Optimizing the design of complex energy conversion systems by Branch and Cut

Turang Ahadi-Oskui[1], Stefan Vigerske[2,3],
Ivo Nowak[2], and Georgios Tsatsaronis[1]

Abstract

The paper examines the applicability of mathematical programming methods to the simultaneous optimization of the structure and the operational parameters of a combined-cycle-based cogeneration plant. Thus, the optimization problem is formulated as a highly non-convex mixed-integer nonlinear problem (MINLP) and solved by the MINLP solver LaGO. The algorithm generates a convex relaxation of the MINLP and applies a Branch and Cut algorithm to the relaxation. Numerical results for different demands for electric power and process steam are discussed and a sensitivity analysis is performed.

1 Introduction

The design of large-scale energy conversion systems is a highly complex process even when only the steady-state case is considered. Design optimizations for new projects are usually limited to sensitivity analyses of existing plants or application of heuristic rules [3]. The increasing computing power and the further development of optimization algorithms in the last years allow now the application of novel computer-aided tools. This paper examines the applicability of mathematical programming methods to the optimization of the design of a combined-cycle-based cogeneration plant. The optimization is not limited to operational parameters alone, but also searches for an appropriate structure of the plant. Thus, a highly non-convex mixed-integer nonlinear optimization problem (MINLP) is formulated. The goal of the optimization is to find a plant design with minimum levelized total cost that fulfills the user specified demands for electric power and process steam.

Due to the presence in the MINLP of integer variables (to model the structure of the plant) and nonconvex equations (to model the thermodynamic behavior of the components), an optimization by local search methods is not sufficient. Thus, traditional methods for convex global optimization might fail to find a feasible point or stop in a local optimum

[1]Technical University Berlin, Institute for Energy Engineering, Marchstr. 18, 10587 Berlin, Germany
[2]Humboldt University Berlin, Department of Mathematics, Unter den Linden 6, 10099 Berlin, Germany
[3]Corresponding author. E-Mail: stefan@math.hu-berlin.de

that is far from a best possible point. To overcome the dependency from the starting point in local search methods, stochastic global optimization is based on the idea that the more local searches are started from different starting points, the higher is the probability to find a good local optimal point of the problem. Hence, these methods generate starting points randomly in the search space, where the amount of local searches that are carried out is reduced by clustering the sample set [12]. In neighborhood search algorithms, the sample points are generated in growing neighborhoods of a prior found best local optimal point [9]. Even though stochastic methods have proven to be a powerful heuristic for the global optimization of small nonconvex problems, they suffer on the curse of dimensionality when applied to high dimensional MINLPs with a highly nonconvex feasible set.

For the deterministic global optimization of a nonconvex MINLP [13, 14], mainly two approaches exist. In successive outer-approximation algorithms [5, 6, 21] an initial relaxation is iteratively solved and improved until a feasible point of the MINLP is found, which is then also a globally optimal point. In branching methods the feasible set is subdivided into smaller subregions (branching). If an optimal point of a subproblem is not found or global optimality (for the subproblem) cannot be verified, the subregion is further subdivided. In Branch and Bound algorithms [1], lower bounds for each subregion are compared with an upper bound on the global optimum. Lower bounds from a linear relaxation that is generated by cutting planes lead to Branch and Cut algorithms [17–19]. In this paper, the Branch and Cut algorithm that is implemented in the MINLP solver LaGO (**La**grangian **G**lobal **O**ptimizer) [14–16] was used to optimize the design of an energy conversion system.

In the next section, we introduce our model of a combined-cycle-based cogeneration plant. Section 3 describes the Branch and Cut algorithm that was applied here. Section 4 presents some numerical results.

2 Optimization Problem

2.1 Combined-cycle based cogeneration plant

The energy conversion system is based on a combined-cycle process, which is a combination of a gas turbine process with a steam turbine process, see Figure 1 for a simple configuration. The entire fuel is, in general, fed to the gas turbine which represents the topping cycle and produces about 2/3 of the electric power of the overall process. The energy of the gas turbine exhaust gas is used within a heat recovery steam generator (HRSG) to produce superheated steam. The steam is fed to a steam turbine to generate additional electric power. Since the steam turbine process operates at a lower temperature level, it is called the bottoming cycle. The combination of both processes working at different temperature levels allows a very efficient utilization of the fuel energy. With an efficiency of up to 59%, combined-cycle processes reach the highest efficiency for the production of electric power from fossil fuels today. Looking at the environmental perspective, in addition to the high efficiency, the use of natural gas as fuel contributes to relatively low specific CO_2-emissions.

A combined-cycle process can easily be converted into a cogeneration plant, in which

process steam is produced in addition to electric power. The required steam is extracted from the steam cycle, in general from a steam turbine stage, at an appropriate pressure.

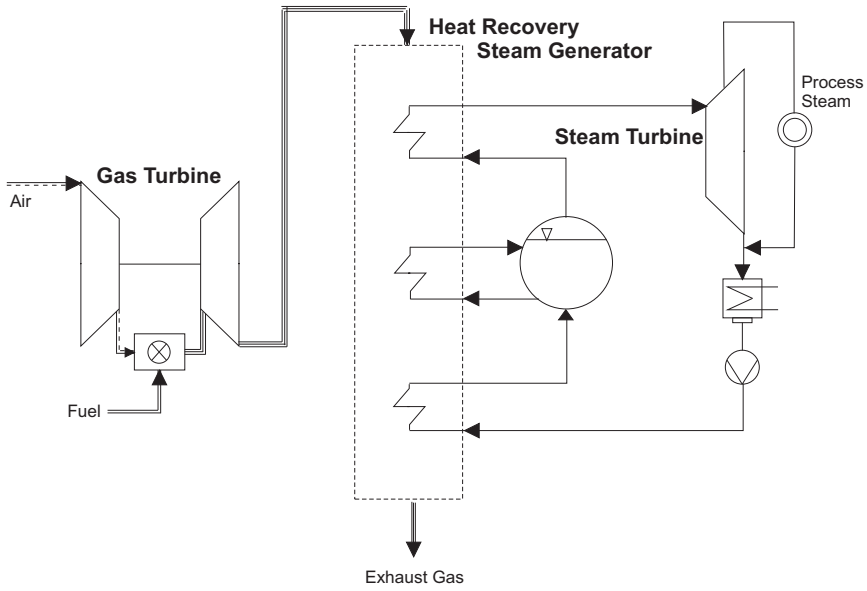

Figure 1: Schematic of a combined-cycle power plant with steam extraction

2.2 Superstructure

The goal of the optimization is to find a design of the combined-cycle-based cogeneration plant with minimum levelized total costs. Starting point for the simultaneous optimization of the structure and the process variables of the design is a so-called superstructure. The superstructure of the cogeneration plant represents a superior process flow sheet which combines a variety of different plant designs. It contains all possible plant components, necessary for accomplishing the predefined task, and all possible connections between them. Depending on the user-specified electric power and process steam demands, the optimization algorithm finds an optimized structure within the superstructure and the associated values of the process variables. The superstructure was developed by combining various existing combined-cycle plant designs. The superstructure was designed for an electric power output between 50 MW and 400 MW and a process steam production of up to a total of 500 t/h at up to three different pressure levels. A specific design is determined by a set of 28 binary structural variables and 48 continuous process variables. A structural variable decides over the existence of a plant component or a stream connection whereas the process variables specify mass flow rates, temperatures, and pressures of process streams as well as efficiencies of plant components.

