a subregion or fix binary variables.

In LaGO, two boxreduction techniques are currently implemented. The first method computes a new bounding box on the variables by enclosing the feasible set of (R), i.e., each or only some selected variable is minimized and maximized w.r.t. the constraints of (R) [45].

The second method is a simple constraint propagation method and thus utilizes only one constraint at a time, but works on the original formulation (P). Similar to the interval gradient cuts, this procedure relies on interval arithmetic operations (as they are available within the GAMS [24] interface): For a box $U \subseteq [\underline{x}, \overline{x}]$ assume that $h_i(x)$ can be written as $h_i(x) = g(x) + bx_j$ with x_j not appearing in $g(x)$ and $b \in \mathbb{R}$, $b \neq 0$. Denote by $g(U)$ an interval in $\mathbb{R} \cup \{\pm\infty\}$ s.t. $g(x) \in g(U)$ for all $x \in U$. Let $[\underline{y}_j, \overline{y}_j] = -g(U)/b$. If $b > 0$, $\overline{x}_j$ can be updated to $\min\{\overline{x}_j, \overline{y}_j\}$. Otherwise, if $b < 0$, $\underline{x}_j$ can be updated to $\max\{\underline{x}_j, \underline{y}_j\}$. In case that the new bounds define an empty box, infeasibility of the subproblem with box U is detected. After the bounds on x_j have been reduced, other constraints depending on x_j might yield further box reductions for other variables. Thus, the same procedure is applied to these constraints. This process iterates until the box stops to reduce significantly or infeasibility is detected.

1.3.4 Branch and Cut algorithm

To search for a global optimum of (P), the algorithm follows a Branch and Bound scheme. Lower bounds on the global optimal value are computed by solving the linear relaxation, while upper bounds are given by the objective function value of incumbent solutions. These are points that are feasible for (P) and are found by a local search, that is the discrete variables are fixed in (P) and a descent algorithm is applied to the resulting nonlinear program. The fundamental idea of Branch and Bound is that partitioning the search space allows to improve the underestimating functions on each subregion and thus tightens the linear relaxation. An improved linear relaxation then results in higher lower bounds and new incumbent solutions due to better starting points for the local search.

The algorithm starts with considering the problem on its complete feasible region. This problem is also called the root problem. Solving (R) yields a lower bound and a starting point for a local search in (P). If the local search is successful, the first incumbent solution has been found and an upper bound can be computed. Otherwise, the upper bound is initialized with $+\infty$. If lower and upper bounds match, a globally optimal solution has been found and

the procedure terminates. Otherwise, two new problems are constructed by dividing the feasible region of (P) using a subdivision of the box $[\underline{x}, \overline{x}]$ (*branching*). For each child the linear relaxation (R) is improved by adding further linearizations of nonlinear convex or convexified functions, interval gradient cuts for nonconvex constraints, and mixed-integer-rounding cuts. The new problems become children of the root problem, and the algorithm is applied recursively on each subproblem. This process constructs a tree of subproblems, the Branch and Bound tree. Since each node of the Branch and Bound tree has its own linear relaxation, the generated cutting planes need to be valid (i.e., underestimating the original functions) only on the corresponding subregion of the original feasible space. Hence, linearizations of convexified functions are generated with respect to α-underestimators that are valid for the corresponding subbox only, cf. (1.26).

The decision on how to subdivide a part of the search space (the *branching rule*) is based on the infeasibility of the solution of the linear relaxation, i.e., the fractionality of discrete variables and the distance between a quadratic function and its convexification [45]. Subdividing w.r.t. a variables x_j, $j \in B$, means to create two nodes with additional restrictions $x_j \leq \lfloor \hat{x}_j \rfloor$ and $x_j \geq \lceil \hat{x}_j \rceil$, respectively, where $\hat{x}_j \notin \mathbb{Z}$ is the value of x_j in a solution of (R). Subdividing w.r.t. a continuous variable x_j means to create one node with increased lower bound $\hat{x}_j$ for x_j and one node with decreased upper bound $\hat{x}_j$ for x_j. Thus, the α-underestimator (1.26) and the derived linearizations improve in the new nodes.

The choice of the next node to be processed (the *node selection rule*) is guided by the gap between the lower bounds of the nodes and the (uniform) upper bound [45].

Since the quadratic underestimators $q(x)$ are not updated during the algorithm and are computed by a heuristic method (Section 1.3.2), convergence of the gap between lower and upper bound to zero and locating a global optimum cannot be ensured for MINLPs with nonquadratic nonconvex terms. However, as our results in the next section show, LaGOs Branch and Cut algorithm is able to compute good local optimal points for difficult MINLPs. For some simplified models, LaGO is even able to reduce the gap below 1%.

