This is a post-peer-review, pre-copyedit version of an article published in *Journal of Global Optimization*. The final authenticated version is available online at:
https://doi.org/10.1007/s10898-017-0547-4
Received: 03 January 2017 / Accepted: 12 July 2017 / Published: 05 August 2017

**Journal of Global Optimization manuscript No.**
(will be inserted by the editor)

# Deterministic Global Optimization of Process Flowsheets in a Reduced Space Using McCormick Relaxations

**Dominik Bongartz** · **Alexander Mitsos**

**Abstract** Deterministic global methods for flowsheet optimization have almost exclusively relied on an equation-oriented formulation where all model variables are controlled by the optimizer and all model equations are considered as equality constraints, which results in very large optimization problems. A possible alternative is a reduced-space formulation similar to the sequential modular infeasible path method employed in local flowsheet optimization. This approach exploits the structure of the model equations to achieve a reduction in problem size. The optimizer only operates on a small subset of the model variables and handles only few equality constraints, while the majority is hidden in externally defined functions from which function values and relaxations for the objective function and constraints can be queried. Tight relaxations and their subgradients for these external functions can be provided through the automatic propagation of McCormick relaxations. Three steam power cycles of increasing complexity are used as case studies to evaluate the different formulations. Unlike in local optimization or in previous sequential approaches relying on interval methods, the solution of the reduced-space formulation using McCormick relaxations enables dramatic reductions in computational time compared to the conventional equation-oriented formulation. Despite the simplicity of the implemented branch-and-bound solver that does not fully exploit the tight relaxations returned by the external functions but relies on further affine relaxation at a single point using the subgradients, in some cases it can solve the reduced-space formulation significantly faster without any range reduction than the state-of-the-art solver BARON can solve the equation-oriented formulation.

**Keywords** Global optimization · Process design · Sequential modular · Branch-and-bound · Relaxation of algorithms

Process Systems Engineering (AVT.SVT), RWTH Aachen University, Forckenbeckstr. 51, 52074 Aachen, Germany
Corresponding Author: A. Mitsos
Tel.: +49-241-80-94704
Fax: +49-241-80-92326
E-mail: amitsos@alum.mit.edu

**Acknowledgements** This work received funding through the "Competence Center Power to Fuel" of RWTH Aachen University and project "Power to Fuel" of JARA Energy, both of which are funded by the Excellence Initiative by the German federal and state governments to promote science and research at German universities, as well as from the German Federal Ministry of Education and Research (BMBF) under grant number 03SFK2A. The responsibility for the content lies with the authors. The authors would also like to thank Jaromił Najman, Hatim Djelassi, and Wolfgang Huster for helpful discussions.

## 1 Introduction

Over the past decades, mathematical programming has become a widely used tool in the design of chemical or energy conversion processes [10,24]. Flowsheet optimization, where the operating conditions and sizes of the units of a process are to be optimized with respect to economic or other (e.g., ecological) objectives, has been drawing particular interest because of the scale of the potential savings. If the flowsheet configuration itself, i.e., the selection of the units and their interconnections, is to be optimized as well, superstructure formulations are employed that in general result in mixed integer nonlinear programs (MINLP) including nonconvex functions [11,35]. However, even the nonlinear programs (NLP) obtained when optimizing the design of processes with fixed configurations can be difficult to solve and are often multimodal.

Local methods have long become state of the art in flowsheet optimization and they are implemented in many commercial process simulators. Solvers have been developed that treat the equations describing the process either in the equation-oriented mode or the sequential modular mode of flowsheet simulation (as well as some intermediate approaches) [11,24]. In the former approach, all variables are handled by the optimizer and all equations are enforced as equality constraints. The latter approach subdivides the process into a number of separate modules that can be evaluated in a specified order to compute all quantities of interest, with the optimizer operating on a subset of the variables corresponding to the degrees of freedom and variables introduced to decouple the equations [12,22,47]. In this case, only few equations are left as equality constraints, while the majority is treated within the modules using specialized solvers, which is a known advantage of this approach. Both methods have been applied successfully to very large process flowsheets.

Methods for deterministic global optimization have also seen increasing interest in process engineering, both for special sub-problems such as heat exchanger network synthesis (see, e.g., [42,46]) and for flowsheet optimization for fixed process configurations or superstructures [1,3,4,8,35,53,64]. However, almost all of these approaches are based on the equation-oriented mode of flowsheet simulation, because the solvers used require direct access to the nonconvex model equations in order to supply convex relaxations for providing lower bounds to the branch-and-bound (B&B) [26,29] and related algorithms [41,55,56]. This can result in very large optimization problems for complex flowsheets that can easily involve several thousand model variables [24], which is problematic because of the exponential worst-case runtime of B&B-based solvers. In general, it also requires the user to provide bounds for a large number of variables, which may be hard to guess (cf., e.g., [3,4]). Therefore, methods that operate in a reduced space of variables seem particularly interesting for global optimization.

Several approaches have been suggested for operating in a reduced space by means of selective branching, but these are only applicable to problems containing certain types of equations [25,54], and while they reduce the dimensionality of the space to branch on, the size of the subproblems to be solved remains unchanged and so does the number of variables for which bounds need to be provided [43]. Therefore, it is desirable to eliminate variables

from the optimization problem instead. The extreme is to consider a method in which only the degrees of freedom are optimization variables. However, providing relaxations is challenging in this case because in general, the evaluation of the objective function and constraints involves the solution of a nonlinear equation system [54,63]. Byrne and Bogle [19] first introduced the idea of using a sequential modular approach for global flowsheet optimization by implementing modules as template functions that can be called with different data types to supply the required bounds for a B&B algorithm. They particularly stressed the advantages of a modular approach from a modeling perspective. However, using interval methods to obtain bounds, they found this approach to be less computationally efficient than equation-oriented formulations [6,7,13,18,19].

In this work, we present an approach for deterministic global flowsheet optimization that is similar to that of Byrne and Bogle [19] in the sense that it exploits problem structure at the modeling level in analogy to the sequential modular infeasible path method in local optimization. However, instead of interval methods, we rely on the automatic propagation of McCormick relaxations and their subgradients [43], which have been shown to have favorable convergence properties [14,44].

In the following, we first provide some background on McCormick relaxations in Section 2 before summarizing different formulations for deterministic global flowsheet optimization and discuss their suitability for the automatic propagation of McCormick relaxations and subgradients in Section 3. In Section 4, the implementation of a basic B&B solver is presented that can handle the suggested reduced-space formulation using external function definitions. Since it is meant for demonstration purposes, it relies on further affine relaxation at a single point and thus does not fully exploit the tight McCormick relaxations obtained from the external functions. Finally, in Section 5 three case studies for flowsheets of increasing complexity (specifically, steam power cycles for use in combined cycle power plants) are considered in order to demonstrate the significant savings in computational time enabled by the reduced-space formulation in comparison to the conventional equation-oriented formulation because of the reduction in problem size. The latter formulation is solved both with the same basic solver and with the state-of-the-art solver BARON [56]. The models used in the case studies are available as Online Resources.

## 2 Background material

Global optimization using B&B-based approaches requires convex and concave relaxations of the nonconvex functions involved.

**Definition 1 (Relaxation of functions)** Given a convex set $Z \subset \mathbb{R}^{n_z}$ and a function $f : Z \to \mathbb{R}$, a convex function $f^{\mathrm{cv}} : Z \to \mathbb{R}$ is called a convex relaxation of $f$ on $Z$ if $f^{\mathrm{cv}}(\mathbf{z}) \leq f(\mathbf{z})$ for all $\mathbf{z} \in Z$, and similarly a concave function $f^{\mathrm{cc}} : Z \to \mathbb{R}$ is called a concave relaxation of $f$ on $Z$ if $f^{\mathrm{cc}}(\mathbf{z}) \geq f(\mathbf{z})$ for all $\mathbf{z} \in Z$.

McCormick [40] introduced a method for the construction of convex and concave relaxations of factorable functions, i.e., functions defined by a finite recursive composition using binary sums, binary products, and a library of univariate intrinsic functions. Tsoukalas and Mitsos [57] extended the composition theorem to multivariate intrinsic functions. Being a generalization of the univariate composition theorem of McCormick [40], the multivariate composition theorem also allows deriving the known relaxations for binary sums, as well as tighter relaxations for binary products. Therefore, we only repeat the multivariate composition here for the sake of completeness.

**Theorem 1 (McCormick relaxation of multivariate composition [57])** *Let $Z \subset \mathbb{R}^{n_z}$ and $X \subset \mathbb{R}^{n_x}$ be nonempty compact convex sets. Consider the composite function defined as $g = F(\mathbf{f}(\mathbf{z}))$, where $F : X \to \mathbb{R}$ and for $i \in I = \{1,\dots,n_x\}$, $f_i : Z \to \mathbb{R}$ are continuous functions, and let*

$$\{(f_1(\mathbf{z}),\dots,f_m(\mathbf{z})) \mid \mathbf{z} \in Z\} \subset X.$$

*Suppose that convex and concave relaxations $f_i^{cv}, f_i^{cc} : Z \to \mathbb{R}$ of $f_i$ on $Z$ are known for every $i \in I$. Let $F^{cv}, F^{cc} : X \to \mathbb{R}$ be convex and concave relaxations of $F$ on $X$. Then the functions $g^{cv}, g^{cc} : Z \to \mathbb{R}$ defined as*

$$g^{cv}(\mathbf{z}) = \min_{\mathbf{x} \in X} \{F^{cv}(\mathbf{x}) \mid f_i^{cv}(\mathbf{z}) \leq x_i \leq f_i^{cc}(\mathbf{z}),\ \forall i \in I\}$$

$$g^{cc}(\mathbf{z}) = \max_{\mathbf{x} \in X} \{F^{cc}(\mathbf{x}) \mid f_i^{cv}(\mathbf{z}) \leq x_i \leq f_i^{cc}(\mathbf{z}),\ \forall i \in I\}$$

*are convex and concave relaxations, respectively, of $g$ on $Z$.*

Mitsos et al. [43] demonstrated how the propagation of McCormick relaxations can be automated to compute relaxations for functions defined by computer codes (similar to the calculation of derivatives via automatic differentiation) implementing factorable functions. This includes a broad class of algorithms with the main restriction that they are either a sequence of explicit function evaluations or involve a finite number of iterations known a priori (such as, e.g., the solution of a system of linear equations or numerical integration using an explicit scheme with fixed step length).

Unlike the auxiliary variable method (AVM) [52,53,55,56], which is also used in the state-of-the-art solver BARON, the recursive application of the composition rules yields relaxations in the original space of variables. However, these relaxations can be weaker than those generated in AVM in case the latter recognizes terms occurring repeatedly [57], and they are known to be nonsmooth. Methods for propagating subgradients of the McCormick relaxations can be found in [43,57] and are not repeated here for brevity. These subgradients can be used either directly for solving the resulting lower bounding problems using nonsmooth optimization methods, or for deriving affine relaxations.

**Definition 2 (Subgradient)** Given a nonempty convex set $Z \subset \mathbb{R}^{n_z}$, a convex function $f^{\mathrm{cv}} : Z \to \mathbb{R}$, and a concave function $f^{\mathrm{cc}} : Z \to \mathbb{R}$, a vector $\nabla_s f^{\mathrm{cv}} \in \mathbb{R}^{n_z}$ is called a *subgradient* of $f^{\mathrm{cv}}$ at $\bar{\mathbf{z}} \in Z$ if $f^{\mathrm{cv}}(\mathbf{z}) \geq f^{\mathrm{cv}}(\bar{\mathbf{z}}) + (\nabla_s f^{\mathrm{cv}})^{\intercal}(\mathbf{z} - \bar{\mathbf{z}})$ for all $\mathbf{z} \in Z$, and a vector $\nabla_s f^{\mathrm{cc}} \in \mathbb{R}^{n_z}$ is called a *subgradient* of $f^{\mathrm{cc}}$ at $\bar{\mathbf{z}} \in Z$ if $f^{\mathrm{cc}}(\mathbf{z}) \leq f^{\mathrm{cc}}(\bar{\mathbf{z}}) + (\nabla_s f^{\mathrm{cc}})^{\intercal}(\mathbf{z} - \bar{\mathbf{z}})$ for all $\mathbf{z} \in Z$.

Further recent extensions that are beyond the scope of the present work include methods for computing relaxations of bounded functions with discontinuities [62], and for providing relaxations for the solution of ordinary differential equations and nonlinear equation systems [50,54,63]. Recently, a differentiable modification of McCormick relaxations was also introduced [34].

## 3 Optimization Formulations

Flowsheets of chemical processes or energy systems typically contain a large number of variables that describe, for example, pressures, temperatures, mass flow rates, and compositions of the process streams, heat and work transfer occurring in different units of the process, or quantities related to the size and cost of the units. When designing (and operating) a process, one or more of these variables are degrees of freedom or *design variables*

$\mathbf{d} \in D \subset \mathbb{R}^{n_d}$ that can be chosen within certain bounds (i.e., $D = [\mathbf{d}_\mathrm{L}, \mathbf{d}_\mathrm{U}]$, where $\mathbf{d}_\mathrm{L}$ and $\mathbf{d}_\mathrm{U}$ denote the vectors of lower and upper bounds, respectively, on the components of $\mathbf{d}$). These may, for example, include sizes of components, split fractions, but also pressures or temperatures at certain points in the process. Once these design variables have been fixed, the values of the remaining *dependent model variables* $\mathbf{x} \in X \subset \mathbb{R}^{n_x}$ can be determined by solving the corresponding nonlinear *model equations* $\mathbf{h}(\mathbf{d}, \mathbf{x}) = \mathbf{0},\ \mathbf{h} : D \times X \to \mathbb{R}^{n_x}$ (assuming such a solution exists for that particular $\mathbf{d}$), which typically consist of the physical relationships describing the different units and the flowsheet connectivity, of cost correlations, as well as design specifications (cf., e.g., [11,24]). The division of variables into design and dependent model variables is usually not unique but can be chosen to facilitate the solution of the design problem.

In general, the objective function (e.g., production cost of a chemical, or net power output of a power plant) $f(\mathbf{d}, \mathbf{x}),\ f : D \times X \to \mathbb{R}$ depends on the design and model variables. Additional inequality constraints $\mathbf{g}(\mathbf{d}, \mathbf{x}) \leq \mathbf{0},\ \mathbf{g} : D \times X \to \mathbb{R}^{n_g}$ can arise from limitations that the design needs to fulfill, such as, e.g., material limits on pressures and temperatures or required product purities, from physical constraints, e.g., for ensuring a positive temperature difference in heat transfer, or from limitations on model validity. From this information, there are several ways to formulate the desired optimization problem for deterministic global optimization.

### 3.1 Full-space formulation

A natural way of formulating a flowsheet optimization problem is to treat both design variables and dependent model variables as decision variables, and interpret the model equations as equality constraints:

$$\begin{aligned} \min_{\mathbf{d} \in D, \mathbf{x} \in X} \quad & f(\mathbf{d}, \mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{d}, \mathbf{x}) = \mathbf{0} \\ & \mathbf{g}(\mathbf{d}, \mathbf{x}) \leq \mathbf{0} \end{aligned} \tag{FS}$$

This full-space formulation is conceptually similar to the *equation-oriented mode* of flowsheet simulation, in which all model equations are solved simultaneously [11,24]. It has become increasingly popular for local optimization and can, at least in principle, also be solved with available deterministic global solvers such as BARON [49,55,56] or ANTIGONE [41]. For lower bounding, these solvers substitute the functions $f$, $\mathbf{h}$, and $\mathbf{g}$ by convex relaxations by means of, e.g., McCormick relaxations [40] or $\alpha$BB relaxations [2,5], possibly introducing auxiliary variables. For this reason, they require direct access to all equations and variables. However, as discussed above, this leads to very large optimization problems that can result in long if not prohibitive run times of global solvers, as well as pose difficulties in providing sufficiently tight bounds on variables without excluding potential solutions.

