Systems & Control Transactions

Research Article
Foundations of Computer Aided Process Design (FOCAPD 2024)
Breckenridge, Colorado, USA. July 14-18, 2024
Peer Reviewed Conference Proceeding

PSE PRESS

Optimal Process Synthesis Implementing Phenomena-based Building Blocks and Structural Screening

David Krone[a], Erik Esche[a]*, Mirko Skiborowski[b], and Jens-Uwe Repke[a]

[a] Technische Universität Berlin, Process Dynamics and Operations Group, Berlin, Germany
[b] Hamburg University of Technology, Institute of Process Systems Engineering, Hamburg, Germany
* Corresponding Author: erik.esche@tu-berlin.de.

ABSTRACT

Superstructure optimization for process synthesis is a challenging endeavour typically leading to large scale MINLP formulations. By the combination of phenomena-based building blocks, accurate thermodynamics, and structural screening we obtain a new framework for optimal process synthesis, which overcomes prior limitations regarding solution by deterministic MINLP solvers in combination with accurate thermodynamics. This is facilitated by MOSAICmodeling's generic formulation of models in MathML / XML and subsequent decomposition and code export to GAMS and C++. A branch & bound algorithm is implemented to solve the overall MINLP problem, wherein the structural screening penalizes instances, which are deemed nonsensical and should not be further pursued. The general capabilities of this approach are shown for the distillation-based separation of a ternary system.

Keywords: Process Synthesis, Optimization, Distillation, Phase Equilibria, Phenomena Building Block

MOTIVATION & INTRODUCTION

Optimal Process Synthesis

Thermal separation processes make up a large part of energy consumption in the US and worldwide [1]. Given the drive to reduce greenhouse gas emissions, it is imperative to explore more energy efficient solutions, e.g., through heat integration, novel separation process concepts, or novel equipment, etc. Exploring these options is complex and a continuing challenge. Mathematical methods with mechanistic models for process synthesis help overcome this hurdle. However, process synthesis is quite a challenging field, for which Chen et al. [2] emphasize the delicate trade-off between generality, fidelity, and tractability in process synthesis methods. So far, methods for process synthesis try to reduce the search space either by focusing on individual synthe-sis tasks (lower generality) or by simplifying models (lower fideli-ty).

Synthesis methods based on rigorous superstructure opti-mization promise to overcome these shortcomings. Formulating and solving these process synthesis problems as large-scale mixed-integer nonlinear programming (MINLP) or generalized disjunctive programming (GDP) problems is tough. While, it is well understood how these should be formulated, tractability is the main challenge here.

Three approaches for superstructure optimization build on phenomena-based formulations. The group of Pistikopoulos developed the generalized modular framework (GMF), which builds on a multipurpose mass / heat transfer module. GMF has, e.g., been applied for synthesis of distillation processes [3, 4] and even reactive distillation [5].

Similarly, the group of Hasan has derived abstract building blocks (ABB), which are arranged in a chessboard-like two-dimensional block superstructure. The boundaries to adjacent blocks and physical attributes of the blocks can be modified during optimization [6, 7]. The ABB formulation is highly versatile. However, application to synthesis of more complex processes so far requires simplification of the ABB model [8] or specialized iterative solution sequences [9].

Finally, Kuhlmann and Skiborowski [10] developed a Phe-nomena-based Building Blocks (PBB) approach. These building blocks consist of thermodynamically accurate equilibrium stage models and kinetics to form a general state-space superstructure. The capabilities of

this approach have been demonstrated for the synthesis of a membrane reactor [11] and membrane-assisted reactive distillation [12]. So far, solving the formulated MINLPs relied on an evolutionary strategy for the mixed-integer side and local solution of the nonlinear sub-problems in Aspen Custom Modeler. This heavily limits the capabilities of the solvers and poses challenges for the formulation, initialization, and solution of larger examples.

