Article:
Gopinath, S. and Adjiman, C.S. (2025) Superstructure optimization with rigorous models via an exact reformulation. Computers & Chemical Engineering, 194. 108972. ISSN: 0098-1354

https://doi.org/10.1016/j.compchemeng.2024.108972

Superstructure optimization with rigorous models via an exact reformulation

Smitha Gopinath[1] and Claire S. Adjiman[2]

[1]School of Chemical, Materials and Biological Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, United Kingdom

[2]Imperial College London, Department of Chemical Engineering, Sargent Centre for Process Systems Engineering and Institute for Molecular Science and Engineering, London SW7 2AZ, United Kingdom

March 7, 2025

Abstract

The applicability of superstructure optimization to process synthesis is often limited to simple models and flowsheets. The state operator network (SON) (Smith, E. and Pantelides, C. (1995). *Comput. & Chem. Eng.*, 19:83) overcomes some limitations via a mixer-splitter network, allowing the use of rigorous unit models. However, setting flowrates to zero for non-selected units can result in numerical issues. Here, the modified state operator network (MSON), a new exact reformulation with modified mixers and splitters, is introduced. When a unit is deselected, a fictitious, strictly positive, mixer inlet flow ensures the unit model is easily solved. A corresponding fictitious splitter outlet counteracts this inlet, resulting in correct flowsheet behaviour. When applied to a toy flowsheet, the MSON outperforms standard formulations. When applied to the synthesis of a reactor-separator network and to a challenging

counter-current column synthesis problem, the MSON offers a systematic and robust approach to superstructure optimization.

1 Introduction

Process synthesis is an important step in computational flowsheet design (Cremaschi, 2015). It is the activity of finding:

1. the process units (from a set of allowed process units),
2. the connectivity between the selected process units, and
3. the operating conditions of each of the selected process units,

that minimize the design objective and satisfy constraints. The main approaches to process synthesis are i) decomposition-based approaches (Douglas, 1985) in which the list of flowsheet alternatives to be considered is iteratively narrowed down by using heuristics or ii) optimization-based approaches known as superstructure optimization (SO) in which all flowsheet alternatives are first embedded in a superstructure and the superstructure is then optimized. Approaches that combine heuristics and mathematical programming have also been proposed (Tula et al., 2015). The focus of this paper is on approaches to process synthesis based on superstructure optimization.

The representation of a superstructure is not unique (Demirel et al., 2019, Kondili et al., 1993, Liñán et al., 2020, Mencarelli et al., 2020, Yeomans and Grossmann, 1999). In this paper, using the nomenclature given in Yeomans and Grossmann (2000b) and Smith (1996), we take as a starting point the State Operator Network (SON) (Smith, 1996, Smith and Pantelides, 1995) representation of a superstructure. In the SON paradigm, each process unit-operator (or process unit) is represented by its detailed, rigorous model. That is, each process unit is described by its constitutive equations (including the mass balances, equilibrium relations, mole/mass fraction summations, and energy balances (MESH equations)), detailed thermodynamic model(s) as well as sizing and costing correlations. A network

of mixers and splitters allows complete connectivity between process units. The decision variables include the discrete (binary) decisions corresponding to each conditional process unit (whether it is selected or not in the optimal flowsheet); the continuous flowrates from the outlets of process units to the inlets of process units; the (usually) continuous design variables of all the process units selected to be in the flowsheet. These design variables are the degrees of freedom for a given flowsheet topology.

The SON superstructure is appealing as the number of binary decision variables is small (equal to the number of conditional process units). It does not suffer from combinatorial explosion when the number of components considered in the flowsheet increases, unlike, say, the State Task Network (STN) representation (Kondili et al., 1993). The SON does not require the modeller to make assumptions on the performance of process units, e.g., that sharp splits are attainable in each distillation column, and is thus applicable to rigorous models. Thus, the designs obtained by the use of the SON may be realizable in practice when high-fidelity process unit models are used. A current limitation of the SON formalism introduced in Smith and Pantelides (1995) is that it is only applicable to isobaric flowsheets. This is because connections between two units at different pressures would also require the addition of pressure-change equipment (e.g., pumps, compressors or throttling valves) between these units. Additionally, as the SON necessitates models of all units to be applicable to all components, the modelling of reactions can be more complex in the SON than with other approaches. Finally, numerical solution of the SON can be challenging as result of the use of rigorous models that are highly nonlinear and coupled. Smith and Pantelides (1995) evaluated the optimal flowsheet in the SON superstructure by enumeration of all discrete alternatives.

Rigorous models for each process unit may be implemented in a chemical process simulator (PS), or a general-purpose modelling and optimization (MO) environment, or a hybrid framework in which a subset of constraints is implemented in the MO environment and the rest in the PS (Javaloyes-Antón et al., 2022). MO environments, such as GAMS (GAMS Development Corporation, 2024), AMPL (Fourer et al., 1989) and PYOMO (Bynum et al., 2021), are widely used across multiple application domains and offer in-

terfaces to several optimization algorithms. Typically, simple, less-detailed process models have been adopted in MO environments (Dowling, 2018). PS environments, such as Aspen Plus (AspenTech, 2015) and gPROMS (Siemens, 2024), offer built-in process libraries, initialization strategies, detailed thermodynamic models and a small choice of optimizers. Furthermore, interfacing other optimization algorithms to PS environments is not only cumbersome but also inefficient as the algebraic form of the model equations cannot be accessed.

Rigorous models usually involve a large number of nonlinear equality constraints that are highly coupled. Obtaining a point that is feasible with respect to all the constraints that describe a single process unit is, in itself, challenging. To exacerbate the issue, in the SON, the constraints of all process units are linked by the mixer-splitter network. Superstructure optimization problems, like other non-convex mixed-integer nonlinear programs (MINLPs), are prone to non-convergence in the absence of good initial guesses. In MO environments, several authors have developed process-specific initialization strategies (Barttfeld et al., 2003, Dowling and Biegler, 2015b). Global optimization of these problems is an immense challenge (Burre et al., 2023, Jiang et al., 2019) that is outside the purview of this paper.

In this paper, we address an open challenge in the formulation and solution of superstructure optimization problems. During the optimization of a superstructure, the mass flows into any process unit that is not selected must be constrained to zero. However, the equality constraints that describe a process unit are typically well-defined only at strictly positive input flowrates. At zero flows, a number of numerical challenges arise due to either non-differentiability of the constraints or singular Jacobian matrices (Navarro-Amorós et al., 2014, Yeomans and Grossmann, 2000b). For example, the Jacobian of the mass-balance constraints that describe a separator or mixer can become rank deficient (Dowling and Biegler, 2015b). Additional issues are encountered for the many types of process units that require the solution of phase equilibrium (e.g., a flash unit): the constraints for these process units may be feasible and well-defined only when input flows and their intensive properties allow solutions with the desired (implied) number of phases (Cavalcanti and Barton, 2020, Gopinath et al., 2016, Skiborowski et al., 2015). Finally, mass and heat

transfer correlations, as well as sizing and costing correlations, are often nonlinear functions of flowrates (such as logarithms or powers with exponents less than one) that may be non-differentiable at zero flowrates and lead to ill-conditioned Jacobians when the flowrate approaches zero. Beyond flowrates, constraints on other design variables can cause singularities when the process unit is not selected. For instance, consider a process unit in which the heating load is a design variable that is constrained to be zero when the process unit is not selected. In such a case, zero-valued heating loads may cause singularities in the utility-cost correlations.

To overcome the issue of singularities, several reformulations of the constraints of the process unit have been proposed. To illustrate these reformulations, let us consider a conditional toy process unit with associated binary variable z. It is described by the following model equations,

$$c = \alpha\sqrt{f^{\text{in}}} \tag{1}$$

$$f^{\text{out}} = f^{\text{in}} \tag{2}$$

$$\epsilon z \leq f^{\text{in}} \leq Mz \tag{3}$$

where c is the cost of operating the unit, α, ϵ and M are constants, f^{in} is the mass flowrate into the toy unit and f^{out} is the mass flowrate from the unit. Note that in the SON, f^{in} takes the value zero when $z = 0$. Constraint (3) also forces f^{in} to be strictly positive when $z = 1$. It is easy to see that the derivative of c with respect to f^{in} is undefined at $f^{\text{in}} = 0$ and becomes very large as f^{in} approaches zero.

As the constraints of the process units that are not selected are redundant, the use of "big-M" (Glover, 1975) reformulations to turn on/off process unit equality constraints is customary. However, the application of the standard big-M reformulation to the equality constraints of a process unit does not necessarily address the issue of numerical singularities.

The oft-used big-M reformulation of Equation (1) of the toy process unit yields

$$-M(1-z) \leq c - \alpha\sqrt{f^{\text{in}}} \leq M(1-z) \tag{4a}$$

$$-Mz \leq c \leq Mz. \tag{4b}$$

It is easy to see that derivatives of the constraints (4a) with respect to f^{in} are undefined at $f^{\text{in}} = 0$. In practice, the choice of modelling platform, optimization algorithm and even initial guess can affect the success of optimization with the big-M formulation in cases where the constraints of a deselected conditional unit are singular. To arrive at a reformulation that is guaranteed to be differentiable and free of singularities, one may need to perform additional reformulation steps. For instance, in the toy problem, one may set $f' = f^{\text{in}} + \epsilon(1-z)$, where ϵ is strictly positive, and substitute f' for f^{in} in Equation (4a). The use of big-M reformulation has a few other drawbacks: it entails the modification of model constraints and its use is restricted to MO environments. Moreover, optimization of big-M formulations can be numerically challenging for several reasons, e.g., due to the the addition of large coefficients, the violation of constraint qualifications (Dowling and Biegler, 2015a) and the difficulty in finding valid and tight big-M parameters for all equality constraints (Trespalacios and Grossmann, 2015).

As an alternative to big-M, it is sometimes possible to use a nonlinear reformulation, although this approach is seldom used. A nonlinear reformulation of Equation (1), studied by (Burre et al., 2023)), is

$$c = z\alpha\sqrt{f^{\text{in}}}. \tag{5}$$

The derivative of c with respect to to f^{in} is mathematically undefined at $f^{\text{in}} = 0$, even when this nonlinear reformulation is used. In practice, when $z = 0$, singularities may occur during the evaluation of derivatives. Thus, the success of SO with the nonlinear reformulation is based on choice of solver and optimization platform. Further, the nonlinear reformulation entails the examination and modification of a subset of model constraints.

A systematic and robust alternative is to formulate the problem as a generalized disjunctive program (GDP) (Raman and Grossmann, 1994). The toy process unit may be replaced by the following disjunction:

$$\begin{bmatrix} Z \\ c = \alpha\sqrt{f^{\text{in}}} \\ f^{\text{out}} = f^{\text{in}} \\ \epsilon \leq f^{\text{in}} \end{bmatrix} \vee \begin{bmatrix} \neg Z \\ c = 0 \\ f^{\text{out}} = 0 \\ f^{\text{in}} = 0 \end{bmatrix} \tag{6}$$

where Z is a logical variable associated with the decision to use the toy process unit or not.

Note that logic-based disjunction (6) may be cast as a set of mixed-integer constraints by using either the big-M reformulation which yields (4a) or the convex hull reformulation as given in Lee and Grossmann (2003). Alternatively, SO problems with nonlinear disjunctive constraints, such as Equation (6), may be solved via logic-based outer-approximation (LBOA) (Türkay and Grossmann, 1996) which averts numerical singularities. In the LBOA algorithm, as used in the context of SO, the master problem is first initialized and thereafter primal problems (with process unit selection fixed) and master problems (the solution of which is used to select process units) are solved alternately. Model constraints corresponding to a given process unit are added to the primal problem only if that process unit is selected, resulting in a reduced primal problem. For example, in the LBOA approach, the toy process unit constraints are enforced only when the unit is selected (Z=true which implies f^{in} is strictly positive). The success of this approach has been demonstrated for several problems in MO environments (Yeomans and Grossmann, 2000a,b). However, it has been observed (Caballero, 2015, Caballero et al., 2005, Navarro-Amorós et al., 2014) that simulation-based GDP may entail expensive initialization of the master problem when all the conditional units are identical, e.g., in distillation column synthesis where the conditional units are identical column trays. Further, the solution of each reduced primal requires the generation of a PS flowsheet with only the units that are selected at that

iteration and this requires considerable interfacing between a process simulator and the LBOA algorithm (Navarro-Amorós et al., 2014).

While the reformulations considered until now (big-M, nonlinear reformulation and GDP approaches) are exact, some authors have introduced small non-zero flows to units that are not selected to avert singularities (Lee et al., 2016). However, these are neither exact nor do they guarantee feasibility of the model constraints of the process units that are not selected. In other approaches, continuous reformulations of the problem have been solved (Dowling and Biegler, 2015b, Kossack et al., 2006). The use of surrogate models has also been proposed by several authors (Bugosen et al., 2023, McBride and Sundmacher, 2019), but such approaches seldom provide optimality guarantees (Vollmer et al., 2021). There have been several other works that combine automated flowsheet generation and optimization. This has included early work on the use of genetic algorithms for the design of heat exchanger networks (Androulakis and Venkatasubramanian, 1991) and more recent work using reinforcement learning (Göttl et al., 2022, Reynoso-Donzelli and Ricardez-Sandoval, 2024). Hybrid methods for flowsheet generation that combine data-driven techniques and domain knowledge have also been recently proposed (Mann et al., 2024).