### 3.2 Reduced-space formulations

The fact that typically $n_d \ll n_x$ motivates alternative formulations that operate in a reduced space of variables.

### 3.2.1 Implicit function formulation

Since the model equations $\mathbf{h}(\mathbf{d},\mathbf{x}) = \mathbf{0}$ can be solved for the dependent model variables $\mathbf{x}$ for given values of $\mathbf{d}$, they define an implicit function $\hat{\mathbf{x}} : \mathbf{d} \mapsto \hat{\mathbf{x}}(\mathbf{d})$ with the property $\mathbf{h}(\mathbf{d},\hat{\mathbf{x}}(\mathbf{d})) = \mathbf{0}\ \forall \mathbf{d} \in D$, assuming a unique solution exists for all $\mathbf{d} \in D$. In this case, the dependent model variables can be eliminated from problem (FS) to obtain (cf. [54,63])

$$\begin{aligned} \min_{\mathbf{d}\in D}\ & f(\mathbf{d},\hat{\mathbf{x}}(\mathbf{d})) \\ \text{s.t.}\ & \mathbf{g}(\mathbf{d},\hat{\mathbf{x}}(\mathbf{d})) \leq \mathbf{0}. \end{aligned}$$

This leads to a significant reduction in the dimensionality of the space of decision variables while also eliminating all equality constraints. In local optimization, similar approaches have been known as *feasible path methods* that fully converge a flowsheet simulation at each iteration of the NLP solver (i.e., they solve for $\mathbf{x}$ given the current values of $\mathbf{d}$ in that iteration) [24]. An obvious advantage of this approach is that operation only in the degrees of freedom leads to the smallest possible problems in terms of number of optimization variables. However, since in general the functional form of $\hat{\mathbf{x}}(\mathbf{d})$ is not known (i.e., it is truly implicit), providing relaxations for $f(\mathbf{d},\hat{\mathbf{x}}(\mathbf{d}))$ and $\mathbf{g}(\mathbf{d},\hat{\mathbf{x}}(\mathbf{d}))$ on $D$ is challenging and it is not clear that these will be as tight as in explicit formulations. Additional questions arise when there are multiple solution branches to $\mathbf{h}(\mathbf{d},\mathbf{x}) = \mathbf{0}$ [54,63]. While methods for providing relaxations of implicit functions recently have been introduced [50,54,63], no implementation of these methods is publicly available, and this formulation is thus beyond the scope of this work.

### 3.2.2 Sequential modular formulations

An alternative is to make use of the fact that the equation systems arising from process modeling are known to be highly structured. In flowsheet simulation, this is exploited in the so-called *sequential modular mode*, where the equations are grouped into modules that describe single units (or groups of units) of a flowsheet and define a specific input-output relationship [11,19,24]. The remaining equations mainly describe the connectivity between the modules. Typically, both the equation systems within the modules and the equations describing flowsheet connectivity are only coupled via few equations that need to be solved iteratively. On the flowsheet level, iterative solution can be required because of loops in the flowsheet, or for satisfying design specifications that prescribe a value for a variable that is part of the output of a module rather than its input, which is to be achieved by varying a variable belonging to a different module. Within the modules, typical examples include flash calculations that are solved by iterating on one or few variables, depending on the property model employed and the specified variables (see, e.g., [11]).

When simulating a flowsheet in sequential modular mode, the latter cases are treated with specialized solution procedures within the modules as part of their input-output relationship. To treat loops in the flowsheet, certain streams (so-called *tear streams*) within the loops are selected and the equations are decoupled by introducing tear variables $\mathbf{x}_t \in X_t \subset \mathbb{R}^{n_t}$ that are duplicates of the model variables associated with the tear streams. The equations that need to be solved iteratively at the flowsheet level then reduce to the fixed point equations $\mathbf{x}_t = \hat{\mathbf{h}}_t(\mathbf{d},\mathbf{x}_t), \hat{\mathbf{h}} : D \times X_t \to X_t$ that enforce the actual connectivity of the tear streams, where $\hat{\mathbf{h}}_t$ returns the values for the variables in the tear stream calculated by sequentially evaluating the modules given values for the design and tear variables in an order that usually corresponds to the flow of material through the process.

In analogy, flowsheet optimization in sequential modular mode can be formulated as

$$\begin{aligned} \min_{\mathbf{d}\in D,\mathbf{x}_t\in X_t} \quad & \hat{f}(\mathbf{d},\mathbf{x}_t) \\ \text{s.t.} \quad & \hat{\mathbf{h}}_t(\mathbf{d},\mathbf{x}_t) = \mathbf{x}_t \\ & \hat{\mathbf{g}}(\mathbf{d},\mathbf{x}_t) \leq \mathbf{0}. \end{aligned} \tag{RS}$$

Here, the objective function and inequality constraints have been reformulated in a similar manner as the model equations. This way, most of the original model variables $\mathbf{x}$ and model equations $\mathbf{h}(\mathbf{d},\mathbf{x}) = \mathbf{0}$ have been moved to the modules and are effectively hidden from the optimizer, which only controls the design and tear variables (note that usually $n_t \ll n_x$). This approach is widely used in *local* optimization of flowsheets (termed *infeasible path method*, since convergence of the flowsheet simulation is only enforced as the optimizer converges) and is implemented in commercial flowsheet simulators [11, 12, 24]. Compared to a fully implicit approach, this leads to larger optimization problems that do contain equality constraints. However, in local optimization it avoids the computational effort required for conducting full flowsheet simulations at each iteration, and is therefore often considered a good compromise [10, 24]. In global optimization, it avoids having to treat the entire model as an implicit function as discussed in Section 3.2.1. Nevertheless, some method is still needed for providing relaxations of the functions $\hat{f}$, $\hat{\mathbf{h}}_t$ and $\hat{\mathbf{g}}$ that are defined by some computer code that sequentially evaluates the modules involved. While Byrne and Bogle [19] used interval methods, we choose to utilize McCormick relaxations [40] that can be automatically propagated through such codes along with their subgradients [43], as long as the codes implement factorable functions[1] (cf. Section 2). An additional advantage of this approach is that bounds on intermediate variables (i.e., those not visible to the optimizer) are not required, since they are propagated along with the relaxations. In case no equations within the modules and no design specifications require iterative solution, problem (RS) can be solved with a B&B solver by using these methods to propagate relaxations through the module implementations and deriving the required bounding information.

In case some modules do contain equation systems that need to be solved iteratively, the equations coupling these systems along with the corresponding variables can be handed to the optimizer in addition to the ones contained in problem (RS). Since by assumption the remaining module equations can be solved explicitly for the remaining module variables, this ensures the applicability of the propagation of McCormick relaxations without resorting to implicit functions. The resulting problem is

$$\begin{aligned} \min_{\mathbf{d}\in D,\mathbf{x}_t\in X_t,\mathbf{x}_m\in X_m} \quad & \tilde{f}(\mathbf{d},\mathbf{x}_t,\mathbf{x}_m) \\ \text{s.t.} \quad & \tilde{\mathbf{h}}_t(\mathbf{d},\mathbf{x}_t,\mathbf{x}_m) = \mathbf{x}_t \\ & \tilde{\mathbf{h}}_m(\mathbf{d},\mathbf{x}_t,\mathbf{x}_m) = \mathbf{0} \\ & \tilde{\mathbf{g}}(\mathbf{d},\mathbf{x}_t,\mathbf{x}_m) \leq \mathbf{0}, \end{aligned} \tag{RS*}$$

where $\mathbf{x}_m \in X_m \subset \mathbb{R}^{n_{x_m}}$ are the additional module variables handled by the optimizer and $\tilde{\mathbf{h}}_m(\mathbf{d},\mathbf{x}_t,\mathbf{x}_m) = \mathbf{0}, \tilde{\mathbf{h}}_m : D \times X_t \times X_m \to X_m$ are the corresponding module equations (i.e., only the residual of the equation coupling the system within the module is returned to the optimizer as a component of $\tilde{\mathbf{h}}_m$). Such additional equations and variables can also serve to satisfy design specifications, which in some infeasible path methods of local optimization

[1] Note that the recent extension of this approach to multivariate outer functions [57] also enables the use of functions that are not typically considered part of this class, such as, e.g., $\min(x, x^2)$.

are also left to the optimizer [11]. To achieve this, the deviation of the calculated value (by sequentially solving the modules) from the specified target value is added as a component of $\mathbf{h}_m$, and the variable to be varied to satisfy this constraint is added as a component of $\mathbf{x}_m$. Note that (RS*) does not introduce additional multimodality compared to (FS) as is shown in Appendix A.

While these formulations were motivated with a modular implementation of the model in mind (c.f. also [19]), this is neither a prerequisite nor is it specific to this formulation. From a solution point of view, the essential feature of formulation (RS*) is that it collects all explicit parts in externally defined functions that the optimizer has no direct access to. These functions can internally have a modular structure in the sense that they utilize sub-models that have been implemented separately, but they can also just contain the model equations arranged to allow sequential explicit evaluation. On the other hand, a modular implementation could also be utilized for the full-space formulation (FS) given a suitable implementation of the external functions.

## 4 Implementation

Since available deterministic global solvers to our knowledge do not currently support the use of external function definitions, a simple solver was implemented to compare formulations (FS) and (RS*). Consider the NLP

$$\begin{aligned} \min_{\mathbf{p}\in P=[\mathbf{p}_\mathrm{L},\mathbf{p}_\mathrm{U}]} \quad & \bar{f}(\mathbf{p}) \\ \text{s.t.} \quad & \bar{\mathbf{h}}(\mathbf{p}) = \mathbf{0} \\ & \bar{\mathbf{g}}(\mathbf{p}) \leq \mathbf{0}, \end{aligned} \tag{NLP}$$

where $\bar{f} : P \to \mathbb{R}$, $\bar{\mathbf{h}} : P \to \mathbb{R}^{n_{\bar{h}}}$, and $\bar{\mathbf{g}} : P \to \mathbb{R}^{n_{\bar{\mathrm{g}}}}$ are bounded factorable functions (also including multivariate outer functions, cf. [57]) that can be defined by some computer code not visible to the optimizer. In this sense, to the solver they act as a black box, very similar to the use of external equations in modeling systems like GAMS [27]. The full-space formulation (FS) can be achieved by letting the vector of optimization variables be $\mathbf{p} = (\mathbf{d}^\mathsf{T}, \mathbf{x}^\mathsf{T})^\mathsf{T}$ and implementing the external functions to return the desired values for $f$, $\mathbf{h}$ (i.e., the residuals of all model equations), and $\mathbf{g}$. The reduced-space formulation (RS*) can be achieved by letting $\mathbf{p} = (\mathbf{d}^\mathsf{T}, \mathbf{x}_t^\mathsf{T}, \mathbf{x}_m^\mathsf{T})^\mathsf{T}$ and implementing the functions to return $\tilde{f}$, $\tilde{\mathbf{h}}_t - \mathbf{x}_t$, $\tilde{\mathbf{h}}_m$, and $\tilde{\mathbf{g}}$. In this case, the calculations in the functions can include a multitude of operations conducted sequentially.

The external functions are implemented in C++ as template functions and can thus be called with different data types In the present implementation, they are called either using floating point types to obtain function values of the objective function and all constraints, using the types defined in FADBAD++ [9] to obtain their gradients with respect to $\mathbf{p}$ via automatic differentiation (in case the functions involved are differentiable), or using the types defined in the MC++ library [20] to evaluate their convex (cv) and concave (cc) relaxations along with the corresponding subgradients (denoted by $\nabla_s$). The latter implements the automatic propagation of McCormick relaxations and extensions [40,43,50,57,62].

Since the propagation of relaxations and subgradients (as well as gradients) is automated, the implementation of the model itself is not more involved than that of a regular simulation model, with the exception that care has to be taken not to violate the assumptions required for ensuring the applicability of the propagation (cf. Section 2). The main restriction is that the models do not involve an iterative solution procedure with unknown number

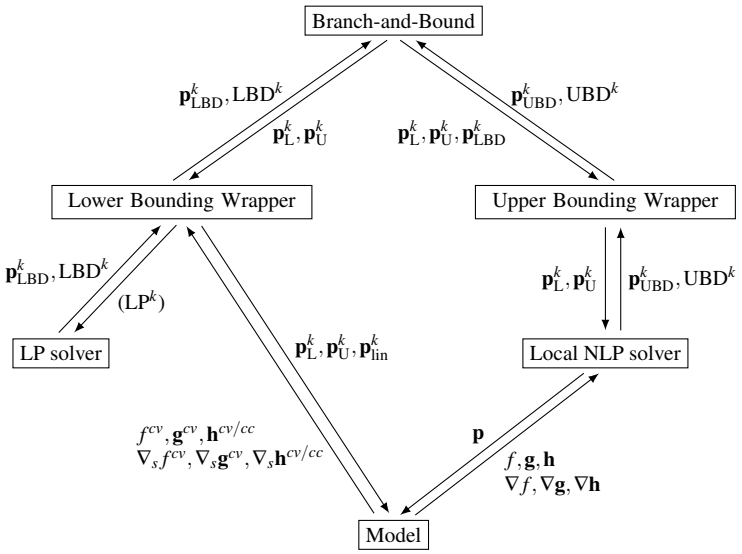


Fig. 1: Structure of the implemented solver for solving (NLP). To the solver, the model is effectively a black box from which function values and gradients or convex and concave relaxations and the corresponding subgradients can be queried for the objective function and constraints.

of iterations (i.e., general loops with a conditional statement for termination), which would require the use of specialized methods [54,63] that are not included in the current implementation (cf. Section 3.2.1). Instead, such implicit modules need to be made explicit by leaving appropriate module variables and equations to the optimizer (cf. Section 3.2.2). Additionally, the calculations must also not involve conditional statements that give nonsmooth or discontinuous functions, since computing valid relaxations always requires information on the entire domain. Such statements can however be reformulated, either through the use of implemented nonsmooth functions like the maximum of two functions, or using the methods for relaxation of discontinuous functions introduced by Wechsung et al. [62]. However, in this case appropriate care has to be taken for upper bounding since the problem itself is in general nonsmooth. Possible solutions include the use of bundle methods, gradient free methods, or simple function evaluation for upper bounding. As an alternative, one could also resort to mixed integer formulations.

The implemented solver is based on the basic B&B algorithm in C++ employed in [43] using a best-first heuristic and bisection along the longest edge (relative to the initial width) for branching (cf. [29,38]). Each node $k$, defined by its lower and upper bounds $\mathbf{p}^k_{\mathrm{L}}$ and $\mathbf{p}^k_{\mathrm{U}}$ on the optimization variables $\mathbf{p}$, is successively handed to wrappers for lower and upper bounding that return the respective lower and upper bounds $\mathrm{LBD}^k$ and $\mathrm{UBD}^k$, as well as the optimal solution points $\mathbf{p}^k_{\mathrm{LBD}}$ and $\mathbf{p}^k_{\mathrm{UBD}}$ (see Fig. 1). The node is fathomed if the lower bounding problem is infeasible, or if its lower bound $\mathrm{LBD}^k$ is not lower than the incumbent objective value by more than the specified optimality tolerance.