Challenges & Approach

In the current work, the limitations of the approach by Kuhlmann and Skiborowski [10] will be addressed by suggesting a new framework for formulation and solution of the MINLP problems. In this contribution, we advance the prior work in two directions: (1) formulating and generating large-scale superstructures and (2) facilitating the solution through structural screening. Towards (1) the modeling environment MOSAICmodeling [13, 14] is here extended for the formulation of superstructures of phenomena-based building blocks and their decomposition and code generation for a target language, i.e., the modeling language for the actual solution, e.g., GAMS, AMPL, Python, etc. The implementation in MOSAICmodeling involves formulation of the superstructure problem as a meta model in MathML / XML. With this formulation, executable code for different program components can be obtained automatically, including a mathematical programming platform, an external CAPE-OPEN thermodynamic engine, and a platform for structural screening. This is achieved through the utilization of separate User-defined Language Specificators (UDLS) [15] within the web-based modeling and optimization tool MOSAICmodeling, developed at TU Berlin [16]. All thermodynamic property calculations are outsourced to a CAPE-OPEN-compliant property package, while the MINLP is exported to GAMS (version 40.4.0).

Towards the structural screening (2), a number of pruning and screening techniques have been previously suggested, e.g., in [10, 17]. These typically operate at a local level, i.e., the connections surrounding a module or PBB. In our contribution, we deviate from this approach and add towards screening / pruning techniques, which also analyse the entire superstructure instance. The wider perspective aligns with the axioms for generating feasible superstructures within the P-graph framework [18].

Here, a branch and bound-type algorithm is implemented in GAMS and augmented with a middle layer, which pre-screens based on binary decision variables of the superstructure. The middle layer performs a set of graph- and rule-based analyses of a structural instance, which are further detailed in section “Structural Screening”. In case of violations of these rules, i.e., an instance will not lead to a physically sensible solution, this structure is penalized without solution of the underlying MINLP. While a similar concept is also present in the approach by Kuhlmann and Skiborowski [10], the MINLP framework extends the screening process with additional rules and algorithms from graph analysis.

