Simultaneous Deterministic Global Flowsheet Optimization and Heat Integration: Comparison of Formulations[1]

Steffen Fahr[a2], Alexander Mitsos[bac], Dominik Bongartz[a*]

[a] Process Systems Engineering (AVT.SVT), RWTH Aachen University, Forckenbeckstraße 51, 52074 Aachen, Germany

[b] JARA-CSD, 52056 Aachen, Germany

[c] Institute of Energy and Climate Research: Energy Systems Engineering (IEK-10), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany.

Abstract: Heat integration can be considered in flowsheet optimization by including the minimum utility demand from pinch analysis in the objective and constraints. This often results in better process performance than conducting these steps separately. However, it comes with increased computational cost, especially for global optimization. This cost depends both on the problem formulation and on the solver. In this work, we compare several existing and new smooth, nonsmooth, and mixed-integer formulations. Furthermore, we test different choices of optimization variables and constraints reaching from full- to reduced-space formulations. In the reduced-space formulations, heat integration can be included with few, one, or even zero additional variables beyond the pure flowsheet optimization problem. For the considered case studies, this can significantly reduce the solution time of various global solvers, in particular for our open-source solver MAiNGO. Depending on the case study and solver, either nonsmooth or mixed-integer formulations are the fastest to solve.

1 Introduction

Optimal heat integration has been an active field of research since decades. In the late 1970s, Linnhoff and Flower [21] introduced pinch analysis and the feasibility table to determine the hot and cold utility demand based on information about the process streams to be heated and cooled for a given minimum temperature approach allowed in each heat exchanger. The temperature-enthalpy-diagram (TH-diagram) represents an equivalent graphical method [16]. Papoulias and Grossmann [27] extended these tools by a mathematical program based on an analogy to the transshipment model. These three tools are all applicable a posteriori, i.e., for a flowsheet with fixed stream conditions.

[1]This is the Authors' Accepted Manuscript of the following article: S. Fahr, A. Mitsos, D. Bongartz, Simultaneous Deterministic Global Flowsheet Optimization and Heat Integration: Comparison of Formulations, Computers & Chemical Engineering 162, pp. 107790, 2022, which has been published in final form at https://doi.org/10.1016/j.compchemeng.2022.107790.

[2]Present address: Institute of Plant and Process Technology, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany

Later, Duran and Grossmann [12] introduced a formulation that embeds pinch analysis in the flowsheet optimization, thus allowing to consider the remaining utility demands after heat integration in the optimization problem. This approach is generally referred to as "simultaneous flowsheet optimization and heat integration" [12, 15, 26]. More recently, optimization of flowsheets with a Heat Exchanger Network (HEN) superstructure embedded has been considered [10, 17, 22]. This approach allows to rigorously take into account the trade-off between capital cost for heat exchanger area and utility cost. However, it is computationally expensive and to date only possible for simple process models and relatively restrictive superstructures. Therefore, this work focuses on the approach suggested by Duran and Grossmann [12].

Solving the simultaneous flowsheet optimization and heat integration problem is challenging, as additional variables and nonsmooth nonconvex constraints are introduced as compared to flowsheet optimization without simultaneous heat integration. Originally, this method had been applied in conjunction with gradient-based local solvers along with a smooth approximation of the nonsmooth functions involved [12]. Some recent works still rely on local solvers with such smoothing approximations [30, 35]. Vikse et al. [30] used a multistart approach to help raise the confidence that the global optimum is found. Other works used genetic algorithms to solve comparable problems [14, 36]. Both approaches, however, cannot guarantee (approximate) global optimality in finite time. Moreover, according to Zhuang et al. [36], the utilized genetic algorithm still required a significant solution time and a more efficient strategy should be developed.

Simultaneous flowsheet optimization and heat integration problems have been solved using deterministic global optimization, as well. Niziolek et al. [26] solved a large optimization problem including heat integration with 10,000's of constraints and variables. The constraints included linear as well as some multilinear (i.e., bilinear, trilinear and quadrilinear) equations. However, the computation was stopped after 100 CPU hours with a large optimality gap remaining. Yang and Grossmann [34] optimized a methanol process using the commercial branch and bound (B&B) solver BARON. Additionally, a more complex bioethanol process was optimized, but various process conditions were fixed a priori to reduce the complexity of the problem. Both works successfully apply global optimization to the simultaneous flowsheet optimization and heat integration problem. Nonetheless, a more efficient formulation or optimization strategy would have allowed to include more degrees of freedom (DOFs), more complex process models, or to achieve a smaller optimality gap. Cassanello et al. [8] tested two formulations of pinch analysis that are applicable in simultaneous flowsheet optimization and heat integration in an effort to improve computational performance of the global optimization. However, three out of their four examples did not have a process model embedded (i.e., no simultaneous flowsheet optimization).

In this work, we aim to accelerate the convergence of the simultaneous deterministic global flowsheet optimization and heat integration problem by testing and comparing new and existing problem formulations. In particular, we exploit the fact that our open-source optimizer MAiNGO [1] can natively handle the two-argument max function [29] that appears in the original pinch formulation [12], thus eliminating the need for smoothing [11, 12] or mixed-integer [15] reformulations. Furthermore, we consider the use of reduced-space optimization formulations [7, 24], which we have shown to be computationally advantageous for global flowsheet optimization [2, 3]. In these formulations, optimization variables and corresponding equality constraints are eliminated by reformulating the constraints such that the eliminated variables can be computed as factorable functions of the remaining variables. In flowsheet optimization, they are a hybrid between equation-oriented and sequential-modular methods [4]. In this work, we show that in reduced-space formulations, heat integration can be included with only few, one, or even no additional variables beyond those of the pure flowsheet optimization problem, depending on the formulation of the pinch analysis. We investigate the effect of these formulations on the solution times of various deterministic global solvers.

The remainder of this article is structured as follows: In Section 2, we briefly introduce the original problem formulations from [12] as well as alternatives focusing on the nonsmooth terms included in the original formulation. In Section 3, four case studies are introduced on which the different formulations are tested. The results of these studies in terms of computational performance achieved with MAiNGO are presented in Section 4.1 and compared to different commercial and open source solvers in Section 4.2.

2 Formulation of the Pinch Problem

In this section, we first summarize the original formulation of the simultaneous flowsheet optimization and heat integration problem. Subsequently, we present continuous and mixed-integer reformulations of that formulation which will later be assessed in terms of their effect on computational performance.

2.1 Original Formulation

Duran and Grossmann [12] proposed a mathematical formulation that allows to embed the pinch calculations in a flowsheet optimization problem. This formulation assumes constant heat capacities over the temperature range of each heat stream as well as a single minimum temperature approach $\Delta T_{\min}$ that must hold for all streams. Previous works have shown that nonlinear enthalpy models may be embedded directly into pinch analysis [37], relaxing the assumption of constant heat capacities. However, this is outside the scope of this work.