A superstructure of the present complexity cannot be displayed concisely in one coherent process flow sheet. Therefore, the superstructure is divided into three parts: the gas

turbine system, the water-steam-cycle, and the exhaust gas path. Figures 2, 3, and 4 show the considered plant components (numbered from 501 to 533) and the process streams in between them. The structural decisions are represented by dots, which mark either an "and" decision or an "or" decision. In the following, they are discussed in more details.

2.2.1 Gas Turbine

Contrary to industrial standards, the gas turbine system is not a fixed system but is composed of individual components to be designed, see Figure 2 for a process flow sheet. This increases the flexibility for the optimization and allows the creation of innovative designs. The first structural option determines whether intercooling with staged compression is included into the design or not. The intercooling can be realized by an injection cooler (Component No. 503 in Figure 2) or by a surface heat exchanger (502). Before entering the combustion chamber, the compressed air can be preheated in an air preheater (505). During the expansion process, reheating of the exhaust gas can be realized in a second combustion chamber (508). The remaining structural variables for the gas turbine system decide over fuel preheating (510) and a possible steam injection into each combustion chamber. The streams at the bottom part of the flowsheet represent the cooling air for the turbine blades. The exhaust gas stream is finally fed to the HRSG.

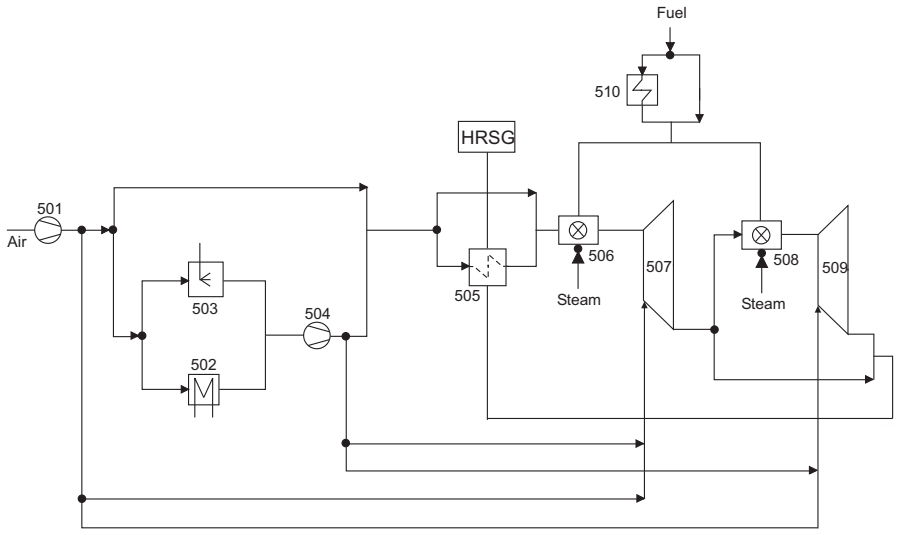

Figure 2: Gas turbine part of the superstructure

2.2.2 Water/Steam Cycle

The water-steam-cycle consists of the water-side of the HRSG, the steam turbines, and the condensing part including the feedwater tank. The flowsheet for the water/steam cycle is shown in Figure 3. Steam can be produced at up to three different pressure levels (high, medium, and low). The medium and low-pressure levels are optional and their existence is determined by structural variables. Each pressure level consists of a feedwater pump, an economizer, an evaporator, and a superheater. The economizer of the low-pressure level (522) and the superheaters of the low and medium-pressure levels (520 and 517) represent also structural decisions. The generated steam is expanded in up to three steam turbines whereas the medium-pressure steam turbine and the associated reheater are optional. If required, process steam can be extracted at four different locations (PD1 to PD4). After the condensate pump, there is an option to implement a condensate reheater (523).

2.2.3 Exhaust Gas Path

The exhaust gas path represents the gas-side of the HRSG. In Figure 4, every rectangle denotes a surface heat exchanger. The structural variables determine basically the configuration of the heat exchangers (order, in series or in parallel). The path through the HRSG is, of course, coupled to the existence of steam at the low or medium-pressure levels. Finally, there is an option to implement up to two duct burners (511, 513) into the HRSG.

2.3 Model of the superstructure

The model of the superstructure describes the thermodynamic behavior of the cogeneration plant, performs the economic analysis and yields the levelized total costs for the overall plant. The equations of the model can be divided into different categories. The first category of equations describes the logic of the superstructure, i.e., how the plant components are connected to each other, under which conditions the different setups are possible and which restrictions have to be fulfilled in each case.

The second category of equations consists of the models that specify the thermodynamic behavior of the plant components. The models of the plant components should on one side be kept as simple as possible, to facilitate the work of the optimization algorithm and to shorten the total computing time. On the other side, they have to guarantee a certain level of accuracy. The model of a plant component consists of mass and energy balances as well as characteristic functions which describe the thermodynamic behavior of the component.

To simulate the behavior of the plant components, thermodynamic property equations for all substances and mixtures used in the plant (e.g., exhaust gas, water, and steam) are needed. The exhaust gas is treated as an ideal mixture of ideal gases and the properties are calculated using the equations given in [11].