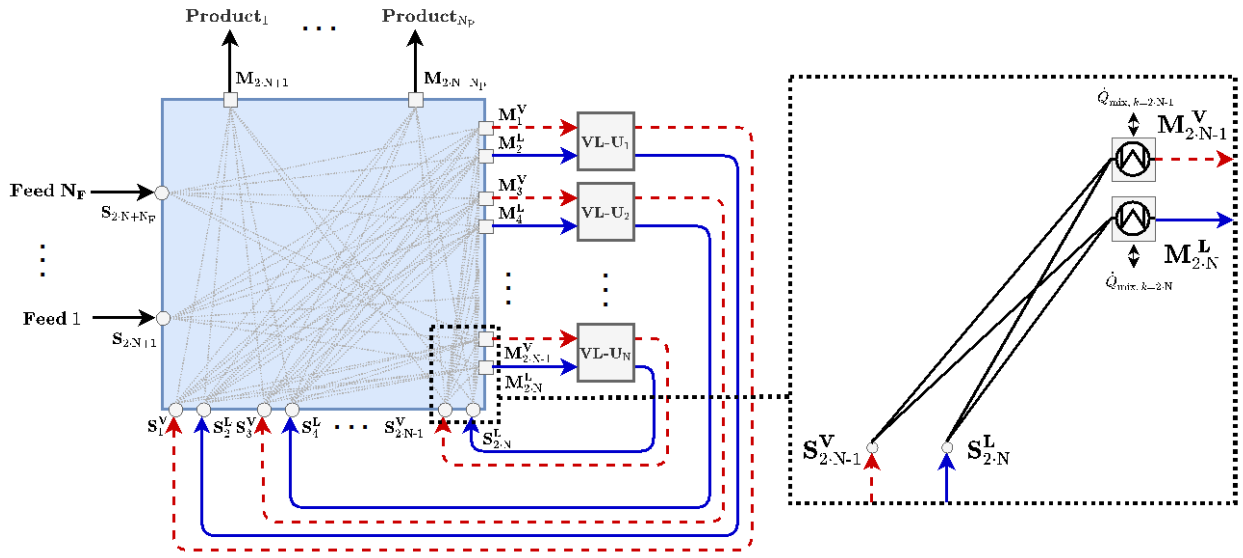

Figure 1. Generic superstructure with vapour-liquid equilibria blocks (VL-U). The left-hand side shows the distribution network with the connected VL-U PBB and the right-hand side a close-up of the connection of splitter nodes to mixer nodes. Red dashed lines signals vapor phase, blue solide lines liquid phase.

METHODOLOGY & IMPLEMENTATION

Generic Superstructure Model

The superstructure formulated here is based on the work of Kuhlmann and Skiborowski [10], which combines a distribution network connecting feed and product streams (blue box in Fig. 1) with a fixed number of PBBs (light grey boxes in Fig. 1). The following extensions are made compared to prior work:

In the distribution network, separate energy flows for pressure manipulation and heat exchange are present. This should allow for deactivation of equipment based on (in-)existence of preceding or subsequent blocks. For the current contribution, we focus on PBBs featuring vapor-liquid equilibrium units (VL-U). In the distribution network, splitter nodes connect the recycles and inlets to mixer nodes and product nodes. Restrictions are implemented on the mixer nodes based on each connected VL-U PBB: By default, the outlet must either be in boiling liquid state or saturated vapor. Binary variables govern the distribution of splitter outlets to mixer nodes. A stream is (for now) split into at most two streams to limit computational complexity. This shall be relaxed later but is deemed for now of low importance. The distribution network changes temperatures before streams enter a PBB.

Here, the formulation for the distribution network is implemented as given on the right-hand side of Fig. 1 (dashed box). Each mixed stream flows through a heat exchanger. The temperature is adjusted according to the input requirements of the connected VL-U, i.e., heated or cooled to the respective boiling or dewpoint temperature of the mixture. Compression and expansion are implemented as isentropic state changes with fixed efficiencies. Each VL-U block is a countercurrent multi-stage contactor in accordance with Kuhlmann and Skiborowski [10]. On each stage, mixing of liquid and vapor and separation take place.

As stated before, phase constraints are enforced on both inlets of a VL-U PBB. This implies that the preceding heat exchangers must be active to ensure that the inlet is either in vapor or liquid state. Nevertheless, as a results of the superstructure optimization, combinations of the VL-U should be able to form conventional equipment, e.g., distillation towers. For these, the heat exchangers at the connecting point of two VL-U sitting “on top of” one another, should be deactivated (see, e.g., VL-U I and VL-U II in Fig. 2).

For this purpose, additional binary variables are included in the model, which (de-)active these heat exchangers and hence also the phase constraints at the respective inlet of the VL-U PBB. This allows for increased flexibility of the overall superstructure. An inactive heat exchanger is bypassed, and the fluid is supplied to the VL-U in whatever state it might currently be in.