We now summarize the formulation. For the full original formulation please refer to the supplementary information (SI) or to Duran and Grossmann [12]. For the pinch calculation, a set of hot streams H to be cooled and cold streams C to be heated are defined. Moreover, the set of pinch candidates P containing all possible (hot-end) pinch temperatures T^p is defined as $P = \{T_i^{\text{in}} \ \forall \ i \in H,\ T_j^{\text{in}} + \Delta T_{\min} \ \forall \ j \in C\}$. Therein, $T_s^{\text{in}},\ s \in H \cup C$ denote the temperatures at which the process streams enter the HEN. The heat transferred to the HEN from hot streams above (i.e., at a higher temperature than) each pinch candidate is then calculated as

$$Q_{\text{in},A}^p = \sum_{i \in H} \text{FC}_i(\max(T_i^{\text{in}} - T^p, 0) - \max(T_i^{\text{out}} - T^p, 0)) \ \ \forall \ p \in P, \tag{1}$$

and the heat transferred from the HEN to cold streams above each pinch candidate as

$$\begin{aligned} Q_{\text{out},A}^p = \sum_{j \in C} \text{FC}_j(&\max(T_j^{\text{out}} - (T^p - \Delta T_{\min}), 0) \\ &- \max(T_j^{\text{in}} - (T^p - \Delta T_{\min}), 0)) \ \ \forall \ p \in P. \end{aligned} \tag{2}$$

Therein, the variables $\text{FC}_s,\ s \in H \cup C$ describe heat capacity flowrates. The inequality constraints

$$Q_{\text{out},A}^p - Q_{\text{in},A}^p - Q_H \leqslant 0 \ \ \forall \ p \in P, \tag{3}$$

where Q_H is the net demand for hot utility, guarantee that the minimum temperature approach is not violated at any of the pinch candidates. The given formulation is also called *explicit* because Q_H is explicitly included as an optimization variable in the problem [19]. For typical objectives, Q_H will be driven down towards the minimum possible hot utility that is limited by one of the inequality constraints given in Eq. (3). Duran and Grossmann [12] also proposed an alternative formulation that was later referred to as the *implicit* formulation [19], where Eq. (3) is replaced with

$$Q_H = \max_{p \in P} (Q_{\text{out},A}^p - Q_{\text{in},A}^p). \tag{4}$$

This eliminates one variable at the cost of introducing the multi-argument max function. Many solvers do not natively support this function, but the n-argument max function can be reformulated as a composition of $n-1$ two-argument max functions.

Both the explicit and implicit formulation use the same formulation to obtain the heat flow rate that individual hot and cold streams contribute above a given pinch candidate, which is based on the max function, Eqs. (1) and (2). These equations introduce a large number of nonsmooth nonconvex terms into the overall problem. The nonsmooth nature of these terms is problematic when using gradient-based solvers; moreover, many global optimization solvers or even methods inherently require twice-continuously differentiable functions. This is not the case in MAiNGO [1] as well as the MC++ library [9] it uses to compute relaxations. On the other hand, the introduced nonconvexities potentially lead to weak relaxations in the context of global optimization. The core of these terms is discussed in more detail in the remainder of this work and will be referred to as the “pinch expression”, defined in its original form as

$$\Delta T \coloneqq \max(T_H - T^p, 0) - \max(T_C - T^p, 0). \tag{5}$$

T_H and T_C refer to the respective hot and cold end stream temperatures, i.e., for hot streams, $T_H = T_i^{\text{in}}$ and $T_C = T_i^{\text{out}}$, and vice versa for cold streams. ΔT denotes the temperature difference that is multiplied with the heat capacity flowrate in Eqs. (1) and (2) to calculate the heat transferred above the pinch candidate T^p.

To use nonsmooth functions like the pinch expression in global optimization, they have to be treated accordingly in the lower and upper bounding problems of the B&B algorithm. For lower bounding, relaxations of the nonsmooth functions are required. In MAiNGO, this is the case since relaxations of, e.g., the two-argument max function [29] are available through the MC++ library [9]. For upper bounding, the challenge is that no derivatives exist, while they are required for the commonly used gradient-based local solvers. In MAiNGO, we heuristically use one of the one-sided derivatives of these piecewise differentiable functions, e.g., for the function $\max(x, y)$, the derivative at points where $x = y$ is always taken to be 1 with respect to x and 0 with respect to y. Our numerical experiments showed no practical issues. Watson et al. [31] described a rigorous method relying on generalized derivatives to handle nonsmooth functions in the simultaneous flowsheet optimization and heat integration problem. However, implementing such a strategy is not trivial and we consider it beyond the scope of our work. Rigorous handling of the derivatives promises more reliable performance of local solvers. This is important when relying exclusively on these solvers, as Watson et al. [31] do. However, in the context of a global optimization strategy, such as ours, occasional failure of the local solvers used for upper bounding may be more acceptable, and in particular it cannot result in convergence to suboptimal solutions.

2.2 Nonsmooth Reformulations

The previously defined pinch expression can be formulated in other ways that also use nonsmooth functions such as the max function, but potentially enable tighter relaxations that accelerate global optimization. For instance, as an alternative to Eq. (5), the max function can be applied in a slightly different way, while yielding the same function value:

$$\Delta T = \max(T_H, T^p) - \max(T_C, T^p). \tag{6}$$

We compared the McCormick relaxations of the alternative pinch expression in Eq. (6) to the one of the original pinch expression given in Eq. (5) using the MC++ library [9], and found the relaxations of the expression in Eq. (6) to be tighter than those of the one in Eq. (5) in some intervals and weaker in others.

Having two equivalent formulations with evidently different relaxations motivates to implement an intrinsic function that uses the respective tighter relaxation of these two at any given point, which we call the *pinch function.* Furthermore, based on the monotonicity properties of this pinch function, we can compute exact interval bounds for it. In contrast, when using one of the previously mentioned expressions from Eqs. (5) and (6), these bounds are calculated using natural interval extensions, which can be substantially weaker. The pinch function as a mathematical function can be defined as

$$\text{pinch}\colon \mathbb{R}^3 \to \mathbb{R},\ \text{pinch}\,(T_H, T_C, T^p) \to \max(T_H - T^p, 0) - \max(T_C - T^p, 0). \tag{7}$$

Note that in the following, when we refer to the pinch function, we refer to this intrinsic function. In contrast, we use the term "pinch expression" as a generic term for any expression calculating ΔT, including but not limited to the expressions previously given in Eqs. (5), (6), and the pinch function.

Another alternative formulation can be envisioned based on the mid function, which selects the median of its three arguments. We implemented the mid function of two variables and a constant as an intrinsic function in MC++ [9]. The necessary convex and concave envelopes of the mid function was derived in analogy to those of the max function [29]. Using the mid function, we replace Eq. (5) by

$$\Delta T = \text{mid}(T_H - T^p, T_H - T_C, 0). \tag{8}$$

Note that the resulting term is equal to the ones using the max function only if $T_H \geqslant T_C$. This should always be the case at the solution of the problem by the definition of hot and cold streams. However, during the optimization this is not necessarily always the case, and therefore we cannot combine the obtained relaxations with the previous ones, as we did in the pinch function. When defining the problem as either a DNLP or an MINLP, GAMS accepts the max function when using LINDOGLOBAL or SCIP. However, BARON and ANTIGONE do not accept the max function directly, whereas they can handle the absolute value function. For those solvers, we reformulate the max function exploiting that

$$\max(x, 0) = \frac{1}{2}(x + |x|). \tag{9}$$

Using the above reformulation and calculating the relaxations of the resulting expression using the MC++ library results in weaker relaxation than the envelope of the max function [29].

2.3 Smooth Formulations

Typical gradient-based solvers cannot handle nonsmooth functions. A smoothed version of the max function has been used in previous works in order to avoid this issue [11, 12, 34]. One possible smoothed max function is given by

$$\widetilde{\max}(x, 0) = \frac{1}{2}(x + \sqrt{x^2 + \epsilon}) \tag{10}$$

with a smoothing parameter $\epsilon > 0$ [11, 34]. It is not immediately clear if this also improves the solution time of the global optimization. While it likely improves the performance of the local gradient-based solvers used for upper bounding, it may weaken the relaxations. Also note that in contrast to the previously presented reformulations, using this smoothed max function does not result in an exact reformulation of the original problem but in an approximation.