1.4 Optimization results

We consider the power plant model presented in Section 1.2. At full load, the plant has an output of $\dot{W} = 750 \, \text{MW}$ electric power and $\dot{m}_{32} = 133.1 \, \text{t/h}$ process steam. Further assumptions are made:

- The *existence* of two heat-recovery steam generators is provided. They can be operated independently.
- To simplify the model, the nominal isentropic efficiencies $\eta_{s,N}$ of the steam turbines are fixed.

- The steam quality x_{32} must be within the steam vapor region and thus $0 \leq x_{32} \leq 1$. Due to this assumption there is no need for further binary variables that describe the thermodynamic properties of this stream.
- The **L**ower **H**eating **V**alue of methane is $LHV = 50.015 \, \text{kJ/kg}$. The fuel (natural gas) is approximated as methane.
- The fuel cost is $c_\text{f} = 4 \, \text{€/GJ}_\text{LHV}$, its real rate of increase is set to 1.0% per year, and the rate of inflation is set to $r_\text{i} = 2.0\%$.

1.4.1 Design optimization with consideration of partial-load behavior

The mathematical program that describes the plant design and plant operation at full and partial load consists of 2204 continuous, and 43 binary variables, and 2517 equations. LaGO computes 433 quadratic underestimators and 834 α-underestimators in the preprocessing (cf. Section 1.3.2).

The independent decision variables were given in Table 1.2, the respective optimization results for the design optimization considering the partial-load behavior are shown in the Tables 1.3 and 1.4. Note, that LaGO was not able to close the gap and thus prove global optimality for this model. Instead, we stopped the optimization after 24 hours (ca. 8000 Branch and Bound iterations on a Linux 2.6 AMD Athlon64 X2 6000+ computer with 3 GHz clock frequency and 4 GB RAM) at a gap between lower and upper bound of 15%. The presented design was found by LaGO after approx. seven hours.

Some of the results shown in Table 1.3 might be unexpected, since apparently the pinch temperature difference ΔT_{PINCH1} and the temperature difference in the superheater ΔT_{SPHT1} are rather high while the subcooling at the economizer outlet $\Delta T_{\text{Sub,ECON1,N}}$ is low. The first two aspects are not required for the operation of the plant at full-load. However, since we included also three partial-load operation points into the model, we forced the solver to find a design that is flexible enough to operate at the considered partial-load operation points.

Although the requested electric power output can be satisfied by the use of only two gas turbines and a subsequent water-steam cycle, the existence of four gas turbines enables a more flexible partial-load operation. The additional burners AB1 and AB2 and the water injector TMX were incorporated in the plant structure but should not be operated in the full-load case OP1. The operation of the gas turbines, the additional burners and the water injector in the respective partial-load cases is shown in Table 1.4.

The use of the additional burners and the water injector enables a plant operation without evaporation of the working fluid in the economizer during partial-load operation. The relatively high pressure in the condenser is necessary to get acceptable steam qualities x_{13} at the steam turbine outlet at partial-load operation. Some of the four existing gas turbines are shut down successively as partial load decreases. Thus, each gas turbine is tried to be operated at conditions as close as possible to the respective full-load conditions.

Table 1.3. Decision variables for the design optimization (nominal values, numbers refer to Figure 1.2). Operating points are considered according to Table 1.1. The first part shows the binary variables and the respective power output of the gas turbines in the full-load case, the second part gives the continuous variables. $\Delta T_{\mathrm{SPHT1}} = T_{22} - T_{12}$, $\Delta T_{\mathrm{PINCH1}} = T_6 - T_{11}$, $\Delta T_{\mathrm{Sub,ECON1}} = T_{\mathrm{saturated}}(p_{10}) - T_{10}$, $\Delta T_{\mathrm{cw}} = T_{16} - T_{15}$

Decision Variable	Value
V94.3A$_{\mathrm{a}}$	–
V94.3A$_{\mathrm{b}}$	1 (177.3 MW)
V94.3A$_{\mathrm{c}}$	1 (253.5 MW)
V94.2$_{\mathrm{a}}$	1 (93.1 MW)
V94.2$_{\mathrm{b}}$	1 (75.3 MW)
AB1 ($\dot{m}_{\mathrm{fuel}}$)	1 (0.0 kg/s)
AB2 ($\dot{m}_{\mathrm{fuel}}$)	1 (0.0 kg/s)
TMX1 ($\dot{m}_{\mathrm{water}}$)	1 (0.0 kg/s)
$p_{14,\mathrm{N}}$ [bar]	53.5
$p_{8,\mathrm{N}}$ [bar]	0.12
$\Delta T_{\mathrm{SPHT1,N/SPHT2,N}}$ [K]	102.3 / 89.7
$\Delta T_{\mathrm{PINCH1,N/PINCH2,N}}$ [K]	32.9 / 22.4
$\Delta T_{\mathrm{Sub,ECON1,N/Sub,ECON2,N}}$ [K]	2.2 / 1.5
$\Delta T_{\mathrm{cw,N}}$ [K]	4.0

Table 1.4. Operation of the four gas turbines, the additional burners AB1 and AB2 and the water injector at the different operation points.