To obtain a lower bound, a relaxation of (NLP) on the current node can be constructed by replacing equalities by two inequalities each, and using the relaxations of the objective function and constraints returned by the external functions. However, in general the Mc-

Cormick relaxations obtained are nonsmooth, and therefore a further affine relaxation based on the subgradients is conducted to obtain a linear program (LP) and thus allow the use of robust LP solvers. Examples illustrating the construction of affine relaxations derived from McCormick relaxations and their subgradients can be found in [43]. For simplicity, we evaluate relaxations and subgradients at a single linearization point ($\mathbf{p}^k_{\text{lin}}$) equal to the centerpoint of node k, and thus get the following LP relaxation:

$$
\begin{aligned}
\min_{\mathbf{p}\in[\mathbf{p}^k_{\text{L}},\mathbf{p}^k_{\text{U}}],\eta\in\mathbb{R}} \quad & \eta \\
\text{s.t.} \quad & f^{cv}(\mathbf{p}^k_{\text{lin}}) + \nabla_s f^{cv}(\mathbf{p}^k_{\text{lin}}) \cdot (\mathbf{p} - \mathbf{p}^k_{\text{lin}}) \leq \eta \\
& h^{cv}_i(\mathbf{p}^k_{\text{lin}}) + \nabla_s h^{cv}_i(\mathbf{p}^k_{\text{lin}}) \cdot (\mathbf{p} - \mathbf{p}^k_{\text{lin}}) \leq 0 \quad i = 1,...,n_{\bar{h}} \\
& h^{cc}_i(\mathbf{p}^k_{\text{lin}}) + \nabla_s h^{cc}_i(\mathbf{p}^k_{\text{lin}}) \cdot (\mathbf{p} - \mathbf{p}^k_{\text{lin}}) \geq 0 \quad i = 1,...,n_{\bar{h}} \\
& g^{cv}_k(\mathbf{p}^k_{\text{lin}}) + \nabla_s g^{cv}_k(\mathbf{p}^k_{\text{lin}}) \cdot (\mathbf{p} - \mathbf{p}^k_{\text{lin}}) \leq 0 \quad k = 1,...,n_{\bar{g}}
\end{aligned}
\tag{LP$^k$}
$$

Tighter relaxations could be obtained by linearizing at multiple points (cf., e.g., [55]), at the expense of increasing the size of the LP and requiring more function evaluations, which may be expensive for large models. Problem (LP$^k$) is solved using CPLEX v12.5 via the C++ API [30]. Optionally, simple range reduction (RR) can be conducted by successively maximizing and minimizing each component of $\mathbf{p}$ subject to the relaxed constraints in (LP$^k$) and the constraint that the relaxed objective be better than the current upper bound (termed *standard range reduction* by Locatelli and Schoen [38]), as well as employing bound tightening based on the dual multipliers returned by CPLEX [48]. While other techniques might also be applicable in this case, including probing [48,55] or the methods introduced in [63], these are beyond the scope of this work.

For upper bounding, the original problem (NLP) subject to $\mathbf{p} \in [\mathbf{p}^k_{\text{L}},\mathbf{p}^k_{\text{U}}]$ is handed to a local NLP solver, using the optimal solution of the lower bounding problem as initial point. The current implementation uses IPOPT [61] initially. Once a feasible point has been found, few iterations (typically 2–5) of the SLSQP algorithm [36,37] implemented in the NLopt library v2.4.2 [31] are conducted instead at each node. This is often sufficient to find feasible or even locally optimal solutions as the bounds on $\mathbf{p}$ within the nodes get tighter through branching, while avoiding excessive computational effort for upper bounding at every node.

## 5 Case Studies

In order to compare the formulations (FS) and (RS*), we consider the bottoming cycle of a combined-cycle power plant (CCPP) as an example. Combined-cycle plants running on natural gas already account for more than 10% of the world electricity production and enjoy increasing popularity due to their high efficiency and low emissions [33]. In a CCPP, the exhaust gas of a gas turbine is run through a heat recovery steam generator (HRSG) that powers a steam (Rankine) cycle. The optimal design of the steam cycle for a given stream of hot exhaust gas is subject of ongoing research. Previous studies have employed different local and heuristic global optimization methods, some using mixed-integer formulations to account for different cycle configurations [3,16,32,39,60]. With respect to formulation (RS*), steam power cycles represent a special type of flowsheet because in many cases they can be modeled in a completely sequential manner, i.e., they do not require the introduction of tear streams (but possibly handing other equations and variables to the optimizer as described in Section 3.2.2).

In the following, we consider three versions of bottoming cycles for CCPP of increasing complexity for a given stream of hot gas turbine exhaust $G$ (characterized by a heat capacity flow rate $(\dot{m}\cdot c_\mathrm{p})_\mathrm{G}$ and an inlet temperature $T_\mathrm{G,in}$) and fixed cycle configurations. As objective function, we consider either the maximization of the net power output ($\dot{W}_\mathrm{net}$) of the steam cycle, or the minimization of the levelized cost of electricity (LCOE) of the CCPP. Even for a fixed configuration and with the simplest thermodynamic models, the NLPs resulting from the optimization of, e.g., operating pressures, temperatures, and mass flow rates with respect to such thermodynamic or economic objectives are usually nonconvex [39], and can be multimodal, as will also be shown in the following.

### 5.1 Case Study I: Basic Rankine Cycle

To demonstrate the multimodality, we first consider a simple single pressure cycle with a fixed outlet temperature of the gas stream $T_\mathrm{G4} = T_\mathrm{G,out}$ (see Fig. 2), as might be the case, e.g., when burning a sulfur-containing fuel [33]. A stream of water in the saturated liquid state leaves the condenser and is pumped to the upper cycle pressure $p_2$. In the HRSG, it is first preheated in the economizer to a temperature slightly below its saturation temperature before entering the steam drum of the evaporator. It is then fully evaporated and superheated to the live steam temperature $T_5$ before entering the turbine, where it is expanded back to the condenser pressure. Since the condenser pressure $p_1$ and the gas outlet temperature $T_\mathrm{G,out}$ are assumed to be fixed, the problem has only two degrees of freedom, namely the upper cycle pressure $p_2$ and the cycle mass flow rate $\dot{m}$ (cf. Table 1; other choices are also possible).

In the sense of the reduced-space formulation (RS*), the cycle can be simulated sequentially to evaluate the objective function and the constraints without requiring tear streams. The model equations and the calculation sequence for the maximization of $\dot{W}_\mathrm{net}$ along with the fixed parameter values can be found in Appendix B.1. The LCOE is calculated based on fuel cost, operating cost and capital investment estimated using sizing data derived from the

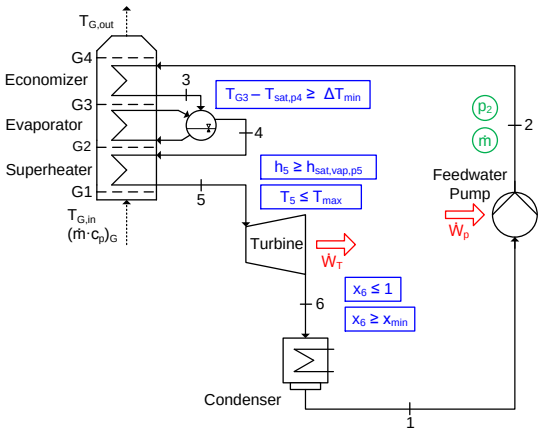


Fig. 2: Case Study I: Basic Rankine cycle for extracting work from a given gas stream G. In formulation (RS*), only the variables shown in circles and the constraints shown in boxes are handled by the optimizer in order to either maximize $\dot{W}_\mathrm{net}$ or minimize LCOE. The problem thus has two degrees of freedom.

Table 1: Optimal objective values and ranges and optimal solutions for the optimization variables of formulation (RS*) for Case Study I.

| Symbol | Description | Unit | Range | max $\dot{W}_{\text{net}}$ | min LCOE |
|---|---|---|---|---|---|
| Optimization variables | | | | | |
| $p_2$ | Upper cycle pressure | bar | [3, 100] | 54.6 | 34.3 |
| $\dot{m}$ | Cycle mass flow rate | kg/s | [5, 100] | 29.5 | 30.7 |
| Objective functions | | | | | |
| $\dot{W}_{\text{net}}$ | Net power output | MW | | 30.0 | |
| LCOE | Levelized cost of electricity | \$/MWh | | | 50.3 |

cycle simulation. Details on the economic evaluation are given in Appendix C. Note that if the cycle is modeled in an equation-oriented way, the optimization problem contains more than 44 variables (cf. Tables 2 and 3).

Four inequality constraints arise from physical limitations (cf. Fig. 2). First, the live steam leaving the HRSG needs to be at least fully evaporated (i.e., its enthalpy $h_5$ must be equal to or greater than the corresponding saturated vapor enthalpy $h_{\text{sat,vap,p5}}$), and its temperature must not exceed the maximum allowable temperature $T_{\text{max}}$ for the materials used. Second, there is a minimum temperature difference $\Delta T_{\text{min}}$ between exhaust gas and water that must be maintained throughout the HRSG. In this example, the only potential pinch point where this could be violated is in the evaporator at point $G3$ (here the gas temperature is $T_{\text{G3}}$ and the water temperature equals the saturation temperature $T_{\text{sat,p4}}$). Finally, the vapor quality $x_6$ at the turbine outlet must be higher than a minimum value $x_{\text{min}}$ to avoid erosion due to droplet formation [33]. To ensure the validity of the chosen model, an additional constraint is introduced that requires the turbine outlet state 6 to be in the two-phase region, i.e., its vapor quality should be at most unity.

The feasible region of this problem is shown in Fig. 3 along with two level sets of the objective function $\dot{W}_{\text{net}}$. The problem exhibits two local optima. The pinch constraint in the HRSG is active for both of them, but one of them exhibits a live steam temperature at its maximum allowable value, whereas for the other the vapor quality at the turbine outlet reaches its minimum allowable value. While the existence of multiple local optima is an open question for more complicated cycles, this example shows that this can be the case even for simple processes.

The problem was solved using the implementation described in Section 4 both in the reduced-space formulation (RS*) and in the full-space formulation (FS). For the latter, reasonable bounds had to be provided for all variables while ensuring not to cut off potential solutions. These bounds were derived manually using physical insight and partially exploiting the model equations. The full-space formulation was also implemented in GAMS 24.8.4 [27] and solved with BARON v17.4.1 [56]. To improve comparability, we confined the modeling to those features available in GAMS/BARON. This excludes some interesting features of the proposed framework. For example, since the standard definition of the logarithmic mean temperature difference (LMTD) can result in a division by zero, Chen's approximation [21] was used instead (cf. Appendix C), although a well-defined formulation of the actual LMTD with tighter relaxations is available within the present framework [42, 45]. The relative optimality tolerance as well as all constraint feasibility tolerances were set to $10^{-6}$. The calculations were conducted on a Intel® Core™ i3-3240 with 3.4 GHz and 16 GB RAM.

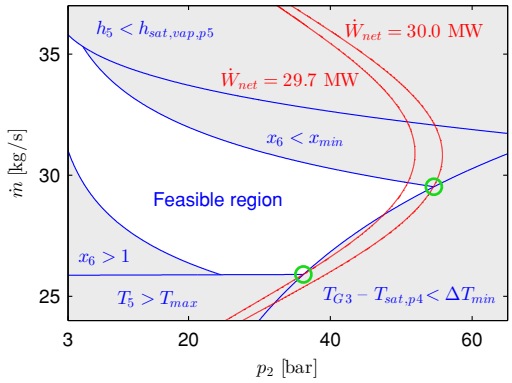


Fig. 3: The basic Rankine cycle (Case Study I) exhibits two local optima shown as circles when maximizing the net power output ($\dot{W}_{\text{net}}$, red level sets). The only degrees of freedom are the upper cycle pressure ($p_2$) and the cycle mass flow rate ($\dot{m}$). The gray areas denote regions that are infeasible (at least one inequality constraint is violated), while the blue lines represent the points where the respective constraints are active.

Table 2: Problem size and solution times for Case Study I maximizing $\dot{W}_{\text{net}}$. 'Fomul.' refers to the problem formulations in Section 3. 'Time' is the CPU time needed for solving the problem to the given tolerance. 'Iterations' denotes the total number of B&B nodes treated, and 'Max. nodes' denotes the maximum number of B&B nodes held in memory at any given time during solution.

| Formul. | Problem size | Solver | Time [s] | Iterations | Max. nodes |
|---|---|---|---|---|---|
| (FS) | 44 variables<br>42 equalities<br>2 inequalities | | | | |
| | | Present solver (no RR) | 4637 | $2.26 \times 10^6$ | $3.14 \times 10^5$ |
| | | Present solver (with RR) | 0.72 | 19 | 6 |
| | | BARON | 0.16 | 1 | 1 |
| (RS*) | 2 variables<br>0 equalities<br>5 inequalities | | | | |
| | | Present solver (no RR) | 0.07 | 299 | 32 |
| | | Present solver (with RR) | 0.03 | 45 | 12 |

Figure 4 shows the incidence matrices for the problem of maximizing $\dot{W}_{\text{net}}$ using formulations (FS) and (RS*). It can be seen that while the incidence matrix of (FS) is sparse, this structure is exploited in (RS*) to achieve a much smaller problem (cf. also Tab. 2) with a dense incidence matrix. It should be emphasized again that herein this reduction in problem size is achieved at the modeling level through physical insight in the system at hand and in analogy to established strategies for flowsheet simulation and local optimization. A possible alternative could be to derive it from the full-space structure automatically. However, this is beyond the scope of the present work.

Using the present simple B&B solver without any range reduction for maximizing $\dot{W}_{\text{net}}$, formulation (RS*) is solved orders of magnitude faster than formulation (FS), obviously because of the much smaller problem size (see Table 2). Note that in formulation (FS), some

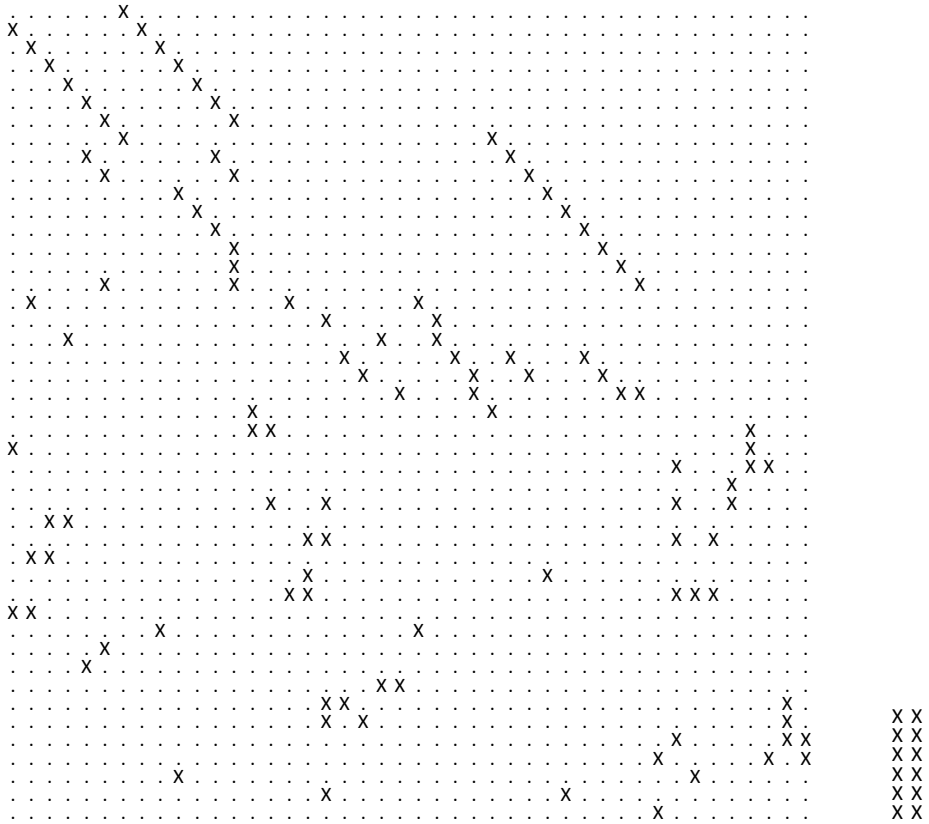

Fig. 4: Incidence matrix for Case Study I maximizing $\dot{W}_{\text{net}}$ using (a) formulation (FS) and (b) formulation (RS*). The columns correspond to the optimization variables, while the rows correspond to equality constraints, inequality constraints, and objective (in that order). 'X' indicates a variable is present in the respective equation, while '.' indicates it is not. Variable and equation names are omitted for better readability.

inequalities from formulation (RS*) can be omitted because they can be enforced as part of the variable bounds. The simple range reduction techniques described in Section 4 enable significant savings both in the number of iterations and computational time, particularly for (FS). In this example, BARON with its sophisticated branch-and-reduce algorithm can solve (FS) in a single iteration, i.e., without any branching, and it is almost as fast in solving (FS) as the present solver can solve (RS*). To confirm the importance of the multimodality, the problem was also solved with the local solvers CONOPT [23], IPOPT [61], and KNITRO [17], all of which do converge to the suboptimal local solution (cf. Fig. 3) for certain initial guesses.