An additional reformulation that avoids nonsmooth functions altogether was proposed by Cassanello et al. [8]. The formulation requires $3|P| \times (|C| + |H|)$ additional continuous variables compared to the original formulation presented by Duran and Grossmann [12]. The formulation is referred to as NLP (nonlinear programming) formulation in [8] for its lack of binary variables as opposed to the proposed alternative in that work. For the full formulation, please refer to the formulation given in Section 2.1 of [8]. The second formulation in [8], named “Multi-M”, requires preprocessing based on bounds of the temperature variables. This is not possible with our formulation in which some of the temperatures are calculated variables that we do not have explicit bounds on. It is in theory possible to derive bounds by bound propagation, but this is not straightforward to implement since it requires a dedicated preprocessing step to construct the problem formulation. Therefore, we do not consider this strategy in this work.

2.4 Mixed-Integer Reformulations

Previous works have employed an exact reformulation of the original formulation of the pinch problem as presented by Duran and Grossmann [12] based on a linear (big-M) formulation [15, 33]. For the full formulation, please refer to [15], [33] or our SI. The formulation adds $|P| \times |S|$ continuous variables with the set of all streams $S = H_{\text{noniso}} \cup C_{\text{noniso}} \cup H_{\text{iso}} \cup C_{\text{iso}}$. The sets labeled with the indices “noniso” and “iso”contain the nonisothermal and isothermal streams, respectively. We define isothermal streams as those streams from or to which heat is transferred at a single temperature (e.g., evaporating or condensing single-component streams), rather than a temperature range. If each stream contributes one new pinch candidate (worst case), the formulation adds $|S|^2$ continuous variables. Additionally, the formulation adds $|P| \times (3|C_{\text{noniso}}| + 3|H_{\text{noniso}}| + |H_{\text{iso}}| + |C_{\text{iso}}|)$ binary variables.

As an alternative, we derive another mixed-integer formulation by reformulating the max function using a bilinear mixed-integer formulation. In contrast to the approach of [15], [33], this introduces additional nonlinearities, but only $|P| \times (2|C_{\text{noniso}}| + 2|H_{\text{noniso}}| + |H_{\text{iso}}| + |C_{\text{iso}}|)$ binary and no continuous variables. In the context of superstructure optimization, we recently showed that such alternative formulations that add nonconvex terms but result in smaller problems can be beneficial if they lead to small problems and the problem is already nonconvex because of the model equations [6]. The constraints for calculating $Q^p_{\text{in},A}$ in this formulation are

$$
\begin{aligned}
Q^p_{\text{in},A} = & \sum_{i \in H_{\text{noniso}}} \text{FC}_i((z_{i,p,1} T_i^{\text{in}} + (1 - z_{i,p,1}) T_i^{\text{out}}) \\
& - (z_{i,p,2} T_i^{\text{out}} + (1 - z_{i,p,2}) T^p)) \\
& + \sum_{i \in H_{\text{iso}}} Q_{H,\text{iso},i} z_{\text{iso},i,p} \ \forall\, p \in P \\
& 0 \geqslant T_i^{\text{in}} - T^p - M z_{i,p,1} \ \forall\, p \in P,\ i \in H_{\text{noniso}} \\
& 0 \geqslant T^p - T_i^{\text{in}} - M(1 - z_{i,p,1}) \ \forall\, p \in P,\ i \in H_{\text{noniso}} \\
& 0 \geqslant T_i^{\text{out}} - T^p - M z_{i,p,2} \ \forall\, p \in P,\ i \in H_{\text{noniso}} \\
& 0 \geqslant T^p - T_i^{\text{out}} - M(1 - z_{i,p,2}) \ \forall\, p \in P,\ i \in H_{\text{noniso}} \\
& 0 \geqslant T_i - (T^p + \epsilon) + -M z_{i,p} \ \forall\, p \in P,\ i \in H_{\text{iso}} \\
& 0 \geqslant (T^p + \epsilon) - T_i - M(1 - z_{i,p}) \ \forall\, p \in P,\ i \in H_{\text{iso}}
\end{aligned} \tag{11}
$$

The term calculating $Q^p_{\text{out},A}$ is derived analogously. The variables $z_{i,p,1}$ and $z_{i,p,2},\ i \in H_{\text{noniso}}$ are binary variables corresponding to the non-isothermal streams. They indicate which of the arguments in the original formulation Eqs. (1) and (2) corresponds to the function value of the respective max function: $z_{i,p,1} = 1$ means that $T_i^{\text{in}} \geqslant T^p$ and $z_{i,p,2} = 1$ means that $T_i^{\text{out}} \geqslant T^p$. M is a parameter that is chosen larger than the largest possible temperature difference in the process. For isothermal hot streams, the binary variables $z_{i,p},\ i \in H_{\text{iso}}$ are introduced that indicate if the isothermal stream lies above ($z_{i,p} = 1$) or below ($z_{i,p} = 0$) the pinch temperature. A small parameter $\epsilon > 0$ is necessary to ensure pinch candidates lie on the correct side of each stream, e.g., at the vapor side of a condensing hot stream [15, 33].

2.5 Isothermal Streams

In previous works, isothermal streams were routinely treated by introducing a dummy temperature change and then treating them as non-isothermal streams [11, 12]. Duran and Grossmann

[12] introduced a fixed dummy temperature change of 1 K, while Dowling and Biegler [11] introduced the parameter α to represent the dummy temperature change. For this work, we adopt the parameter α from Dowling and Biegler [11] in a way that assumes hot isothermal streams to exit at a lower temperature and the cold ones to exit at a higher temperature as suggested by Duran and Grossmann [12]. This is more conservative than adding α to the hot end of both hot and cold isothermal streams as proposed by Dowling and Biegler [11]. With dummy temperature change, all of the reformulations listed in Sections 2.2 and 2.3 can also be used for isothermal streams. As an alternative, the mixed-integer formulations in Section 2.3 contain constraints specifically designed for isothermal streams. Using these constraints results in zero dummy temperature change. Finally, we consider an additional formulation based on the step function step(x), which returns 1 if $x \geqslant 0$, and otherwise returns zero. This formulation also corresponds to zero dummy temperature change. The step function and its relaxations presented by Wechsung and Barton [32] are available in MC++. The corresponding constraints calculating $Q^p_{\mathrm{in},A}$ and $Q^p_{\mathrm{out},A}$ read as follows:

$$\begin{aligned} Q^p_{\mathrm{in},A} = & \sum_{i \in H_{\mathrm{noniso}}} \mathrm{FC}_i (\max(T_i^{\mathrm{in}} - T^p, 0) \\ & - \max(T_i^{\mathrm{out}} - T^p, 0)) \\ & + \sum_{i \in H_{\mathrm{iso}}} Q_{H,\mathrm{iso},i} \ (1 - \mathrm{step}(T^p - T_i)) \ \forall \ p \in P \\ Q^p_{\mathrm{out},A} = & \sum_{j \in C_{\mathrm{noniso}}} \mathrm{FC}_j (\max(T_j^{\mathrm{out}} - (T^p - \Delta T_{\min}), 0) \\ & - \max(T_j^{in} - (T^p - \Delta T_{\min}), 0)) \\ & + \sum_{j \in C_{\mathrm{iso}}} Q_{C,\mathrm{iso},j} \ \mathrm{step}(T_j - (T^p - \Delta T_{\min})) \ \forall \ p \in P \end{aligned} \tag{12}$$

The motivation for using alternative formulations to those using dummy temperature change lies primarily in accelerating the convergence of the optimization rather than improving accuracy, which is at most a welcome side effect. A dummy temperature change of $1K$ only rarely changes the degree to which two streams can exchange heat and represents a reasonably accurate assumption in the context of pinch analysis.