Component	Variable	Unit	OP1	OP2	OP3	OP4
V94.3A$_{\mathrm{a}}$	$\dot{W}$	MW	–	–	–	–
V94.3A$_{\mathrm{b}}$	$\dot{W}$	MW	177.3	192.4	0.0	0.0
V94.3A$_{\mathrm{c}}$	$\dot{W}$	MW	253.5	253.5	216.3	0.0
V94.2$_{\mathrm{a}}$	$\dot{W}$	MW	93.1	149.3	149.3	149.3
V94.2$_{\mathrm{b}}$	$\dot{W}$	MW	75.3	0.0	149.3	132.5
$\dot{m}_{\mathrm{f,AB1}}$	$\dot{m}$	kg/s	0.00	0.02	0.00	0.00
$\dot{m}_{\mathrm{f,AB2}}$	$\dot{m}$	kg/s	0.00	0.00	0.00	0.02
$\dot{m}_{23}$	$\dot{m}$	kg/s	0.00	0.00	1.80	1.80

Note that conventional optimization procedures (e.g. [3, 7, 13, 16, 54–57]) might easily have failed to find a design that is feasible for some of the considered partial-load cases due to a violation of the pinch and steam quality constraints. That is why conventional optimization approaches with consideration of only one operation point require a subsequent variation of some decision variables using heuristic methods. These methods do not guarantee cost optimality of the final design obtained after these corrections, e.g., the subcooling at the economizer outlet $\Delta T_{\mathrm{Sub,ECON,N}}$ would be too high.

The objective function value is the levelized total revenue requirement TRR_{lev}. The economic assumptions were presented at the beginning of this section. Table 1.5 shows the PEC, the levelized fuel cost flow rate $\dot{C}_{\text{f,i}}$ for the respective load cases, and the TRR_{lev}. The fuel costs are calculated by

$$\dot{C}_{\text{f,i}} = c_{\text{f}}\, \text{LHV}\, \dot{m}_{\text{f,OP,i}}\, oh_{\text{OP,i}}, \tag{1.27}$$

where $\dot{m}_{\text{f,OP,i}}$ denotes the fuel mass flow rate and $oh_{\text{OP,i}}$ the operating hours in the respective load case. The respective costs for the different products (process steam and electricity) can be calculated by using an exergy-based cost allocation method [8].

Table 1.5. Operation costs for the first year with an optimized design. Operation according to Table 1.1. Fuel costs $\dot{C}_{\text{f}}$ are calculated by (1.27).

Costs	Unit	Value
PEC	Mio. €	197.2
$\dot{C}_{\text{f,OP1}}$	Mio. €/a	44.7
$\dot{C}_{\text{f,OP2}}$	Mio. €/a	32.3
$\dot{C}_{\text{f,OP3}}$	Mio. €/a	28.1
$\dot{C}_{\text{f,OP4}}$	Mio. €/a	23.4
TRR_{lev}	Mio. €/a	236.2

1.4.2 Sensitivity of an optimal design with respect to uncertain investment cost

In the following we present a first attempt to analyse the effect that uncertainty in the investment cost of a power plant component has on the optimal plant design. For this purpose we have considered only the design operation point (full-load case) with an operation of 8000 hours full load equivalent per year. We have chosen the two academic cost models shown in Figure 1.4 (cf. Section 1.2.2) for the high-, intermediate-, and low-pressure sections of the steam turbine and compared the designs that are found by the solver when either one of the models is applied or when a weighted average (A/B) is used for calculating the PEC of a steam turbine. Table 1.6 presents the results.

The PEC and TRR_{lev} values depend on the cost function that was used. But the differences among the results (structure and operation variables) of the three cases (A, B, and A/B) are practically negligible. Due to this, the TRR_{lev} in this analysis is basically influenced by the PEC of the turbines. Altogether, in this example, the design is significantly influenced by the fuel cost, i.e., the cost optimal design tends towards a thermodynamic optimal design. Hence, the results of the optimized designs recalculated with different

Table 1.6. Comparison of optimization and simulation results with different cost models: A, B (cf. Section 1.2.2), and a weighted average of the cost functions (A/B).