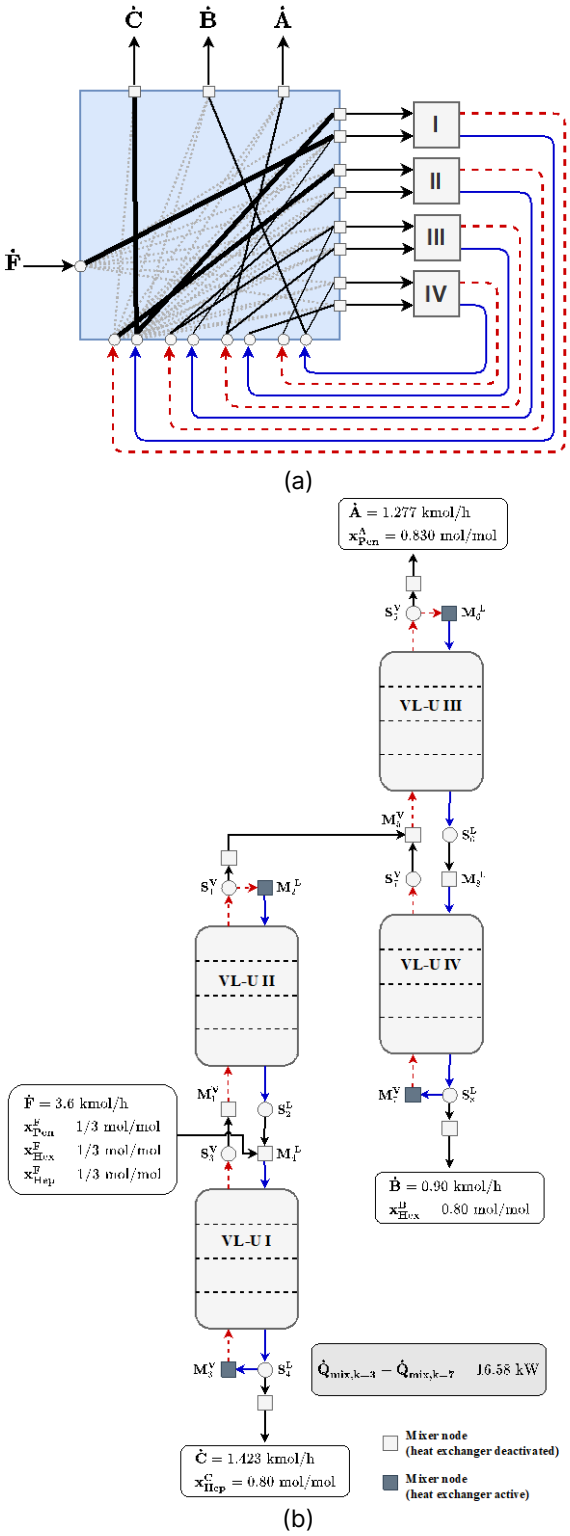

Figure 2. Optimization results for the separation of a feed stream of n-pentane, n-hexane, and n-heptane by 4 VL-U: (a) resulting connections in the superstructure, (b) interpretation as a sequence of two columns.

Further details regarding the model formulation, e.g., activation / deactivation of heat exchangers, can be

found in [19].

As a result of deactivated heat exchangers, the superstructure also allows, e.g., a vapor stream to enter a VL-U by its dedicated liquid inlet. In addition to this, further uncommon situations within each VL-U may arise during superstructure optimization. Therefore, the VLE formulation on each stage of the VL-U is augmented by relaxation through complementarity constraints [20, 21] as shown in Krone et al. [13] to account for liquid-only or vapor-only scenarios on each stage.

For a given synthesis task, we implement an objective function, which minimizes the total annualized costs consisting of investment costs for heat exchangers, and operation costs for heating utilities. A penalty term is added to the objective for regularization of the complementarity constraints. The penalty converges to zero at a feasible solution.

Thermodynamic Properties

Within the MathML / XML model in MOSAICmodeling, all thermodynamic properties are denoted as external function calls, i.e., enthalpies, entropies, equilibrium coefficients, temperatures of boiling or dew point. For these, only the appropriate input variables are assigned to compute the desired output, e.g., enthalpy as a function of temperature, pressure, and composition. These are linked in accordance with the CAPE-OPEN standard for thermodynamic engines using COBIA (version 1.2.0.8) as architecture. TEA (version 3.5) provided by AmsterChem is used as thermodynamic engine.

Code Generation and MINLP Framework

The novel MINLP framework is illustrated in Fig. 3. Starting from MOSAICmodeling, code is automatically generated for GAMS and C++ (see Krone et al. [13]). The GAMS code then contains the entire MINLP including the state-space superstructure. The thermodynamic properties are marked as external equations and further detailed in the linked C++ code. Therein, the prerequisites for CAPE-OPEN function calls and the COBIA architecture are implemented, function values and derivatives are obtained and returned to GAMS. The overall system is a mixed integer nonlinear programming (MINLP) problem. There are two types of binary variables for the superstructure and for activation or deactivation of phase constraints of the mixers in the network. The superstructure variables by themselves cause a very high computational complexity.