2.6 Choice of Optimization Variables

In addition to the formulation used for the pinch expression, we consider different choices of optimization variables. Typically, when implementing a problem, e.g., in GAMS, all model variables are implemented as optimization variables and all model equations are implemented as equality constraints. We call this type of formulation full-space (FS) in the following.

As an alternative, we consider *factorable reduced space formulations* (RS) [2, 7, 24]. In these formulations, variables for which explicit relationships exist as equality constraints in the FS formulation are eliminated by replacing each occurrence of the variables with these explicit relationships [24]. RS formulations lead to a small number of optimization variables and constraints but at the same time usually result in highly nonlinear and complex objective and constraints. The variables in RS only include the DOFs of the problem and those variables for which no explicit relationship exists. The concept of RS formulations and its resemblance to sequential-modular formulations of flowsheet optimization is comprehensively discussed in Chapter 3 of [2]. Previous work showed that using RS can lead to a significant speed-up, e.g., in the context of flowsheet [3, 4] and superstructure optimization [6].

Both approaches can be applied to the simultaneous heat integration and flowsheet optimization problem. In an FS formulation, all equations occurring in the chosen formulation from Sections 2.1–2.5 become equality constraints (along with all equations of the flowsheet model), and all variables therein become optimization variables. The simultaneous problem is thus significantly larger than the pure flowsheet optimization problem. In an RS formulation, the equations of the chosen formulation from Sections 2.1–2.5 are instead used to eliminate variables where possible. For example, Eq. (1) can be used to compute the $Q^p_{\text{in},A}$ as a factorable function of the temperatures and the heat capacity flowrates. Therefore, the $Q^p_{\text{in},A}$ do not become optimization variables, and Eq. (1) does not become an equality constraint, since this part of the model can simply be evaluated given the other variables. Therefore, only a few additional variables and (in-)equality constraints are required for the pinch analysis. For example, the *explicit formulation* in Section 2.1 requires only Q_H as additional variable and Eq. (3) as additional inequalities. In contrast, the *implicit formulation* in the same section does not require any additional variables beyond those of the flowsheeting problem, since now Q_H is computed from Eq. (4).

Naturally, there is a spectrum reaching from the minimum possible number of variables, RS, to FS. One additional option will be considered for the simultaneous flowsheet optimization and heat integration problem, called reduced-space + temperatures (RS+T). This formulation corresponds to RS, but with all temperatures occurring in pinch analysis added as optimization variables along with the corresponding equality constraints. Adding optimization variables and the corresponding constraints leads to higher dimensionality but may help provide tighter relaxations [4].

3 Case Studies

Four case studies from the literature are chosen for comparing the computational performance of different formulations of the simultaneous optimization and heat integration problem. Although for some of the cases the optimal heat exchanger matching is rather obvious and known a priori, we use pinch analysis for the sake of analyzing the computational performance.

Table 1: Overview of key indicators of problem size for case studies. *OV: number of optimization variables **Here, we define DOFs as the number of variables minus the number of equality constraints. ***temperatures that occur in pinch analysis and are not declared optimization variables or constants, but rather computed as a function of other variables (see Section 2.6) †RS and RS+T are identical for this case.

Name	OV*	DOFs**	Hot streams (isothermal)	Cold streams (isothermal)	Hot/cold utilities	Non-OV temp.***
Regenerative Rankine RS	7	6	2 (1)	3	1 / 1	3
Regenerative Rankine RS+T	10	6	2 (1)	3 (1)	1 / 1	0
Two-Pressure Rankine RS+T	16	9	2 (1)	6 (2)	1 / 1	0
Methanol production†	68	6	6 (2)	3 (1)	1 / 3	0
LNG RS	7	6	4 (0)	7 (0)	1 / 1	3
LNG RS+T	10	6	4 (0)	7 (0)	1 / 1	0

An overview of the size of the four processes used for case studies is given in Table 1. The number of optimization variables and DOFs include the additional DOFs introduced with the pinch analysis in its original formulation as given by Duran and Grossmann [12]. The FS formulations result in larger numbers of optimization variables, 91 for the Regenerative Rankine and 49 for the LNG case study. These are solved in the solver comparison case study for the Regenerative Rankine and the LNG case, but not in the formulation comparison which focuses on MAiNGO (see Section 4.1). The composite curves at the optimal solution of the four processes can be found in the SI. The C++ and the GAMS models of these processes used in this work are available via our website[1].

3.1 Regenerative Rankine Cycle

The first case study is the Regenerative Rankine Cycle presented in [3]. The optimization aims at maximizing the net power output of a Rankine cycle driven by a fixed stream of hot gas turbine exhaust. To transform the original model to the one including pinch analysis, all heat transfers between flue gas and Rankine cycle are decoupled. The original model with matched heat streams and without pinch analysis can be used as a reference case to assess the increase in computational time caused by including pinch analysis. Although the process is not expected to require hot utility, we still introduce it, as setting it to zero requires physical insight and is not always possible. In

[1] The C++ and GAMS implementations of the case studies are available at `http://permalink.avt.rwth-aachen.de/?id=198813`.

the objective function, it is priced as electric heating, so in the optimal solution, the process does not use any hot utility but runs on the provided heat from the flue gas. Additionally, a small penalty for cold utility use is added to the objective function to avoid cooling down the flue gas stream with cold utility, which would otherwise be indifferent to the objective. The composite curves and operation points are identical for the simultaneous optimization and the original model without pinch analysis, which indicates that in the original model, the heat streams were matched optimally.

3.2 Two-Pressure Rankine Cycle

As a second case study, the Two-Pressure Rankine Cycle from the same publication is used [3]. The additional turbine stage introduces three additional cold streams and three DOFs compared to the Regenerative Rankine Cycle. The composite curves of the process are identical for the original model without pinch analysis and the simultaneous optimization and heat integration model, once again indicating that the heat exchanger matching in the original model was chosen optimally.

3.3 Methanol Production Process

As the third case study used herein, we consider a methanol production process. The process is based on [4], but with an added distillation column for separating methanol and water and a purge combustor, which are both adopted from [5]. Moreover, we assume that the reactor heat is available at the reactor effluent temperature (i.e., the heat stream is isothermal), and that this stream may be integrated like any other stream.

For the methanol process, the exergetic efficiency of the process is chosen as objective function. Medium and low pressure export steam are added to the process models in the form of additional cold utilities, which adds two isothermal cold streams, each associated with one DOF. In contrast to the two previously introduced models, the optimal heat exchanger matching is not known a priori. Therefore, there is no heat integrated process to be used as a reference case. Instead, as a reference case, the process is optimized without heat integration and all heat streams contribute to the exergetic efficiency with their respective exergy, analogous to the approach taken in [5]. By coincidence, all temperatures occurring in pinch analysis are in the set of the minimum required optimization variables in the process model, i.e., they are optimization variables in RS. Accordingly, RS and RS+T are identical.

3.4 Liquefied Natural Gas (LNG) Process

As the last case study, we consider the natural gas liquefaction (LNG) process presented by Wechsung et al. [33]. The process relies on liquid nitrogen and liquid CO_2 and a combination of compressors and turbines to liquefy an incoming natural gas stream. For this purpose, it uses multistream heat exchangers that are modeled using pinch analysis, since no clear heat exchanger matching exists. Therefore, no reference case can be provided that does not use pinch analysis. The optimization aims at minimizing the liquid nitrogen consumption. We consider the last scenario of Wechsung et al. [33] in which the process operates as a standalone unit, i.e., no external electricity supply and no hot or cold utility are available.

4 Results

In this section, we first discuss the solution times achieved with different formulations using MAiNGO. Next, we compare the performance with other solvers. Throughout the studies conducted, we consider RS, RS+T and FS.