To address the challenges highlighted so far, we propose a new formulation of the superstructure optimization problem. An initial version of this approach, restricted to the case of optimization of pressure in counter-current columns, was briefly introduced in the proceedings of the 2024 Conference on Foundations of Computer-Aided Process Design (Gopinath and Adjiman, 2024). In the current work, a detailed presentation of the proposed modelling framework is given for the case of general process systems and its performance is assessed based on several examples. The approach is based on the introduction of a modified state operator network, MSON, such that no numerical singularities occur when a process unit is deselected. The MSON does not require the modification of process unit constraints. Furthermore, it includes a mathematically well-defined solution to the constraints of deselected units, unlike the Big-M and NL approaches. While the MSON was developed to address shortcomings in existing PS-based and hybrid optimization ap-

proaches, it may also be used in MO. A further advantage of the formulation is that it is amenable to use with black-box/grey box models of process units. The formulation relies on a simple modification of flowsheet mixers and splitters. A fictitious stream is introduced into each mixer such that the input flowrates into process units that are not selected are always strictly positive. In turn, splitters are modified so that the output flowrates that arise from the fictitious input streams are not propagated to the rest of the flowsheet. Our proposed approach provides a systematic framework to assign the states of the fictitious mixer streams such that the feasibility of the equality constraints representing deselected units is assured. The additional computational complexity arising from the MSON is small. The MSON, in its current iteration, is applicable to isobaric flowsheets at fixed pressures.

In Section 2, we formally introduce the MSON and the mixer-splitter network, expanding on the ideas Smith and Pantelides (1995) and defining the notation used to derive a mathematical formulation of the MSON. In Section 3 we investigate the case of flowsheet synthesis and present simple case studies to illustrate the approach, including a flowsheet based on the toy process unit and a flowsheet from the literature. Next, we develop an MSON superstructure for the synthesis of a counter-current column in Section 4 and show the application of the formulation for the synthesis of an absorber for the separation of carbon dioxide from a natural gas stream.

2 The Modified State Operator Network

We propose a new and exact reformulation of the superstructure optimization problem, that does not exhibit any singularities due to a process unit not being selected. It is based on a modified state operator network. Specifically, the MSON-based formulation eliminates those singularities that are due to zero-valued mass flows into a deselected process unit. The MSON approach also avoids numerical singularities that may arise from other specifications in the model of a deselected unit, e.g, a heating load that is set to zero, by eliminating the need for such specifications. The MSON-based problem may be solved via either MINLP or GDP algorithms and the MSON framework is amenable to be used in

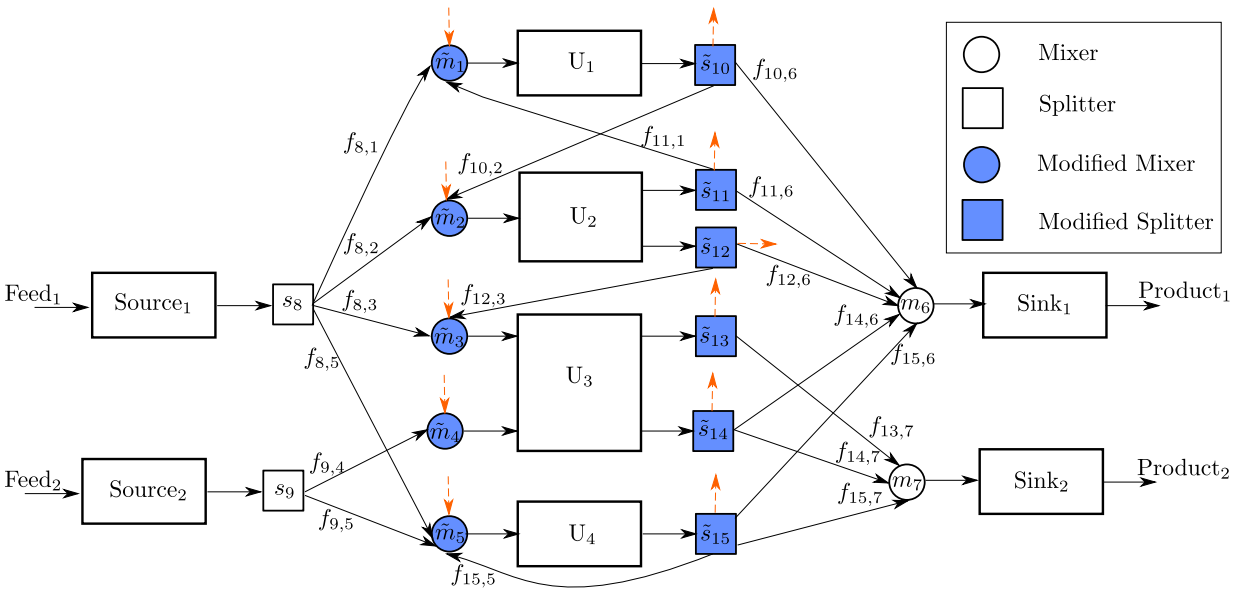


Figure 1: An example SON superstructure shows 2 feeds, 2 products and 4 process units: U_1, U_2, U_3 and U_4. Each feed (product) is followed (preceded) by a source (sink). Key elements of the flowsheet are the set of units $\mathcal{U} = \{\text{Source}_1, \text{Source}_2, \text{Sink}_1, \text{Sink}_2, U_1, U_2, U_3, U_4\}$, the set of process units $\mathcal{L} = \{U_1, U_2, U_3, U_4\}$, the set of sources $\mathcal{F} = \{\text{Source}_1, \text{Source}_2\}$ and the set of sinks $\mathcal{P} = \{\text{Sink}_1, \text{Sink}_2\}$. A set of conceptual mixers (circles) and splitters (squares) connect the units and allow various flowsheet topologies including recycles (e.g., see unit U_4). Each splitter is labelled by the corresponding outlet number. U_1 and U_4 each have one mass inlet and one mass outlet. U_2 has one mass inlet and two mass outlets, whereas U_3 has 2 mass inlets and 2 mass outlets. $\mathcal{IN}_{U_1} = \{1\}$, $\mathcal{IN}_{U_2} = \{2\}$, $\mathcal{IN}_{U_3} = \{3, 4\}$, $\mathcal{IN}_{U_4} = \{5\}$, $\mathcal{IN}_{\text{Sink}_1} = \{6\}$ and $\mathcal{IN}_{\text{Sink}_2} = \{7\}$. $\mathcal{OUT}_{U_1} = \{10\}$, $\mathcal{OUT}_{U_2} = \{11, 12\}$, $\mathcal{OUT}_{U_3} = \{13, 14\}$, $\mathcal{OUT}_{U_4} = \{15\}$, $\mathcal{OUT}_{\text{Source}_1} = \{8\}$ and $\mathcal{OUT}_{\text{Source}_2} = \{9\}$.

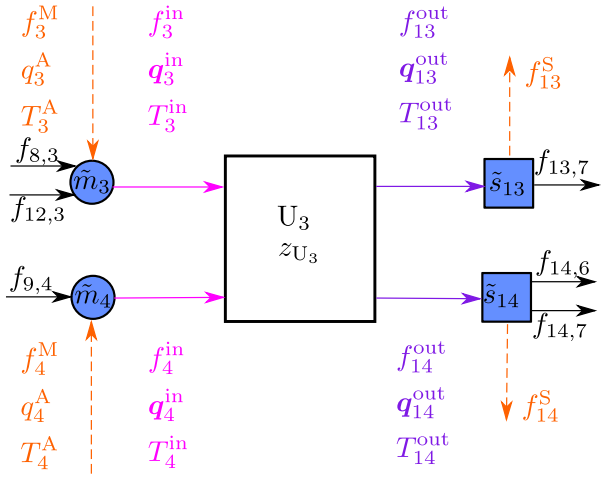


Figure 2: Unit U_3 from Figure 1 with detailed labelling of the streams leaving the splitters and entering the mixers.

both MO and PS environments. The approach is general and easy to implement in that it does not require examining or reformulating any of the equations representing the process units, irrespective of the complexity of the process unit model.

The basic idea behind the MSON approach is simple. We introduce the possibility of bypassing process units through modifications to the representation of mixers and splitters. Process units that are deselected then benefit from fictitious, non-zero, inlet and outlet flows, a feature that prevents numerical errors due to zero-flows. At the level of the flowsheet and corresponding optimization problem, these flows have no impact on the optimality or feasibility of the selected flowsheet structure, thanks to the modified mixer and splitter models and this makes the formulation exact and robust.

In the remainder of this section, we introduce the key sets, variables and equations in the formulation, illustrating this with the flowsheet shown in Figure 1.

2.1 Description of the Modified State Operator Network

2.1.1 Superstructure flowsheet: units and streams

We begin by introducing the relevant notation. Let $\mathcal{M} = \{1, \ldots, m\}$ and define the index set $\mathcal{M}' \subseteq \mathcal{M}$. Consider a vector $\boldsymbol{y} \in \mathbb{R}^m$ where y_i, $i \in \mathcal{M}$, denotes the ith component of $\boldsymbol{y}$. Let $\boldsymbol{y}_{\mathcal{M}'} \in \mathbb{R}^{|\mathcal{M}'|}$ be a sub-vector of $\boldsymbol{y}$ defined as the vector of all the components y_i of $\boldsymbol{y}$ such that $i \in \mathcal{M}'$. For a given stream defined by a subscript i and (optional) superscript j, let $f_i^j \in \mathbb{R}_+$ denote its mass flowrate, $\boldsymbol{q}_i^j \in [0,1]^K$, where K is the number of species in the flowsheet, the vector of mass fractions, T_i^j its temperature and P its pressure (the same for all streams in an isobaric flowsheet). Furthermore, let $\bar{f}$ denote an upper bound on all flowrates. Finally, let M denote an upper bound on the magnitude of any variable ($\bar{f} \leq M$).

Consider the set $\mathcal{L}$ of process units allowed in the flowsheet. We use the term "process unit" to indicate both process equipment (such as an absorber or a condenser) as well as any discrete subset of the process equipment (such as a stage in an absorber). Each process unit $u \in \mathcal{L}$ is described by its rigorous model equations (MESH, sizing, costing). In Figure 1, $\mathcal{L} = \{\mathrm{U}_1, \mathrm{U}_2, \mathrm{U}_3, \mathrm{U}_4\}$. $\mathcal{L}$ is partitioned into the set of conditional (or optional) units $\mathcal{T}$ and permanent units $\mathcal{L} \backslash \mathcal{T}$.

The *feeds* to the flowsheet are all the raw materials and utilities that enter the flowsheet. The *products* are all the streams, including wastes, that leave the flowsheet. Each feed (product) is modelled as a permanent process unit, a *source* (*sink*), with one inlet and one outlet. The sets of all sources and sinks are given by $\mathcal{F}$ and $\mathcal{P}$, respectively. In Figure 1, $\mathcal{F} = \{\mathrm{Source}_1, \mathrm{Source}_2\}$ and $\mathcal{P} = \{\mathrm{Sink}_1, \mathrm{Sink}_2\}$. The set of all "flowsheet units" (including process units, sources and sinks) is given by $\mathcal{U} = \mathcal{L} \cup \mathcal{F} \cup \mathcal{P}$.

Each flowsheet unit $u \in \mathcal{U}$ is assumed to have one or more mass inlets (from where mass enters the unit) and each flowsheet unit $u \in \mathcal{U}$ is assumed to have one or more mass outlets (from where mass leaves the unit). Each inlet (outlet) stream is assigned a unique index, shown as a subscript. The set of all mass inlet indices in $\mathcal{L} \cup \mathcal{P}$ is given by $\mathcal{I}$. The set of all mass outlet indices in $\mathcal{L} \cup \mathcal{F}$ is given by $\mathcal{O}$. In Figure 1, $\mathcal{I} = \{1, 2, 3, 4, 5, 6, 7\}$

and $\mathcal{O} = \{8, 9, 10, 11, 12, 13, 14, 15, 16\}$, with corresponding flowrates thus denoted by f_i^{in}, $i \in \mathcal{I}$ and f_o^{out}, $o \in \mathcal{O}$, respectively. The set of inlets corresponding to each unit $u \in \mathcal{L} \cup \mathcal{P}$ is described by $\mathcal{IN}_u$, where $\mathcal{IN}_u \subset \mathcal{I}$. Similarly, the set of outlets corresponding to each unit $u \in \mathcal{L} \cup \mathcal{F}$ is described by $\mathcal{OUT}_u$, where $\mathcal{OUT}_u \subset \mathcal{O}$. For example, process unit U_3 in Figure 1 has inlet set $\mathcal{IN}_{\text{U}_3} = \{3, 4\}$ and outlet set $\mathcal{OUT}_{\text{U}_3} = \{13, 14\}$.