	Variable	Unit	Optimized with cost approach		
			A	B	A/B
	PEC	Mio €	225.42	159.96	192.70
	$\dot{C}_{\mathrm{f}}$	Mio €/a	170.74	170.62	170.68
	TRR_{lev}	Mio €/a	300.19	278.57	289.38
	p_{14}	bar	38.17	38.17	38.17
	T_{21}	°C	415.64	415.64	415.64
	$\dot{W}_{\mathrm{HPST}}$	MW	108.94	109.18	109.07
	$\dot{W}_{\mathrm{LPST}}$	MW	85.21	85.40	85.31
TRRlev	A	Mio €/a	*300.19*	300.20	300.20
calculated with	B	Mio €/a	278.58	*278.57*	278.57
cost approach	A/B	Mio €/a	289.39	289.39	*289.38*

cost models are almost the same (last three lines of Table 1.6). Apparently the impact of the different steam turbine cost functions is not strong enough to influence the optimized structure or the operating parameters.

1.4.3 Comparison with other MINLP solvers

Finally, we depict our experience with other MINLP solvers on the problem instance from Section 1.4.1. This instance will be made available in the MINLPLib [15]. We note that the results presented here do not allow conclusions about the performance of the considered solvers in general. Since the power plant model has been developed with having the solver LaGO in mind, it is in some sense tailored for LaGO, e.g., a block-separable formulation where nonconvex functions have only low dimensionality has been emphasized. For other solvers, a different but equivalent formulation might be advantageous. Further, the choice of LaGOs parameters is based on a long experience with similar power plant optimization models, while for the other solvers we have made only small adjustments to the default setting.

For this study, we have chosen the MINLP solvers that are available with GAMS 22.6 [24]. These are AlphaECP, BARON, Bonmin, DICOPT, LINDOGlobal, OQNLP, and SBB. Note that only BARON and LINDOGlobal can explicitly handle nonconvex problems. Unfortunately, testing our model on LINDOGlobal was not possible because the model size exceeded the restrictions on the GAMS/LINDOGlobal license.

We have run each solver once without providing any feasible starting point and once with the solution discussed in Section 1.4.1 as starting point in order to see whether the solver is able to improve it further. Recall, that this solution has an objective function value of 236.24 Mio. €/a and LaGO reported a lower

bound of 201.59 Mio. €/a when it was stopped. Each solver was run with a time limit of three hours and a relative gap tolerance of 1%. Iteration, node, or memory limits were turned off. The computer was a Linux 2.6 Intel Core 2 Duo T7500 laptop with 2.2 GHz clock frequency and 2 GB RAM. The results are summarized in Table 1.7 and discussed in more detail in the following.

AlphaECP [62] implements an extended cutting plane algorithm which guarantees global optimality for pseudo-convex problems. The algorithm constructs and improves a MIP approximation of the problem by constructing a (possibly shifted) linearization of violated constraints. If a (partial) solution of the MIP approximation is feasible to the original problem, an upper bound might be updated. Otherwise AlphaECP can do a local search in the MINLP, i.e., discrete variables are fixed and a NLP solver is called using the MIP solution as starting point. When AlphaECP was run with default parameter values no feasible point was found. Also setting the option `callnlpiter` to increase the number of NLP subsolver calls did not improve the situation.

BARON [52,53] implements a branch and reduce algorithm that is related to LaGOs methodology. Here, exact convex underestimators are constructed for nonconvex functions using a factorable reformulation of the model. These underestimators are used to generate a linear approximation that yields lower bounds. Further, constraint propagation and duality techniques are used to tighten the bounding box on the variables. On the power plant model, BARON with default parameters spend most of the time to do probing on variable bounds and did not find any feasible point. Thus, we rerun BARON with the restriction to do probing for at most 10 variables (option `PDo 10`). Now a feasible point with objective function value 255.13 Mio. €/a was found. The lower bound at the end of the timeperiod coincides with LaGOs lower bound of 201.59 Mio. €/a. Providing BARON with LaGOs solution as starting point did not result in a better point.

COIN-OR/Bonmin [11] implements both outer approximation and branch and bound algorithms. We have decided for the branch and bound algorithm since it seems to be better suited for nonconvex problems. Here, lower bounds are computed from solutions of the relaxed MINLP, i.e., the MINLP with dropped integrality restrictions, and upper bounds from solving the MINLP with fixed discrete variables. Bonmin found a feasible solution with objective function value 242.50 Mio. €/a after 41 minutes (1016 nodes). Unfortunately a further improvement was not possible because it stopped 10 minutes later due to a failing solve of a lower bounding problem by the NLP subsolver Ipopt. Running Bonmin with the provided starting point and with or without the options `num_resolve_at_root` and `num_resolve_at_node` set to 10 did not improve the situation.