Tractability of the problem is achieved through an augmented solution strategy: The MINLP is solved by a branch and bound algorithm combined with a middle layer that analyzes individual structures. For details on the branch and bound implementation see [19] and [22]. This is implemented in GAMS. The middle layer with the structural screening, however, is formulated within MATLAB (version R2023b).

In each iteration of the branch and bound solver, the current structural instance is passed on to MATLAB, where the screening is carried out regarding the rules specified in the section on “Structural Screening” below. These subproblems are still MINLP as there are further binaries, which are not elemental to the superstructure but form further options within the model formulations, e.g., phase constraints (see above).

Structural Screening

As a novelty of our contribution, we employ graph- or network-based constraints in addition to algebraic

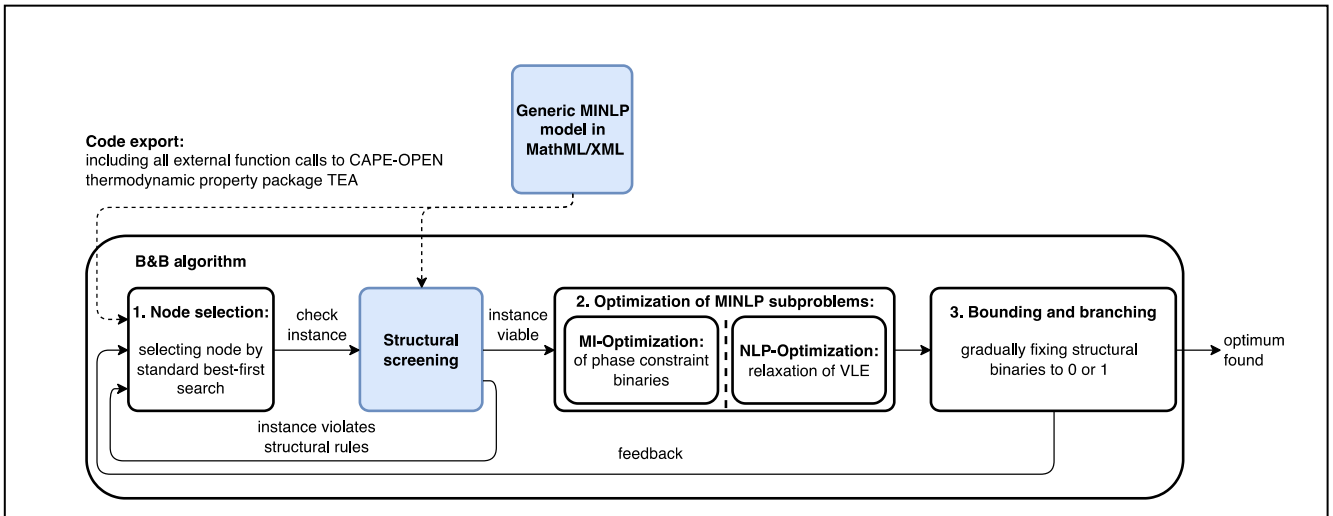

Figure 3. Structure of the developed MINLP-framework with its core features: (1) code export from the generic MINLP-model in MathML/XML, (2) the adapted branch and bound (B&B) algorithm as the outer layer of the MINLP optimization, as well as (3) an inserted middle layer performing structural screening of the MINLP subproblems. (Figure published in [19].)

constraints. At the moment, we assume that the graph-based constraints can only be resolved (efficiently) algorithmically and not implemented as algebraic constraints in a straightforward fashion. This remains to be analyzed in future work.