4.1 Formulation Comparison

The problems for this study were implemented using the C++ interface of MAiNGO. All problems were solved using the solver SLSQP [18] instead of the default IPOPT in the multistart during preprocessing; this avoided long preprocessing times which would distort the results. Otherwise, we used default settings and a time limit of 1,000 s. All results presented in this section were obtained using an Intel Core i7-10510U CPU at 2.3 GHz.

The results achieved with some formulations have been excluded from this section, as, with MAiNGO, they frequently did not converge within the time limit. These include FS formulations as well as full mixed-integer reformulations, i.e., those eliminating all max functions by using integer variables. Both of these strategies introduce a large number of additional variables and constraints, i.e., lead to high dimensionality, which was found to lead to poor performance when using MAiNGO. Please refer to Section 4.2 and the SI for some of the according results.

Figure 1 shows the effect of the problem formulation and the choice of optimization variables on the CPU time required to solve the Regenerative Rankine Cycle study. The results for the LNG case study are qualitatively similar and can be found in the SI. All isothermal streams were modeled with a dummy temperature change of $\alpha = 1\ K$. In the figure legend, "max" refers to using the original pinch expression Eq. (5), whereas "alt max" refers to using the alternative pinch expression based on the max function Eq. (6). "max implicit" refers to using the max function and the implicit formulation and "smooth" uses the original formulation with the max function replaced by Eq. (10) with the given ϵ. Finally, "max-mid" refers to using the pinch expression based on the max function Eqs. (5) for non-isothermal streams and the one based on the mid function Eq. (8) for isothermal streams with dummy temperature change.

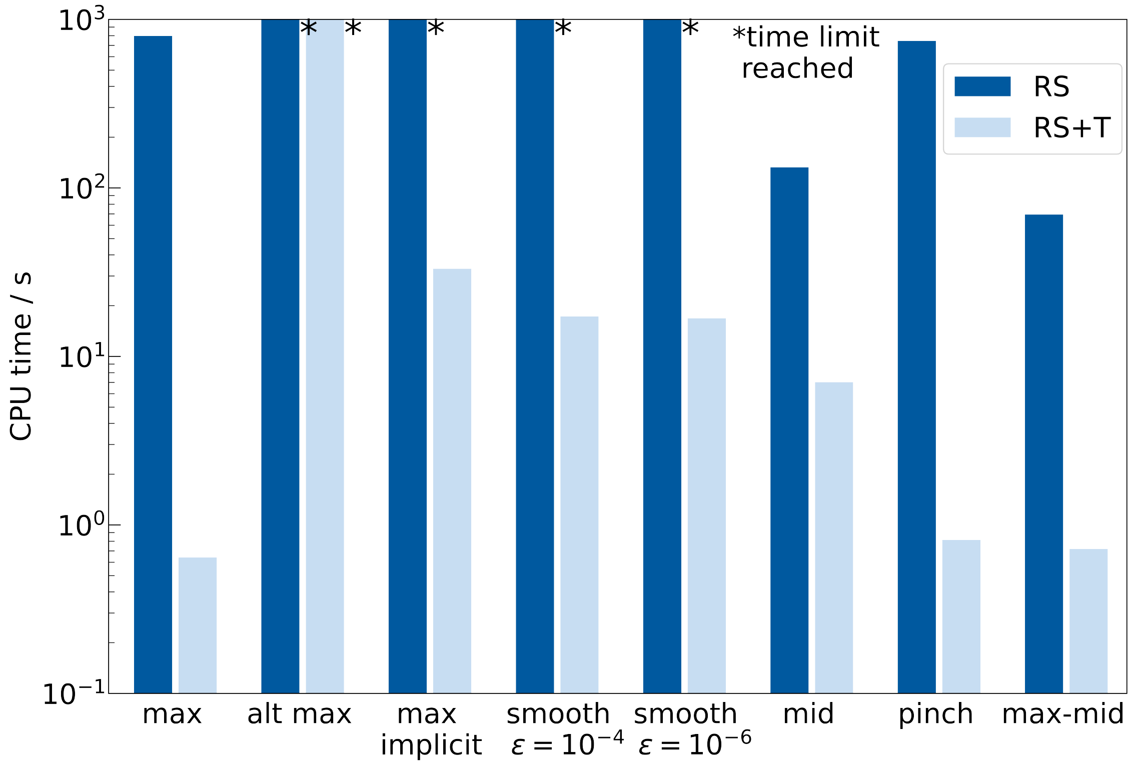

Figure 1: Effect of problem formulation and choice of optimization variables on CPU time for the Regenerative Rankine Cycle case study with MAiNGO. Including the temperatures as optimization variables along with suitable constraints (RS+T) is advantageous compared to RS.

The formulations using RS+T always performed better than the respective equivalent RS formulations. This is likely due to the fact that the temperatures used in the pinch expression are reoccurring nonlinear factors; it is well-known that this leads to potentially weak McCormick relaxations, see, e.g., Tsoukalas and Mitsos [29]. While MAiNGO has a feature for heuristically adding some variables to reduced-space formulations to achieve tighter relaxations [25], these were not yet sufficient to achieve the same benefit obtained by manually adding the temperatures as optimization variables.

The standard formulation using the real (nonsmooth) max function, the pinch function, and the combination of max and mid function (for non-isothermal and isothermal streams, respectively) with RS+T performed approximately equally well and outperformed the remaining formulations. The better relaxations of the pinch function and the max-mid combination came with a practical benefit only in RS. The alternative formulation using the max function in Eq. (6) did not appear to have any advantage over the standard one. In fact, for unknown reasons it did not converge within 1,000 s with the Regenerative Rankine Cycle case, even with added optimization variables. The implicit formulation as well as the one based on the smoothed max function with the considered

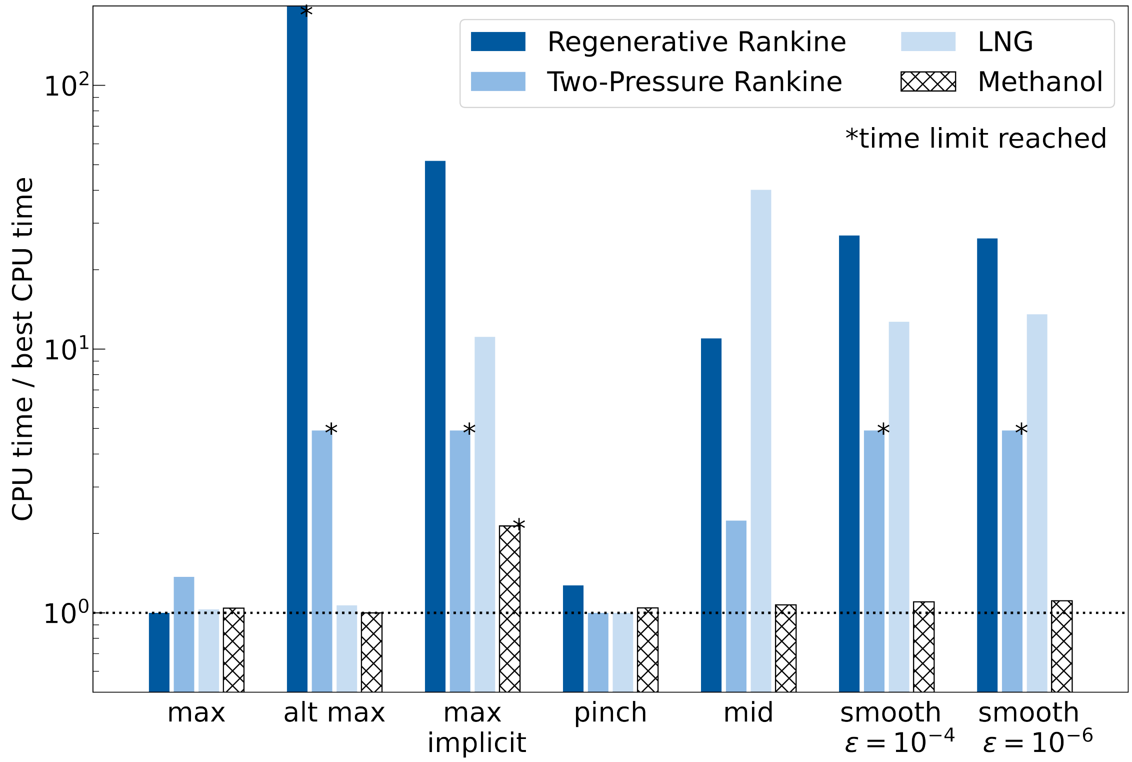

Figure 2: Effect of problem formulation on CPU time in RS+T. Computational performance is measured in terms of CPU times divided by the best CPU time achieved on the respective case study. For all case studies, the original formulation using the max function performs very well in RS+T.