DICOPT [18] implements an extended outer approximation algorithm. It iteratively solves MIP approximations generated by linearization of the MINLP and NLP subproblems obtained by fixing discrete variables in the MINLP. We have run DICOPT with stopping rule 0 and a very high value for `maxcycles`, so that it does not stop before the time limit is reached.

Unfortunately no feasible point was found. Providing a feasible starting point did not improve the situation.

OQNLP [59] is a heuristic multistart algorithm. The solver generates starting points via a scatter search or by random and uses them as starting points for local searches. For the power plant optimization model, OQNLP did not find any feasible point within the timelimit when run without a starting point. However, if we provide OQNLP with the feasible starting point, it was able to find an improved point with objective function value 235.97 Mio. €/a after a few seconds, but then was not able to improve this point further.

SBB [6] is a branch and bound algorithm similar to the one used in Bonmin. Lower bounds are computed by solving NLP subproblems obtained by partly fixing and partly relaxing integrality restrictions in the MINLP. If the solution of such an NLP is feasible for the MINLP, a new upper bound has been found. We run SBB with the option `acceptnonopt 1` to increase tolerance with NLP solver failures. For the power plant optimization model, SBB found a feasible solution with objective function value 236.76 Mio. €/a and reported a lower bound of 202.49 Mio. €/a. When we provided LaGOs solution as starting point, it found a feasible solution with objective function value 235.73 Mio. €/a and reported a lower bound of 203.91 Mio. €/a. Note, that SBBs lower bounds are only guaranteed for convex models.

Table 1.7. Best objective function values when running the power plant optimization instance from Section 1.4.1 with different MINLP solvers and a time limit of three hours. The third column gives the best values when a solution with objective function value 236.24 Mio. €/a is provided as a starting point to the solver.

Solver	best solution Mio. €/a	best solution with starting point Mio. €/a
LaGO	244.17	236.24
AlphaECP	fail	fail
BARON	255.13	236.24
Bonmin	242.50	fail
DICOPT	fail	fail
OQNLP	fail	235.97
SBB	236.76	235.73

From the results presented here, it can be observed that algorithms based on a linearization of nonlinear constraints like AlphaECP and DICOPT are not suited for this problem. We presume, that this is due to the disregard of nonconvex behavior in the construction of the MIP approximation.

On the other hand, NLP based branch and bound algorithms like Bonmin and SBB seem to behave well on this problem instance. Even though the NLP relaxations solved in these algorithms constitute nonconvex problems, the NLP solver (CONOPT in the SBB run and Ipopt in the Bonmin run)

seem to be able to find good solutions. The proper treatment of the 45 binary variables is then the remaining task of the MINLP solver. The moderate size of this combinatorial part (when compared to the 2204 continuous variables) is probably advantageous for SBB and Bonmin.

The performance of BARON and LaGO indicate that the extra effort for the convexification of nonconvex functions in order to compute true lower bounds results in longer running times, either because the linear relaxation does not give useful information for branching decisions or because the starting points used for local searches are worse when compared with an NLP based branch and bound algorithm.

Finally, the slightly improved solution points found by OQNLP and SBB when started with LaGOs solution indicate that LaGO could benefit from a kind of local branching heuristic that searches for solutions that improve a recently found incumbent solution.

1.5 Conclusions

We presented a MINLP formulation of a model for a complex cogeneration plant and the Branch and Cut based solver LaGO that is used to solve the model. The operation under different load conditions and under uncertainties in the investment cost are discussed. It has been shown, that the consideration of only one typical design point is not sufficient, since the operation at some partial-load operation points may become impossible due to thermodynamic constraints of the plant so that heuristics methods would be required to satisfy the further requirements that are posed by partial-load operation. The presented method overcomes this problem by considering required partial-load conditions already in the design optimization and thus allows to find a cost optimal plant design that is feasible for the full-load and the considered partial-load cases. Of course, this enhancement comes in hand with a significantly more complex optimization model.

Further, we have seen that the optimized design is insensitive to changes in the steam turbine cost function. An approach to extend this sensitivity analysis towards the computation of a design that is robust w.r.t. changes in a cost function is to use a (weighted) average of cost function scenarios. However, in order to avoid that this approach collapses to the computation of a design that is just optimal for the averaged costs, one should also introduce recourse decisions into the model, i.e., allow the off-design variables to take different values in each considered cost scenario.

Due to the formulation of the problem as a system of equations, an extension of the model, such as the consideration of more partial-load operation points or more complex energy conversion plants is possible without any significant modelling effort. In particular there is no need to define a calculation order for the streams and components because the MINLP solver optimizes the entire problem simultaneously. We have seen that even though a typical

engineering problem, when formulated as mathematical program, requires sophisticated solution algorithms, the investment in a complex MINLP model and powerful solver can be rewarded by realistic solutions of good quality.