When minimizing LCOE, the problem gets more computationally challenging, since the equations used for estimating heat exchanger areas and equipment cost contain many highly nonlinear terms (cf. Appendix C). In case of formulation (FS), the problem also increases in size because of the added variables and equations. In formulation (RS*), only

Table 3: Problem size and solution times for Case Study I minimizing LCOE and a time limit of $10^5$ s.

| Formul. | Problem size | Solver | Time [s] | Iterations | Max. nodes |
|---|---|---|---|---|---|
| (FS) | 87 variables<br>85 equalities<br>2 inequalities | | | | |
| | | Present solver (no RR) | $>10^5$ | $>9.2\times10^6$ | $1.26\times10^6$ |
| | | Present solver (with RR) | 37 | 345 | 98 |
| | | BARON | $1.08\times10^4$ | $1.44\times10^5$ | $1.67\times10^4$ |
| | | BARON (sel. branching) | 4178 | $4.20\times10^4$ | 9139 |
| (RS*) | 2 variables<br>0 equalities<br>9 inequalities | | | | |
| | | Present solver (no RR) | 0.55 | 1677 | 206 |
| | | Present solver (no RR, no MV) | 0.54 | 1745 | 213 |
| | | Present solver (with RR) | 0.25 | 379 | 94 |
| | | Present solver (with RR, no MV) | 0.26 | 415 | 96 |

some inequalities have to be added to ensure the validity of the cost correlations employed by ensuring a minimum area of the heat exchangers, which in (FS) is again part of the variable bounds[2]. In this case, the present simple solver is not able to solve formulation (FS) within the time limit of $10^5\,\text{s} \approx 28\,\text{h}$ when relying on pure B&B (no feasible point is found, and the lower bound is almost 40% below the optimal objective value), while it can solve (RS*) in less than half a second (cf. Table 3).

BARON takes significantly longer for solving (FS) when minimizing LCOE than the present solver. This suggests that when minimizing LCOE, the problems contain some expressions for which the relaxations in BARON are relatively weak. Both the number of branch-and-reduce iterations and the CPU time needed by BARON can be reduced significantly by means of selective branching (cf. [25,43,54]), i.e., setting the branching priorities to force BARON to branch only on $p_2$ and $\dot{m}$. However, although branching is only performed on the two degrees of freedom, the subproblems involved still have the large number of variables and constraints of formulation (FS), and the reductions in computational time are by far not as pronounced as when changing from (FS) to (RS*) with the present solver.

The solution with the present solver was also repeated *without* using the multivariate McCormick relaxations (MV) introduced by Tsoukalas and Mitsos [57] by disabling the corresponding option in MC++ , which implements the multivariate composition rules for binary products, divisions, and the minimum or maximum of two variables (cf. Table 3). The multivariate relaxations do reduce the number of iterations for the present example, since the resulting relaxations are tighter [44,57]. Because they are also more expensive to compute, the CPU time does not decrease for all cases.

To provide some first insight into the influence of more complex thermodynamic models, Case Study I was also repeated using temperature-dependent sub-models, namely a quadratic model for the ideal gas heat capacity and the Watson equation for the enthalpy of vaporization. In this case, the expressions for enthalpy and entropy of gaseous or liquid streams cannot be solved analytically for temperature any more, which makes both the pump

[2] Note that it might be more reasonable not to restrict the area itself, but rather set the heat exchanger cost to a constant below a certain threshold. However, this would lead to a nonsmooth problem requiring special care for upper bounding. Also, BARON does not currently support the max function.

Table 4: Optimal objective values and ranges and optimal solutions for the optimization variables of formulation (RS*) for Case Study I with temperature dependent thermodynamic sub-models.

| Symbol | Description | Unit | Range | max $\dot{W}_{\text{net}}$ | min LCOE |
|---|---|---|---|---|---|
| Optimization variables | | | | | |
| $p_2$ | Upper cycle pressure | bar | [3, 100] | 100 | 100 |
| $\dot{m}$ | Cycle mass flow rate | kg/s | [5, 100] | 27.6 | 29.2 |
| $T_2$ | Temperature Stream 2 | K | [300, 873] | 350 | 350 |
| $T_5$ | Temperature Stream 5 | K | [300, 873] | 873 | 784 |
| Objective functions | | | | | |
| $\dot{W}_{\text{net}}$ | Net power output | MW | | 34.2 | |
| LCOE | Levelized cost of electricity | \$/MWh | | | 49.0 |

Table 5: Problem size and solution times for maximizing $\dot{W}_{\text{net}}$ in Case Study I with temperature dependent thermodynamic sub-models.

| Formul. | Problem size | Solver | Time [s] | Iterations | Max. nodes |
|---|---|---|---|---|---|
| (FS) | 72 variables<br>69 equalities<br>2 inequalities | | | | |
| | | BARON | 0.36 | 5 | 2 |
| (RS*) | 4 variables<br>2 equalities<br>4 inequalities | | | | |
| | | Present solver (no RR) | 0.15 | 291 | 34 |
| | | Present solver (with RR) | 0.05 | 11 | 4 |

and boiler models implicit. To treat these in the context of formulation (RS*), their outlet temperatures $T_2$ and $T_5$ are added as additional optimization variables and the corresponding energy balances are added as equality constraints. The model formulation and results are summarized in Appendix B.4.

The optimal solution points and objective values for maximum power output and minimum LCOE are given in Table 4. The maximum power output differs from that predicted with the simpler model by 13%, and both for maximum power output and minimum LCOE the upper bound is now reached for the upper cycle pressure. Note also that the upper bound on the steam temperature $T_5$ can now be enforced as part of the variable bounds, while it had to be added as an actual inequality constraint for the simpler model.

The main observations in terms of computational performance are similar to the ones made with the simple thermodynamic model used above. Formulation (RS*) is solved faster with the present solver than (FS) is solved in BARON, especially for minimizing LCOE (cf. Tables 5 and 6). Interestingly, when minimizing LCOE BARON can solve the model with the more complex thermodynamics faster than the one with the simple thermodynamics. However, it is still slower than the present solver solving (RS*) by a factor of 880 and 5480 for pure B&B and with simple range reduction in the latter, respectively.

Table 6: Problem size and solution times for minimizing LCOE in Case Study I with temperature dependent thermodynamic sub-models.

| Formul. | Problem size | Solver | Time [s] | Iterations | Max. nodes |
|---|---|---|---|---|---|
| (FS) | 112 variables<br>110 equalities<br>2 inequalities | | | | |
| | | BARON | 2026 | $1.02 \times 10^5$ | 7889 |
| (RS*) | 4 variables<br>2 equalities<br>8 inequalities | | | | |
| | | Present solver (no RR) | 2.3 | 3465 | 470 |
| | | Present solver (with RR) | 0.37 | 211 | 56 |

## 5.2 Case Study II: Regenerative Rankine Cycle

A slightly more complex example is considered next (see Fig. 5), where the gas outlet temperature $T_{G4}$ is not fixed any more [33]. Instead, the outlet enthalpy of the superheater $h_7$ is added as a degree of freedom for formulation (RS*). An inequality constraint is added to prevent temperature crossover at the economizer inlet. Furthermore, the cycle now contains a deaerator for removing non-condensable gases, which allows operation of the condenser at lower pressure for increased cycle efficiency. In the deaerator, the water is mixed with a two-phase bleed stream extracted from the turbine. The pressure of the turbine bleed is equal to that of the deaerator [65]. This pressure $p_2$ along with the fraction $k_{Bl}$ of the cycle

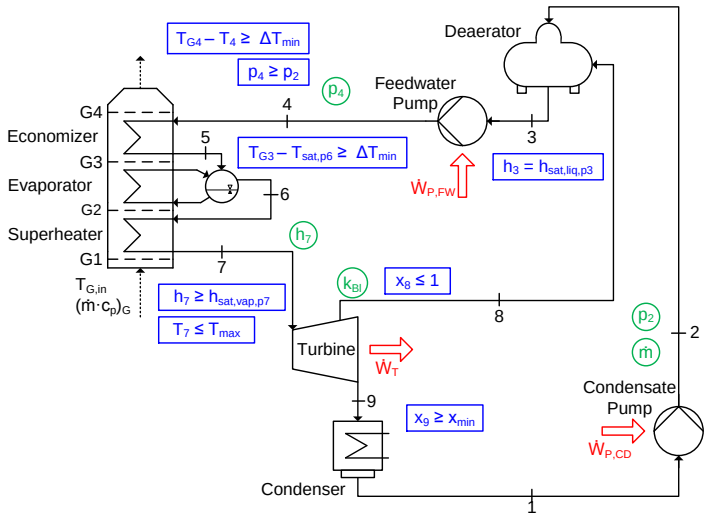


Fig. 5: Case Study II: Regenerative Rankine cycle. In formulation (RS*), only the variables shown in circles and the constraints shown in boxes are handled by the optimizer in order to either maximize $\dot{W}_{net}$ or minimize LCOE. Since there is one equality constraint in this formulation, the problem has four degrees of freedom.

Table 7: Optimal objective values and ranges and optimal solutions for the optimization variables of formulation (RS*) for Case Study II.

| Symbol | Description | Unit | Range | $\max \dot{W}_{\text{net}}$ | min LCOE |
|---|---|---|---|---|---|
| Optimization variables | | | | | |
| $p_2$ | Deaerator pressure | bar | [ 0.2, 5] | 0.2 | 0.2 |
| $p_4$ | Upper cycle pressure | bar | [ 3, 100] | 45.3 | 41.1 |
| $\dot{m}$ | Cycle mass flow rate | kg/s | [ 5, 100] | 25.4 | 26.3 |
| $h_7$ | Live steam enthalpy | kJ/kg | [2480, 3750] | 3640 | 3350 |
| $k_{\text{Bl}}$ | Fraction of mass flow extracted | - | [ 0.01, 0.2] | 0.0328 | 0.0348 |
| Objective functions | | | | | |
| $\dot{W}_{\text{net}}$ | Net power output | MW | | 34.4 | |
| LCOE | Levelized cost of electricity | \$/MWh | | | 49.0 |

Table 8: Problem size and solution times for Case Study II maximizing $\dot{W}_{\text{net}}$ and a time limit of $10^5$ s.

| Formul. | Problem size | Solver | Time [s] | Iterations | Max. nodes |
|---|---|---|---|---|---|
| (FS) | 73 variables<br>69 equalities<br>4 inequalities | | | | |
| | | Present solver (no RR) | $>10^5$ | $>1.93 \times 10^7$ | $2.36 \times 10^6$ |
| | | Present solver (with RR) | 39 | 547 | 115 |
| | | BARON | 1.7 | 191 | 22 |
| | | BARON (sel. branching) | 2.7 | 217 | 26 |
| (RS*) | 5 variables<br>1 equality<br>7 inequalities | | | | |
| | | Present Solver (no RR) | 4.2 | $1.17 \times 10^4$ | 1442 |
| | | Present Solver (with RR) | 0.62 | 475 | 108 |

mass flow rate that is extracted from the turbine are also added as optimization variables (cf. Table 7). The requirement that the liquid leaving the deaerator be in the saturated liquid state to enable the removal of gases is enforced as an equality constraint. From a flowsheet point of view, this corresponds to a design specification. A description of the calculation sequence can be found in Appendix B.2.

When maximizing $\dot{W}_{\text{net}}$, the present solver is again not able to solve (FS) with pure B&B within the time limit, while it can solve (RS*) in a similarly short time (or even faster when using simple range reduction) as BARON needs for solving (FS) (cf. Table 8). Interestingly, in this example selective branching on the variables of formulation (RS*) when solving (FS) with BARON *increases* both the number of iterations and CPU time. We currently do not know why this is the case.[3] When minimizing LCOE, (RS*) is solved in 7.5 min, or less than 3 min when using simple range reduction (cf. Table 9). BARON still has 6% gap remaining after reaching the time limit when solving (FS) with default settings, while in this case selective branching reduces the gap to 1% when reaching the time limit (cf. Figure 6).

[3] Note also that while Epperly and Pistikopoulos [25] give a convergence proof for a selective branching strategy, Stuber et al. [54] argue that this is limited to linear equations for equality constrained problems.

Table 9: Problem size and solution times for Case Study II minimizing LCOE and a time limit of $10^5$ s.

| Formul. | Problem size | Solver | Time [s] | Iterations | Max. nodes |
|---|---|---|---|---|---|
| (FS) | 120 variables<br>116 equalities<br>4 inequalities | | | | |
| | | BARON | $>10^5$ | $>7.35 \times 10^5$ | $2.87 \times 10^5$ |
| | | BARON (sel. branching) | $>10^5$ | $>4.48 \times 10^5$ | $1.45 \times 10^5$ |
| (RS*) | 5 variables<br>1 equality<br>12 inequalities | | | | |
| | | Present Solver (no RR) | 446 | $7.59 \times 10^5$ | $9.31 \times 10^4$ |
| | | Present Solver (with RR) | 158 | $1.23 \times 10^5$ | $2.03 \times 10^4$ |

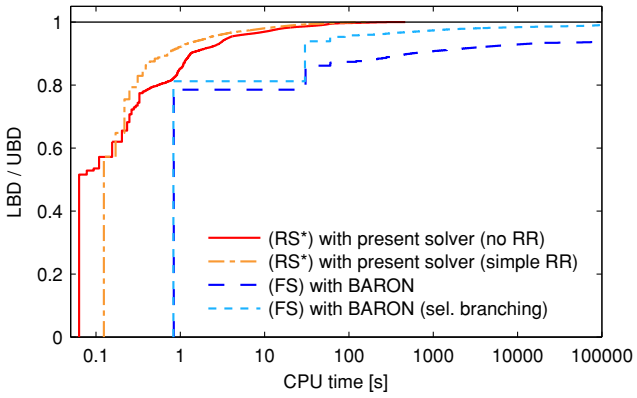


Fig. 6: Convergence (indicated by the ratio of lower (LBD) and upper (UBD) bounds on the objective) of Case Study II minimizing LCOE. While the reduced-space formulation (RS*) can be solved with a simple solver within a few minutes, the solution of the full-space formulation (FS) takes several hours. In each case, the optimal solution (i.e., optimal UBD) is found relatively quickly, but it takes long for LBD to converge.