To enable the representation of alternative flowsheet topologies, a "conceptual mixer" is placed before (at) each mass inlet $i \in \mathcal{IN}_u$, $u \in \mathcal{L} \cup \mathcal{P}$. Similarly, a "conceptual splitter" is located after (at) each outlet $o \in \mathcal{OUT}_u$, $u \in \mathcal{L} \cup \mathcal{F}$. These mixers and splitters are not necessarily physical process units and are not included in sets $\mathcal{L}$ and $\mathcal{U}$, but they enable different process units to be connected or not. When the mixers and splitters are completely connected all possible flowsheet topologies involving the process units are represented by the superstructure. A mixer associated with the inlet i of a permanent unit is known as a standard mixer and labelled m_i whereas a mixer associated with the inlet i' of a conditional unit is known as a modified mixer and labelled $\tilde{m}_{i'}$. Similarly, a splitter that is associated with the outlet o of a permanent unit is known as standard splitter s_o whereas that at outlet o' of a conditional unit is known as modified splitter $s_{o'}$. Streams that flow from outlet o (via splitter s_o or splitter $\tilde{s}_o$) to inlet i (via mixer via mixer m_i or mixer $\tilde{m}_i$) are known as inter-unit streams. A stream that flows from an outlet o (via splitter s_o or splitter $\tilde{s}_o$) into an inlet i (via mixer m_i or mixer $\tilde{m}_i$) is labelled (o, i), with mass flowrate $f_{o,i} \in \mathbb{R}_+$, composition vector $\boldsymbol{q}_{o,i} \in [0, 1]^K$, temperature $T_{o,i}$ and pressure P. An example of a inter-unit stream is stream $f_{8,2}$ that connects splitter s_8 to mixer $\tilde{m}_2$ in Figure 1.

At each inlet $i \in \mathcal{IN}_u$ to process unit u, the mixer (standard or modified) combines $|\mathcal{M}_i|$ inter-unit streams, where $\mathcal{M}_i \subseteq O$ is a set of indices that denote all possible outlets that may be connected (via splitters) to mixer m_i. In addition to the inter-unit streams, a modified mixer also has a fictitious inlet stream, shown with a dashed orange line in Figure 1. The fictitious stream is associated with a set of variables, in a similar way to all other streams, i.e., a mass flowrate f_i^{M}, a constant mass fraction vector $\boldsymbol{q}_i^{\text{A}}$, constant temperature T_i^{A} and constant pressure P. The mixed stream that leaves mixer m_i or $\tilde{m}_i$ is described by the mass flowrate f_i^{in}, mass fraction vector $\boldsymbol{q}_i^{\text{in}}$, temperature T_i^{in} and pressure

P, and enters process unit u.

Analogously, at each outlet $o \in \mathcal{OUT}_u$ from process unit u, a stream whose state is described by mass flowrate f_o^{out}, mass fraction vector $\boldsymbol{q}_o^{\text{out}}$, temperature T_o^{out} and pressure P leaves unit u. This outlet stream enters the splitter s_o or $\tilde{s}_o$ at outlet o. A number $|\mathcal{S}_o|$ of inter-unit streams (with identical temperatures, pressures and compositions), where $\mathcal{S}_o \subseteq I$ is the set of indices that denote all the inlets to which splitter s_o may be connected (via mixers). In addition to the inter-unit streams, a modified splitter also has a fictitious outlet stream, shown with a dashed orange line in Figure 1. The fictitious stream is associated with mass flowrate f_o^{S}, mass fraction vector $\boldsymbol{q}_o^{\text{S}}$, temperature T_o^{S} and pressure P. For example, let us consider conditional process unit U_3 in Figure 1 and Figure 2. The set of outlets that may connect to mixer $\tilde{m}_3$ is described by $\mathcal{M}_3 = \{8, 12\}$ and those that may connect to mixer $\tilde{m}_4$ is given by $\mathcal{M}_4 = \{9\}$. The set of inlets that splitter $\tilde{s}_{13}$ is connected to is given by $\mathcal{S}_{13} = \{7\}$ and those that leave splitter $\tilde{s}_{14}$ may be connected to inlets in $\mathcal{S}_{14} = \{6, 7\}$. The robustness of our formulation arises from the addition of a modified mixer at the inlet of each conditional unit and a modified splitter at its outlet, with the associated fictitious flows. Hence, a splitter can only be connected to the inlet of a process unit indirectly, via a mixer (and correspondingly, a mixer can only be connected to the outlet of a process unit indirectly, via a splitter).

Once either the sets $\mathcal{M}_i$, $\forall i \in \mathcal{I}$, or the sets $\mathcal{S}_o$, $\forall o \in \mathcal{O}$, have been specified, the allowed connectivity of the flowsheet is fully defined. Given that each $i \in \mathcal{I}$ is associated with one flowsheet unit only (similarly for each $o \in \mathcal{O}$), the stream (o, i) corresponds to a pair of units (u, v), such that $i \in \mathcal{IN}_v$, $v \in \mathcal{L} \cup \mathcal{P}$ and $o \in \mathcal{OUT}_u$, $u \in \mathcal{L} \cup \mathcal{F}$. Thus, units u and v are connected, with directionality (u, v), if there is a strictly positive flowrate between a splitter at an outlet of process unit u and a mixer at an inlet of unit v. In Figure 1, U_1 is connected to U_2, and feeds into it, if flowrate $f_{10,2}$ is strictly positive. Similarly, U_2 is connected to U_1, and feeds into it, if flowrate $f_{11,1}$ is strictly positive.

The sets introduced in this section are summarized in Table 1.

Set	Description
$\mathcal{L}$	Process units in superstructure
$\mathcal{T}$	Conditional process units in superstructure
$\mathcal{F}$	Sources in superstructure
$\mathcal{P}$	Sinks in superstructure
$\mathcal{U}$	$\mathcal{L} \cup \mathcal{F} \cup \mathcal{P}$
$\mathcal{I}$	Mass inlet indices of all process units $u \in \mathcal{L} \cup \mathcal{P}$
$\mathcal{O}$	Mass outlet indices of all process units $u \in \mathcal{L} \cup \mathcal{F}$
$\mathcal{IN}_u$	Mass inlet indices corresponding to a unit u, where $u \in \mathcal{L} \cup \mathcal{P}$
$\mathcal{OUT}_u$	Mass outlet indices corresponding to a unit u, where $u \in \mathcal{L} \cup \mathcal{F}$
$\mathcal{M}_i$	Mass outlet indices that are connected to mixer m_i or $\tilde{m}_i$
$\mathcal{S}_o$	Mass inlet indices that are connected to splitter s_o or $\tilde{s}_o$

Table 1: Summary of sets that define the superstructure flowsheet

2.1.2 Variable vectors and sets

A vector of binary variables $\boldsymbol{z}$, of dimensionality $|\mathcal{T}|$ is defined to represent unit selection / deselection. The binary variable z_u is thus associated with each conditional process unit $u \in \mathcal{T}$ to indicate its presence in the process ($z_u = 1$) or its absence from the process ($z_u = 0$). In addition, any further discrete design variables in the flowsheet (such as feed-stage location for a distillation column) is included in a vector $\boldsymbol{y} \in \{0, 1\}^r$.

We define a vector of continuous process variables denoted by $\boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{R}^n$. Several types of variables can be distinguished within $\boldsymbol{x}$. The continuous variables that appear in the model equations for a process unit u are referred to as process unit-level variables. Within these we also define a (usually small) subset of "output variables" corresponding to process unit u. These are the process unit-level variable that appear either in the objective function or in at least one flowsheet-level constraint, i.e., the constraints that are used to define flowsheet-wide performance metrics (e.g., the total annualized cost or total energy demand). The flowsheet-level constraints are not unit-specific but a subset of these depend on some variables (known as output variables) corresponding to one or more process units. That is, the output variables are visible at the level of the flowsheet model. The value of an output variable is usually obtained by solving the process unit-level constraints that describe unit u. Examples of output variables include the cost c_u of unit u (which is used

Set	Description
$\mathcal{N}$	Indices of continuous variables, $\mathcal{N} = \{1, \ldots, n\}$
$\mathcal{Y}$	Indices of discrete design variables, $\mathcal{Y} = \{1, \ldots, r\}$
$\mathcal{D} \subseteq \mathcal{N}$	Indices of continuous design (independent) variables e.g., reboiler duty in a distillation column e.g., inter-unit flowrate $f_{o,i}$ between splitter s_o and mixer m_i
$\mathcal{A}_u \subseteq \mathcal{D}$	Indices of continuous design variables that correspond to unit $u \in \mathcal{L}$ only e.g., reboiler duty of the distillation column u is indexed by $\mathcal{A}_u$
$\mathcal{Y}_u \subset \mathcal{Y}$	Indices of discrete design variables that correspond to unit $u \in \mathcal{L}$ only
$\mathcal{C}_u \subseteq \mathcal{N}$	Indices of output variables that correspond to unit $u \in \mathcal{L}$ only
$\mathcal{C} \subseteq \mathcal{N}$	Indices of output variables corresponding to all conditional units $\mathcal{C} = \cup_{u \in \mathcal{T}} \mathcal{C}_u$

Table 2: Index sets of the variables in the MSON formulation

to compute a flowsheet-wide cost) and its cooling duty, Q_u. In addition, vector $\boldsymbol{x}$ also contains variables that only appear in equations at the flowsheet level, and not in models of any process units. For example, this includes variables obtained from the outputs of multiple units, such as a total capital cost, as well as inter-unit variables $f_{(i,o)}$. In addition, we define an auxiliary vector $\boldsymbol{x}^{\mathrm{S}} \in \mathbb{R}^n$ of modified continuous variables. Each component of $\boldsymbol{x}^{\mathrm{S}}$ and $\boldsymbol{x}$ describes the same variable. However, when x_i is an output variable of a conditional unit u, x_i^{S} and x_i can take different values reflecting whether the unit u is present or not in the flowsheet as described in 2.2.4.

Lastly, we define the sets that correspond to the variables in the MSON, including the design variables and output variables, in Table 2. Furthermore, we note that $\mathcal{C}_u$ and $\mathcal{A}_u$ may have elements in common so that it is possible that $\mathcal{C}_u \cap \mathcal{A}_u \neq 0$.

2.2 MSON model equations

The following assumptions are made in deriving the MSON equations:

1. The process is assumed to be isobaric at a constant pressure P that is fixed *a priori*.

2. All flows are assumed to be frictionless and isenthalpic.

3. Chemical reactions do not occur in mixers or splitters.

4. All units are at steady-state.

5. The standard and modified mixers and splitters introduced for the purpose dealing with conditional units are isenthalpic (i.e., they are adiabatic and shaft work, e.g., work of mixing, is zero).

6. The total mass flow into each sink is strictly positive.

2.2.1 Flow positivity constraints

When a conditional unit u is not selected to be in the flowsheet, flow-validity constraints ensure that all inter-unit flows directed to this unit are zero. The flow-validity constraints

$$f_{o,i} \leq \bar{f} z_u \qquad \forall o \in \mathcal{M}_i,\, \forall i \in \mathcal{IN}_u,\, \forall u \in \mathcal{T} \tag{7}$$

thus set the flow from splitter s_o or $\tilde{s}_o$ to mixer $\tilde{m}_i$ to zero (as $f_{o,i}$ can only take positive values) if mixer $\tilde{m}_i$ is associated with a unit u that is deselected. In Figure 2, when U_3 is not selected, the flow-validity constraints drive flowrates $f_{8,3}$, $f_{12,3}$ and $f_{9,4}$ to zero.

We also enforce that the inlet flowrates into all process units are strictly positive.

$$\epsilon \leq f_i^{\text{in}} \qquad \forall i \in \mathcal{IN}_u, \forall u \in \mathcal{L} \tag{8}$$

It is evident that both a permanent unit and a selected conditional unit should have strictly positive inlet mass flows. However, we go further by imposing the positivity of inlet flowrates for all process units in the MSON, including those that are deselected. We explain this choice further in the next subsection.

2.2.2 Mixers in the MSON

The constraints of both a standard mixer (Figure 3a) and the modified mixer that is introduced in this work (Figure 3b) are shown in this section.