For graph-based analysis, the current superstructure is exported as a directed graph. The following seven rules are implemented to screen structures at intermediate iterations of the branch and bound algorithm; three of the rules implemented for structural screening are formulated similarly to [10]:

Rule 1: An active VL-U PBB without a connected stream to the inlet mixers needs to be avoided. This PBB will be considered degenerate and the according structure is discarded.

Rule 2 (cf. [10]): During superstructure optimization, it could occur that a subsection of a structural instance is not connected to any feed or product streams – not even indirectly, i.e., a structure which is fully closed to the outside. Such a substructure might cause a host of numerical issues, definitely does not contain any added value, and will hence be discarded.

Rule 3: Within the superstructure several equivalent formulations exist. One example is the interchange of two split streams. To limit the number of equivalencies, rule 3 is implemented to ensure that a connected product node is always served by the first split stream coming from a connected splitter node.

Rule 4: Connections to the product outlets, which appear nonsensical, should be ruled out. For example, this includes connection of two splitter nodes in the distribution network to a product node. This is deemed to be unnecessary entropy generation.

Rule 5 (cf. [10]): A stream that leaves a VL-U PBB should not be fed back to an inlet of the same VLU- PBB. This is considered a “direct recycle”, which also does not appear to be logical from a thermodynamic point of view.

Rule 6: The two inlets of a VL-U PBB should not originate from the same splitter node. Yet again, this rule should be well-founded in thermodynamic considerations and always apply to PBBs of type VL-U.

Rule 7 (cf. [10]): Two outlet streams of a VL-U PBB should not be fed back to the same mixer node. This would counteract any previously achieved separation and hence also amounts to entropy generation.

CASE STUDY

The formulation described above is applied on a process synthesis task to separate a mixture of n-pentane, n-hexane, and n-heptane by a network of up to six building blocks each representing a counter-current cascade of five vapor-liquid equilibrium stages (VL-U).

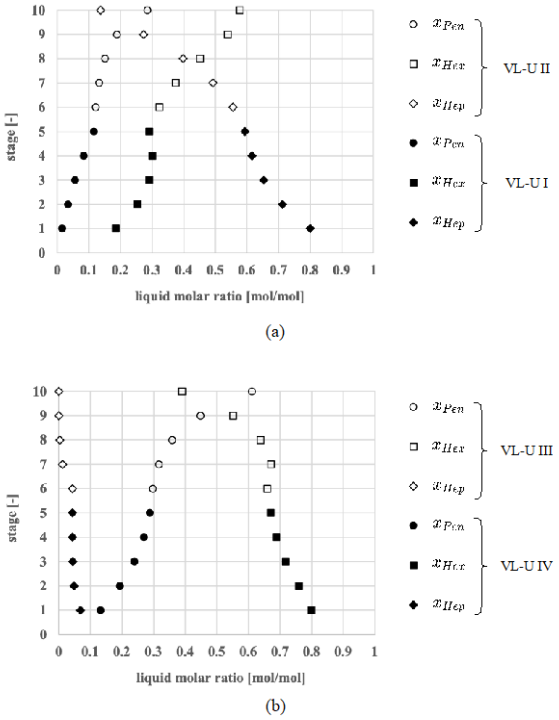

Figure 4. Liquid composition profiles inside of the optimal design determined for the separation of an n-pentane, n-hexane, and n-heptane mixture by 4 VL-U: (a) liquid molar ratios on the stages of VL-U I & II, (b) liquid molar ratios on the stages of VL-U III & IV.

This leads to an optimization problem with 1303 (in-)equalities, 62 binary, and 1123 continuous variables. The problem is implemented in MOSAICmodeling and solved in GAMS combining the branch and bound solver with DICOPT [23] for MINLP subproblems, CONOPT3 (version 3.17) [24] for relaxed MINLPs, and IPOPT (version 3.14) [25] for NLP subproblems. The structural screening in Matlab is performed in each iteration of the branch and bound solver.

For now, the system pressure is fixed at 100 kPa and the desired product qualities are specified at purities of 80 mol/mol % each for n-pentane, n-hexane, and n-heptane. The feed stream is specified at 3.6 kmol/h as saturated vapor and features equal amounts of all three components.