### 5.3 Case Study III: Two-Pressure Cycle

Finally, a more sophisticated cycle configuration using two pressure levels in the HRSG is considered, which is one of the preferred layouts for newly installed CCPP [33,39,60] (see Fig. 7). After the deaerator, the water is pumped to an intermediate pressure level in the low-pressure (LP) pump and is preheated close to saturation in the LP economizer. Part of the water is then further pumped to a higher pressure level in the high-pressure (HP) pump and again preheated, evaporated, and superheated before entering the HP turbine, which expands the fluid back to the LP pressure level. The remaining water is evaporated and superheated at the LP pressure level and mixed with the HP turbine outlet before entering the LP turbine, which corresponds to the turbines in the previous case studies. In this case, additional degrees of freedom are the outlet pressure $p_8$ of the HP pump, the outlet enthalpy $h_{11}$ of the HP superheater, and the fraction $k_{\mathrm{LP}}$ of the cycle mass flow rate that is sent through the LP rather than the HP part of the HRSG (cf. Table 10). Additional inequalities

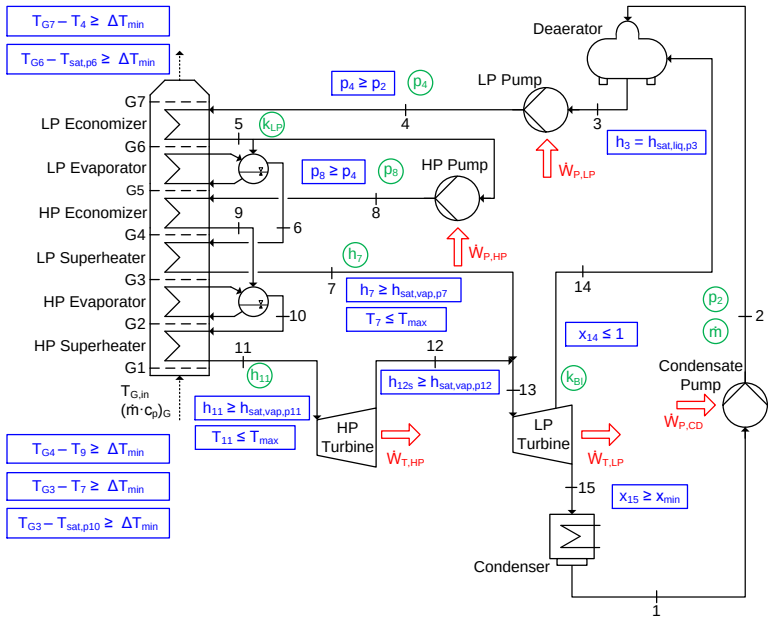


Fig. 7: Case Study III: Two-pressure cycle. In formulation (RS*), only the variables shown in circles and the constraints shown in boxes are handled by the optimizer in order to either maximize $\dot{W}_{net}$ or minimize LCOE. Since there is one equality constraint in this formulation, the problem has seven degrees of freedom.

are required to ensure a minimum temperature difference in the HRSG, full evaporation in both the LP and HP part of the HRSG, and for ensuring validity of the selected model at the HP turbine outlet (in this case, the enthalpy of the hypothetical isentropic turbine outlet state 12*s* is required to be above the saturated vapor enthalpy). A description of the calculation sequence can be found in Appendix B.3.

For this more complex case study, the relative optimality tolerance was set to $10^{-2}$, which is still well below the model uncertainty given the simplicity of the thermodynamic models and the inherent uncertainty of the investment cost correlations [58]. In this case, BARON is considerably faster solving (FS) than the present solver solving (RS*) when maximizing $\dot{W}_{net}$ (see Table 11). A solution of (FS) with the present solver was not attempted given the long solution times for (RS*). When minimizing LCOE, the solution of both formulations with the respective solvers takes longer than $10^5$ s (cf. Table 12). However, in the solution of (RS*) with the present solver, the optimality gap is closed faster after the first 0.5 to 3 h, leading to a remaining gap of around 6% (or less than 5% with simple range reduction) as compared to 15% (or 13% with selective branching) for solving (FS) with BARON when reaching the time limit (see Fig. 8).

Interestingly, the best solution found by the present solver is 0.4% better than that found by BARON (cf. Table 10). The fact that the values of the optimization variables differ significantly between the two solutions suggest the existence of multiple local optima. In fact, solving (FS) from different starting points with the local solvers CONOPT [23], IPOPT [61],

Table 10: Optimal objective values and ranges and optimal solutions for the optimization variables of (RS*) for Case Study III. Since for the minimization of LCOE the solutions returned by BARON and the present solver differ (substantially in the variables, insignificantly in the objective), the latter are given in parentheses.

| Symbol | Description | Unit | Range | $\max \dot{W}_{\text{net}}$ | $\min$ LCOE |
|---|---|---|---|---|---|
| Optimization variables | | | | | |
| $p_2$ | Deaerator pressure | bar | [ 0.2, 5] | 0.2 | 0.2 (0.2) |
| $p_4$ | LP pressure level | bar | [ 3, 15] | 9.20 | 4.78 (15.0) |
| $p_8$ | HP pressure level | bar | [ 10, 100] | 100 | 47.7 (47.0) |
| $\dot{m}$ | Cycle mass flow rate | kg/s | [ 5, 100] | 29.4 | 29.1 (28.0) |
| $h_7$ | LP steam enthalpy | kJ/kg | [2480, 3750] | 3040 | 2715 (2919) |
| $h_{11}$ | HP steam enthalpy | kJ/kg | [2480, 3750] | 3640 | 3480 (3489) |
| $k_{\text{Bl}}$ | Fraction of mass extracted | - | [ 0.01, 0.2] | 0.0347 | 0.0347 (0.0348) |
| $k_{\text{LP}}$ | Fraction of mass in LP part | - | [ 0.05, 0.5] | 0.235 | 0.216 (0.248) |
| Objective functions | | | | | |
| $\dot{W}_{\text{net}}$ | Net power output | MW | | 39.3 | |
| LCOE | Levelized cost of electricity | \$/MWh | | | 49.8 (49.6) |

Table 11: Problem size and solution times for Case Study III maximizing $\dot{W}_{\text{net}}$.

| Formul. | Problem size | Solver | Time [s] | Iterations | Max. nodes |
|---|---|---|---|---|---|
| (FS) | 114 variables<br>107 equalities<br>10 inequalities | | | | |
| | | BARON | 202 | $1.50 \times 10^4$ | 1429 |
| | | BARON (sel. branching) | 196 | $1.22 \times 10^4$ | 1655 |
| (RS*) | 8 variables<br>1 equality<br>14 inequalities | | | | |
| | | Present solver (no RR) | 8502 | $1.11 \times 10^7$ | $1.66 \times 10^6$ |
| | | Present solver (with RR) | 624 | $2.65 \times 10^5$ | $6.32 \times 10^4$ |

Table 12: Problem size and solution progress for Case Study III minimizing LCOE and a time limit of $10^5$ s.

| | | | | LBD after | | |
|---|---|---|---|---|---|---|
| Formul. | Problem size | Solver | 100 s | 1 h | 10 h | 24 h |
| (FS) | 190 variables<br>183 equalities<br>10 inequalities | | | | | |
| | | BARON | 38.6 | 41.1 | 41.9 | 42.2 |
| | | BARON (sel. branching) | 40.2 | 42.1 | 43.1 | 43.4 |
| (RS*) | 8 variables<br>1 equality<br>22 inequalities | | | | | |
| | | Present solver (no RR) | 26.8 | 40.1 | 45.0 | 46.4 |
| | | Present solver (with RR) | 31.7 | 42.1 | 46.5 | 47.2 |

and KNITRO [17] returned several different solutions declared as locally optimal with objective values ranging from 49.6 \$/MWh to 56.3 \$/MWh.

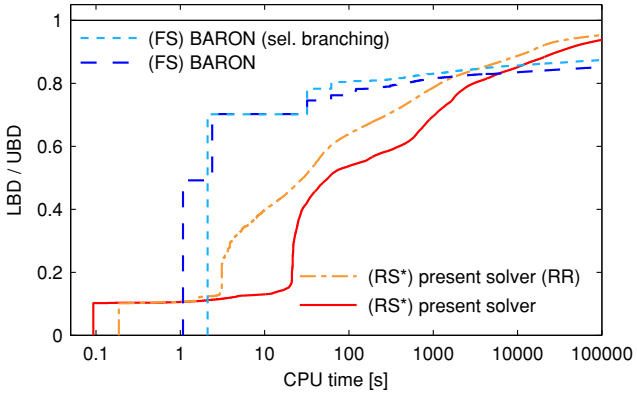


Fig. 8: Convergence of Case Study III minimizing LCOE. After the first 0.5 to 3 h, the optimality gap is closed faster when solving the reduced-space formulation (RS*) with a simple solver than when solving the full-space formulation (FS) with the state-of-the-art solver BARON.

Finally it should be noted that in many cases the present solver requires more iterations and also holds more nodes in memory than BARON. When minimizing LCOE for the present case study, it fully utilizes the available RAM towards the end of the runtime. This is a known possible drawback of pure best-first heuristics in B&B [38], and it is probably exacerbated for the present solver by the extremely simple relaxation technique using linearization at a single point (cf. Section 4).

## 6 Conclusions

A problem formulation for deterministic global optimization of process flowsheets analogous to the sequential modular mode of flowsheet simulation was presented that ensures the applicability of the automatic propagation of McCormick relaxations and their subgradients. This formulation operates in a reduced space containing only the design variables and those variables that are required for decoupling the model equations that would require iterative solution. Only the equations coupling the systems are left to the optimizer as equality constraints, while all others are hidden in externally defined functions. Depending on the process considered, this approach leads to significantly smaller problems compared to the conventional full-space formulation. Furthermore, the number of variables for which bounds need to be provided is reduced significantly, while some additional inequalities may be required instead to enforce physically important restrictions on intermediate variables that are not controlled directly by the optimizer. While not necessarily linked to a reduced-space formulation, the approach of providing model equations externally is also attractive from a modeling perspective since it facilitates the construction of model libraries for process units or thermodynamics calculations.

The two formulations were tested on three steam power cycles of increasing complexity. It was shown that the reduced-space formulation can enable deterministic global flowsheet optimization even with a simple B&B algorithm without any range reduction. Both the number of iterations and the CPU time are significantly lower than for solving the full-space formulation with the same solver. In several examples, the solution of the reduced-space

formulation with a simple solver can be as fast as or even faster than the solution of the full-space formulation in the state-of-the-art solver BARON, which uses a much more sophisticated branch-and-reduce scheme. The observed differences in performance are greater when minimizing LCOE than when maximizing power output for the given cycles. This can be attributed to the increased complexity and nonlinearity when adding component sizing and investment cost calculation, and apparently some of the added equations are particularly difficult for the current version of BARON, probably resulting in loose relaxations. Since the cycles considered as case studies herein occur in similar form in many more complex systems such as, e.g., advanced zero emission power cycles [28], the present calculations show that this reduced-space approach can be used in these cases as a fast and robust way to globally solve these subproblems.

It should be emphasized that the relaxations used in the present simple solver are relatively weak because of the linearization at a single point. Further improvements could thus likely be achieved in this framework by making better use of the propagated relaxations, e.g., by linearizing at multiple points chosen in a suitable way, or using the nonsmooth relaxations returned from the external functions directly in a nonsmooth NLP solver. However, while tighter relaxations will reduce the number of iterations, it remains to be shown how it affects the solution time, since in the former case the lower bounding problems become larger (in case of multiple linearization points) and in the latter case more challenging due to their nonlinearity. One could also rely on the recently developed smooth modifications of McCormick relaxations [34] in order to utilize regular NLP solvers for lower bounding. A nonsmooth NLP solver could also be used for upper bounding to extend the applicability of the present solver to nonsmooth unit operations. The present approach could also be integrated with more sophisticated branch-and-reduce solvers as soon as they allow the use of external functions for supplying relaxations. However, in this case it would have to be evaluated which of the features for range reduction are still applicable when there is no direct access to all equations and which are not.

Future work should investigate to what extent the advantages of the reduced-space formulation still hold for larger flowsheets, in particular if the difference in problem size to the full space formulation decreases, i.e., for flowsheets requiring many tear streams and containing many implicit unit operations. A first study based on the present work indicates that in such cases careful selection of tear locations as well as variables and equations to be left to the optimizer is required to fully exploit the benefits of the reduced-space formulation [15]. The implications of such decisions should be clarified in order to identify suitable approaches for model formulation. This is expected to be particularly relevant when applying the present approach to chemical processes that exhibit complex recycle structures as well as challenging unit operations because the mixtures involved often require iterative solution to calculate phase equilibria. It might also be possible to automate the construction of such reduced-space formulations from the structure of the full-space problem, thus avoiding the effort of having to find a calculation sequence manually at the modeling level. Finally, it would be interesting to compare the present approach to a fully implicit treatment of the model equations, as well as explore ways of incorporating the methods for handling implicit functions [50,54,63] into similar approaches. Since one could choose which equations to eliminate by treating them as an implicit function within the external functions and which to leave to the optimizer, this opens up even more interesting alternatives.

## A Formulation (RS*) does not introduce multimodality

In the following, we show that (RS*) does not introduce additional local minima compared to (FS). For ease of analysis, we recast the formulations in slightly different (and more general) form.

Let $\mathbf{x} \in X \subset \mathbb{R}^{n_x}$, $\mathbf{y} \in Y \subset \mathbb{R}^{n_y}$, with $X, Y$ nonempty compact convex sets, and the functions $f : X \times Y \to \mathbb{R}$, $\mathbf{h}_{\text{exp}} : X \to Y$, $\mathbf{h}_{\text{imp}} : X \times Y \to \mathbb{R}^{n_{h,\text{imp}}}$, $\mathbf{g} : X \times Y \to \mathbb{R}^{n_g}$ continuous. Consider the NLP

$$\begin{aligned} \min_{(\mathbf{x},\mathbf{y}) \in X \times Y} \quad & f(\mathbf{x},\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{h}_{\text{exp}}(\mathbf{x}) \\ & \mathbf{h}_{\text{imp}}(\mathbf{x},\mathbf{y}) = \mathbf{0} \\ & \mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0} \end{aligned} \tag{A-FS}$$

Let $\mathbf{y}$ and $\mathbf{h}_{\text{exp}}$ be selected such that the equality constraints $\mathbf{y} = \mathbf{h}_{\text{exp}}(\mathbf{x})$ are of the form

$$\begin{aligned} y_1 &= \hat{h}_{\text{exp},1}(\mathbf{x}), \\ y_i &= \hat{h}_{\text{exp},i}(\mathbf{x}, \hat{h}_{\text{exp},1}(\mathbf{x}), ..., \hat{h}_{\text{exp},i-1}(\mathbf{x})), \qquad i = 2, ..., n_y, \end{aligned}$$

where $\hat{h}_{\text{exp,i}}, i = 1, ..., n_y$ consist only of compositions of binary sums, binary products, or univariate or multivariate functions from a given library as discussed in Section 2. In total, the constraints $\mathbf{y} = \mathbf{h}_{\text{exp}}(\mathbf{x})$ can thus be evaluated sequentially to compute a unique vector $\mathbf{y} \in Y$ for a given $\mathbf{x} \in X$. Note, however, that this mapping need not be injective nor surjective.