Standard mixer For each standard mixer m_i, $i \in \mathcal{IN}_u, u \in (\mathcal{L} \setminus \mathcal{T}) \cup \mathcal{P}$, we have

$$f_i^{\text{in}} = \sum_{o \in \mathcal{M}_i} f_{o,i} \tag{9a}$$

$$f_i^{\text{in}} q_{i,c}^{\text{in}} = \sum_{o \in \mathcal{M}_i} f_{o,i} q_{o,i,c} \qquad \forall c \in \{1, \ldots, K\} \tag{9b}$$

$$f_i^{\text{in}} h_{en}(T_i^{\text{in}}, \text{P}, \boldsymbol{q}_i^{\text{in}}) = \sum_{o \in \mathcal{M}_i} f_{o,i} h_{en}(T_{o,i}, \text{P}, \boldsymbol{q}_{o,i}) \tag{9c}$$

where $h_{en} : \mathbb{R}_+^{2+K} \to \mathbb{R}$ is the enthalpy (relative to a standard) per unit mass as a function of temperature T, pressure P and composition on a mass basis $\boldsymbol{q}$. Equations (9a)-(9c) describe the overall mass, component-mass and energy balances. It is assumed that the constraints $\sum_{c=1}^{K} q_{o,i,c} = 1, \forall o \in \mathcal{M}_i$, are satisfied due to mass balance equations on other units or input specifications from which it follows from Equations (9a) and (9b) that $\sum_{c=1}^{K} q_{i,c}^{\text{in}} = 1$. Thanks to Equation (8), the flow at the outlet of the standard mixer is strictly positive, irrespective of whether it is associated with a permanent process unit or a sink.

Modified mixer We now present the modified mixer at each inlet i, $\forall i \in \mathcal{IN}_u$, $\forall u \in \mathcal{T}$.

When a unit $u \in \mathcal{T}$ is deselected, inter-unit flows into the unit are driven to zero by the flow validity constraint (7). To avoid any issues with the solution of the process-unit level equality constraints when unit $u \in \mathcal{T}$ is deselected, the flowrate of the fictitious stream takes on a pre-defined, non-zero, value f_i^{A}, and corresponding mass fraction vector $\boldsymbol{q}_i^{\text{A}}$, temperature T_i^{A} and pressure P. When the unit is selected, the flowrate of the fictitious stream is forced to zero by virtue of Equation (10d) and the fictitious stream has no impact on the mixer.

The corresponding model equations for each modified mixer $\tilde{m}_i$, $\forall i \in \mathcal{IN}_u$, $\forall u \in \mathcal{T}$

are:

$$f_i^{\text{in}} = \sum_{o \in \mathcal{M}_i} f_{o,i} + f_i^{\text{M}} \tag{10a}$$

$$f_i^{\text{in}} q_{i,c}^{\text{in}} = \sum_{o \in \mathcal{M}_i} f_{o,i} q_{o,c} + f_i^{\text{M}} q_{i,c}^{\text{A}}, \qquad \forall c \in \{1, \ldots, K\} \tag{10b}$$

$$f_i^{\text{in}} h_{en}(T_i^{\text{in}}, \text{P}, \boldsymbol{q}_i^{\text{in}}) = \sum_{o \in \mathcal{M}_i} f_{o,i} h_{en}(T_{o,i}, \text{P}, \boldsymbol{q}_{o,i}) + f_i^{\text{M}} h_{en}(T_i^{\text{A}}, \text{P}, \boldsymbol{q}_i^{\text{A}}) \tag{10c}$$

$$f_i^{\text{M}} = f_i^{\text{A}}(1 - z_u) \tag{10d}$$

In addition, we set the continuous and discrete design variables corresponding to unit u, $\boldsymbol{x}_{\mathcal{A}_u}$ and $\boldsymbol{y}_{\mathcal{A}_u}$, to the constant vectors $\boldsymbol{x}_{\mathcal{A}_u}^{\text{A}}$ and $\boldsymbol{y}_{\mathcal{A}_u}^{\text{A}}$, respectively:

$$-Mz_u \leq \boldsymbol{x}_{\mathcal{A}_u} - \boldsymbol{x}_{\mathcal{A}_u}^{\text{A}} \leq Mz_u \tag{11a}$$

$$-Mz_u \leq \boldsymbol{y}_{\mathcal{A}_u} - \boldsymbol{y}_{\mathcal{A}_u}^{\text{A}} \leq Mz_u \tag{11b}$$

where f_i^{A}, $\boldsymbol{q}_i^{\text{A}}$, T_i^{A}, $\forall i \in \mathcal{IN}_u$, $\boldsymbol{x}_{\mathcal{A}_u}^{\text{A}}$ and $\boldsymbol{y}_{\mathcal{A}_u}^{\text{A}}$ are user-defined constants. These can be assigned any values that ensure that the equality constraints of unit u are feasible, that the desired number of phases are obtained at the solution of the unit and that $f_i^{\text{A}} \geq \epsilon$, $\forall i \in \mathcal{IN}_u$. Thus, the flows at the inlet of the unit are strictly positive and satisfy Equation (8), even when the associated unit is deselected. If the states (flowrate, temperature and composition) at any of the inlets of unit u are design variables (that is, indexed by set $\mathcal{A}_u$), then the assignment of these constants must be self-consistent. The independent variables of the deselected unit are assigned values in the MSON such that singularities of the equations of the unit are guaranteed not to occur. Thus, when a process unit is deselected, the feasibility of the unit model equations and the correctness of the phase behaviour are guaranteed thanks to the introduction of the modified mixer.

(a) Standard mixer

(b) Modified Mixer

Figure 3: Schematic of (a) a standard mixer m_i with its inlet and outlet streams and (b) a modified mixer $\tilde{m}_i$ with its inlet and outlet streams. A fictitious stream f_i^{M} denoted by the dashed arrow (in orange) has been added to the modified mixer.

2.2.3 Splitters in the MSON

We next show the equations of a standard splitter (Figure 4a) as well those of the modified splitters introduced in the MSON (Figure 4b).

Standard splitter For each standard splitter s_o, $\forall o \in \mathcal{OUT}_u$, $\forall u \in (\mathcal{L} \setminus \mathcal{T}) \cup \mathcal{F}$ we have

$$f_o^{\text{out}} = \sum_{i \in \mathcal{S}_o} f_{o,i} \tag{12a}$$

$$T_o^{\text{out}} = T_{o,i} \qquad \forall i \in \mathcal{S}_o \tag{12b}$$

$$q_{o,c}^{\text{out}} = q_{o,i,c} \qquad \forall i \in \mathcal{S}_o, \quad \forall c \in \{1, \ldots, K\}, \tag{12c}$$

where $\boldsymbol{q}_{o,i}^{out} \in [0,1]^K,\ \forall i \in \mathcal{S}_o$ and $\boldsymbol{q}_o^{out} \in [0,1]^K$. The splitter equations set the composition, temperature and pressure of each stream leaving the splitter equal to that at the inlet of the splitter.

Modified splitter The modified mixer ensures that flowrates at the inlets of conditional units are strictly non-zero but this results in non-zero values of the flowrates as well as other output variables at the outlets of units that are not selected. In addition, due to Equation (11a), all other unit-specific independent variables are assigned to non-zero values. Without modifications to the splitter, this would result in an erroneous flowsheet model. In order to correct for this, we make use of the fictitious stream that leaves the

modified splitter.

For each modified splitter $\tilde{s}_o$, $\forall o \in \mathcal{OUT}_u$, $\forall u \in \mathcal{T}$ we have

$$f_o^{\text{out}} = \sum_{i \in \mathcal{S}_o} f_{o,i} + f_o^{\text{S}} \tag{13a}$$

$$T_o^{\text{out}} = T_{o,i}, \qquad \forall i \in \mathcal{S}_o, \tag{13b}$$

$$q_{o,c}^{\text{out}} = q_{o,i,c} \qquad \forall i \in \mathcal{S}_o. \quad \forall c \in \{1, \ldots, K\} \tag{13c}$$

$$T_o^{\text{out}} = T_o^{\text{S}} \tag{13d}$$

$$q_{o,c}^{\text{out}} = q_{o,c}^{\text{S}} \qquad \forall c \in \{1, \ldots, K\} \tag{13e}$$

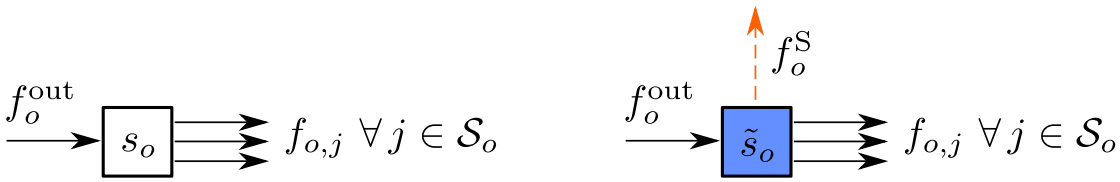
We further ensure that $f_o^{\text{S}} = f_o^{\text{out}}$ when $z_u = 0$ as follows:

$$f_o^{\text{out}} - f_o^{\text{S}} \leq \bar{f} z_u \tag{14}$$

However, when $z_u = 1$, f_o^S takes the value of zero:

$$0 \leq f_o^{\text{S}} \leq \bar{f}(1 - z_u) \tag{15}$$

Hence, no mass flow is propagated from process unit u to other units when $z_u = 0$. We also note that the mass, composition and enthalpy balances are conserved for the process unit both when the unit is selected and deselected. The indicator constraints (14)-(15) have been given in big-M form here, via the use of the upper bound on flowrates, $\bar{f}$. We re-emphasize that the big-M form is only used on the output flowrates and output variables of the splitter and the constraints of the process unit itself are unchanged. Lastly, the constraints (14)-(15) may be expressed by equivalent disjunctions, convex hull reformulations and so on (see Appendix A).



(a) Standard splitter

(b) Modified Splitter

Figure 4: Schematic of (a) a standard splitter s_o with associated inlet and outlet streams and (b) a modified splitter $\tilde{s}_o$ with associated inlet and outlet streams, including a fictitious stream f_o^{S} denoted by the dashed arrow (in orange).

2.2.4 Logical relations for continuous variables

In the MSON formulation, deselected units, with their fictitious inlet and outlet streams and fixed design variables, can themselves be viewed as fictitious. The key requirement for superstructure optimization is that such units should not impact on flowsheet-level metrics and decisions. In this section we show the constraints that drive the contributions from deselected units to zero in the MSON objective and flowsheet-level constraints. Specifically, consider a conditional unit $u, u \in \mathcal{T}$, that is characterized by output variables $x_i, i \in \mathcal{C}_u$. We set the value of the corresponding modified output variables $x_i^{\text{S}}, i \in \mathcal{C}_u$, to a constant vector $\boldsymbol{c}_{u_i}$ (typically of zeros) when unit u is deselected and to $x_i, i \in \mathcal{C}_u$, when it is selected. This is expressed as:

$$-M(1-z_u) \leq x_i - x_i^{\text{S}} \leq M(1-z_u) \qquad \forall i \in \mathcal{C}_u, \quad \forall u \in \mathcal{T}, \tag{16}$$

$$-Mz_u \leq x_i^{\text{S}} - c_{u_i} \leq Mz_u \qquad \forall i \in \mathcal{C}_u, \quad \forall u \in \mathcal{T}. \tag{17}$$

As previously, alternatives to the big-M formulation can be used instead of Equations (16) and (17) (see Appendix A).

For all other components of variable vector $\boldsymbol{x}$ that do not represent an output variable linked to a conditional unit, i.e., $x_i, i \in \mathcal{N} \setminus \mathcal{C}$, the values of $\boldsymbol{x}^{\text{S}}$ and $\boldsymbol{x}$ are set equal:

$$x_i^{\text{S}} = x_i, \qquad \forall i \in \mathcal{N} \backslash \mathcal{C}. \tag{18}$$

Constraint class	Description
Flow positivity	Ensures zero inter-unit flows into deselected units Equation (7) Ensures flows into all units are strictly positive Equation (8)
Mixer / splitter	Equations for conceptual mixers and splitters (9a)-(15)
Process unit-level equalities for unit u	MESH equations and sizing and costing equations Imposed for each unit $u \in \mathcal{U}$
Process unit-level inequalities for unit u	Impose requirements on the performance of u Imposed for each unit $u \in \mathcal{U}$
Flowsheet-level equalities	Equations to define flowsheet-wide metrics Usually depend on multiple process units and/or flowsheet-wide quantities, e.g., constraints to compute the total annualized cost
Flowsheet-level inequalities	Impose requirements on flowsheet-wide metrics e.g., a constraint on the energy demand of the flowsheet
Unit-selection	Logical relationships related to selection of process units e.g., at least one reactor must be selected
Bounds	Lower and upper bounds on $\boldsymbol{x}$

Table 3: Constraint classes used in the MSON formulation

The constraints (18) are shown for ease of exposition only. In practice, we use identity elimination to remove variables $\boldsymbol{x}^{\mathrm{S}}_{\mathcal{N}\setminus\mathcal{C}}$ from the problem formulation and thus do not unnecessarily increase problem dimensionality.

Finally, the objective function F and the flowsheet-level equality and inequality constraints (see Table 3), $\boldsymbol{h}_p$ and $\boldsymbol{g}_p$ respectively, are all expressed in terms of the modified continuous variables $\boldsymbol{x}^{\mathrm{S}}$, rather than in terms of the variables $\boldsymbol{x}$.