Results

Fig. 2 shows the results for four VL-U PBB in the network (a) and an interpretation as a sequence of two columns (b) and Fig. 4 shows the respective liquid mole fractions of all three components. The results appear to be consistent and are cost-optimal compared to alternative configurations.

This result is obtained after 94 iterations of the Branch & Bound algorithm with roughly 172 h of CPU time. Computation is done on an Intel® Core™ i7-2600

processor (3.8 GHz and 4 cores) with 16 GB of RAM. This is still quite computationally expensive but shows the general feasibility of the overall approach. It should be noted at this point that no particular initialization strategy is applied for the continuous nonlinear variables during the superstructure optimization. Each MINLP and NLP subproblem is solved with the currently available prior values. We assume that a considerable speedup could be achieved by implementing some kind of rigorous initialization strategy or homotopy to guide the NLP solver to the solution.

We would like to stress at this point that all these results are obtained using no specific / manual initialization of the state variables. At the moment, all variables are initialized at the mid-points of their intervals. This mediocre initialization will cause a large part of the CPU time. Furthermore, the VLE formulation including the interfacing via CAPE-OPEN is not yet very efficient. In future work, we shall evaluate different formulations for the external computation to achieve a speedup. Prior work by [11, 5] promise a sizable speedup using different formulations.

The inclusion of the structural screening procedure proved to be vital for the overall set-up. Without the middle layer, the optimizer failed to solve the system at all. In our case studies, we observed behavior which we interpret in two different ways: First of all, some of the otherwise penalized nodes in the branch & bound algorithm consume a lot of time and typically the solution is aborted based on our maximum of around 1000 h. Second, solution to local infeasibility of some of the otherwise penalized subsystems leads to a degradation of the initial values of all continuous variables, which causes infeasibility in a subsequent node, which might otherwise have been successfully solved.

Of the 94 iterations, 23 iterations involved penalization of the current structure based on the screening rules. In these cases, the lower-level solvers would otherwise have probably failed to solve to any reasonable solution, so this also implies a great save in computational expenditure. Overall, we can observe a great contribution of the structural screening and pruning towards ensuring feasibility of these type of MINLP problems. Given that without screening, the optimization did not converge to a feasible solution within the given time frame, we cannot further quantify how great the speed-up due to the structural screening is.

CONCLUSIONS & OUTLOOK

The presented, novel approach for formulation and solution of process synthesis problems through superstructure optimization can reliably solve the challenging MINLP determining the optimal structure of a separation train and find its energetic optimum. The implemented structural screening is vital to ensure solvability depending on the problem formulation at hand.

The main advantage in using the MathML / XML-based model formulation lies in flexibility regarding setting up even larger superstructure problems and in flexibility regarding the solution environment. A switch from the GAMS / MATLAB / C++ combination to, e.g., a Python-based setting should be straight forward.

In future work, we shall attempt to further decrease the computational complexity by employing GDP-type formulations for the superstructure and export the model to other platforms, e.g., PYOMO, to speed up the solution and make use of state-of-the-art GDP solvers. Also, we will investigate reduced order models to implement the initialization of the structural instances during the optimization.

ACKNOWLEDGEMENTS

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 523327609.