The full-space formulation (A-FS) can be converted to the following reduced-space formulation, which is a generalization of (RS*):

$$\begin{aligned} \min_{\mathbf{x} \in X} \quad & f(\mathbf{x}, \mathbf{h}_{\text{exp}}(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{h}_{\text{imp}}(\mathbf{x}, \mathbf{h}_{\text{exp}}(\mathbf{x})) = \mathbf{0} \\ & \mathbf{g}(\mathbf{x}, \mathbf{h}_{\text{exp}}(\mathbf{x})) \leq \mathbf{0} \end{aligned} \tag{A-RS*}$$

Note that in (A-RS*) the substitution of $\mathbf{y}$ by $\mathbf{h}_{\text{exp}}(\mathbf{x})$ need not be done symbolically, but can rather be deferred to the time of function evaluation.

**Proposition 1** *A point $\mathbf{x}^* \in X$ is a local solution of* (A-RS*) *if and only if there is a $\mathbf{y}^* \in Y$ such that $(\mathbf{x}^*, \mathbf{y}^*)$ is a local solution of* (A-FS).

*Proof* The feasible regions of (A-FS) and (A-RS*) are

$$\begin{aligned} \mathscr{F}_{\text{FS}} &:= \{(\mathbf{x},\mathbf{y}) \in X \times Y : \ \mathbf{y} = \mathbf{h}_{\text{exp}}(\mathbf{x}),\ \mathbf{h}_{\text{imp}}(\mathbf{x},\mathbf{y}) = \mathbf{0},\ \mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}\}, \\ \mathscr{F}_{\text{RS}} &:= \{\mathbf{x} \in X : \ \mathbf{h}_{\text{imp}}(\mathbf{x}, \mathbf{h}_{\text{exp}}(\mathbf{x})) = \mathbf{0},\ \mathbf{g}(\mathbf{x}, \mathbf{h}_{\text{exp}}(\mathbf{x})) \leq \mathbf{0}\}, \end{aligned}$$

respectively, and it holds that

$$\mathbf{x} \in \mathscr{F}_{\text{RS}} \wedge \mathbf{y} = \mathbf{h}_{\text{exp}}(\mathbf{x}) \iff (\mathbf{x},\mathbf{y}) \in \mathscr{F}_{\text{FS}}. \tag{1}$$

Assume a point $(\mathbf{x}^*, \mathbf{y}^*)$ is a local solution of (A-FS). Then by definition (cf., e.g., [38]) there exists $\varepsilon > 0$ such that

$$\begin{aligned} & (\mathbf{x}^*, \mathbf{y}^*) \in \mathscr{F}_{\text{FS}}, \\ & f(\mathbf{x}^*, \mathbf{y}^*) \leq f(\mathbf{x}, \mathbf{y}) \quad \forall \ (\mathbf{x},\mathbf{y}) \in \mathscr{F}_{\text{FS}} \cap \mathscr{N}_{\text{FS},\varepsilon,(\mathbf{x}^*,\mathbf{y}^*)}, \\ & \mathscr{N}_{\text{FS},\varepsilon,(\mathbf{x}^*,\mathbf{y}^*)} := \{(\mathbf{x},\mathbf{y}) \in X \times Y : \ \|(\mathbf{x},\mathbf{y}) - (\mathbf{x}^*,\mathbf{y}^*)\| \leq \varepsilon\}. \end{aligned} \tag{2}$$

From (1) it follows that $\mathbf{x}^* \in \mathscr{F}_{\text{RS}}$ and $\mathbf{y}^* = \mathbf{h}_{\text{exp}}(\mathbf{x}^*)$. Thus, from (2) it follows

$$f(\mathbf{x}^*, \mathbf{h}_{\text{exp}}(\mathbf{x}^*)) \leq f(\mathbf{x}, \mathbf{h}_{\text{exp}}(\mathbf{x})) \quad \forall \ (\mathbf{x},\mathbf{y}) \in \mathscr{F}_{\text{FS}} \cap \mathscr{N}_{\text{FS},\varepsilon,(\mathbf{x}^*,\mathbf{y}^*)}.$$

Take $\mathscr{N}_{\text{RS},\hat{\varepsilon},\mathbf{x}^*} := \{\mathbf{x} \in X : \ \|\mathbf{x} - \mathbf{x}^*\| \leq \hat{\varepsilon}\}$ with $\hat{\varepsilon} := \max_{(\mathbf{x},\mathbf{y}) \in \mathscr{F}_{\text{FS}} \cap \mathscr{N}_{\text{FS},\varepsilon,(\mathbf{x}^*,\mathbf{y}^*)}} \|\mathbf{x} - \mathbf{x}^*\|$. Note that $\hat{\varepsilon}$ exists by continuity of $\mathbf{h}_{\text{exp}}$, $\mathbf{h}_{\text{imp}}$, and $\mathbf{g}$ and compactness of $X$ and $Y$. It follows

$$f(\mathbf{x}^*, \mathbf{h}_{\text{exp}}(\mathbf{x}^*)) \leq f(\mathbf{x}, \mathbf{h}_{\text{exp}}(\mathbf{x})) \quad \forall \ \mathbf{x} \in \mathscr{F}_{\text{RS}} \cap \mathscr{N}_{\text{RS},\hat{\varepsilon},\mathbf{x}^*}, \tag{3}$$

which together with $\mathbf{x}^* \in \mathscr{F}_{\mathrm{RS}}$ shows that $\mathbf{x}^*$ is a local solution of (A-RS*).

Assume, on the other hand, that $\mathbf{x}^*$ is a local solution of (A-RS*), thus satisfying $\mathbf{x}^* \in \mathscr{F}_{\mathrm{RS}}$ and (3) for some $\hat{\varepsilon} > 0$. We define the vector $\mathbf{y}^* \in Y$ as $\mathbf{y}^* = \mathbf{h}_{\mathrm{exp}}(\mathbf{x}^*)$ and by (1) we obtain that $(\mathbf{x}^*, \mathbf{y}^*) \in \mathscr{F}_{\mathrm{FS}}$ and furthermore

$$f(\mathbf{x}^*, \mathbf{y}^*) \leq f(\mathbf{x}, \mathbf{h}_{\mathrm{exp}}(\mathbf{x})) \quad \forall\ \mathbf{x} \in \mathscr{F}_{\mathrm{RS}} \cap \mathscr{N}_{\mathrm{RS},\hat{\varepsilon},\mathbf{x}^*}.$$

Since for any $(\mathbf{x}, \mathbf{y}) \in X \times Y$ we have $\|(\mathbf{x}, \mathbf{y}) - (\mathbf{x}^*, \mathbf{y}^*)\| \geq \|\mathbf{x} - \mathbf{x}^*\|$, using (1) it follows for $\mathscr{N}_{\mathrm{FS},\hat{\varepsilon},(\mathbf{x}^*,\mathbf{y}^*)} :=$ $\{(\mathbf{x}, \mathbf{y}) \in X \times Y :\ \|(\mathbf{x}, \mathbf{y}) - (\mathbf{x}^*, \mathbf{y}^*)\| \leq \hat{\varepsilon}\}$ that

$$f(\mathbf{x}^*, \mathbf{y}^*) \leq f(\mathbf{x}, \mathbf{y}) \quad \forall\ (\mathbf{x}, \mathbf{y}) \in \mathscr{F}_{\mathrm{FS}} \cap \mathscr{N}_{\mathrm{FS},\hat{\varepsilon},(\mathbf{x}^*,\mathbf{y}^*)},$$

which together with $(\mathbf{x}^*, \mathbf{y}^*) \in \mathscr{F}_{\mathrm{FS}}$ shows that $(\mathbf{x}^*, \mathbf{y}^*)$ is a local solution of (A-FS). □

## B Process models

The following sections provide details on the model equations and the calculation sequences for simulating the cycles by sequential evaluation of these equations to obtain the desired thermodynamic quantities, in particular the power output $\dot{W}_{\mathrm{net}}$. Enthalpies and entropies in the process models are computed using the ideal gas and ideal liquid equations of state with constant heat capacities, and saturation temperatures are computed using the Antoine equation (cf., e.g., [11]). Pressure losses in components other than the pumps and turbines are neglected. Tables 13 and 14 summarize the fixed model parameter values for the thermodynamic calculations used in the case studies. A list of all symbols and subscripts used is given in Table 15.

Table 13: Fixed model parameters for the cycles in the case studies. The gas outlet temperature is only fixed for Case Study I. The condenser pressure is 0.2 bar for Case Study I, and 0.05 bar for Case Studies II and III.

| Symbol | Description | Unit | Value |
|---|---|---|---|
| $T_{\mathrm{G,in}}$ | Gas inlet temperature | K | 900 |
| $T_{\mathrm{G,out}}$ | Gas outlet temperature | K | 448 |
| $(\dot{m}c_{\mathrm{p}})_{\mathrm{G}}$ | Heat capacity flow rate of the gas | kJ/K | 200 |
| $\Delta T_{\mathrm{min}}$ | Minimum temperature difference in HRSG | K | 15 |
| $\Delta T_{\mathrm{ap}}$ | Approach to saturation in economizers | K | 10 |
| $\eta_{\mathrm{P}}$ | Isentropic efficiency of pumps | - | 0.8 |
| $\eta_{\mathrm{T}}$ | Isentropic efficiency of turbines | - | 0.9 |
| $T_{\mathrm{max}}$ | Maximum live steam temperature | K | 873 |
| $x_{\mathrm{min}}$ | Minimum vapor quality in turbine | - | 0.85 |
| $p_1$ | Condenser pressure | bar | 0.2 or 0.05 |
| $x_1$ | Vapor quality condenser outlet | - | 0 |

Table 14: Fixed model parameters for the thermodynamic properties of water.

| Symbol | Description | Unit | Value |
|---|---|---|---|
| $c_{\mathrm{if}}$ | Specific heat capacity (liquid) | kJ/(kg K) | 4.18 |
| $c_{\mathrm{p,ig}}$ | Specific heat capacity (gas) | kJ/(kg K) | 2.08 |
| $R$ | Specific gas constant | kJ/(kg K) | 0.462 |
| $v_{\mathrm{if}}$ | Specific volume (liquid) | $\mathrm{m}^3$/kg | 0.001 |
| $p_0$ | Reference pressure | bar | 0.01 |
| $\Delta h_{\mathrm{evap,p0}}$ | Enthalpy of vaporization at $p_0$ | kJ K/kg | 2480 |
| A | Antoine parameter | - | 3.5595 |
| B | Antoine parameter | K | 643.748 |
| C | Antoine parameter | K | -198.043 |

Table 15: List of symbols and subscripts used in the process models of the case studies.

| | | | |
|---|---|---|---|
| *Symbols* | | LMTD | Logarithmic mean temperature difference [K] |
| $\dot{m}$ | Mass flow rate [kg/s] | | |
| $\dot{Q}$ | Heat flow rate [kW] | *Subscripts* | |
| $\dot{W}$ | Power [kW] | ap | Approach to saturation |
| $\eta$ | Isentropic efficiency [-] | B | Boiler |
| $\phi$ | Fixed operating cost factor [-] | Bl | Bleed stream |
| $\Psi$ | Annuity factor [-] | CCPP | Combined cycle power plant |
| $A$ | Heat transfer area [$m^2$] | CD | Condensate |
| $A,B,C$ | Parameters for Antoine equation [K,-] | Chen | Chen approximation |
| $C$ | Parameter for pressure factor [-] | Cond | Condenser |
| $C_{\text{Fuel}}$ | Specific fuel cost [$/MWh] | crit | Critical point |
| $c_{\text{if}}$ | Ideal liquid heat capacity [kJ K/kg] | cw | Cooling water |
| $c_{\text{p,ig}}$ | Ideal gas isobaric heat capacity [kJ K/kg] | Dae | Deaerator |
| $C_p$ | Base purchase cost [US-$] | evap | Evaporation |
| $F_p$ | Pressure factor [-] | FW | Feedwater |
| $h$ | Specific enthalpy [kJ/kg] | G | Exhaust gas of gas turbine |
| $K$ | Cost parameter [-] | GT | Gas turbine |
| $k$ | Heat transfer coefficient [kW K/$m^2$] | HP | High pressure |
| $p$ | Pressure [bar] | i | Index for streams |
| $R$ | Specific gas constant [kJ K/kg] | in | Inlet |
| $s$ | Specific entropy [kJ K/kg] | j | Index for heat exchangers |
| $T$ | Temperature [K] | liq | Liquid |
| $T_{eq}$ | Equivalent utilization time at rated power [h/a] | LP | Low pressure |
| | | max | Maximum allowable value |
| $TCI$ | Total capital investment [$/MWh] | min | Minimum allowable value |
| $u_{\text{var}}$ | Variable cost factor [$/MWh] | net | Net |
| $V$ | Volume [$m^3$] | out | Outlet |
| $v_{\text{if}}$ | Specific volume (liquid) [$m^3$/kg] | P | Pump |
| $w$ | Specific work [kJ/kg] | pi | Evaluated at pressure $p_i$ |
| $x$ | Vapor quality [-] | ref | Reference state for Watson equation |
| a | Parameter Watson equation [-] | s | Hypothetical isentropic state |
| c | Parameter ideal gas heat capacity [K,kJ,kg] | sat | Saturated state |
| | | T | Turbine |
| Inv | Investment cost [$] | Ti | Evaluated at temperature $T_i$ |
| LCOE | Levelized cost of electricity [$/MWh] | vap | Vapor |

## B.1 Case Study I: Basic Rankine Cycle

For every stream $i$, the saturation temperature is computed from the pressure $p_i$ of the stream via the Antoine equation:

$$T_{\text{sat,pi}} = \frac{B}{A - \log_{10}(\frac{p_i}{\text{bar}})} - C.$$

Where required, the saturated vapor and liquid enthalpy and entropy can be computed via:

$$\begin{aligned}
h_{\text{sat,vap,pi}} &= \Delta h_{\text{evap,p0}} + c_{\text{p,ig}} \cdot (T_{\text{sat,pi}} - T_0) \\
h_{\text{sat,liq,pi}} &= c_{\text{if}} \cdot (T_{\text{sat,p1}} - T_0) + v_{\text{if}} \cdot (p_1 - p_0) \\
s_{\text{sat,vap,pi}} &= \frac{\Delta h_{\text{evap,p0}}}{T_0} + c_{\text{p,ig}} \cdot \ln\left(\frac{T_{\text{sat,pi}}}{T_0}\right) - R \cdot \ln\left(\frac{p_i}{p_0}\right) \\
s_{\text{sat,liq,pi}} &= c_{\text{if}} \cdot \ln\left(\frac{T_{\text{sat,pi}}}{T_0}\right),
\end{aligned}$$

where the reference temperature $T_0$ is also obtained from the Antoine equation.