The third category of equations is formed by the functions approximating the purchase equipment costs. These functions represent the most imprecise part of the model since real cost data are hardly available. In this work we use cost functions either from the

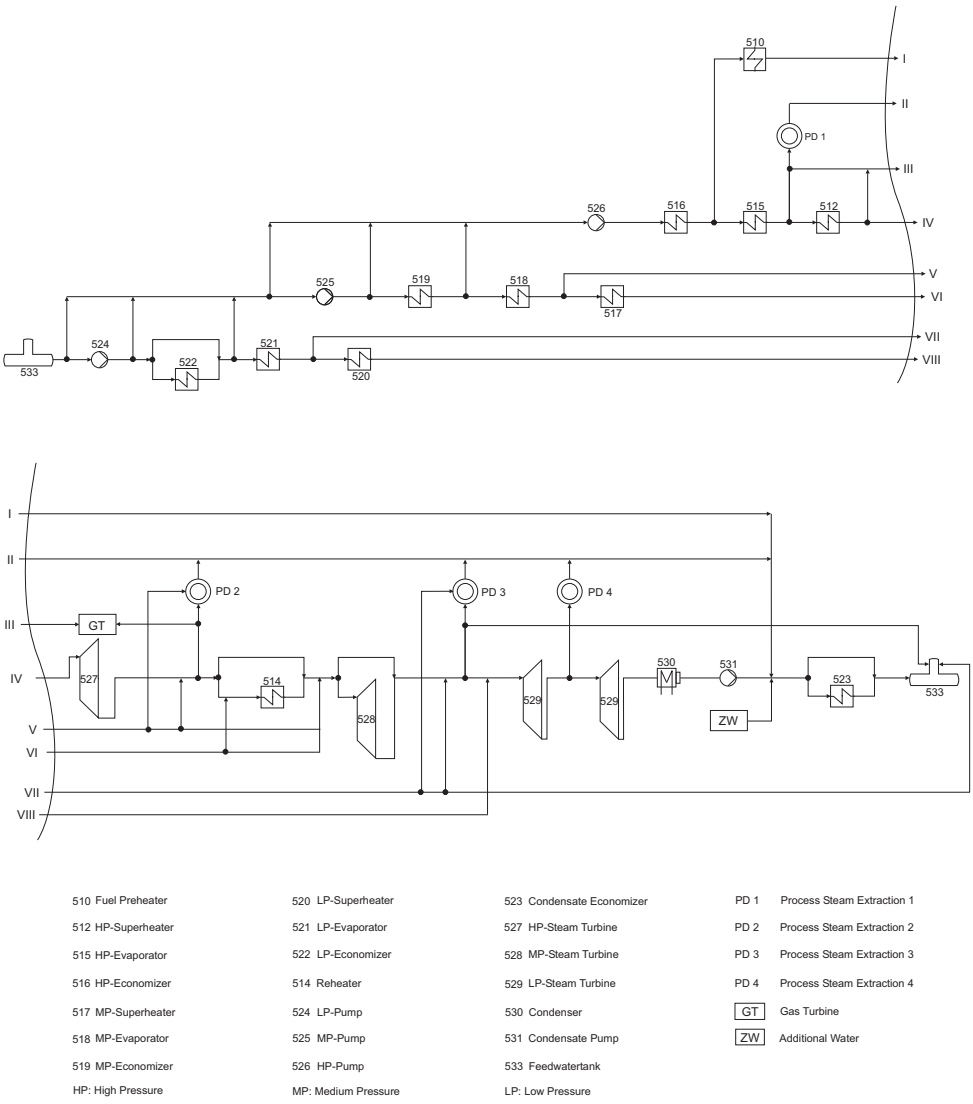

Figure 3: Water-steam-cycle of the superstructure

literature [3, 20] or derived from available cost data [4, 8].

The last category of equations belongs to financial mathematics. With these equations the economic analysis, which yields the levelized total costs for the cogeneration plant, is conducted. The cost of the plant represent the total revenue which is required to maintain a sound economic operation of the plant. The total cost consists of investment cost, operating and maintenance cost, fuel cost, depreciation, taxes, insurance, interests, and a minimum return on investment. The economic analysis is performed according to the Total Revenue Requirement (TRR) method [3]. The levelization of the costs takes the time value of money into account.

The model for the MINLP solver LaGO is programmed as one coherent system of

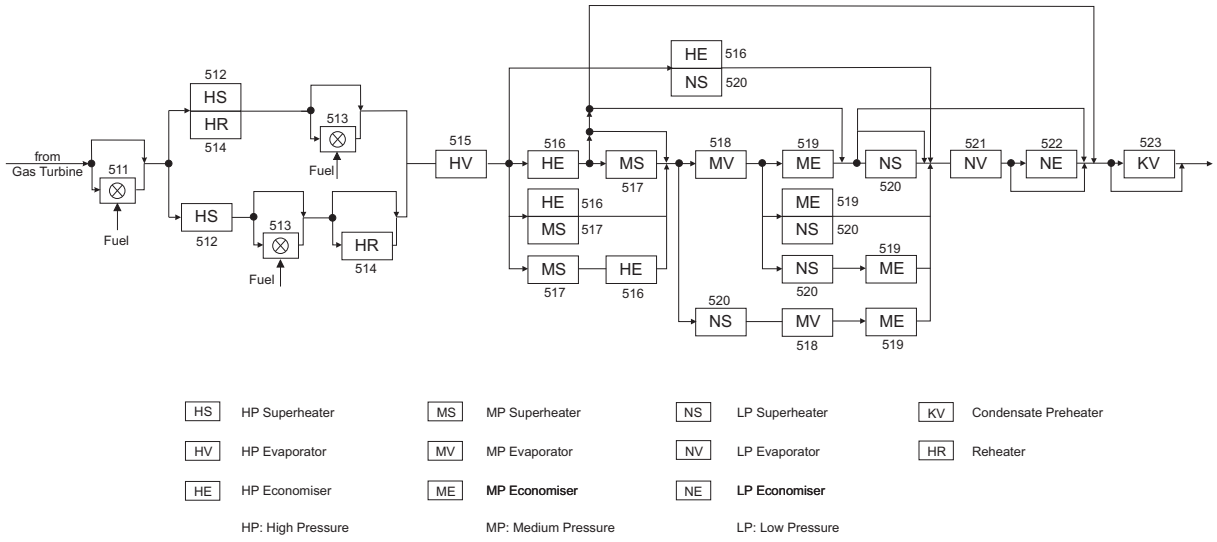

Figure 4: Exhaust gas path through the HRSG of the superstructure

equations using the mathematical modeling language GAMS [7]. Since the MINLP solver requires first and second derivatives of the equations of the model, external software (e.g., for the simulation of single components of the plant) cannot be employed. The complexity of the integrated nonlinear and nonconvex functions is limited to a certain degree to assure the convergence of the solver. Therefore, for calculating the properties of water and steam in LaGO, the IAPWS-IF 97 functions [22] with up to 60 terms and exponents of ± 50 are approximated by less complicated polynomials of lower degree.

Another restriction of the mathematical programming algorithm is that the designer cannot determine a calculation sequence because the entire model is computed simultaneously. Here, distinction of cases can only be realized by additional binary variables, which increase the complexity of the problem. Moreover, even though inactive parts of the model have to be decoupled from the objective function, the equations of inactive plant components have to be satisfied anyhow.

Modeling the overall plant performance with a system of equations has advantages in conjunction with recirculated streams because there is no need for the designer to determine a calculation order. Moreover, additional plant components can be easily integrated into the model by connecting them with the upstream and downstream components.