REFERENCES

1. Sholl DS, Lively RP. Seven chemical separations to change the world. *Nat* 532, 435–427 (2016)
2. Chen Q, Liu Y, Seastream G, Siirola JD, Grossmann, IE. Pyosyn: A new framework for conceptual design modelling and optimization. *Comp Chem Eng* 153, 107414 (2021)
3. Papalexandri KP, Pistikopoulos EN. Generalized modular representation framework for process synthesis. *AIChE J* 42:4, 1010–1032 (1996)
4. Proios P, Pistikopoulos EN. Generalized modular framework for the representation and synthesis of complex distillation column sequences. *Ind & Eng Chem Res* 44:13, 4656–4675 (2005)
5. Ismail SR, Proios P, Pistikopoulos EN. Modular synthesis framework for combined separation / reaction systems. *AIChE J* 47:3, 629–649 (2001)
6. Demirel SE, Li J, Hasan MF. Systematic Process Intensification using building blocks. *Comp & Chem Eng* 105, 2–38 (2017)
7. Li J, Demirel SE, Hasan MF. Process Integration Using Block Superstructure. *Ind & Eng Chem Res* 57:12, 4377–4398 (2018)
8. Demirel SE, Li J, Hasan MF. Systematic Process Intensification. *Curr Opin Chem Eng* 25, 108–113 (2019)
9. Demirel SE, Li J, El-Halwagi M, Hasan MF. Sustainable Process Intensification Using Building Blocks. *ACS Sustain Chem Eng* 8:48, 17664–17679 (2020)
10. Kuhlmann H, Skiborowski M. Optimization-based

approach to process synthesis for process intensification: General approach and application to ethanol dehydration. *Ind Eng Chem Res* 56, 13461–13481 (2017)
11. Kuhlmann H, Möller M, Skiborowski M. Analysis of tba-based etbe production by means of an optimization-based process-synthesis approach. *Chem Ing Tech* 32:3, 336-348 (2018)
12. Kuhlmann H, Veith H, Möller M, Nguyen KP, Górak A, Skiborowski M. Optimization-based approach to process synthesis for process intensification: Synthesis of reaction-separation processes. *Ind Eng Chem Res* 57, 3639–3655 (2018)
13. Krone D, Esche E, Asprion N, Skiborowski M, Repke JU. Enabling optimization of complex distillation configurations in GAMS with CAPE-OPEN thermodynamic models. *Comp Chem Eng* 157, 107626 (2022)
14. Esche E, Hoffmann C, Illner M, Müller D, Fillinger S, Tolksdorf G, Bonart H, Wozny G, Repke JU. Mosaic - enabling large-scale equation-based flow sheet optimization. *Chem Ing Tech* 89, 620–635 (2017)
15. Tolksdorf G, Esche E, Wozny G, Repke JU. Customized code generation based on user specifications for simulation and optimization. *Comp Chem Eng* 121, 670–684 (2019)
16. Merchan VA, Esche E, Fillinger S, Tolksdorf G, Wozny G, Computer-aided process and plant development. A review of common software tools and methods and comparison against an integrated collaborative approach. *Chem Ing Tech* 88, 50-69 (2016)
17. Wu W, Henao C, Maravelias C. A superstructure representation, generation, and modeling framework for chemical process synthesis. *AIChE J* 62:9, 3199–3214 (2016)
18. Friedler F, Tarjan K, Huang YW, Fan LT. Combinatorial algorithms for process synthesis. *Comp Chem Eng* 16, 313-320 (1992)
19. Krone D, Esche E, Skiborowski M, Repke, JU. Optimization-Based Process Synthesis by Phenomena-Based Building Blocks and an MINLP-Framework Featuring Structural Screening, submitted to *Comp Chem Eng* (2023)
20. Gopal V, Biegler LT. Smoothing methods for complementarity problems in process engineering. *AIChE J* 45, 1535–1547 (1999)
21. Lin GH, Fukushima M. A modified relaxation scheme for mathematical programs with complementarity constraints. *Ann Oper Res* 133, 63–84 (2005)
22. Kalvelagen E. Branch-and-bound methods for an MINLP model with semi-continuous variables, http://www.amsterdamoptimization.com/pdf/bb.pdf (2003)
23. Viswanathan J, Grossmann IE. A combined penalty function and outer-approximation method for MINLP optimization. *Comp Chem Eng* 14:7, 769-782 (1990)
24. Drud AS. CONOPT - A Large-Scale GRG Code. *ORSA J Comput* 6:2, 207-216 (1994)
25. Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Program* 106, 25–57 (2006)