2.3 MSON formulation

The MSON framework is used to address the following problem: Given the sets of process units $\mathcal{L}$ and $\mathcal{T}$, feeds $\mathcal{F}$ and products $\mathcal{P}$, inlets $\mathcal{I}$, outlets $\mathcal{O}$ and either set $\mathcal{M}_i$, $\forall i \in \mathcal{I}$ or set $\mathcal{S}_o$, $\forall o \in \mathcal{O}$ that allows connectivity between flowsheet units $\mathcal{U}$, determine: i) the set of conditional units, $\mathcal{T}^* \subseteq \mathcal{T}$, to be present in the flowsheet, ii) the value of the flowrates $f_{o,i}$, $\forall o \in \mathcal{M}_i$, $\forall i \in \mathcal{I}$ and iii) the values of all other design variables, so that the process is feasible and the flowsheet objective is minimized.

The problem formulation includes constraints of different types as summarized in Table 3. The corresponding MSON formulation is summarized in Problem (MSON) defined

by Equations (19a)-(19v):

$$\min_{\boldsymbol{x},\boldsymbol{x}^{\mathrm{S}},\boldsymbol{y},\boldsymbol{z}} F(\boldsymbol{x}^{\mathrm{S}}) \tag{19a}$$

$$\text{s.t. } \boldsymbol{h}_p(\boldsymbol{x}^{\mathrm{S}}) = \boldsymbol{0} \tag{19b}$$

$$\boldsymbol{g}_p(\boldsymbol{x}^{\mathrm{S}}) \leq \boldsymbol{0} \tag{19c}$$

$$\boldsymbol{h}_u(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0} \qquad \forall u \in \mathcal{U} \tag{19d}$$

$$\boldsymbol{g}_u(\boldsymbol{x},\boldsymbol{y}) \leq \boldsymbol{0} \qquad \forall u \in \mathcal{U}\backslash\mathcal{T} \tag{19e}$$

$$\boldsymbol{g}_u(\boldsymbol{x},\boldsymbol{y}) \leq M(1-z_u) \qquad \forall u \in \mathcal{T} \tag{19f}$$

$$\epsilon \leq f_i^{\mathrm{in}} \qquad \forall i \in \mathcal{IN}_u, u \in \mathcal{L}\backslash\mathcal{T} \tag{19g}$$

$$f_{o,i} \leq \bar{f} z_u \qquad \forall o \in \mathcal{M}_i,\ \forall i \in \mathcal{IN}_u,\ \forall u \in \mathcal{T} \tag{19h}$$

$$\text{Standard Mixer Eqs (9)} \qquad \forall i \in \mathcal{IN}_u,\ \forall u \in (\mathcal{L} \setminus \mathcal{T}) \cup \mathcal{P} \tag{19i}$$

$$\text{Standard Splitter Eqs (12)} \qquad \forall i \in \mathcal{OUT}_u,\ \forall u \in (\mathcal{L} \setminus \mathcal{T}) \cup \mathcal{F} \tag{19j}$$

$$\text{Modified Mixer Eqs (10a)} - \text{(10c)} \qquad \forall i \in \mathcal{IN}_u,\ \forall u \in \mathcal{T} \tag{19k}$$

$$\text{Modified Mixer Conditional Eqs (10d), (11)} \qquad \forall i \in \mathcal{IN}_u,\ \forall u \in \mathcal{T} \tag{19l}$$

$$\text{Modified Splitter Eqs (13)} \qquad \forall i \in \mathcal{OUT}_u,\ \forall u \in \mathcal{T} \tag{19m}$$

$$\text{Modified Splitter Conditional Eqs (14), (15)} \qquad \forall i \in \mathcal{OUT}_u,\ \forall u \in \mathcal{T} \tag{19n}$$

$$\boldsymbol{x}^{\mathrm{S}} \text{ Assignment Eqs (16), (17)} \qquad \forall k \in \mathcal{T} \tag{19o}$$

$$\boldsymbol{x}^{\mathrm{S}} \text{ Assignment Eqs (18)} \qquad \forall i \in \mathcal{N} \setminus \mathcal{C} \tag{19p}$$

$$Q_y \boldsymbol{y} + Q_z \boldsymbol{z} \leq \boldsymbol{d} \tag{19q}$$

$$\boldsymbol{x} \in X \subseteq \mathbb{R}^n, \boldsymbol{x}^{\mathrm{S}} \in X \subseteq \mathbb{R}^n \tag{19r}$$

$$\boldsymbol{y} \in \{0,1\}^{|\mathcal{Y}|}, \boldsymbol{z} \in \{0,1\}^{|\mathcal{T}|} \tag{19s}$$

$$f_i^{\mathrm{M}} \in \mathbb{R}_+ \qquad \forall i \in \mathcal{IN}_u,\ \forall u \in \mathcal{T} \tag{19t}$$

$$f_i^{\mathrm{S}} \in \mathbb{R}_+ \qquad \forall i \in \mathcal{OUT}_u,\ \forall u \in \mathcal{T} \tag{19u}$$

$$f_{o,i} \in \mathbb{R}_+ \qquad \forall o \in \mathcal{M}_i,\ \forall i \in \mathcal{I} \tag{19v}$$

where $F : \mathbb{R}^n \to \mathbb{R}$ denotes a flowsheet-level objective to be minimized, $\boldsymbol{h_p}(\boldsymbol{x}^{\mathrm{S}})$ represents

flowsheet-level equality constraints (e.g., a total capital cost), $\boldsymbol{g_p}(\boldsymbol{x}^{\mathrm{S}})$ represents flowsheet-level inequality constraints (e.g., a constraint on the maximum allowable total heat duty of a flowsheet), $\boldsymbol{h}_u(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{g}_u(\boldsymbol{x}, \boldsymbol{y})$ denote the process unit-level equality and inequality constraints for unit u, respectively. Equation (19f) enforces the process-unit level inequality constraints only when the unit is selected.

Equation (19g) constrains the flows at the inlets of both permanent and conditional process units to be strictly positive (given the constant $\epsilon > 0$). Equation (19h) drives the mass flows into deselected units to be zero. The problem solution must also respect the flow-validity constraints and equality constraints of the mixer-splitter network as represented by Equations (19i)-(19n). The behavior of the modified vector of continuous variables $\boldsymbol{x}^{\mathrm{S}}$ is governed by Equations (19o) and (19p). Equation (19q) denotes linear relationships between the discrete variables and can be used to enforce unit-selection constraints, with Q_y and Q_z as constant matrices and $\boldsymbol{d}$ a real vector. These constraints include Big-M type inequalities that are used to enforce disjunctive logic related to the selection of units, e.g., a logic expression such as "if process unit U_1 is selected, then process units U_2 and U_3 cannot be selected". Note that the equality constraints that describe each process unit are not modified in this reformulation.

To illustrate the effect of the MSON on a process unit and flows to and from the unit, we consider example unit U_3 in Figure 2. Figures 5(a) and 5(b) show the constraints on the flowrates, input variables and output variables when $z_{\mathrm{S}_1} = 1$ and when $z_{\mathrm{S}_1} = 0$, respectively. In Figure 5(a), at any feasible solution to the MSON, flows f_3^{M} and f_4^{M} are zero. Similarly, the fictitious flows leaving the splitters, f_{13}^{S} and f_{14}^{S}, are zero. The variables $\boldsymbol{x}^{\mathrm{S}}_{\mathcal{C}_{\mathrm{U}_3}}$ are set equal to the output variables $\boldsymbol{x}_{\mathcal{C}_{\mathrm{U}_3}}$ computed for the unit. In Figure 5(b), at any feasible solution to the MSON, flows f_3^{M} and f_4^{M} are strictly positive and equal to f_3^{A} and f_4^{A}, respectively. The variables $\boldsymbol{x}_{\mathcal{A}_{\mathrm{U}_3}}$ are set equal to $\boldsymbol{x}^{\mathrm{A}}_{\mathcal{A}_{\mathrm{U}_3}}$ so that the model equations for unit U_3 are satisfied and no singularities are encountered. The fictitious flows leaving the splitters, f_{13}^{S} and f_{14}^{S}, are strictly positive. No mass is propagated to other units since flows $f_{13,7}$, $f_{14,6}$ and $f_{14,7}$ are all equal to zero. The variables $\boldsymbol{x}^{\mathrm{S}}_{\mathcal{C}_{\mathrm{U}_3}}$ are set to $\boldsymbol{c}_{\mathrm{U}_3}$.

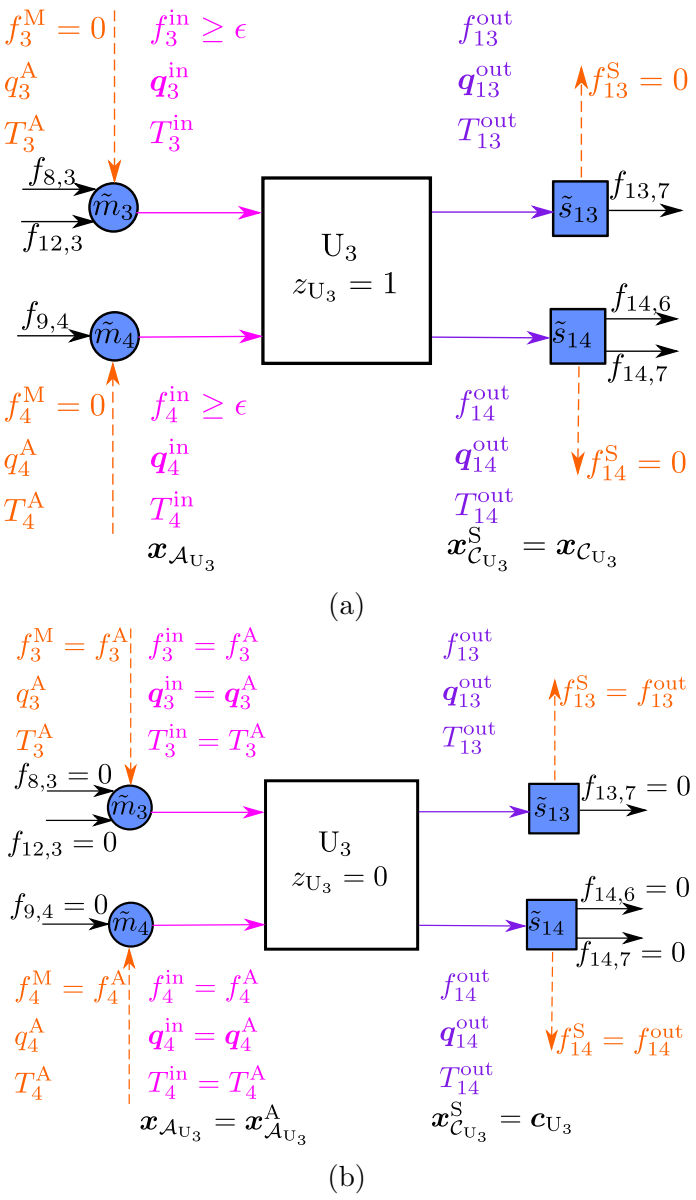


Figure 5: (a) Unit U_3 and associated variables when $z_{\mathrm{U}_3} = 1$; (b) Unit U_3 and associated variables when $z_{\mathrm{U}_3} = 0$. In this case, the flowrates f_{13}^{out} and f_{14}^{out} are guaranteed to be strictly positive, thanks to the values assigned to the fictitious inlet streams and the unit-specific design variables.

It is important to note that the MSON framework only addresses issues related to zero-valued input flows that arise when a process unit is deselected. However, in process simulation and optimization, flows of streams within units (e.g., the vapour flowrate from a

stage) can sometimes take zero value for some assignments of the design variables, usually due to a failure of the phase-equilibrium assumptions (Raghunathan and Biegler, 2003). Numerical singularities that arise in these situations are not mitigated by the use of the MSON.

The partitioning of process units in the MSON superstructure into conditional or permanent is based on problem-specific considerations. For example, when a reaction in a plug flow reactor is to be followed by a downstream separation whose structure is to be determined, the reactor is a permanent unit and the separators (each of which may utilize different separation technologies) are conditional.

In summary, to use the MSON, a modeller needs to have knowledge of the inputs and outputs of each conditional process unit. That is, for each conditional process unit $u \in \mathcal{T}$, the modeller needs to know i) the sets $\mathcal{IN}_u$, $\mathcal{OUT}_u$ that define the mass inlets and outlets; ii) the sets $\mathcal{A}_u$ and $\mathcal{Y}_u$ that index the continuous and discrete design variables, respectively, for each process unit; iii) the set $\mathcal{C}_u$ that indexes the output variables from each process unit; and iv) an assignment of the design variables, denoted by constant vectors $\boldsymbol{x}^{\mathrm{A}}_{\mathcal{A}_u}$and $\boldsymbol{y}^{\mathrm{A}}_{\mathcal{Y}_u}$, and of the states at the inlets, denoted by f_i^{A}, $\boldsymbol{q}_i^{\mathrm{A}}$, T_i^{A} $\forall i \in \mathcal{IN}_u$. This assignment must be feasible with respect to the equality constraints of the process unit, it must meet any user-specified lower bound on flowrates, and it must result in the expected phase behaviour. This latter requirement should be easy to satisfy and is important for the robustness and computational efficiency of the MSON-based problem solution. The flowsheet-wide constraints and the objective function need to be expressed in terms of the auxiliary variables $\boldsymbol{x}^{\mathrm{S}}$ that are introduced in the MSON. In contrast to the MSON, units in the SON are connected by standard mixers and splitters only and the flowrates to deselected units are zero in the latter.