Due to great differences in the order of magnitude of the variables, the entire model of the superstructure has to be scaled to ensure an equal treatment by the solver regarding the compliance of all equations. In addition, to facilitate the work of the solver and to improve the quality of the results upper and lower bounds for all variables of the model are needed. This causes significant additional work for a model with 1308 variables and 1640 constraints. The optimization is controlled by the MINLP solver LaGO which accesses the GAMS model of the superstructure, performs the optimization, and stores the best

solutions of a run in a data file for a subsequent analysis.

3 MINLP optimization by Branch and Cut

In this section we briefly describe the algorithm that is used to solve the design problem described in Section 2. The model can be formulated as the following mixed-integer nonlinear program (MINLP):

$$\begin{array}{rl} \min & b_0^T x \\ \text{such that} & h(x) \leq 0, \\ & x \in [\underline{x}, \overline{x}], \\ & x_j \in \{0,1\}, \quad j \in B, \end{array} \tag{P}$$

where $B \subseteq \{1,\ldots,n\}$, c, $\underline{x}$, $\overline{x} \in \mathbb{R}^n$, $\underline{x} \leq \overline{x}$, and $h : \mathbb{R}^n \to \mathbb{R}^m$ is twice continuously differentiable. Without loss of generality, we have assumed a linear objective function here and replaced equality constraints by two inequalities in this general formulation.

We assume to have procedures for evaluating function values, gradients, and Hessians of the functions $h_i(x)$, $i = 1,\ldots,m$. The restriction to black-box functions has the advantage that our algorithm can handle more general functions than other deterministic solvers for nonconvex MINLPs. On the other hand, we are not able to use advanced convexification and box reduction techniques (as in [13,19]). Hence, for some components of our algorithm we are restricted to sampling methods.

The proposed algorithm follows a Branch and Bound scheme to search for a global optimum of (P). It starts by considering the original problem with its complete feasible region, which is called the root problem. A lower bound on the global optimum of (P) is computed by solving a linear outer-approximation of (P). An upper bound $\overline{v}$ is computed by finding a local optimum of (P). If the 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) (branching). 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.

The gap between the lower bound $\underline{v}(U)$ of a node U and the global upper bound $\overline{v}$ is diminished by improving the linear outer-approximation and by computing local optimal points. If such a point is found and the upper bound $\overline{v}$ is improved, nodes of the tree, the lower bound of which exceeds $\overline{v}$, are pruned. The process of branching and bounding is performed until no unprocessed nodes are left or the gap has been sufficiently reduced.

The outer-approximation is improved by cutting planes that are derived from a (nonlinear) convex outer-approximation of (P). Further, the efficiency of the algorithm is enhanced by box reduction techniques that allow to tighten the box $[\underline{x}, \overline{x}]$ (or a subbox) and can discover infeasibility.

In the following, we briefly explain the above mentioned components of the algorithm. For more details we refer to [16]. We start with a reformulation of (P) into a block-separable form (Subsection 3.1). Subsection 3.2 depicts the steps to the linear outer-approximation

of (P). Box reduction algorithms are explained in Subsection 3.3. Finally, the components are brought together in a Branch and Cut algorithm (Subsection 3.4).

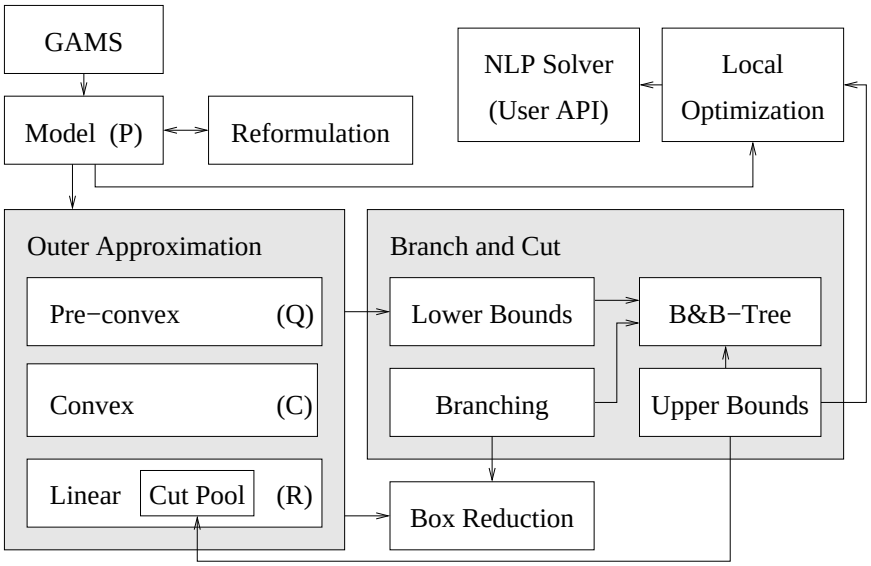

Figure 5: Structure of the MINLP solver LaGO.

3.1 Block-separable reformulation

Many real-world optimization problems have a natural separable structure, which is often related to components of the underlying model. This structure allows all functions of (P) to be represented as a sum of sub-functions each one of which depends on a small number of variables. Functions having such a property are called *block-separable*. For the model of the cogeneration plant (Section 2), the sub-functions are the thermodynamic equations for each component of the plant. Since each of them depends only on a small number of variables, this model has a clear separable structure that can be exploited by the optimization algorithm.

LaGO automatically identifies a block-separable structure of the black-box functions of (P) and reformulates them as

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

where $Q_{i,k}$ and $N_{i,k}$ are pairwise disjunct index sets (also denoted as *block*) of quadratic and nonlinear nonquadratic variables that appear in $h_i(x)$. The block-separable structure allows to distinguish between quadratic and nonquadratic parts of a function, and to treat each block separately if advantageous.

Furthermore, the sparsity graph E_i^{sparse} of the Hessian for each function is computed. This graph has the set $V_i := \bigcup_{k=1}^{q_i} Q_{i,k} \bigcup_{k=1}^{p_i} N_{i,k}$ of nonlinear variables as nodes and there is an edge between nodes j and j' if there is a point $\hat{x} \in [\underline{x}, \overline{x}]$ such that $(\nabla^2 h_i(\hat{x}))_{j,j'} \neq 0$, i.e., the variables x_j and $x_{j'}$ are *coupled* in $h_i(x)$.

The set V_i is provided by the GAMS interface. To partition it into sets of quadratic ($Q_{i,k}$) and nonquadratic variables ($N_{i,k}$) and for determining the sparsity graph E_i^{sparse}, the Hessian of $h_i(x)$ is evaluated at sample points. Nonzero entries in the Hessian yield an edge in the sparsity graph, and constant columns in the Hessian indicate quadratic variables. Since we only need the information whether entries of the Hessian are constant and nonzero, but not the actual values, this sampling approach yields correct results for functions that are common in practical applications.