To simulate the cycle, we start at the condenser outlet. Since the condenser pressure $p_1$ is treated as a fixed parameter and by assumption the fluid leaves the condenser in the saturated liquid state, we have

$$h_1 = h_{\text{sat,liq,p1}}.$$

For the pump, we can compute the specific pump work and power consumption using the isentropic efficiency $\eta_\mathrm{P}$:

$$w_\mathrm{P} = \frac{v_\mathrm{if} \cdot (p_2 - p_1)}{\eta_\mathrm{P}}$$
$$\dot{W}_\mathrm{P} = \dot{m} \cdot w_\mathrm{P}.$$

The outlet state then follows from the energy balance:

$$h_2 = h_1 + w_\mathrm{P}. \tag{4}$$

Since the gas outlet temperature is fixed, we first calculate the overall heat transfer rate in the boiler, which is assumed to be isobaric ($p_5 = p_4 = p_3 = p_2$):

$$\dot{Q}_\mathrm{B} = (\dot{m}c_\mathrm{p})_\mathrm{G} \cdot (T_\mathrm{G,in} - T_\mathrm{G,out})$$

In the economizer, the outlet enthalpy of the water can be computed using the known outlet temperature $T_3 = T_\mathrm{sat,p3} - \Delta T_\mathrm{ap}$:

$$h_3 = c_\mathrm{if} \cdot (T_3 - T_0) + v_\mathrm{if} \cdot (p_3 - p_0).$$

Using an energy balance around the economizer, the gas temperature between economizer and evaporator can be calculated as

$$T_\mathrm{G3} = T_\mathrm{G,out} + \frac{\dot{m} \cdot (h_3 - h_2)}{(\dot{m}c_\mathrm{p})_\mathrm{G}}.$$

In the evaporator, the water leaving the steam drum towards the superheater is in the saturated vapor state, so that we obtain similarly:

$$T_4 = T_\mathrm{sat,p4}$$
$$h_4 = h_\mathrm{sat,vap,p4}$$
$$T_\mathrm{G2} = T_\mathrm{G,in} - \frac{\dot{m} \cdot (h_5 - h_4)}{(\dot{m}c_\mathrm{p})_\mathrm{G}}.$$

In the superheater, the live steam enthalpy can be computed from the energy balance since the gas inlet temperature is known:

$$h_5 = h_2 + \frac{\dot{Q}_\mathrm{B}}{\dot{m}}. \tag{5}$$

In the turbine, the inlet temperature and entropy can be computed as

$$T_5 = T_0 + \frac{h_5 - \Delta h_\mathrm{evap,p0}}{c_\mathrm{p,ig}}$$
$$s_5 = \frac{\Delta h_\mathrm{evap,p0}}{T_0} + c_\mathrm{p,ig} \cdot \ln\left(\frac{T_5}{T_0}\right) - R \cdot \ln\left(\frac{p_5}{p_0}\right).$$

From this, the specific turbine work and power output are computed using the isentropic efficiency $\eta_\mathrm{T}$, similar to the pump, with the hypothetical isentropic turbine outlet state 6$s$ being in the two-phase region.

$$p_6 = p_1$$
$$p_{6s} = p_1$$
$$x_\mathrm{6s} = \frac{s_5 - s_\mathrm{sat,liq,p6}}{s_\mathrm{sat,vap,p6} - s_\mathrm{sat,liq,p6}}$$
$$h_\mathrm{6s} = h_\mathrm{sat,liq,p6} + x_\mathrm{6s} \cdot (h_\mathrm{sat,vap,p6} - h_\mathrm{sat,liq,p6})$$
$$w_\mathrm{T} = \eta_\mathrm{T} \cdot (h_5 - h_\mathrm{6s})$$
$$\dot{W}_\mathrm{T} = \dot{m} \cdot w_\mathrm{T}.$$

An energy balance then yields the true outlet state 6:

$$h_6 = h_5 - w_\mathrm{T}$$
$$x_6 = \frac{h_6 - h_{\mathrm{sat,liq,p6}}}{h_{\mathrm{sat,vap,p6}} - h_{\mathrm{sat,liq,p6}}}.$$

Note that when computed this way, the vapor quality $x_6$ is greater than unity if the enthalpy $h_6$ is greater than the saturated vapor enthalpy at $p_6$. Therefore, the condition $x_6 \leq 1$ can be used to ensure the validity of the assumption of 6 (and hence also 6$s$) being in the two-phase region.

Finally, the net power output of the cycle is

$$\dot{W}_\mathrm{net} = \dot{W}_\mathrm{T} - \dot{W}_\mathrm{P}.$$

## B.2 Case Study II: Regenerative Rankine Cycle

In this case, the simulation of the cycle starts at the turbine inlet since the pressure ($p_7 = p_4$) and enthalpy ($h_7$ is an optimization variable) are known. The turbine with bleed extraction can be modeled as two separate turbines in parallel, each of which is treated as described in Section B.1. The one associated with the bleed stream expands to the bleed pressure that is equal to the deaerator pressure $p_2$, while the other one expands to the condenser pressure $p_1$. The power output of the turbine can be obtained as the sum of these two parts ($\dot{W}_\mathrm{T} = \dot{W}_\mathrm{T,Bl} + \dot{W}_\mathrm{T,Cond}$). The condenser outlet state and the condensate (CD) pump can be modeled as above, with the exception that only the mass flow that is not extracted as a bleed contributes to the pump power consumption:

$$\dot{W}_\mathrm{P,CD} = \dot{m} \cdot (1 - k_\mathrm{Bl}) \cdot w_\mathrm{P}.$$

The deaerator is assumed to be isobaric and its outlet enthalpy follows from the energy balance:

$$p_3 = p_2$$
$$h_3 = k_\mathrm{Bl} \cdot h_8 + (1 - k_\mathrm{Bl}) \cdot h_2.$$

The calculation of the feedwater (FW) pump is analogous to the condensate pump, but with the entire cycle mass flow. Since in this case study the gas outlet temperature is not fixed any more, the overall heat transfer rate is determined using the known inlet and outlet enthalpies of the water:

$$\dot{Q}_\mathrm{B} = \dot{m} \cdot (h_7 - h_4)$$
$$T_\mathrm{G4} = T_\mathrm{G,in} - \frac{\dot{Q}_\mathrm{B}}{(\dot{m} c_\mathrm{p})_\mathrm{G}}.$$

From this, the missing quantities for the economizer and evaporator can be calculated as described above.

Finally, the net power output of the cycle is

$$\dot{W}_\mathrm{net} = \dot{W}_\mathrm{T} - \dot{W}_\mathrm{P,CD} - \dot{W}_\mathrm{P,FW}.$$

## B.3 Case Study III: Two-Pressure Cycle

For convenience, variables for the mass flow rates through the LP and HP parts of the HRSG and the ones of the turbine bleed and that being expanded to the condenser are defined as

$$\dot{m}_\mathrm{LP} = \dot{m} \cdot k_\mathrm{LP}$$
$$\dot{m}_\mathrm{HP} = \dot{m} \cdot (1 - k_\mathrm{LP})$$
$$\dot{m}_\mathrm{Bl} = \dot{m} \cdot k_\mathrm{Bl}$$
$$\dot{m}_\mathrm{Cond} = \dot{m} \cdot (1 - k_\mathrm{Bl}).$$

We start at the HP turbine inlet, since its state is known (cf. above). For the HP turbine, the outlet state is assumed to be in the vapor region (which is ensured by the constraint $h_{12\mathrm{s}} \geq h_{\mathrm{sat,vap,p12}}$). Therefore, the

temperature and enthalpy of the isentropic outlet state is calculated based on the corresponding ideal gas equations, while the rest of the equations remains the same as above:

$$\begin{aligned}
p_{12} &= p_4 \\
p_{12\mathrm{s}} &= p_4 \\
s_{12\mathrm{s}} &= s_{11} \\
T_{12\mathrm{s}} &= T_0 \cdot \exp\left(\frac{s_{12\mathrm{s}} + R \cdot \ln(p_{12s}/p_0) - \Delta h_{\mathrm{evap,p0}}/T_0}{c_{\mathrm{p,ig}}}\right) \\
h_{12\mathrm{s}} &= \Delta h_{\mathrm{evap,p0}} + c_{\mathrm{p,ig}} \cdot (T_{12\mathrm{s}} - T_0) \\
w_{\mathrm{T,HP}} &= \eta_{\mathrm{T}} \cdot (h_{11} - h_{12\mathrm{s}}) \\
h_{12} &= h_{11} - w_{\mathrm{T,HP}} \\
\dot{W}_{\mathrm{T,HP}} &= \dot{m}_{\mathrm{HP}} \cdot w_{\mathrm{T,HP}}.
\end{aligned}$$

The mixing of the HP turbine outlet stream 12 with the outlet stream 7 of the LP superheater is analogous to the model of the deaerator described above. The models for the LP turbine, condenser, condensate pump, deaerator, and LP pump (corresponding to the feedwater pump) are the same as described above and can be evaluated in this order. The HP pump is analogous to the LP pump but uses only the mass flow of the HP part of the cycle. The HRSG can then be evaluated in a similar manner as described above, starting from the HP superheater and working back to the LP economizer. The net power output finally follows as

$$\dot{W}_{\mathrm{net}} = \dot{W}_{\mathrm{T,HP}} + \dot{W}_{\mathrm{T,LP}} - \dot{W}_{\mathrm{P,CD}} - \dot{W}_{\mathrm{P,LP}} - \dot{W}_{\mathrm{P,HP}}.$$

## B.4 Case Study I with Temperature-Dependent Sub-Models

Case Study I was also repeated using temperature-dependent sub-models for the ideal gas heat capacity and enthalpy of vaporization:

$$\begin{aligned}
c_{\mathrm{p,ig}}(T) &= c_1 + c_2 \cdot T + c_3 \cdot T^2, \\
\Delta h_{\mathrm{evap}}(T) &= \Delta h_{\mathrm{evap,Tref}} \cdot \left(\frac{1 - T/T_{\mathrm{crit}}}{1 - T_{\mathrm{ref}}/T_{\mathrm{crit}}}\right)^a.
\end{aligned}$$

The corresponding parameters are given in Table 16. Note that since we decided to use the ideal gas heat capacity (rather than liquid), for convenience the reference state is shifted to the dew curve at $p_0$. Using these sub-models, the enthalpy and entropy for streams in the gas phase can be computed for given $p_i$ and $T_i$ as

$$\begin{aligned}
h_{\mathrm{i}} &= \int_{T_0}^{T_i} c_{\mathrm{p,ig}}(T)dT, \\
s_{\mathrm{i}} &= \int_{T_0}^{T_i} \frac{c_{\mathrm{p,ig}}(T)}{T}dT - R \cdot \ln\left(\frac{p_i}{p_0}\right),
\end{aligned}$$

while for liquid streams they are given by

$$\begin{aligned}
h_{\mathrm{i}} &= \int_{T_0}^{T_i} c_{\mathrm{p,ig}}(T)dT - \Delta h_{\mathrm{evap}}(T_i) + v_{\mathrm{if}} \cdot (p_i - p_{\mathrm{sat,Ti}}), \\
s_{\mathrm{i}} &= \int_{T_0}^{T_i} \frac{c_{\mathrm{p,ig}}(T)}{T}dT - R \cdot \ln\left(\frac{p_{\mathrm{sat,Ti}}}{p_0}\right) - \frac{\Delta h_{\mathrm{evap}}(T_i)}{T_i}.
\end{aligned}$$

The vapor pressure at the stream temperature is again obtained from the Antoine equation:

$$p_{\mathrm{sat,Ti}} = 10^{A - \frac{B}{C + T_i}}\,\mathrm{bar}.$$

Table 16: Fixed model parameters for the thermodynamic properties of water using temperature dependent sub-models.

| Symbol | Description | Unit | Value |
|---|---|---|---|
| $c_1$ | Parameter ideal gas heat capacity | $\mathrm{kJ}/(\mathrm{kg\,K})$ | 1.995 |
| $c_2$ | Parameter ideal gas heat capacity | $\mathrm{kJ}/(\mathrm{kg\,K}^2)$ | $-7.027\times10^{-4}$ |
| $c_3$ | Parameter ideal gas heat capacity | $\mathrm{kJ}/(\mathrm{kg\,K}^3)$ | $8.476\times10^{-7}$ |
| $T_{\mathrm{crit}}$ | Critical temperature of water | K | 647 |
| $T_{\mathrm{ref}}$ | Reference temperature for Watson equation | K | 273 |
| $\Delta h_{\mathrm{evap,Tref}}$ | Enthalpy of vaporization at $T_{\mathrm{ref}}$ | kJ/kg | 2501.3 |
| $a$ | Parameter Watson equation | - | 0.38 |

Unlike for the simple thermodynamic model used in the other case studies, these expressions for enthalpy and entropy cannot be solved for the temperature analytically[4]. Therefore, some additional variables and equality constraints have to be introduced when optimizing the cycle using formulation (RS*). This is the case for Streams 2 and 5, the state of which is determined from energy balances (cf. Section B.1). Thus, their temperatures $T_2$ and $T_5$ are handed to the optimizer as additional module variables ($\mathbf{x}_m$), and Equations (4) and (5) are added as additional module equations ($\tilde{\mathbf{h}}_m = \mathbf{0}$). Note that for the two-phase streams 6 and 6$s$, the vapor fraction can still be computed from the given enthalpies or entropies as described in Section B.1 so that no additional variables are needed.

## C Economic analysis

The LCOE of the CCPP is calculated according to the equation [51,33]

$$\mathrm{LCOE} = \frac{TCI\cdot\Psi\cdot\varphi}{\dot{W}_{\mathrm{CCPP}}\cdot T_{\mathrm{eq}}} + \frac{C_{\mathrm{Fuel}}}{\eta_{\mathrm{CCPP}}} + u_{\mathrm{var}},$$

where $TCI$, $\dot{W}_{\mathrm{CCPP}}$ and $\eta_{\mathrm{CCPP}}$ denote the total capital investment, the net power output, and the efficiency of the CCPP, respectively. The remaining quantities are constant parameters that can be found in Table 17. The annuity factor was determined as described in ref. [51] assuming a depreciation period of 20 years and a construction time of 2 years, as well as their values for interest and inflation rates.

While the gas turbine is not considered in the optimization itself since its design is assumed to be fixed, some data is required for evaluating the aforementioned quantities. To this end, it was simulated in AspenPlus® assuming a pressure ratio of 20 with a turbine inlet temperature of 1620 K and isentropic compressor and turbine efficiencies of 0.8 and 0.9, respectively[5]. These conditions and the mass flow rate through the gas turbine were selected to be in a typical range while matching the assumptions on the exhaust gas flow rate and heat capacity made for the simulation of the bottoming cycle. The resulting net power output of the gas turbine is $\dot{W}_{\mathrm{GT}} = 69.7\,\mathrm{MW}$ while consuming $\dot{Q}_{\mathrm{Fuel}} = 182\,\mathrm{MW}$ (based on lower heating value) of natural gas. From this, the power output of the CCPP follows as the sum of net power output of the gas turbine and that of the steam cycle obtained as described in Appendix B, and the CCPP efficiency can be calculated as $\eta_{\mathrm{CCPP}} = \dot{W}_{CCPP}/\dot{Q}_{\mathrm{Fuel}}$.

The capital investment for the gas turbine as well as the steam cycle is calculated using the cost correlations given in ref. [51] for pumps, steam turbines, and generators, as well as the gas turbine including compressor and combustor which are based on their power and mass flow rates as well as pressures and temperatures. For heat exchangers and deaerators, the more detailed correlations from ref. [58] are used. To this end, the heat transfer areas $A_j$ of the heat exchangers involved (i.e., condenser, economizer, evaporator, and superheater) are computed based on their heat flows $\dot{Q}_j$, inlet and outlet temperature differences $\Delta T_{\mathrm{in,j}}$

[4] While the gas enthalpy can in principle still be solved for temperature, e.g., using Cardano's method, this is challenging without the use of conditional statements and is thus not attempted here.