smoothing parameters performed worse than the standard formulation using the max function. In conjunction with a look at the solver logs, this indicates that our heuristic used for upper bounding in the presence of nonsmooth functions works well with the explicit formulation. However, the nested max functions used for the implicit formulation appears to complicate both upper and lower bounding.

The previously presented study was repeated with the remaining case studies considering only RS+T. Again, the standard formulation using the (real, nonsmooth) max function Eq. (5) performs approximately equally well or better than the alternatives (see Fig. 2).

Finally, we can consider dedicated formulations for handling isothermal streams. For this purpose, different strategies were tested for handling isothermal streams, while using the pinch function for non-isothermal streams. The formulations for isothermal streams cannot be tested on the LNG example, as it does not contain any isothermal streams.

Figure 3 shows the CPU time required for optimizing the Rankine Cycles and the methanol process using different strategies for isothermal streams. In each formulation, the pinch function

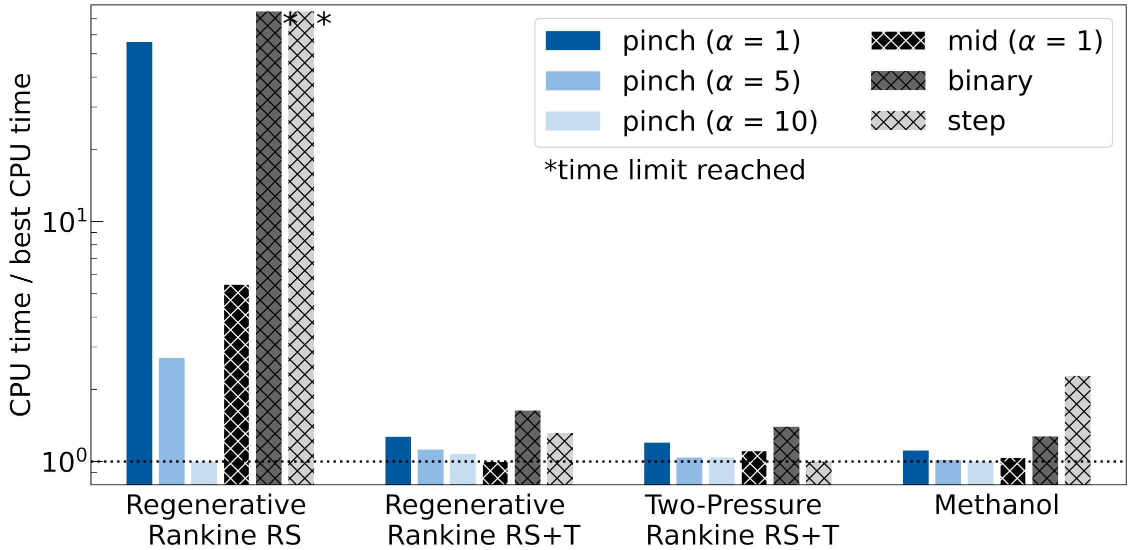

Figure 3: Computational performance of different formulations for isothermal streams relative to best performing case on the respective study. Using dummy temperature change performs well independent of the numerical value of α in RS+T.

is used for non-isothermal streams and only the strategy for isothermal streams is varied. The pinch and mid function can be used in conjunction with a dummy temperature change α, whereas binary variables and the step function as used in Eqs. (12) do not assume any dummy temperature change. The binary formulation used is the one presented in Eqs. (11) for isothermal streams (non-isothermal streams are still modeled using the pinch function). Recall that RS and RS+T are identical for the methanol case (see Section 3.3). RS is not considered for the Two-Pressure Rankine Cycle, as it takes significantly longer than 1,000 s to solve regardless of the formulation used for isothermal streams.

Overall, the methods using dummy temperature change and the pinch function or the original formulation based on the max function (not depicted in Fig. 3) converged quickly and did not produce any deviation from the result without this assumption. The parameter α had a significant effect on the solution time with RS, but almost none with RS+T. Moreover, in RS, using the mid function for isothermal streams led to significantly faster convergence than using the pinch function when considering the same α. Using binary variables or the step function generally increased the solution time. In contrast to the other formulations, when using the formulation containing binary variables, the best solution is not found immediately during preprocessing, but during the B&B procedure after the binaries are fixed in every node. Therefore, we believe that finding a good upper bound is what slows down the convergence of the mixed-integer formulations. This effect may be less severe with solvers using more elaborate strategies to find feasible solutions in the presence of binary variables, as will be shown in the following section.

4.2 Solver Comparison

In this section, we compare the solution times achieved with MAiNGO with those achieved using other solvers. For this purpose, we use the parser tool of MAiNGO to translate the problems into GAMS format. We then use the solvers ANTIGONE [23], BARON [28], LINDOGLOBAL [20] and SCIP [13] in GAMS to solve the problems to a relative optimality gap of 1%. For ANTIGONE and BARON, all max functions are reformulated by the parser using the absolute value functions, as in Eq. (9), because the max function is not implemented in these solvers. All solutions discussed in this section were obtained on an Intel Xeon CPU E5-2640 v3 at 2.60 GHz. The results are presented for the Regenerative Rankine case only, but additional results for the LNG case are available in the SI.

In terms of the choice of optimization variables, we consider RS, RS+T, and FS for all solvers and problem formulations. Note that when directly implementing a problem with GAMS, many of the reduced-space formulations would be cumbersome to write, since they include few optimization variables and constraints but very long and complex equations in the constraints. All tested formulations are summarized in Table 2.

Figure 4 shows the wall clock time required for the Regenerative Rankine Cycle. The wall clock time is used since not all solvers return their CPU time. For each solver, the best results achieved with any formulation from Table 2 are given for RS, RS+T and FS, respectively. For reference, the case without pinch analysis (best performing among all choices of optimization variables) is also given. ANTIGONE apparently encountered an error while reading some of the problems and terminated immediately with a nonzero exit code and without returning a solution.