3.2 Relaxations

We now describe the steps which lead to a polyhedral relaxation of problem (P).

First, for each function $h_{i,k}(x_{N_{i,k}})$ and $x_{Q_{i,k}}^T A_{i,k} x_{Q_{i,k}}$ (cf. (1)) it is determined whether it is convex over $[\underline{x}, \overline{x}]$. For a function $h_{i,k}(x_{N_{i,k}})$ the minimal eigenvalue of $\nabla^2 h_{i,k}$ is evaluated at sample points. 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. For a quadratic form $x_{Q_{i,k}}^T A_{i,k} x_{Q_{i,k}}$ it suffices to compute the minimal eigenvalue of $A_{i,k}$.

Next, convex underestimators are constructed in a two-step approach. In the first step, nonconvex functions $h_{i,k}(x_{N_{i,k}})$ are underestimated by (possibly nonconvex) quadratic functions. In the second step, quadratic nonconvex functions are replaced by convex α-underestimators as introduced in [2]. Even though the direct application of the α-underestimator technique to the original function would result in a convex underestimator, the proposed quadratic underestimator is often tighter because the α-convexification depends only on the curvature of the function and not on the function behavior, cf. Figure 6.

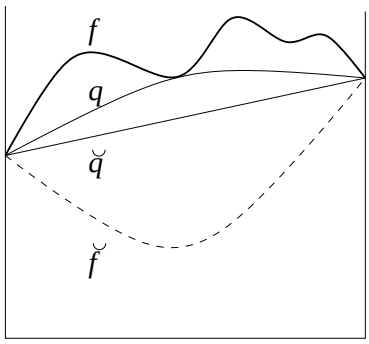

Figure 6: α-underestimator $\breve{f}$ of f versus the convexification $\breve{q}$ of the quadratic underestimator q of f.

Finally, the functions of the convex relaxation are linearized to obtain a polyhedral outer-approximation.

3.2.1 Quadratic underestimators

A quadratic underestimator

$$q(x) = x^T A x + b^T x + c$$

of a nonconvex function $f : \mathbb{R}^r \to \mathbb{R}$ over a box $[\underline{x}, \overline{x}]$ is constructed by solving the following linear program in the variables A, b, and c:

$$\begin{array}{rl} \min\limits_{A,b,c} & \sum\limits_{x \in S} f(x) - q(x) \\ \text{such that} & q(x) \leq f(x), \qquad x \in S, \end{array} \tag{2}$$

where S is a sample set of points from $[\underline{x}, \overline{x}]$. Thereby, the sparsity pattern of A and b are chosen according to that of f, i.e., the matrix A and the Hessian $\nabla^2 f$, and the vector b and the gradient ∇f, respectively, have the same zero entries. Information about the "shape" of the function $f(x)$ is inherited to $q(x)$ by minimizing additionally the distances of the gradient and Hessian between $f(x)$ and $q(x)$ in some of the sample points.

This method requires only function evaluations, and can thus be applied to black-box functions for which no analytic expressions are known. The quality of the quadratic underestimator depends thereby strongly on the sample set S. In our numerical experiments, the choice $S = \text{vert}([\underline{x}, \overline{x}]) \cup \{x_{\min}, (\underline{x} + \overline{x})/2\} \cup M$, where $\text{vert}([\underline{x}, \overline{x}])$ are the vertices of the box $[\underline{x}, \overline{x}]$, $x_{\min}$ is a local minimizer of $f(x)$, and M a set of randomly generated points, led to robust results. Note, that in practical applications such as the cogeneration plant presented in Section 2, the nonconvex functions $h_{i,k}(x_{N_{i,k}})$ depend on rather small sets $N_{i,k}$ of variables and hence allow for a sufficiently large sample set S while keeping the linear program (2) still efficiently solvable.

The relaxation (Q) of (P) is obtained by replacing nonconvex functions $h_{i,k}(x_{N_{i,k}})$ by quadratic underestimators $q(x_{N_{i,k}})$ computed by means of (2). Finally, the binary conditions on the variables x_B are dropped.

3.2.2 Convex relaxation

The relaxation (C) of (Q) is obtained by replacing nonconvex quadratic forms in (Q) by α-underestimators as introduced by Adjiman and Floudas [2]. An α-underestimator of a quadratic form $f(x) = x^T A x + b^T x + c$ (for $x \in \mathbb{R}^r$) is the function

$$\breve{f}(x) = f(x) + \alpha^T \operatorname{Diag}(x - \underline{x})(x - \overline{x})$$

where $\operatorname{Diag}(\cdot)$ denotes a diagonal matrix and the parameter $\alpha \in \mathbb{R}^r$ is computed according to $\alpha = \max\{0, -\lambda_1(\operatorname{Diag}(w) A \operatorname{Diag}(w))\} \operatorname{Diag}(w)^{-2} e$, where $e \in \mathbb{R}^r$ is the vector of ones, $w = \overline{x} - \underline{x}$, and $\lambda_1(\cdot)$ denotes the minimum eigenvalue of a matrix. It is clear that $\breve{f}$ is convex and $\breve{f}(x) \leq f(x)$ for all $x \in [\underline{x}, \overline{x}]$. The convex relaxation takes now the form

$$\begin{array}{rl} \min & b_0^T x \\ \text{such that} & \breve{h}(x) \leq 0, \\ & x \in [\underline{x}, \overline{x}], \end{array} \tag{C}$$

where $\breve{h}_i(x) \equiv h_i(x)$ if the function $h_i(x)$ is convex in (Q).

3.2.3 Linear relaxation

The linear relaxation (R) of (P) is generated by linearization of the nonconvex functions $\breve{h}_i(x)$ in (C) in an optimal point of (C). In the Branch and Cut algorithm, (R) is augmented by further linearizations in local optimal points of (P).

For a box $U \subseteq [\underline{x}, \overline{x}]$, we denote by (R[$U$]) the linear relaxation where the variables are restricted to take values in U,

$$\begin{array}{lll} \min & b_0^T x & \\ \text{s.t.} & \breve{h}_i(x) \le 0, & i \in \{1,\dots,m\} \text{ with } \breve{h}_i \text{ linear}, \\ & \breve{h}_i(x^*) + \nabla \breve{h}_i(x^*)(x - x^*) \le 0, \quad x^* \in X^*, & i \in \{1,\dots,m\} \text{ with } \breve{h}_i \text{ nonlinear}, \\ & x \in U, & \end{array} \tag{R[U]}$$

where X^* is a set of local optimal points of (P) or (C).