[5] We decided to use Aspen rather than a basic thermodynamic model to avoid mistakes in setting up the equations and to utilize the temperature dependent property data (in particular heat capacities) that get more important at the high temperatures involved.

and $\Delta T_{\text{out,j}}$, and heat transfer coefficients $k_j$ using Chen's approximation of the logarithmic mean temperature difference (LMTD) [21]:

$$\text{LMTD}_{\text{Chen},j} = \left( \Delta T_{\text{in},j} \cdot \Delta T_{\text{out},j} \cdot \frac{\Delta T_{\text{in},j} + \Delta T_{\text{out},j}}{2} \right)^{1/3}$$

$$A_j = \frac{\dot{Q}_j}{k_j \cdot \text{LMTD}_{\text{Chen},j}}$$

For the heat transfer coefficients, average values are used that depend on the state of the fluids involved (cf. Table 17). For the condenser, water cooling is assumed with specified cooling water inlet and outlet temperatures. The investment cost $\text{Inv}_j$ of the heat exchangers is calculated via the base purchase cost $C_{p,j}$ and the pressure factor $F_{p,j}$ [58]:

$$C_{p,j} = 10^{K_1 + K_2 \cdot \log_{10}(A_j/\text{m}^2) + K_3 \cdot \log_{10}(A_j/\text{m}^2)^2}$$

$$F_{p,j} = 10^{C_1 + C_2 \cdot \log_{10}(p_j/\text{bar}) + C_3 \cdot \log_{10}(p_j\text{bar})^2}$$

$$\text{Inv}_j = 1.18 \cdot (1.63 + 1.66 \cdot 2.75 \cdot F_{p,j}) \cdot C_{p,j}$$

The deaerator is treated as a process vessel, the volume of which is determined for a 10 min liquid holdup with another 50% added for vapor in the vessel [33]:

$$V_{\text{Dae}} = 1.5 \cdot \dot{m}_{\text{Dae,out}} \cdot v_{\text{if}} \cdot 600\,\text{s}$$

$$C_{p,\text{Dae}} = 10^{K_4 + K_5 \cdot \log_{10}(V_{\text{Dae}}/\text{m}^3) + K_6 \cdot \log_{10}(V_{\text{Dae}}/\text{m}^3)^2}$$

$$F_{p,\text{Dae}} = 1.25$$

$$\text{Inv}_{DAE} = 1.18 \cdot (1.49 + 1.52 \cdot F_{p,\text{Dae}}) \cdot C_{p,\text{Dae}}$$

Table 17: Parameters for the economic evaluation taken from refs. [33,51,58,59].

| Symbol | Description | Unit | Value |
|---|---|---|---|
| $\Psi$ | Annuity factor | - | 0.1875 |
| $\varphi$ | Fixed operating cost factor | - | 1.06 |
| $T_{\text{eq}}$ | Equivalent utilization time at rated power | h/a | 4000 |
| $C_{\text{Fuel}}$ | Specific fuel cost | $/MWh$_{\text{fuel}}$ | 14 |
| $u_{\text{var}}$ | Variable operating cost | $/MWh | 4 |
| $k_{\text{gas,gas}}$ | Heat transfer coefficient | kW/(m$^2$ K) | 0.03 |
| $k_{\text{gas,liquid}}$ | Heat transfer coefficient | kW/(m$^2$ K) | 0.06 |
| $k_{\text{gas,evaporating}}$ | Heat transfer coefficient | kW/(m$^2$ K) | 0.06 |
| $k_{\text{liquid,condensing}}$ | Heat transfer coefficient | kW/(m$^2$ K) | 0.35 |
| $T_{\text{cw,in}}$ | Cooling water inlet temperature | K | 298 |
| $T_{\text{cw,out}}$ | Cooling water out temperature | K | 303 |
| $K_1$ | Cost parameter for heat exchangers | - | 4.3247 |
| $K_2$ | Cost parameter for heat exchangers | - | -0.3030 |
| $K_3$ | Cost parameter for heat exchangers | - | 0.1634 |
| $K_4$ | Cost parameter for deaerator | - | 3.5565 |
| $K_5$ | Cost parameter for deaerator | - | 0.3776 |
| $K_6$ | Cost parameter for deaerator | - | 0.0905 |
| $C_1$ | Parameter for pressure factor | - | 0.03881 |
| $C_2$ | Parameter for pressure factor | - | -0.11272 |
| $C_3$ | Parameter for pressure factor | - | 0.08183 |
| $A_{\text{min}}$ | Minimum heat exchanger area | m$^2$ | 10 |
| $V_{\text{min}}$ | Minimum deaerator volume | m$^3$ | 1 |

## References


1. Adjiman, C.S., Androulakis, I.P., Maranas, C.D., Floudas, C.A.: A global optimization method, $\alpha$BB, for process design. Comput. Chem. Eng. **20**, S419–S424 (1996)
2. Adjiman, C.S., Dallwig, S., Floudas, C.A., Neumaier, A.: A global optimization method, $\alpha$BB, for general twice-differentiable constrained NLPs–I. Theoretical advances. Comput. Chem. Eng. **22**(9), 1137–1158 (1998)
3. Ahadi-Oskui, T., Vigerske, S., Nowak, I., Tsatsaronis, G.: Optimizing the design of complex energy conversion systems by branch and cut. Comput. Chem. Eng. **34**(8), 1226–1236 (2010)
4. Ahmetović, E., Grossmann, I.E.: Global superstructure optimization for the design of integrated process water networks. AIChE J. **57**(2), 434–457 (2011)
5. Androulakis, I.P., Maranas, C.D., Floudas, C.A.: $\alpha$BB: A global optimization method for general constrained nonconvex problems. J. Global Optim. **7**(4), 337–363 (1995)
6. Balendra, S., Bogle, I.D.L.: A comparison of flowsheet solving strategies using interval global optimisation methods. Comput. Aid. Chem. Eng. **14**, 23–28 (2003)
7. Balendra, S., Bogle, I.D.L.: Modular global optimisation in chemical engineering. J. Global Optim. **45**(1), 169–185 (2009)
8. Baliban, R.C., Elia, J.A., Misener, R., Floudas, C.A.: Global optimization of a MINLP process synthesis model for thermochemical based conversion of hybrid coal, biomass, and natural gas to liquid fuels. Comput. Chem. Eng. **42**, 64–86 (2012)
9. Bendtsen, C., Stauning, O.: FADBAD++, a flexible C++ package for automatic differentiation. Version 2.1. (2012). URL `http://www.fadbad.com.Accessed18October2016.`
10. Biegler, L.T.: Nonlinear programming: Concepts, algorithms, and applications to chemical processes. MOS-SIAM, Philadelphia (2010)
11. Biegler, L.T., Grossmann, I.E., Westerberg, A.W.: Systematic Methods of Chemical Process Design. Prentice Hall PTR, Upper Saddle River (1997)
12. Biegler, L.T., Hughes, R.R.: Infeasible path optimization with sequential modular simulators. AIChE J. **28**(6), 994–1002 (1982)
13. Bogle, I.D.L., Byrne, R.P.: Global optimisation of chemical process flowsheets. In: G. Dzemyda, V. Saltenis, A. Zilinskas (eds.) Stochastic and Global Optimization, pp. 33–48. Springer Science & Business Media, Dordrecht (2002)
14. Bompadre, A., Mitsos, A.: Convergence rate of McCormick relaxations. J. Global Optim. **52**(1), 1–28 (2012)
15. Bongartz, D., Mitsos, A.: Infeasible path global flowsheet optimization using McCormick relaxations. Computer Aided Chemical Engineering, in press (2017)
16. Bracco, S., Siri, S.: Exergetic optimization of single level combined gas–steam power plants considering different objective functions. Energy **35**(12), 5365–5373 (2010)
17. Byrd, R.H., Nocedal, J., Waltz, R.A.: KNITRO: An integrated package for nonlinear optimization. In: Large-scale nonlinear optimization, pp. 35–59. Springer (2006)
18. Byrne, R., Bogle, I.: Global optimisation of constrained non-convex programs using reformulation and interval analysis. Comput. Chem. Eng. **23**(9), 1341–1350 (1999)
19. Byrne, R.P., Bogle, I.D.L.: Global optimization of modular process flowsheets. Ind. Eng. Chem. Res. **39**(11), 4296–4301 (2000)
20. Chachuat, B.: MC++ (version 2.0): Toolkit for Construction, Manipulation and Bounding of Factorable Functions. (2014). URL `https://omega-icl.bitbucket.io/mcpp/?.Accessed18October2016.`
21. Chen, J.J.J.: Comments on improvements on a replacement for the logarithmic mean. Chem. Eng. Sci. **42**(10), 2488–2489 (1987)
22. Diwekar, U.M., Grossmann, I.E., Rubin, E.S.: An MINLP process synthesizer for a sequential modular simulator. Ind. Eng. Chem. Res. **31**(1), 313–322 (1992)
23. Drud, A.S.: CONOPT – a large-scale GRG code. ORSA J. Comput. **6**(2), 207–216 (1994)
24. Edgar, T.F., Himmelblau, D.M., Lasdon, L.: Optimization of Chemical Processes. McGraw-Hill, New York (2001)
25. Epperly, T.G.W., Pistikopoulos, E.N.: A reduced space branch and bound algorithm for global optimization. J. Global Optim. **11**(3), 287–311 (1997)
26. Falk, J.E., Soland, R.M.: An algorithm for separable nonconvex programming problems. Manag. Sci. **15**(9), 550–569 (1969)
27. GAMS Development Corporation: General Algebraic Modeling System (GAMS) Release 24.6.1. Washington, DC (2016)
28. Gunasekaran, S., Mancini, N.D., Mitsos, A.: Optimal design and operation of membrane-based oxycombustion power plants. Energy **70**, 338–354 (2014)

29. Horst, R., Tuy, H.: Global optimization: Deterministic approaches, 3rd edn. Springer Science & Business Media, Berlin (1996)
30. International Business Machines Corporation: IBM ILOG CPLEX v12.1. Armonk, NY (2009)
31. Johnson, S.G.: The NLopt nonlinear-optimization package. URL `http://ab-initio.mit.edu/nlopt.Accessed18October2016.`
32. Jüdes, M., Tsatsaronis, G.: Design optimization of power plants by considering multiple partial load operation points. In: Proceedings of IMECE2007. ASME International Mechanical Engineering Congress and Exposition. November 11–15, 2007, Seattle, WA, pp. 217–225 (2007)
33. Kehlhofer, R., Hannemann, F., Stirnimann, F., Rukes, B.: Combined-Cycle Gas & Steam Turbine Power Plants, 3rd edn. PennWell Corporation, Tulsa (2009)
34. Khan, K.A., Watson, H.A., Barton, P.I.: Differentiable McCormick relaxations. J. Global Optim. **67**(4), 687–729 (2017)
35. Kocis, G.R., Grossmann, I.E.: Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis. Ind. Eng. Chem. Res. **27**(8), 1407–1421 (1988)
36. Kraft, D.: A software package for sequential quadratic programming. Tech. Rep. DFVLR-FB 88-28, Institut für Dynamik der Flugsysteme, Oberpfaffenhofen (1988)
37. Kraft, D.: Algorithm 733: TOMP–Fortran modules for optimal control calculations. ACM T. Math. Software **20**(3), 262–281 (1994)
38. Locatelli, M., Schoen, F.: Global optimization: theory, algorithms, and applications, vol. 15. MOS-SIAM, Philadelphia (2013)
39. Manassaldi, J.I., Arias, A.M., Scenna, N.J., Mussati, M.C., Mussati, S.F.: A discrete and continuous mathematical model for the optimal synthesis and design of dual pressure heat recovery steam generators coupled to two steam turbines. Energy **103**, 807–823 (2016)
40. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I - convex underestimating problems. Math. Prog. **10**, 147–175 (1976)
41. Misener, R., Floudas, C.A.: ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. J. Global Optim. **59**, 503–526 (2014)
42. Mistry, M., Misener, R.: Optimising heat exchanger network synthesis using convexity properties of the logarithmic mean temperature difference. Comput. Chem. Eng. **94**, 1–17 (2016)
43. Mitsos, A., Chachuat, B., Barton, P.I.: McCormick-based relaxations of algorithms. SIAM J. Optim. **20**(2), 573–601 (2009)
44. Najman, J., Mitsos, A.: Convergence analysis of multivariate McCormick relaxations. J. Global Optim. **66**, 597–628 (2016)
45. Najman, J., Mitsos, A.: Convergence order of McCormick relaxations of LMTD function in heat exchanger networks. In: 26th European Symposium on Computer Aided Process Engineering, pp. 1605–1610 (2016)
46. Quesada, I., Grossmann, I.E.: Global optimization algorithm for heat exchanger networks. Ind. Eng. Chem. Res. **32**(3), 487–499 (1993)
47. Reneaume, J.M.F., Koehret, B.M., Joulia, X.L.: Optimal process synthesis in a modular simulator environment: New formulation of the mixed-integer nonlinear programming problem. Ind. Eng. Chem. Res. **34**(12), 4378–4394 (1995)
48. Ryoo, H.S., Sahinidis, N.V.: Global optimization of nonconvex NLPs and MINLPs with applications in process design. Comput. Chem. Eng. **19**(5), 551–566 (1995)
49. Ryoo, H.S., Sahinidis, N.V.: A branch-and-reduce approach to global optimization. J. Global Optim. **8**(2), 107–138 (1996)
50. Scott, J.K., Stuber, M.D., Barton, P.I.: Generalized McCormick relaxations. J. Global Optim. **51**, 569–606 (2011)
51. Silveira, J.L., Tuna, C.E.: Thermoeconomic analysis method for optimization of combined heat and power systems. Part I. Prog. Energ. Combust. **29**(6), 479–485 (2003)
52. Smith, E.M.B., Pantelides, C.C.: Global optimisation of nonconvex MINLPs. Comput. Chem. Eng. **21**, S791–S796 (1997)
53. Smith, E.M.B., Pantelides, C.C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex minlps. Comput. Chem. Eng. **23**(4), 457–478 (1999)
54. Stuber, M.D., Scott, J.K., Barton, P.I.: Convex and concave relaxations of implicit functions. Optim. Method. Softw. **30**, 424–460 (2015)
55. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Math. Prog. **99**(3), 563–591 (2004)
56. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Math. Prog. **103**(2), 225–249 (2005)
57. Tsoukalas, A., Mitsos, A.: Multivariate McCormick relaxations. J. Global Optim. **59**, 633–662 (2014)

58. Turton, R., Bailie, R.C., Whiting, W.B.: Analysis, Synthesis and Design of Chemical Processes, 4th edn. Prentice Hall PTR, Upper Saddle River (2012)
59. U.S. Energy Information Administration: United States Natural Gas Industrial Price. URL `https://www.eia.gov/dnav/ng/hist/n3035us3m.htm.Accessed6September2016.`
60. Valdés, M., Duran, M.D., Rovira, A.: Thermoeconomic optimization of combined cycle gas turbine power plants using genetic algorithms. Appl. Therm. Eng. **23**(17), 2169–2182 (2003)
61. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Prog. **106**(1), 25–57 (2006)
62. Wechsung, A., Barton, P.I.: Global optimization of bounded factorable functions with discontinuities. J. Global Optim. **58**(1), 1–30 (2014)
63. Wechsung, A., Scott, J.K., Watson, H.A., Barton, P.I.: Reverse propagation of McCormick relaxations. J. Global Optim. **63**(1), 1–36 (2015)
64. Zamora, J.M., Grossmann, I.E.: Continuous global optimization of structured process systems models. Comput. Chem. Eng. **22**(12), 1749–1770 (1998)
65. Zebian, H., Mitsos, A.: A double-pinch criterion for regenerative Rankine cycles. Energy **40**(1), 258–270 (2012)