RS+T tended to converge faster than RS in cases comprising pinch analysis with BARON being the exception. MAiNGO with RS+T resulted in the shortest solution time for both the reference case and the one with pinch analysis. In RS+T, MAiNGO solved the simultaneous flowsheet optimization and heat integration problem about one order of magnitude faster than the

Table 2: Overview of all formulations tested for the solver comparison.

isothermal streams	non-isothermal streams
max (Eq. (5))	max (Eq. (5))
max-smooth ($\epsilon = 10^{-4}$) (Eq. (10))	max-smooth ($\epsilon = 10^{-4}$) (Eq. (10))
max-smooth ($\epsilon = 10^{-6}$) (Eq. (10))	max-smooth ($\epsilon = 10^{-6}$) (Eq. (10))
binary (Eqs. (11))	binary (Eqs. (11))
binary big-M (see [15])	binary big-M [15])
max (Eq. (5))	binary (Eqs. (11))
max (Eq. (5))	binary big-M (see [15])
NLP formulation from [8]	
reference case (no pinch analysis)	

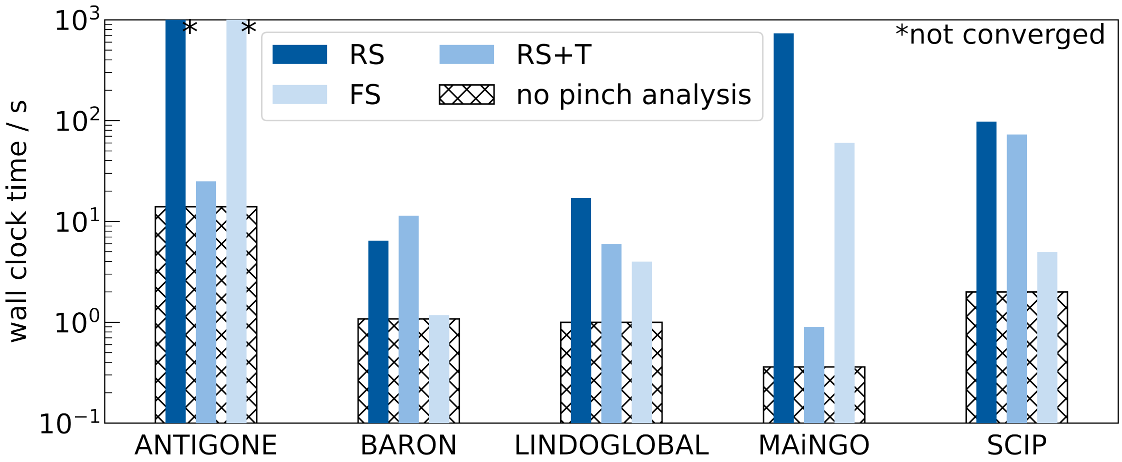

Figure 4: Solver comparison for the Regenerative Rankine Cycle. The respective best formulation in RS, RS+T and FS with pinch analysis is given for each solver. MAiNGO with RS+T requires the least computational time for solving the problem including pinch analysis.

remaining solvers except for a single formulation (mixed-integer from Eq. (11) in FS) solved with BARON. The results of this study underline the different trade-off between dimensionality and tightness of relaxations for the different solvers. Most solvers handled the increased dimensionality associated with the FS formulation well, while MAiNGO (to a point) favors lower dimensionality at the cost of more complex constraints. With all solvers, the simultaneous flowsheet optimization and heat integration problem can be solved within less than four times the time required for the flowsheet optimization without pinch analysis.

A table containing the solution times for all solver - formulation combinations is given in the SI. With MAiNGO, using the standard formulation and RS+T was the fastest option. With the remaining solvers, the mixed-integer formulations generally performed better. For LINDOGLOBAL and SCIP, the computational times were nearly identical regardless of whether the bilinear or the big-M binary-containing formulation was used. Meanwhile, BARON seemed to work well with the bilinear formulation, while ANTIGONE favored the BigM reformulation. Notably, the formulation from [8] did not converge with any solver and choice of optimization variables (smallest achieved optimality gap 13.8 % with LINDOGLOBAL), although the correct global optimum (as an upper bound) was usually found quickly. This may be at least partially due to the increased dimensionality caused by introducing 60 optimization variables and 60 constraints in this particular case study, which is a significant number, when considering RS formulations.

A similar study was conducted using the LNG process, the results of which are qualitatively similar to the ones presented in Fig. 4 and can be found in the SI. There are three notable differences between the results of that study and the one performed on the Regenerative Rankine Cycle. First, the original formulation using the max function and RS+T performed very well with all solvers.

Second, with that formulation, SCIP outperformed MAiNGO, which in turn outperformed the other solvers. Third, the formulation adapted from Cassanello et al. [8] converged within 1,000 s when using SCIP or BARON. Recall that no reference case without pinch analysis exists for that process model.

5 Conclusion

We discussed different strategies for formulating the simultaneous flowsheet optimization and heat integration problem and compared their computational performance when solved with different global optimization solvers. In addition to the original formulation using the max function [12], we tested variations of this formulation that use a smoothed max function, nonsmooth functions, mixed-integer reformulations, or a combination thereof.

We tested three strategies in terms of choosing optimization variables, namely a reduced-space formulation, a reduced-space formulation with the temperatures occurring in pinch analysis added as optimization variables with corresponding constraints, and a full-space formulation. In the reduced-space formulation, introducing pinch analysis into the flowsheet optimization problem requires only few or even no additional variables, depending on the chosen formulation of the pinch problem. For all cases comprising pinch analysis solved with our open-source solver MAiNGO, the reduced-space formulation with temperatures added as optimization variables performed best. This was also the case for all other solvers with the LNG case, but not with the Regenerative Rankine Cycle case. Recall that this formulation corresponds to eliminating as many variables and equality constraints as possible by algebraic transformations and then adding the temperature variables back to the problem along with the corresponding constraints. Our findings underline the known trade-off between tight bounds and low dimensionality: The reduced-space formulation has lower dimensionality but potentially weaker bounds than the formulations with more variables (see [2], Section 3.4).

For all solvers, using the original formulation based on the max function for non-isothermal streams led to good performance if all temperatures occurring in pinch analysis are optimization variables. In MAiNGO, handling isothermal streams by employing a dummy temperature change and handling them as non-isothermal resulted in the best performance. For the remaining solvers, it appeared beneficial to use binary variables for isothermal streams. For SCIP and BARON, using binary variables for all streams sometimes outperformed the remaining formulations. Overall, MINLP formulations worked moderately well for most solvers, but are unsuitable for MAiNGO.

The tested smoothing approaches did not prove useful for global optimization of the considered flowsheets, since a heuristic treatment of the nonsmooth functions for upper bounding worked well in practice, and the smoothing did indeed weaken the relaxations. Based on these finding, we recommend using the original formulation based on the max function for non-isothermal and binary variables for isothermal streams. In terms of dimensionality, the reduced-space formulation with added temperature variables worked best for MAiNGO for all cases and for most other solvers for the LNG case. However, for the Regenerative Rankine Cycle, BARON, LINDOGLOBAL and SCIP performed better when using the full-space formulation.

Overall, in the two studies conducted with multiple solvers, MAiNGO outperformed most of the remaining commercial and open-source solvers tested in both the cases with and without pinch analysis when choosing the respective best-performing formulation for each solver. The only exception is one optimization conducted with SCIP in the LNG case. In general, MAiNGO appears to favor low dimensionality at the cost of more complex constraints more than the other solvers do. This is likely due to the fact that McCormick relaxations allow to exploit the reduced-space formulation better than the auxiliary variable method does. However, additional studies should be conducted to confirm this.

Acknowledgements We thank the anonymous reviewers for their constructive feedback that have led to significant improvements of the manuscript. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, Exzellenzcluster 2186 "The Fuel Science Center". ID: 390919832.

Supplementary Material

The full mathematical problem formulations as well as additional details on the case studies are available as a Supplementary Information. Additionally, the C++ and GAMS implementations of the case study problems are available at `http://permalink.avt.rwth-aachen.de/?id=198813`.

References

[1] D. Bongartz, J. Najman, S. Sass, and A. Mitsos. MAiNGO – McCormick-based Algorithm for mixed-integer Nonlinear Global Optimization. Technical report, Process Systems Engineering (AVT.SVT), RWTH Aachen University, 2018. URL `http://permalink.avt.rwth-aachen.de/?id=729717`. The open-source version of MAiNGO is available at `https://git.rwth-aachen.de/avt-svt/public/maingo`.