3.3 Box reduction

In practice, the bounding box $[\underline{x}, \overline{x}]$ of a given MINLP can be large, which can result in convex underestimators and cuts of bad quality. This drawback might be prevented if a box reduction procedure is applied 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. Two box reduction techniques are currently implemented in LaGO.

The first method utilizes the whole set of constraints of the linear relaxation (R) at once by enclosing the feasible set of the linear relaxation with a new maybe smaller box. The feasible set is thereby further restricted by a level cut that cuts off all points for which the objective function value exceeds the incumbent upper bound $\bar{v}$ on the optimal value of (P). Formally, for a box $U \subseteq [\underline{x}, \overline{x}]$, a new lower (upper) bound on a variable x_j is computed by solving

$$\begin{array}{lll} \min\,(\max) & x_j & \\ \text{s.t.} & \breve{h}_i(x) \le 0, & \text{if } \breve{h}_i \text{ is linear}, \\ & \breve{h}_i(x^*) + \nabla \breve{h}_i(x^*)(x - x^*) \le 0, \quad x^* \in X^*, & \text{if } \breve{h}_i \text{ is nonlinear}, \\ & b_0^T x \le \overline{v}, & i \in \{1,\dots,m\} \\ & x \in U. & \end{array} \tag{B$_j$[U]}$$

This procedure is illustrated in Figure 7. If problem (B$_j$[U]) is infeasible, then there exists no point in U with a better optimal value than the incumbent upper bound. Hence, the subregion U does not need further investigation. Solving (B$_j$[U]) for all variables can be costly, and thus should only be carried out for variables which seem promising for a box reduction, cf. [16].

The second box reduction method is a simple constraint propagation method [13]. It applies interval arithmetic techniques to the constraints of the original formulation (P). Hence, it does not depend on the quality of the relaxation (R[U]), but handles only one

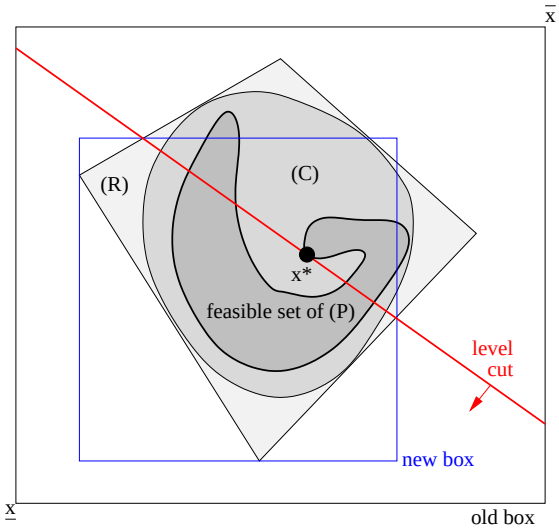

Figure 7: Box reduction using the feasible set of the linear relaxation

constraint at a time. Let $h_i(x) = g(x) + c_j x_j$ for $c_j \neq 0$. For a box $U \subseteq [\underline{x}, \overline{x}]$ we denote by $g(U)$ an interval such that $g(x) \in g(U)$ for all $x \in U$. Then let $[\underline{b}_j, \overline{b}_j] = -g(U)/c_j$. If $c_j > 0$, then the upper bound on x_j can be updated to $\min(\overline{x}_j, \overline{b}_j)$, and if $c_j < 0$, then the lower bound of x_j can be updated to $\max(\underline{x}_j, \underline{b}_j)$. Furthermore, if x_j is a binary variable, i.e., $j \in B$, and one of its bounds was reduced, then x_j can be fixed. In case that the new variable bounds define an empty box, infeasibility of a subproblem with box U is detected.

After reducing the box of one variable x_j, other constraints that depend on x_j might yield further box reductions. The implementation keeps track of these dependencies and selects variables and constraints for further consideration until the box is not (significantly) reduced anymore, cf. [16].

3.4 Branch and Cut algorithm

The Branch and Cut algorithm for solving problem (P) is shown in the Algorithm presented in Table 1. It computes the set X_{cand} of local optimizers.

If the lower bounds $\underline{v}(U)$ are correct and tight, the Algorithm converges to a global optimum of (P). However, LaGO currently does not update the relaxations (Q) and (C) after a branching operation, so that the relaxations (Q), (C), and (R) might not be tight and convergence to a global optimum cannot be ensured. For this reason, we decided to branch on binary variables only, i.e., when a subproblem is considered in which all binary variables are fixed, it is discarded, even when the gap between lower and upper bound is not closed. Another problem arises if the quadratic underestimator of a function $h_{i,k}(x_{N_{i,k}})$ is not rigorous and a wrong lower bound leads to a mistaken pruning of a node. Because of these two reasons, the proposed algorithm can be seen as a heuristic only.

Table 1: Branch and Cut Algorithm

Reformulate all functions into form (1) and get the sparsity graphs.
Box reduction by interval arithmetic and by enclosing the polyhedron defined by the linear constraints in (P) (if any).
Determine which of the functions $h_{i,k}(x_{N_{i,k}})$ and $x_{Q_{i,k}}^T A x_{Q_{i,k}}$ are convex.
Construct the pre-convex relaxation (Q).
Construct and solve the convex relaxation (C). Let x^* be a solution point of (C).
Construct the linear relaxation (R) using $X^* = \{x^*\}$.
Box reduction by solving ($\mathrm{B}_j[U]$) with $U = [\underline{x}, \overline{x}]$ and by interval arithmetic.
Initialize the Branch and Bound tree $\mathcal{U}$ with node $[\underline{x}, \overline{x}]$. Set $\overline{v} = \infty$, $X_{\text{cand}} = \emptyset$.
Set $\underline{v}([\underline{x}, \overline{x}])$ to the optimal value of (C) and $\hat{x}_{[\underline{x},\overline{x}]}$ to the solution point x^* of (C).
repeat
  Take node U from $\mathcal{U}$ with $\underline{v}(U)$ minimal.
  if $U \cap X_{\text{cand}} = \emptyset$ **then** {Update upper bound}
    Start local search from $\hat{x}$ (with rounded binary variables) in (P) where the binary variables are fixed.
    if a new local minimizer x^* of (P) is found **then**
      Update X_{cand}, $\overline{v}$.
      Add cuts to (R) by adding x^* to the set X^*.
    end if
    Solve (R[U]). Update $\underline{v}(U)$ and $\hat{x}_U$.
  end if
  if $\underline{v}(U) < \overline{v}$ **and** not all binary variables are fixed in U **then**
    Let $j \in B$ be such that $\min((\hat{x}_U)_j, 1 - (\hat{x}_U)_j)$ is maximized over all $j \in B$.
    for $t = 0, 1$ **do** {Branching at variable j}
      Let $U_t := \{x \in U | x_j = t\}$.
      repeat
        Box reduction by interval arithmetic and by solving ($\mathrm{B}_j[U_t]$).
      until U_t is not reduced significantly **or** infeasibility is detected
      if infeasibility was not detected **then**
        Solve (R[U_t]), update $\underline{v}(U_t)$, and let $\hat{x}_{U_t}$ be a minimizer of (R[U_t]).
        Put U_t into $\mathcal{L}$.
      end if
    end for
  end if
  Prune $\mathcal{U}$ by deleting $U \in \mathcal{U}$ with $\underline{v}(U) \geq \overline{v}$.
until $\mathcal{U} = \emptyset$ or the gap $\overline{v} - \min_{U \in \mathcal{U}} \underline{v}(U)$ is small enough