From the algorithmic perspective, the main difficulty of the considered MINLP models are the nonlinearities and nonconvexities introduced by the equations for the thermodynamic behavior of the plant. LaGO handles nonconvex functions by computing quadratic underestimators that can be convexified using α-underestimators. While it is possible to compute good local minimal points, closing the gap between lower and upper bounds, and thus proving global optimality of the computed solutions, is still exceptional for complex MINLPs. To bring LaGO closer to this goal, ongoing research focuses on the additional generation of quadratic underestimators during the branch-and-bound process. Additional room for improvement is presented by the branching and node selection rules and a further exploration of block-separability. Also the consideration of a mixed-integer linear relaxation could help to improve convergence.

References

1. C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical advances. *Computers & Chemical Engineering*, 22:1137–1158, 1998.
2. C. S. Adjiman and C. A. Floudas. Rigorous convex underestimators for general twice-differentiable problems. *Journal of Global Optimization*, 9:23–40, 1997.
3. T. Ahadi-Oskui. Optimierung des Entwurfs komplexer Energieumwandlungsanlagen. In *Fortschritt-Berichte*, number 543 in Series 6. VDI-Verlag, Düsseldorf, Germany, 2006.
4. T. Ahadi-Oskui, H. Alperin, I. Nowak, F. Cziesla, and G. Tsatsaronis. A Relaxation-Based Heuristic for the Design of Cost-Effective Energy Conversion Systems. *Energy*, 31:1346–1357, 2006.
5. T. Ahadi-Oskui, S. Vigerske, I. Nowak, and G. Tsatsaronis. Optimizing the design of complex energy conversion systems by branch and cut. Preprint 07-11, Department of Mathematics, Humboldt-University Berlin. available at `http://www.math.hu-berlin.de/publ/pre/2007/P-07-11.pdf` and submitted, 2007.
6. ARKI Consulting & Development A/S and GAMS Inc. SBB. `http://www.gams.com/solvers/solvers.htm#SBB`, 2002.
7. J. Axmann, R. Dobrowolski, and R. Leithner. Evolutionäre Algorithmen zur Optimierung von Kraftwerkskonzepten und Anlagenbauteilen. In *Fortschritt-Berichte*, number 438 in Reihe 6, pages 251–265. VDI-Verlag, Düsseldorf, Germany, 1997.

8. A. Bejan, G. Tsatsaronis, and M. Moran. *Thermal Design and Optimization*. J. Wiley, New York, USA, 1996.

9. M.S. Boddy and D.P. Johnson. A new method for the global solution of large systems of continuous constraints. In Ch. Bliek, Ch. Jermann, and A. Neumaier, editors, *Global Optimization and Constraint Satisfaction*, volume 2861 of *Lecture Notes in Computer Science*, pages 143–156. Springer, 2003.

10. C. G. E. Boender and H. E. Romeijn. Stochastic methods. In R.Horst and P. Pardalos, editors, *Handbook of Global Optimization*, pages 829–869. Kluwer Academic Publishers, 1995.

11. P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. Technical Report IBM Research Report RC23771, Carnegie Mellon University, IBM Research Division, Faculté des Sciences du Luminy, University of Bologna, 2005. to appear in Discrete Optimization.

12. C. Bouvy. Kombinierte Struktur- und Einsatzoptimierung von Energieversorgungssystemen mit einer Evolutionsstrategie. Shaker Verlag, Aachen, 2007.

13. C. Bouvy and S. Herbergs. Mehrkriterielle Optimierung dezentraler Energieversorgungssysteme mit evolutionären Algorithmen. In *Optimierung in der Energiewirtschaft*, number 1908 in Fortschitt-Berichte, pages 265–277. VDI-Verlag, Düsseldorf, Germany, 2005.

14. J. Bruno, F. Fernandez, F. Castells, and I. Grossmann. A Rigorous MINLP Model for the Optimal Synthesis and Operation of Utility Plants. *Transactions of the Institution of Chemical Engineers*,, 76:246–258, 1998.

15. M. R. Bussieck, A. S. Drud, and A. Meeraus. MINLPLib - A Collection of Test Models for Mixed-Integer Nonlinear Programming. *INFORMS Journal on Computing*, 15(1):114–119, 2003.

16. F. Cziesla. Produktkostenminimierung beim Entwurf komplexer Energieumwandlungsanlagen mit Hilfe von wissensbasierten Methoden. In *Fortschritt-Berichte*, number 438 in Reihe 6. VDI-Verlag, Düsseldorf, 2000.