2.4 Implementation: The SO-PS Algorithm

The behaviour of the MSON reformulation, Problem (MSON), is investigated via an in-house C++ implementation of a logic-based variant of the outer-approximation with equal-

ity relaxation and augmented penalty (OA-ER-AP) algorithm (Kocis and Grossmann, 1987, Viswanathan and Grossmann, 1990), the SO-PS algorithm, derived by modifying the algorithm described in Gopinath et al. (2016). The use of an in-house implementation provides maximum flexibility to explore different formulations and initialization strategies.

The optimization algorithm involves the solution of a continuous primal problem and a mixed-integer master problem. The primal problem, a continuous nonlinear program (NLP) derived by fixing the discrete variables in the MINLP Problem (MSON), is implemented and solved in the modelling environment gPROMS 7.1 (Siemens, 2024). The master problem is a mixed integer linear programming problem which is solved using Gurobi 10.0.2 (Gurobi Optimization, Inc., 2024) via the C++ API.

The primal problem is solved in gPROMS using a sequential quadratic programming (SQP) solver. In the primal problem, the values of the binary variables and all other discrete variables are fixed *a priori*. The NLP solver takes a feasible path approach with respect to a subset of the SO problem constraints, namely, those equality constraints that are typically solved by the use of a variant of Newton's method. We refer to these constraints as simulator-only constraints and they are treated implicitly by the optimizer. In our implementation the flowsheet-level equality constraints (19b), the process unit-level equality constraints (Equation(19d)), the mixer equality constraints (Equations (19i), (19k)) and the splitter equality constraints (Equations (19j), (19m)) are solved as simulator-only constraints. The remaining constraints in Problem (MSON) are referred to as optimization-only constraints. The variables chosen as degrees of freedom are declared as optimization variables $\boldsymbol{w}$ and the objective function and optimization-only constraints are treated as functions of the optimization variables. First-order analytical derivatives of the objective function and optimization-only constraints with respect to the optimization degrees of freedom are computed by gPROMS using symbolic manipulations. These are used in the solution of the primal problem as well as to construct the master problem, which includes linearizations of the objective function and optimization-only constraints.

We also note that the simulator model (or flowsheet) remains unchanged at each iteration of the algorithm, except for a change in the vector $\boldsymbol{z}^{(k)}$ which is passed to gPROMS via

the gO:RUN functionality. We also modify the heuristic stopping criteria of the LB-OA-ER-AP algorithm, akin to Bowskill et al. (2020). The algorithm converges to a solution when whichever of the following occurs: i) in three of the iterations the primal problem is feasible and its objective function is greater than the upper bound; ii) the master problem is infeasible and iii) the total number of iterations exceeds 100. Full details of our implementation may be found in the Supporting information.

3 Flowsheet synthesis

Two flowsheet synthesis problems are considered. A synthesis problem featuring the toy unit presented in the introduction of the paper is first studied. The simplicity of the problem enables benchmarking of several problem formulation and solution approaches. We next study a flowsheet synthesis problem from the literature.

3.1 Case study 1: Toy problem

To illustrate the MSON concept and solution, we first consider a flowsheet synthesis problem featuring the toy unit that was presented in the introduction of the paper. The problem is stated as follows:

Given a feed with mass flowrate F and two identical toy units, labelled 1 *and* 2 *and described by Equation* (1)*, select either one or both units (the units may be connected in series or parallel or completely connected), so that the sum of the costs of both units is minimized. Further, if a toy unit is selected, the mass flowrate into the unit must be greater than a positive constant ϵ.*

While the SO problem considered here is trivial, we study it to illustrate the MSON formulation relative to other approaches. We test the numerical performance of different approaches to solving the toy problem, using a range of solvers in both MO and PS environments. We also note that, coincidentally, a similar problem has been recently studied in Burre et al. (2023) in the context of global optimization.

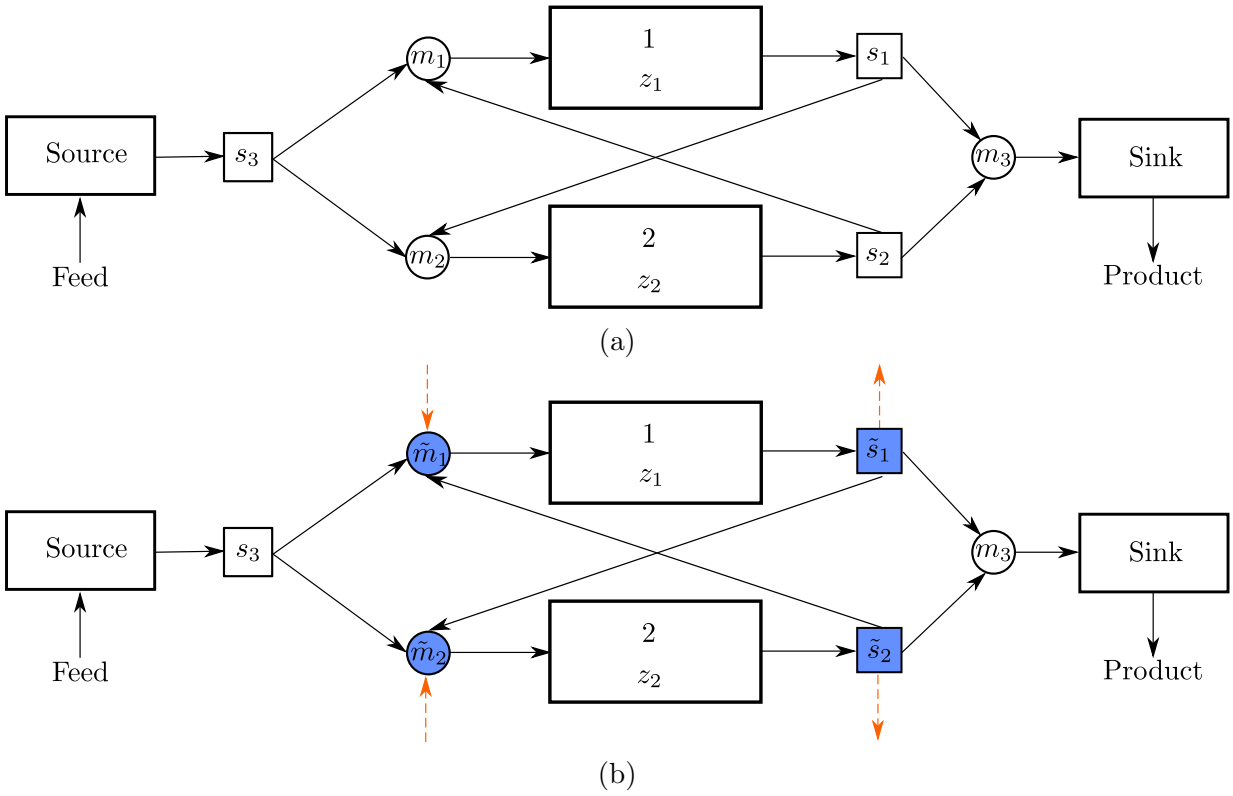


Figure 6: (a) Unit selection problem for case study 1 – SON representation; (b) Unit selection problem for case study 1 – MSON representation.

3.1.1 MSON formulation of the toy problem

We construct the MSON formulation of the toy problem to yield Problem (MSON-toy), which is represented in Figure 6b. To do so, we define $\mathcal{M}_1 = \{2, 3\}$, $\mathcal{M}_2 = \{1, 3\}$ and $\mathcal{M}_3 = \{1, 2\}$ for the mixers and $\mathcal{S}_1 = \{2, 3\}$ $\mathcal{S}_2 = \{1, 3\}$ and $\mathcal{S}_3 = \{1, 2\}$ for the splitters. We note that $\mathcal{C}_1 = \{1\}$ (corresponding to variable c_1) and $\mathcal{C}_2 = \{2\}$ (corresponding to variable c_2). We introduce new flowsheet-level variables c_u^{S}, $u \in \{1, 2\}$ that correspond to variables indexed by $\mathcal{C}$. Only one of the two Big-M inequalities in Equation (16) is used here to arrive at Equation (32). Combining Equation (32) with the objective function is equivalent to including both the inequalities in (16). Thanks to the objective function, Equation(33) is also redundant, however, we include the equation here for illustration

purposes. Problem (MSON-toy) is thus given by:

$$\min \quad c_1^{\mathrm{S}} + c_2^{\mathrm{S}} \tag{20}$$

$$c_u = \alpha\sqrt{f_u^{\mathrm{in}}} \qquad \forall u \in \{1,2\} \tag{21}$$

$$f_u^{\mathrm{out}} = f_u^{\mathrm{in}} \qquad \forall u \in \{1,2\} \tag{22}$$

$$F = f_{3,1} + f_{3,2} \tag{23}$$

$$f_1^{\mathrm{in}} = f_{3,1} + f_{2,1} + f_1^{\mathrm{M}} \tag{24}$$

$$f_2^{\mathrm{in}} = f_{3,2} + f_{1,2} + f_2^{\mathrm{M}} \tag{25}$$

$$f_u^{\mathrm{M}} = f_u^{\mathrm{A}}(1 - z_u) \qquad \forall u \in \{1,2\} \tag{26}$$

$$f_3^{\mathrm{in}} = f_{1,3} + f_{2,3} \tag{27}$$

$$f_1^{\mathrm{out}} = f_{1,2} + f_{1,3} + f_1^{\mathrm{S}} \tag{28}$$

$$f_2^{\mathrm{out}} = f_{2,1} + f_{2,3} + f_2^{\mathrm{S}} \tag{29}$$

$$f_i^{\mathrm{out}} - f_u^{\mathrm{S}} \leq \bar{f} z_u \qquad \forall u \in \{1,2\} \tag{30}$$

$$f_u^{\mathrm{S}} \leq \bar{f}(1 - z_u) \qquad \forall u \in \{1,2\} \tag{31}$$

$$c_u - c_u^{\mathrm{S}} \leq \bar{c}(1 - z_u) \qquad \forall u \in \{1,2\} \tag{32}$$

$$c_u^{\mathrm{S}} \leq \bar{c} z_u \qquad \forall u \in \{1,2\} \tag{33}$$

$$\epsilon \leq f_u^{\mathrm{in}} \qquad \forall u \in \{1,2\} \tag{34}$$

$$f_{2,1} \leq \bar{f} z_1 \tag{35}$$

$$f_{3,1} \leq \bar{f} z_1 \tag{36}$$

$$f_{1,2} \leq \bar{f} z_2 \tag{37}$$

$$f_{3,2} \leq \bar{f} z_2 \tag{38}$$

$$z_1 + z_2 \geq 1 \tag{39}$$

$$z_u \in \{0,1\} \qquad \forall u \in \{1,2\} \tag{40}$$

$$c_u, c_u^{\mathrm{S}} \in [0, \bar{c}] \qquad \forall u \in \{1,2\} \tag{41}$$

$$f_u^{\mathrm{in}}, f_u^{\mathrm{out}}, f_u^{\mathrm{S}} \in [0, F] \qquad \forall u \in \{1,2\} \tag{42}$$

$$f_{1,2}, f_{2,1}, f_{1,3}, f_{2,3}, f_{3,1}, f_{3,2} \in [0, \bar{f}] \tag{43}$$

where $\bar{c}$ is an upper bound on cost and $\bar{f}$ is an upper bound on flowrate. These upper bounds are set to reasonable values (not necessarily the tightest values) based on the input feed flowrate. We make sure that these bounds do not cut off the optimal solution.

3.1.2 Standard formulations of the toy problem

We consider three standard formulations, discussed in the introduction 1:

- Problem (SON-toy): the SON formulation of the toy problem
- Problem (Big-M toy): the Big-M reformulation of the toy problem
- Problem (NL-toy): the nonlinear reformulation of the toy problem

The SON superstructure is shown in Figure 6a. The standard formulations of the toy problem are given in full in Appendix B.

3.1.3 Numerical experiments on case study 1

We investigate numerical performance of Problems (SON-toy), (BigM-toy), (NL-toy) and (MSON-toy) in the gPROMS PS environment and the GAMS MO environment. The values of the constants used in these formulations are given in Table 4. To implement these problem formulations in gPROMS, variables are partitioned into model variables and optimization variables as mentioned in 2.4. $f_{3,1}$, $f_{1,2}$, $f_{2,1}$, z_1 and z_2 are optimization variables in all formulations. For the BigM-toy formulation, the additional optimization variables are c_1 and c_2. For the MSON formulation, the additional optimization variables are $f_{1,3}$, $f_{2,3}$, f_1^{M}, f_2^{M}, c_1^{S} and c_2^{S}. In GAMS we use default initial guesses for all variables, but in gPROMS these need to be specified. We test the following two sets of intuitive initial guesses:

- IG-1: Both binary variables are 1 and all other degrees of freedom are set at the midpoints of their intervals.