4 Results and Discussion

To examine the applicability of the optimization method (Section 3) on the presented model (Section 2) and to investigate the flexibility of the algorithm, optimization runs

for four different cases of electric power and process steam demand were performed. The objective was to find a design of the cogeneration plant with minimum levelized total costs that fulfills the specified requirements.

Due to the complexity of the model, a preprocessing has to be applied for the local optimization step of the algorithm. In the preprocessing, the model is split up and initially, a part of it is made feasible. In the second step, the local optimization of the complete model is carried out taking into account the results of the first step. For an optimization run, 30000 iterations of the Branch-and-Bound algorithm were performed. Within LaGO, the LPs are solved by CPLEX 9.0 [10] and the local search is done by CONOPT 3.14P [7].

In the discussion of the results, no values of the 76 decision variables are presented for reasons of simplicity. Instead, the structure of the resulting cogeneration plant for the respective case is explained briefly. In addition, the most important parameters are presented in Table 2. These are the following: Exergy of the fuel $\dot{E}_F$, exergy destruction $\dot{E}_D$, exergy loss $\dot{E}_L$, exergetic efficiency ϵ, electric power output of the gas turbine $\dot{W}_{GT}$, total electric power output $\dot{W}_{total}$, thermal efficiency η_{th}, and total levelized costs TRR_{lev}. The efficiency values are based on the lower heating value of the fuel.

4.1 Case 1: pure electric power output of 300 MW

In the first case, a design for pure power generation with a capacity of 300 MW is studied. This case represents a relatively simple case because no process steam extraction has to be considered and the power demand is in the middle of the capacity range of the superstructure. The design produced by LaGO consists of a simple gas turbine without intercooling, air preheater, or sequential combustion. The bottoming cycle is designed as a two-pressure-level process with reheating after the high-pressure part of the steam turbine. The characteristic parameters of the design can be found in the first column of Table 2. The thermal efficiency of almost 57% is in line with the efficiencies of existing combined-cycle power plants at this capacity range. Since electricity is the only product of the plant, the levelized total cost of 12674 €/h can easily be converted into the levelized cost of electricity (for 8000 hours of yearly operation) and results in a value of about 4.2 Eurocent/kWh. Compared to the electricity costs of today's combined-cycle power plants and taking the future escalation of the fuel price into account, these results are realistic.

4.2 Case 2: pure electric power output of 400 MW

For the second case, the required electric power output of 400 MW reaches the upper bound of the capacity of the superstructure. Looking at characteristic parameters in the second column of Table 2, it can be seen that the thermal efficiency almost reaches 59%, which corresponds to the thermal efficiency of today's large-scale combined-cycle power plants. The design consists of a simple gas turbine as topping cycle and a three-pressure level steam cycle including a reheater as bottoming cycle. The more complex structure of the steam cycle leads to an increased efficiency compared to the first case. The costs of

electricity amount again to about 4.2 Eurocent/kWh. As in the first case, the gas turbine generates around two-thirds of the total electric power output.

4.3 Case 3: electric power output of 90 MW and process steam extraction of 99.5 t/h

The third case introduces process steam extraction into the design. The specifications of 90 MW electric power generation and process steam extraction of 99.5 t/h at a pressure level of 4.5 bar represent the demand of an existing paper factory. The design of the cogeneration plant consists of a simple gas turbine and a two-pressure steam cycle. Here, a reheating of the medium-pressure steam is not implemented resulting in a lower efficiency of the steam cycle. Compared to the first and second cases, the percentage of electric power contributed by the steam cycle is much lower which is a result of the high amount of steam extracted from the steam cycle. Since steam can be produced with a much higher energetic efficiency than electricity, the overall thermal efficiency of the design outperforms clearly the cases with pure electric power generation. Looking at the exergetic efficiency which takes into account the quality of the energy, it is the lowest of all cases reflecting the lower efficiency of the steam cycle.

4.4 Case 4: electric power output of 290 MW and process steam extraction of 150 t/h at different pressure levels

The fourth case represents again a cogeneration plant but with maximum exploitation of the complexity of the superstructure. Therefore, process steam extraction at three different pressure levels (40 t/h at 50 bar, 40 t/h at 15 bar, and 70 t/h at 3.5 bar) and an electric power output of 290 MW is required. The resulting design consists of a simple gas turbine and a three-pressure-level steam cycle including a reheater. The power-to-steam ratio is higher than in the third case which results in a lower overall thermal efficiency but in a more complex structure of the steam cycle and therewith a higher overall exergetic efficiency.

4.5 Sensitivity Analysis

A sensitivity analysis examines the influence of certain parameters, which are usually kept constant during an optimization run, on the result of the optimization. These parameters describe technical limits of plant components or are estimates of uncertain future economic developments. Knowledge of the effects of such parameters on the resulting design and on the associated costs is very valuable for the development engineer. Among the many possible parameters, the effect of the fuel price, the rate of increase of the fuel price, and the number of planned operating hours are chosen to demonstrate the sensitivity analysis performed with LaGO.

For the base case 1, the price of natural gas is set to 4 €/GJ. In the sensitivity analysis this price is changed to 3.5 €/GJ (representing a lower limit), and to 4.5 €/GJ (repre-

Table 2: Results from the cases considered in the optimization of the superstructure for different demands for electric power and process steam.