17. DEWI, E.ON Netz, EWI, RWE Transportnetz Strom, and VE Transmission. Energiewirtschaftliche Planung für die Netzintegration von Windenergie in Deutschland an Land und Offshore bis zum Jahr 2020. `http://www.eon-netz.com/Ressources/downloads/dena_haupt_studie.pdf`, February 2005. Studie im Auftrag der Deutschen Energie-Agentur GmbH (dena).

18. M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

19. A. Epe, C. Küchler, W. Römisch, S. Vigerske, H.-J. Wagner, C. Weber, and O. Woll. Stochastic programming with recombining scenario trees – optimization of dispersed energy supply. accepted for "Optimization in the Energy Industry" (J. Kallrath, P. Pardalos eds.), 2007.

20. M. Fenski. Prozessanalyse mittels selbstorganisierender neuronaler Karten am Beispiel eines kombinierten Gas- und Dampfturbinenwerks mit Fernwärmeauskopplung. Diplomarbeit, Technische Universität Berlin, 2006.

21. R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Mathematical Programming*, 66(3(A)):327–349, 1994.

22. C. A. Floudas, I. G. Akrotirianakis, C. Caratzoulas, C. A. Meyer, and J. Kall-rath. Global optimization in the 21st century: Advances and challenges. *Computers & Chemical Engineering*, 29(6):1185–1202, 2005.
23. C.A Frangopoulos. Optimal Synthesis and Operation of Thermal Systems by the Thermoeconomic Functional Approach. *Transactions of the ASME, Journal of Engineering for Gas Turbines and Power*, 114:707–714, 1992.
24. GAMS Development Corp. *GAMS - The Solver Manuals*. Washington DC, 2007.
25. Gas Turbine World 2006 Handbook. Pequot Publishing Inc., Fairfield, 2006.
26. M. Gebhardt, H. Kohl, and T. Steinrötter. Preisatlas, Ableitung von Kosten-funktionen für Komponenten der rationellen Energienutzung. `http://www.iuta.de/thermodynamik/Preisatlas_Download.htm`, June 2002.
27. R. Horst and P. Pardalos. *Handbook of Global Optimization*. Kluwer Academic Publishers, 1995.
28. K. Hüttenhofer and A. Lezuo. Cogeneration Power Plant Concepts Using Advanced Gas Turbines. *VGB PowerTech*, 81(6):50–56, 2001.
29. M. Jüdes, F. Cziesla, W. Ahrens, J. Petri, and G. Tsatsaronis. Neuronale Netze als Hilfsmittel zur Betriebsbewertung am Beispiel einer industriellen Kraft-Wärme-Kopplungsanlage. In *Optimierung in der Energieversorgung*, number 1792 in VDI-Berichte, pages 201–213. VDI-Verlag, 2003.
30. M. Jüdes and G. Tsatsaronis. Cost Effective Design Optimization and Maintenance Strategies with Consideration of the Partial Load Behavior of Power Plants. In A. Mirandola, Ö. Arnas, and A. Lazzaretto, editors, *Proceedings of ECOS 2007, Padova, Italy*, volume **I**, pages 251–258, June, 25-28 2007.
31. M. Jüdes and G. Tsatsaronis. Design Optimization of Power Plants by Considering Multiple Partial Load Operation Points. In *2007 ASME International Mechanical Engineering Congress and Exposition, November 11-15, Seattle, Washington USA, IMECE2007*. ASME, 2007.
32. O. Knacke, O. Kubaschewski, and K. Hesselmann. *Thermochemical Properties of Inorganic Substances*. Springer Verlag, Berlin, Germany, 1991.
33. T. Kohonen. *Self-Organizing Maps*. Springer Verlag, Heidelberg, 2001.
34. M. Krämer. Modellanalyse zur Optimierung der Stromerzeugung bei hoher Einspeisung von Windenergie. In *Fortschritt-Berichte*, number 492 in Reihe 6. VDI-Verlag, Düsseldorf, Germany, 2003.
35. H. Marchand and L.A. Wolsey. Aggregation and mixed integer rounding to solve MIPs. *Operations Research*, 49(3):363–371, 2001.
36. J.R. Muñoz and M.R. von Spakovsky. A Decomposition Approach for the Large Scale Synthesis/Design Optimization of Highly Coupled, Highly Dynamic Energy Systems. *International Journal of Applied Thermodynamics*, 4(1):19–33, 2001.
37. J.R. Muñoz and M.R. von Spakovsky. The Application of Decomposition to the Large Scale Synthesis/Design Optimization of Aircraft Energy Systems. *International Journal of Applied Thermodynamics*, 4(2):61–74, 2001.
38. G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, 1988.
39. A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge Univ. Press, Cambridge, 1990.
40. A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. In *Acta Numerica*, volume 13, chapter 4, pages 271–370. Cambridge University Press, 2004.