Table 4: Values of constants in the toy problem

Constant	Value
ϵ	0.1
$\bar{c}$	1
$\bar{f}$	1
f_1^{A}	0.5
f_2^{A}	0.5
F	1

- IG-2: All variables are assigned their respective values at the globally optimal solution of the SO problem. The optimal solution, in which only one of the binary variables takes the value 1, is easy to arrive at.

Several MINLP algorithms (such as outer approximation and generalized Benders decomposition) entail the solution of primal problems in which the discrete variables are fixed to yield nonlinear programming (NLP) problems. NLPs with partially fixed binary variables are also solved in branch and bound-based MINLP algorithms. To illustrate the numerical issues that may arise due to singularities when a process unit is deselected, we first study the numerical performance of the NLP primal problems that arise in the solution of the toy problem. We consider two primal problems: i) Primal-11 in which both units are selected ($z_1 = 1$ and $z_2 = 1$) and ii) Primal-10 in which only unit A is selected ($z_1 = 1$ and $z_2 = 0$). (Primal-01 in which only unit B is selected is symmetric to Primal-10 as the two units are identical).

We implement and solve Primal-11 and Primal-10 that arise in the solution of problems (SON-toy), (MSON-toy), (BigM-toy) and (NL-toy) as follows:

1. formulate the NLPs in GAMS and solve them with all available local NLP solvers, namely CONOPT (ARKI Consulting & Development A/S, 2024a), SNOPT (Gill et al., 2005), IPOPT (Wächter and Biegler, 2005), KNITRO (Byrd et al., 2006), MINOS (Murtagh et al., 2002) and XPRESS (FICO, 2024). All solver parameters and initial guesses are set to default values.

2. formulate the NLPs in gPROMS and solve them using the NLPSQP solver. As

NLPSQP requires initial guesses to be supplied, we initialize the continuous variables in these problems with their respective values in sets IG-1 and IG-2.

We implement and solve MINLP Problems (SON-toy), (MSON-toy), (BigM-toy) and (NL-toy) as follows:

1. formulate the MINLP in GAMS and solve it with all available local MINLP solvers, namely ALPHAECP (denoted here as αECP) (Lastusilta et al., 2009), DICOPT (Kocis and Grossmann, 1989), KNITRO (Byrd et al., 2006), sBB (ARKI Consulting & Development A/S, 2024b), SHOT (Kronqvist et al., 2015), and XPRESS (FICO, 2024). All solver parameters and initial guesses are set to default values.

2. formulate the MINLP in gPROMS and solve it using the native OA/ER/AP (Viswanathan and Grossmann, 1990) algorithm in gPROMS with initial guesses IG1 and IG2.

3.1.4 Results for the toy problem

In Table 5 we show the convergence status of Primal-11 and Primal-10 derived from Problems (MSON-toy), (SON-toy), (BigM-toy) and (NL-toy) in GAMS and gPROMS. We test the performance of each formulation with 6 solvers in GAMS and 2 initial guesses in gPROMS resulting in a total of 8 tests for each primal problem arising from each problem formulation. We find that for Primal-11, convergence to a solution is achieved for all problem formulations for all solvers in GAMS and in gPROMS when the initial guess is IG-1. With gPROMS and initial guess IG-2, convergence to a solution is only achieved for the formulation that corresponds to the MSON. We also note that the solver IPOPT converges to a solution for the toy flowsheet despite Jacobian evaluation errors at the solver-generated initial point for Problems (SON), (BigM-toy) and (NL-toy).

For Primal-10, on the other hand, the solvers fail to converge to a solution in 28% of the 32 tests, illustrating the numerical issues that can arise when process units are deselected even in simple flowsheet synthesis problems. Function or derivative evaluation errors are reported by the solvers in 34% of the total 32 tests in which unit B is deselected as shown

in Table 5. Problem (SON-toy) is the worst-performing formulation because the solution of Primal-10 results in a failure to converge in 5 out of the 8 tests. Similarly, the solution of Primal-10 that corresponds to Problem (BigM-toy) results in a convergence failure in 50% of the tests. Primal-10 problems derived from Problems (MSON-toy) and (NL-toy) are solved successfully across all of the 8 tests, indicating that both these formulations are promising. We note that in gPROMS with initial guess IG-2, the solver fails to converge to a solution of Primal-10 as derived from Problem (NL-toy) when the binary variable post-multiplies the cost function due to an undefined derivative. In contrast, in GAMS with default initial guesses, convergence across the tested solvers is not related to the order of multiplication of the binary variable and the cost function, demonstrating that the successful performance of the nonlinear reformulation may depend on the modelling platform, and specifically the routine used to evaluate derivatives. The results also indicate that a flowsheet in which only one unit is selected is optimal.

In Table 6 we show the results of solving MINLP Problems (MSON-toy), (SON-toy), (BigM-toy) and (NL-toy) in GAMS and gPROMS. We again test the performance of each formulation with 6 solvers in GAMS and 2 initial guesses in gPROMS resulting in a total of 8 tests for each problem formulation. Problem (BigM-toy) is the worst-performing formulation as it fails to converge to a solution in 4 out of the 8 tests. For Problem (SON-toy), failure to converge to a solution at all is seen in 1 out of the 8 tests. For both Problems (MSON-toy) and (NL-toy), convergence to a solution is obtained across all 8 tests. However, for Problem (NL-toy), convergence to a solution is not achieved in gPROMS when the binary variable post multiplies the cost function and the initial guess is at the problem solution, due to a failure to evaluate derivatives. This indicates that the use of the nonlinear formulation does not guarantee removal of numerical singularities. Function or derivative evaluation errors are reported in a total of 5 of the total 32 tests. In our tests some MINLP solvers, for example, XPRESS in Table 6, converge to a solution even when function or derivative evaluation errors are reported.

In this study, we use only local optimization techniques that are not guaranteed to converge to the global minimum of the problem. As solving Problems (BigM-toy), (SON-

Table 5: Results of the primal problems for case study 1 with different fixed values of the binary variables ($z_A = z_B = 1$ for Primal-11 and $z_A = 1.\ z_B = 0$ for Primal-10. The MINLP formulation from which the primal problem is derived is indicated in column MINLP. Each subsequent column corresponds to a different solver (for GAMS) or initial point (for gPROMS). The symbols indicate the following: ✓: Converges to a locally optimal solution; ✗: Does not converge to a locally optimal solution; †: Function/derivative evaluation errors reported; ‡: Fails to converges due to function/derivative evaluation error when the binary variable post multiplies the cost function, i.e., when $c = \alpha\sqrt{f}z$, but succeeds when $c = z\alpha\sqrt{f}$

Primal problem	MINLP formulation	GAMS solver						gPROMS initial point	
		CONOPT	SNOPT	IPOPT	KNITRO	MINOS	XPRESS	IG-1	IG-2
Primal-11	(MSON-toy)	✓	✓	✓	✓	✓	✓	✓	✓
	(SON-toy)	✓	✓	✓†	✓	✓	✓	✓	✗†
	(BigM-toy)	✓	✓	✓†	✓	✓	✓	✓	✗†
	(NL-toy)	✓	✓	✓†	✓	✓	✓	✓	✗†
Primal-10	(MSON-toy)	✓	✓	✓	✓	✓	✓	✓	✓
	(SON-toy)	✓	✓	✗†	✗†	✗†	✓	✗†	✗†
	(BigM-toy)	✓	✓	✓†	✗†	✗†	✓	✗†	✗†
	(NL-toy)	✓	✓	✓†	✓	✓	✓	✓	✓‡

	GAMS						gPROMS	
	αECP	DICOPT	KNITRO	SBB	SHOT	XPRESS	OA-1	OA-2
MSON-toy	✓	✓	✓	✓	✓	✓	✓$^{+}$	✓
SON-toy	✓*	✓	✓	✓*	✓	✓†	✓*	✗†
BigM-toy	✓*	✗	✓	✗	✓	✓	✗†	✗†
NL-toy	✓*	✓	✓*	✓	✓	✓†	✓	✓‡

Table 6: Results of Toy problem MINLP
✓: Converges to a locally optimal and/or feasible solution
✗: Does not converge to a solution
†: Function/derivative evaluation errors reported
*: Identifies sub-optimal solution in which both binaries are equal to 1
‡: Fails to converges due to function/derivative evaluation error when the binary variable post-multiplies the cost function in Equation (92), i.e. when $c = \alpha\sqrt{f}z$.
$^{+}$: A primal problem in which both binary variables are fixed to zero does not converge to a solution. This fixing of binary variables is infeasible w.r.t. (39), nevertheless this is generated by the OA/ER/AP solver in gPROMS due to the use of slack variables in the constraints of the master problem.

toy) and (NL-toy) results in local (not global) solutions in which both units are selected in 6 out of the 24 tests, some problem failures due to the numerical singularities that arise from deselection of a toy unit may have been averted. In all 8 tests, only one unit is selected at the solution of Problem (MSON-toy), illustrating that numerical singularities are eliminated by the use of the formulation. Lastly, we test Problem (MSON-toy) using the SO-PS algorithm. The results obtained from the SO-PS algorithm, the OA/ER/AP algorithm in gPROMS and the GAMS solvers are all identical.

The results in Tables 5 and 6 indicate that the formulation, solver choice and initial guess all impact the convergence of SO. The results illustrate that commonly used literature formulations such as Big-M and SON can be prone to non-convergence even on simple flowsheets. The results also indicate that for both the full process synthesis MINLP and the primal NLPs, the reformulation offered into the MSON is robust and removes numerical singularities due to process unit deselection. Additionally, in the MSON formulation, unlike Problem (NL-toy), simulator-only constraints do not have to be modified. This is especially advantageous as the number of such constraints tends to be large.

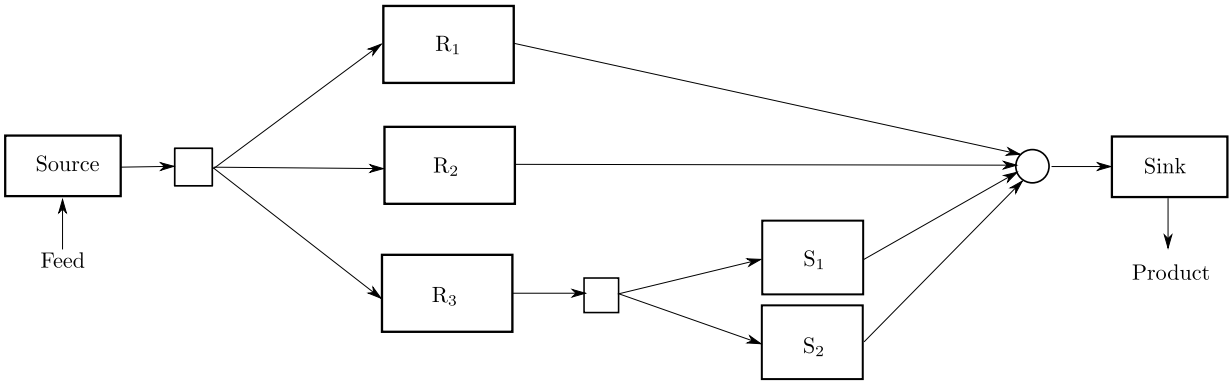


Figure 7: SON representation of the flowsheet synthesis problem in case study 2

3.2 Case study 2: Reactor-separator network

To illustrate the MSON approach on a more challenging problem, we present its application to a reactor-separator network synthesis case study taken from Trespalacios and Grossmann (2014). The GDP formulation of the flowsheet synthesis problem as well as the Big-M and convex hull MINLP formulations of the problem are given in Trespalacios and Grossmann (2014), whereas the nonlinear MINLP formulation is given in Burre et al. (2023).

3.2.1 Problem statement

Consider a mixture with molar flowrate f_{F}, that may flow into up to three reactors: R_1, R_2 and R_3. When reactor R_3 is chosen, one of the two separators S_1 and S_2 must be selected. All streams that exit the units R_1, R_2, S_1 and S_2 are combined (via a mixer) to yield the product stream with molar flowrate f_{P}. The flowsheet superstructure is shown in Figure 7.