	Case 1	Case 2	Case 3	Case 4
$\dot{E}_F$ [MW]	543.3	701.4	203.8	582.8
$\dot{E}_D$ [MW]	226.4	281.3	87.1	222.0
$\dot{E}_L$ [MW]	17.0	20.1	6.8	21.9
ϵ [%]	55.21	57.03	53.9	58.16
$\dot{W}_{GT}$ [MW]	203.8	268.3	71.2	218.7
$\dot{W}_{total}$ [MW]	300	400	90	290
η_{th} [%]	56.72	58.59	77.19	68.45
TRR_{lev} [€/h]	12674	16771	5022	13424

senting a higher limit). The specified capacity for the power plant remains equal to 300 MW. To examine the influence of the fuel price, optimization runs with fuel prices of 3.5 €/GJ, 4 €/GJ, and 4.5 €/GJ are performed. The levelized total costs of the three resulting designs are then calculated with the three different values of the fuel price and are shown in Table 3. For example, the levelized total costs displayed in line 1 of column 3 of this Table result from a design which was optimized for a fuel price of 4.5 €/GJ but here is calculated with a fuel price of 3.5 €/GJ. If the fuel price would have a noticeable influence on the optimization, the design which is calculated with the fuel price for which it is optimized should always yield the lowest costs.

In Table 3, noticeable differences can only be seen between a fuel price of 3.5 €/GJ and 4.5 €/GJ. Nevertheless, the small differences show the influence of the fuel price considering its little relative variation.

Table 3: Effect of the fuel price on the resulting designs showing the TRR_{lev} in €/h

Values calculated with	Design optimized for 3.5 €/GJ	4.0 €/GJ	4.5 €/GJ
3.5 €/GJ	11436	11441	11461
4.0 €/GJ	12678	12674	12677
4.5 €/GJ	13921	13907	13894

To analyse the effect of the rate of increase of the fuel price, the same investigation is performed. In addition to the base rate of 1.0%, the values of 0.5% and 1.5% are considered. Table 4 shows the results of the calculations. Looking at the values of the levelized total costs no noticeable tendency can be observed. Taking into account the large relative variation of the parameter during the analysis, it can be concluded that the rate of increase of the fuel price has no significant influence on the optimization.

The last parameter under investigation is the average number of annual operating hours

Table 4: Effect of the rate of increase of the fuel price on the resulting designs showing the TRR_{lev} in €/h

Values calculated with	Design optimized for 0.5%	1.0%	1.5 %
0.5 %	12290	12296	12287
1.0 %	12667	12674	12661
1.5 %	13467	13477	13455

the plant is designed for. They are varied between 8000, 7000, 5000, and 4000 hours. The results are presented in Table 5. It can be seen that the design which is calculated with the number of operation hours for which it is optimized always yields the lowest cost. Moreover, the greater the difference between the number of hours the design was optimized for and the number of hours used for evaluating the design, the higher the total levelized costs are. Here, the influence of the yearly operation hours on the optimization results is clearly evident.

Table 5: Effect of the number of annual operation hours on the resulting designs showing the TRR_{lev} in €/h

Values calculated with	Design optimized for 4000 h	5000 h	7000 h	8000 h
4000 h	15233	15253	15368	15439
5000 h	14268	14248	14292	14333
7000 h	13167	13101	13062	13069
8000 h	12823	12742	12668	12664

4.6 Discussion

The various optimizations performed with LaGO produce reasonable results in all cases. The thermal efficiencies and the levelized total costs of the resulting designs are within the range of existing combined-cycle power plants of the corresponding capacity. This proves the applicability and functionality of LaGO as well as the plausibility of the model of the superstructure. The sensitivity analysis also produces reasonable and useful results and shows the flexibility of the optimization algorithm.

From the energy engineering point of view the results allow some interpretations in regard to the design of combined-cycle power plants. All investigated cases show a great unity in the gas turbine design. The gas turbine system is always a simple cycle with a pressure ratio of 18. The power output of the gas turbine is determined by the air mass flow. Differences between the several cases can mainly be found in the steam cycle design. For pure electricity generation, the fuel savings due to the efficiency increase outweigh the

investment costs in additional or more efficient plant components and result in a greater complexity of the steam cycle. In the case of significant process steam extraction, the complexity of the steam cycle is not a crucial factor anymore and leads to simpler designs.

5 Conclusions

A superstructure of a combined-cycle-based cogeneration plant was developed and optimized. The model allows the simultaneous optimization of the operational parameters and the structure of the plant. The resulting nonconvex mixed-integer nonlinear problem (MINLP) is solved by LaGO. The solver generates a convex relaxation of the MINLP and applies a Branch and Cut algorithm to the convex relaxation.

Various optimization runs with different requirements for the electric power and process steam demand as well as sensitivity analyses for different parameters were performed. The optimization tool produces reasonable results for all cases which proves its applicability and functionality.

Regarding the design of combined-cycle-based cogeneration plants, the results show that the focus should be set on the configuration of the steam cycle. Moreover, the option of process steam extraction has to be taken into account and decides over the complexity of the design. The fact that only fixed gas turbine systems are available on the market is not a disadvantage because all designs obtained contain the same simple gas turbine process.

Acknowledgements: This work was supported by the German Science Foundation (DFG) under grants NO 421/2-1, TS 79/1-1, and TS 79/1-3.

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 and 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] A. Bejan, G. Tsatsaronis, and M. Moran. *Thermal Design and Optimization*. John Wiley & Sons, Inc., New York, 1996.

[4] D. Dressler. *Kostenabschätzung von ausgewählten Kraftwerkskomponenten*. Studienarbeit, Institut für Energietechnik, TU Berlin, 1994.

[5] 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.

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

[7] GAMS Development Corp. *GAMS - The Solver Manuals*. Washington DC, 2003.

[8] *Gas Turbine World Handbook*. Pequot Publishing Inc., 1997.

[9] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467, 2001.

[10] ILOG, Inc. CPLEX. `http://www.ilog.com/products/cplex`.

[11] O. Knacke, O. Kubaschewski, and K. Hesselmann. *Thermochemical Properties of Inorganic Substances*. Springer-Verlag, 2. edition, 1991.

[12] S. Kucherenko and Y. Sytsko. Application of deterministic low-discrepancy sequences in global optimization. *Computational Optimization and Applications*, 30:297–318, 2005.

[13] 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.

[14] I. Nowak. *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*. Birkhäuser, 2005.

[15] I. Nowak and S. Vigerske. LaGO - Lagrangian Global Optimizer. `https://projects.coin-or.org/LaGO`.

[16] 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 submitted, 2006.

[17] N. Sahinidis and M. Tawarmalani. BARON. `http://archimedes.scs.uiuc.edu/baron/baron.html`, 2002.

[18] M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

[19] 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.

[20] R. Turton, R. C. Bailie, W. B. Whiting, and J. A. Shaeiwitz. *Analysis, Synthesis and Design of Chemical Processes*. Prentice Hall, Upper Saddle River, 1998.

[21] J. Viswanathan and I. E. Grossmann. A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization. *Computers and Chemical Engineering*, 14(7):769–782, 1990.

[22] W. Wagner and A. Kruse. *Properties of Water and Steam - The Industrial Standard IAPWS-IF97*. Springer, 1998.