[2] Dominik Bongartz. *Deterministic Global Flowsheet Optimization for the Design of Energy Conversion Processes*. PhD Thesis, RWTH Aachen University, 2020.

[3] Dominik Bongartz and Alexander Mitsos. Deterministic global optimization of process flowsheets in a reduced space using McCormick relaxations. *Journal of Global Optimization*, 69 (4):761–796, 2017.

[4] Dominik Bongartz and Alexander Mitsos. Deterministic global flowsheet optimization: Be-

tween equation–oriented and sequential–modular methods. *AIChE Journal*, 65(3):1022–1034, 2019.

[5] Dominik Bongartz, Jannik Burre, Anita L. Ziegler, and Alexander Mitsos. Power-to-OME_1 via direct oxidation of methanol: Process design and global flowsheet optimization. In *31th European Symposium on Computer Aided Process Engineering*, volume 50 of *Computer Aided Chemical Engineering*, pages 273–278. Elsevier, 2021.

[6] Jannik Burre, Dominik Bongartz, and Alexander Mitsos. Comparison of MINLP formulations for global superstructure optimization. 2021. Submitted for publication.

[7] RP Byrne and IDL Bogle. Global optimization of modular process flowsheets. *Industrial & engineering chemistry research*, 39(11):4296–4301, 2000.

[8] Matteo Cassanello, Yingzong Liang, and Chi Wai Hui. A shortcut method for simultaneous energy and heat exchange area optimization with variable stream conditions. *Applied Thermal Engineering*, 175:115363, 2020.

[9] Benoit Chachuat, Boris Houska, Radoslav Paulen, Nikola Peric, Jai Rajyaguru, and Mario E. Villanueva. Set-theoretic approaches in analysis, estimation and control of nonlinear systems. *IFAC-PapersOnLine*, 48(8):981–995, 2015. URL `https://omega-icl.github.io/mcpp/`.

[10] Xuan Dong, Zuwei Liao, Jingyuan Sun, Zhengliang Huang, Binbo Jiang, Jingdai Wang, and Yongrong Yang. Simultaneous optimization of a heat exchanger network and operating conditions of organic rankine cycle. *Industrial & Engineering Chemistry Research*, 59(25):11596–11609, 2020.

[11] Alexander W. Dowling and Lorenz T. Biegler. A framework for efficient large scale equation-oriented flowsheet optimization. *Computers & Chemical Engineering*, 72:3–20, 2015.

[12] M. A. Duran and I. E. Grossmann. Simultaneous optimization and heat integration of chemical processes. *AIChE Journal*, 32(1):123–138, 1986.

[13] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 7.0. Technical report, Zuse Institute Berlin, March 2020. URL `http://nbn-resolving.de/urn:nbn:de:0297-zib-78023`.

[14] Emre Gençer and Rakesh Agrawal. Toward supplying food, energy, and water demand: Integrated solar desalination process synthesis with power and hydrogen coproduction. *Resources, Conservation and Recycling*, 133:331–342, 2018.

[15] Ignacio E. Grossmann, Héctor Yeomans, and Zdravko Kravanja. A rigorous disjunctive optimization model for simultaneous flowsheet optimization and heat integration. *Computers & Chemical Engineering*, 22:S157–S164, 1998.

[16] F Huang and R Elshout. Optimizing the heat recovery of crude units. *Chemical Engineering Progress*, 72, 7 1976.

[17] Lingxun Kong, Yaqing Wu, and Christos T. Maravelias. Simultaneous utility and heat exchanger area targeting for integrated process synthesis and heat integration. *Industrial & Engineering Chemistry Research*, 56(41):11847–11859, 2017.

[18] D Kraft. A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany, 1988.

[19] Y.-D. Lang, L. T. Biegler, and I. E. Grossmann. Simultaneous optimization and heat integration with process simulators. *Computers & Chemical Engineering*, 12(4):311–327, 1988.

[20] Youdong Lin and Linus Schrage. The global solver in the LINDO API. *Optimization Methods and Software*, 24(4-5):657–668, 2009.

[21] Bodo Linnhoff and John R. Flower. Synthesis of heat exchanger networks: I. systematic generation of energy optimal networks. *AIChE Journal*, 24(4):633–642, 1978.

[22] Yurong Liu, Minglei Yang, Liang Zhao, Wenli Du, Weimin Zhong, and Feng Qian. Simultaneous optimization and heat integration of an aromatics complex with a surrogate model. *Industrial & Engineering Chemistry Research*, 60(9):3633–3647, 2021.

[23] Ruth Misener and Christodoulos A. Floudas. Antigone: Algorithms for continuous / integer global optimization of nonlinear equations. *Journal of Global Optimization*, 59(2):503–526, 2014.

[24] A. Mitsos, B. Chachuat, and P. Barton. McCormick-based relaxations of algorithms. *SIAM Journal on Optimization*, 20(2):573–601, 2009.

[25] JaromiłNajman, Dominik Bongartz, and Alexander Mitsos. Linearization of mccormick relaxations and hybridization with the auxiliary variable method. *Journal of Global Optimization*, 80:731–756, 2021.

[26] Alexander M. Niziolek, Onur Onel, and Christodoulos A. Floudas. Production of benzene, toluene, and xylenes from natural gas via methanol: Process synthesis and global optimization. *AIChE Journal*, 62(5):1531–1556, 2016.

[27] Soterios A. Papoulias and Ignacio E. Grossmann. A structural optimization approach in process synthesis—II: Heat recovery networks. *Computers & Chemical Engineering*, 7(6): 707–721, 1983.

[28] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

[29] A. Tsoukalas and A. Mitsos. Multivariate McCormick relaxations. *Journal of Global Optimization*, 59(2-3):633–662, 2014.

[30] Matias Vikse, Harry A.J. Watson, Donghoi Kim, Paul I. Barton, and Truls Gundersen. Optimization of a dual mixed refrigerant process using a nonsmooth approach. *Energy*, 196: 116999, 2020.

[31] Harry A. J. Watson, Kamil A. Khan, and Paul I. Barton. Multistream heat exchanger modeling and design. *AIChE Journal*, 61(10):3390–3403, 2015.

[32] Achim Wechsung and Paul I. Barton. Global optimization of bounded factorable functions with discontinuities. *Journal of Global Optimization*, 58(1):1–30, 2014.

[33] Achim Wechsung, Audun Aspelund, Truls Gundersen, and Paul I. Barton. Synthesis of heat exchanger networks at subambient conditions with compression and expansion of process streams. *AIChE Journal*, 57(8):2090–2108, 2011.

[34] Linlin Yang and Ignacio E. Grossmann. Water targeting models for simultaneous flowsheet optimization. *Industrial & Engineering Chemistry Research*, 52(9):3209–3224, 2013.

[35] Haoshui Yu, John Eason, Lorenz T. Biegler, and Xiao Feng. Simultaneous heat integration and techno-economic optimization of organic rankine cycle (ORC) for multiple waste heat stream recovery. *Energy*, 119:322–333, 2017.

[36] Yu Zhuang, Mingxin Li, Lei Zhang, Linlin Liu, Jian Du, and Shengqiang Shen. Simultaneous optimization of a heat-integrated coal-to-SNG/MeOH polygeneration process based on rigorous kinetic models. *Industrial & Engineering Chemistry Research*, 59(51):22247–22257, 2020.

[37] Yu Zhuang, Congcong Zhou, Yachao Dong, Jian Du, and Shengqiang Shen. A hierarchical optimization and design of double Kalina cycles for waste heat recovery. *Energy*, 219:119593, 2021.