41. A. Neumaier and S. Vigerske. personal communication, October 2006.
42. I. Nowak. *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*. Birkhäuser Verlag, Basel, Schweiz, 2005.
43. I. Nowak, H. Alperin, and S. Vigerske. LaGO - an object oriented library for solving MINLPs. In Ch. Bliek, Ch. Jermann, and A. Neumaier, editors, *Global Optimization and Constraint Satisfaction*, volume 2861 of *Lecture Notes in Computer Science*, pages 31–43. Springer, 2003.
44. I. Nowak and S. Vigerske. LaGO - Lagrangian Global Optimizer. `https://projects.coin-or.org/LaGO`.
45. I. Nowak and S. Vigerske. LaGO - a (heuristic) branch and cut algorithm for nonconvex MINLPs. Humboldt-Universität zu Berlin, Institut für Mathematik, Preprint 06-24, and accepted for publication in Central European Journal of Operations Research, 2006.
46. K. Papalexandri, E. Pistikopoulos, and B. Kalitventzeff. Modelling and Optimization Aspects in Energy Management and Plant Operation with Variable Energy Demands-Application to Industrial Problems. *Computers & Chemical Engineering*, 22(9):1319–1333, 1998.
47. D. Paulus. Single-Component Optimal Heat Exchanger Effectiveness using Specific Exergy Costs and Revenues. In S. Kjelstrup, J. E. Hustad, T. Gundersen, A. Røsjorde, and G. Tsatsaronis, editors, *Proceedings of ECOS 2005, Trondheim, Norway*, volume **III**, pages 1407–1414, June, 20-22 2005.
48. R. Peltier. TOP PLANTS: Tenaska Virginia Generating Station, Scottsville, Virginia. *POWER magazine*, 151(9):50–53, 2007.
49. R. Romero, A. Zobaa, E. Asada, and W. Freitas. Mathematical optimisation techniques applied to power systems operation and planning. *International Journal of energy technology and policy*, 5(4):393–403, 2007.
50. T. Savola. Simulation and Optimisation of Power Production in Biomass-Fuelled Small-Scale CHP plants. Licentiate's thesis, Helsinki University of Technology, March 2005.
51. SOFBID. EBSILONProfessional. `http://www.sofbid.com`.
52. M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.
53. M. Tawarmalani and N.V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, 2002.
54. G. Tsatsaronis, F. Cziesla, and Z. Gao. Avoidable Thermodynamic Inefficiencies and Costs in Energy Conversion Systems. Part 1: Methodology. In N. Houbak, B. Elmegaard, B. Qvale, and M.J. Moran, editors, *Proceedings of ECOS 2003, Copenhagen, Denmark* , volume **II**, pages 809–814, June 30 - July 2 2003.
55. G. Tsatsaronis, K. Kapanke, and A.M. Blanco Marigorta. Exergoeconomic estimates for a novel process with integrated CO_2 capture for the production of hydrogen and electric power. In C.A. Frangopoulos, C.D. Rakopoulos, and G. Tsatsaronis, editors, *Proceedings of ECOS 2006, Aghia Pelagia, Crete, Greece*, volume **III**, pages 1581–1591, July, 12-14 2006.
56. G. Tsatsaronis, L. Lin, J. Pisa, and T. Tawfik. Thermoeconomic Design Optimization of a KRW-based IGCC power plant, Final Report submitted to Southern Company Services and the U.S. Department of Energy. DE-FC21-89MC26019, Center for Electric Power, Tennessee Technological University, 1991.

57. G. Tsatsaronis, T. Tawfik, L. Lin, and D.T. Gallaspy. Exergetic Comparison of two KRW-based IGCC Power Plants. *J. Eng. Gas Turbines and Power*, pages 219–299, 1994.
58. R. Turton, R. Bailie, W. Whiting, and J. Shaeiwitz. *Analysis, Synthesis and Desing of Chemical Processes*. Prentice Hall PTR, New Jersey (USA), 1984.
59. Zsolt Ugray, Leon Lasdon, John Plummer, Fred Glover, Jim Kelly, and Rafael Martí. Scatter search and local NLP solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007.
60. G. Ulrich. *A guide to chemical engineering process design and economics*. John Wiley&Sons, Inc., New York (USA), 1984.
61. W. Wagner. *Properties of Water and Steam*. Springer, Berlin, 1998.
62. T. Westerlund and R. Pörn. Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering*, 3:253–280, 2002.