Each unit $u \in \mathcal{T}$, where $\mathcal{T} = \{\mathrm{R}_1, \mathrm{R}_2, \mathrm{R}_3, \mathrm{S}_1, \mathrm{S}_2\}$, is modelled by the following equations.

$$f_u^{\mathrm{out}} = \xi_u f_u^{\mathrm{in}}, \tag{44}$$

$$c_u = \begin{cases} \mu_u {f_u^{\mathrm{in}}}^{\nu_u} & \text{if } u \text{ is a reactor} \\ \mu_u & \text{otherwise} \end{cases} \tag{45}$$

Table 7: Constants ξ_u, μ_u and ν_u for case study 2

Unit u	ξ_u	μ_u	ν_u
R_1	0.7	0.9	0.5
R_2	0.8	0.8	0.3
R_3	0.9	0.7	0.6
S_1	0.9	0.2	-
S_2	0.8	0.1	-

where f_u^{in} and f_u^{out} are the molar flowrates of the feed and the product at the inlet and outlet of unit u, respectively; c_u is the cost of unit u; and ξ_u, μ_u and $\nu_u < 1$ are strictly positive constants. The values of the constants ξ, μ and ν for each of the process units are given in Table 7.

The objective to be minimized is the net cost which is given by $C_2 f_{\text{F}} + \sum_{u \in \mathcal{T}} c_u - C_1 f_{\text{P}}$, where C_1 and C_2 are suitable constants. Note that constraints (45) are not differentiable for the reactors R_1, R_2 and R_3 when the reactor feed has a zero flowrate.

The superstructure optimization problem is given as: *Select between one and three reactors and a separator, if required, such that the net cost is minimized and the problem constraints are satisfied, given the models that describe the process units.*

3.2.2 MSON representation of the superstructure

The superstructure as represented using the MSON framework is shown in Figure 8. The feed to the flowsheet is modelled via a source unit followed by a standard splitter. Similarly, the product from the flowsheet is modelled as a sink unit that is preceded by a standard mixer. We define $\mathcal{F} = \{Source\}$, $\mathcal{P} = \{Sink\}$ and $\mathcal{U} = \mathcal{T} \cup \mathcal{F} \cup \mathcal{P}$. Each conditional process unit $u \in \mathcal{T}$, has an associated binary variable z_u. Since exactly one separator is selected if and only if reactor R_3 is chosen and no separator is selected otherwise, we introduce the logical constraint:

$$z_{R_3} = z_{S_1} + z_{S_2} \tag{46}$$

Each process unit in $\mathcal{T}$ has one inlet and one outlet. At each of these inlets and

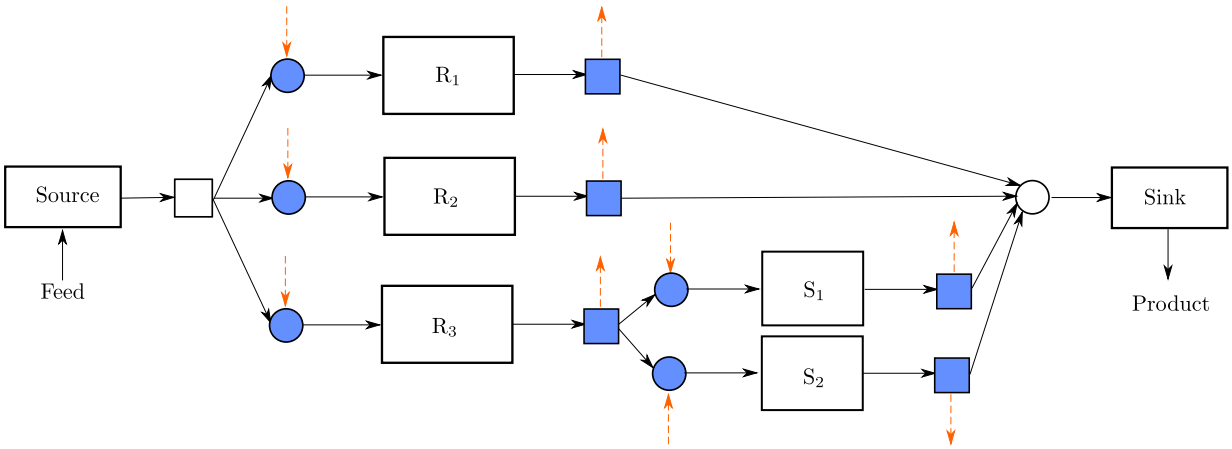


Figure 8: MSON representation of the flowsheet synthesis problem in case study 2

Table 8: Values of constants in Problem (MSON) for case study 2

Constant	Value
ϵ	0.1
$\bar{f}$	5
f_u^{A}, $u \in \mathcal{T}$	0.5
$\bar{c}$, $u \in \mathcal{T}$	5
f_F	1

outlets we replace the standard mixer and the standard splitter with a modified mixer and a modified splitter, respectively. For each process unit $u \in \mathcal{T}$, the set $\mathcal{C}_u$ indexes the cost c_u and the set $\mathcal{A}_u$ is empty. We introduce new variables $c_u^{\mathrm{S}}, u \in \mathcal{T}$ and constraints such that $c_u^{\mathrm{S}} = c_u, \forall u \in \mathcal{T}$ if and only if $z_u = 1$. We rewrite the objective function as $C_2 f_{\mathrm{F}} + \sum_{u \in \mathcal{T}} c_u^S - C_1 f_{\mathrm{P}}$. Note that the process unit constraints themselves are unchanged. The values of the constants in Problem (MSON) for case study 2, including reasonable values for the variable bounds, are shown in Table 8.

3.2.3 Results of case study 2

The reactor-separator network synthesis problem is implemented using the SO-PS algorithm shown in Section 2.4. The progress of the SO-PS algorithm is shown in Table 9. Note that due to the inherent non-convexity of the problem the solution of the master problem, which is a lower bound on the MINLP solution when its continuous relaxation is

Table 9: Progress of the SO-PS algorithm for the reactor-separator network of case study 2. k is the iteration number, z_u is the value of the binary variable that corresponds to unit $u \in \mathcal{T}$ during iteration k. $F^{(k)}$ is the optimal value of the objective function at the solution of the k^{th} primal problem. BUB is the best upper bound, i.e., the lowest primal objective function value up to iteration k. $MPS^{(k)}$ is the master problem solution (objective function value) at iteration k. The penalty factor in the objective function of the master problem of the SO-PS algorithm is set to 5.

k	z_{R_1}	z_{R_2}	z_{R_3}	z_{S_1}	z_{S_2}	$F^{(k)}$	BUB	$MPS^{(k)}$
1	1	1	1	1	0	1.3	1.3	-1e6
2	1	1	1	0	1	1.2	1.2	1.3
3	0	1	1	1	0	0.9	0.9	1.8
4	1	0	0	0	0	0.5	0.5	2.7
5	1	1	0	0	0	0.6	0.5	2.8
6	1	0	1	0	1	1.0	0.5	3.4
7	0	1	0	0	0	0.3	0.3	3.9
8	1	0	1	1	0	0.8	0.3	4.5

convex, is found to be larger than the upper bound in most iterations.

In the initialization phase of the SO-PS algorithm, two primal problems are solved: iteration 1 in which all units except S_2 are selected and iteration 2 in which all units except S_1 are selected. Across the two initialization iterations, all units $u \in \mathcal{T}$ are selected at least once. The two iterations are thus sufficient to initialize the master problem while satisfying constraint (46). A total of 11 flowsheet configurations are feasible with respect to Equation (46) in this reactor-separator network. The optimal flowsheet in which only unit R_2 is selected is identified in the seventh iteration of the SO-PS algorithm. The algorithm takes a total of 8 primal problems to terminate as per the stopping criteria in Section 2.4: condition i) of the stopping criteria is active at algorithm termination. The optimal flowsheet in which only reactor R_2 is selected has a 77% lower cost than the initial flowsheet tested in iteration 1. The deselection of units S_1 and S_2 does not result in singularities. Thus, in this case, the modified mixers and splitters associated with the inlets and outlets of the two separators may be replaced by standard mixers and splitters.

4 Case study 3 – Column synthesis

The third case study is concerned with the design of a counter-current gas-liquid absorption column to separate a binary mixture. It can be stated as follows:

Given a binary mixture of carbon dioxide and methane with flowrate f_F and mass fraction vector $\boldsymbol{q}_F$ at pressure P *and temperature T_F, and a solvent, find the number of stages N and solvent flowrate that minimize the total annualized cost of separation such that the column has at most $\bar{N}$ equilibrium-stages; the vapour (product) flowrate at the top of the column is at least f_p; the mass fraction of methane in the product is at least $\bar{q}_p$; and the area of the column is less than $\bar{a}_c$. We assume that the pressure* P *is constant throughout the column. We also assume that the column is perfectly insulated.*

4.1 An MSON superstructure for a counter column

Here we propose a superstructure representation of a column section that may also be applied to the synthesis of extraction columns and extended to the synthesis of distillation columns (Barttfeld et al., 2003, Ma et al., 2023, Sargent and Gaminibandara, 1976). The superstructure has the following permanent process units: a vapour source F^{V}, a liquid source F^{L}, a vapour sink P^{V} and a liquid sink P^{L}.

The selection of stages in the column is represented by an R-graph (Farkas et al., 2008). The column is divided into $|\mathcal{R}|$ subsections where $|\mathcal{R}| = \mathrm{ceil}(\log_2(\bar{N}+1))$ and $\mathrm{ceil}(x)$ returns the smallest integer greater than or equal to $x \in \mathbb{R}$. Each subsection $u \in \mathcal{R}$, where $\mathcal{R} = \{1, \ldots, |\mathcal{R}|\}$, is a conditional unit in our column superstructure, with 2^{u-1} equilibrium stages in which the liquid and vapour streams are contacted in a counter-current fashion. For example, in Figure 9, the column has at most $\bar{N} = 15$ stages. It may be represented by 4 conditional units R_1, R_2, R_3 and R_4 which have 1, 2 and 4 and 8 stages, respectively. A binary variable z_u is used to denote the existence of conditional unit / section u.

To ensure that at least one subsection is selected, we impose the constraint:

$$\sum_{u \in \mathcal{R}} z_u \geq 1. \tag{47}$$

The optimal number of stages $N = \sum_{r \in \mathcal{R}} 2^{r-1} z_r$ and $N \in \{1, \ldots, \bar{N}\}$. Constraint (48) ensures that at most $\bar{N}$ stages are selected:

$$\sum_{r \in \mathcal{R}} 2^{r-1} z_r \leq \bar{N}. \tag{48}$$

The total set of units (including the permanent units and conditional units) is given by $\mathcal{U} = \mathcal{R} \cup \{\mathrm{F^L}, \mathrm{F^V}, \mathrm{P^L}, \mathrm{P^V}\}$. Mixers and splitters connect the units in $\mathcal{U}$. Inlet and outlet streams and connectivity constraints are defined such that vapour and liquid streams are contacted in a counter-current fashion. As can be seen in Figure 9, source $\mathrm{F^L}$ has one liquid outlet β_0^{L} and one liquid inlet α_0^{L}, wherein solvent enters the column superstructure. $\mathrm{F^L}$ is located at the top of the column superstructure alongside the vapour sink $\mathrm{P^V}$. $\mathrm{P^V}$ has one vapour inlet α_0^{V} and one vapour outlet β_0^{V}. Similarly, $\mathrm{F^V}$ has one vapour outlet $\beta_{|\mathcal{R}|+1}^{\mathrm{V}}$ and one vapour inlet $\alpha_{|\mathcal{R}|+1}^{\mathrm{V}}$ and is located at the bottom of the column superstructure alongside the liquid sink $\mathrm{P^L}$. $\mathrm{P^L}$ has one liquid inlet $\alpha_{|\mathcal{R}|+1}^{\mathrm{L}}$ and one liquid outlet $\beta_{|\mathcal{R}|+1}^{\mathrm{L}}$.

Further, as shown in Figure 9, we arrange column subsections $u \in \mathcal{R}$ top-down with subsection $\mathrm{R_1}$ placed directly below $\mathrm{P^V}$ and $\mathrm{F^L}$ and units arranged in ascending order from top to bottom. Each subsection $u \in \mathcal{R}$ has one liquid inlet at the top of the subsection α_u^{L}, one vapour inlet α_u^{V} at the bottom of the subsection, one liquid outlet β_u^{L} at the bottom of the subsection and one vapour outlet β_u^{V} at the top of the subsection.

Let set $\mathcal{J} = \{0\} \cup \mathcal{R}$ and $\mathcal{K} = \mathcal{R} \cup \{|\mathcal{R}| + 1\}$. Either a mixer m_i^{V} or a modified mixer $\tilde{m}_i^{\mathrm{V}}$ is placed at each vapour inlet α_i^{V}, $\forall i \in \mathcal{J}$. Either a mixer m_i^{L} or a modified mixer $\tilde{m}_i^{\mathrm{L}}$ is placed at each liquid inlet, $\alpha_i^{\mathrm{L}}, \forall i \in \mathcal{K}$. A modified mixer is used when the inlet is associated with a conditional subsection and a standard mixer is used otherwise. A splitter, which may be either standard or modified, s_o^{V} or $\tilde{s}_o^{\mathrm{V}}$, is introduced after each vapour outlet β_o^{V}, $\forall o \in \mathcal{K}$ and a splitter, which is either standard or modified, s_o^{L} or $\tilde{s}_o^{\mathrm{L}}$, at

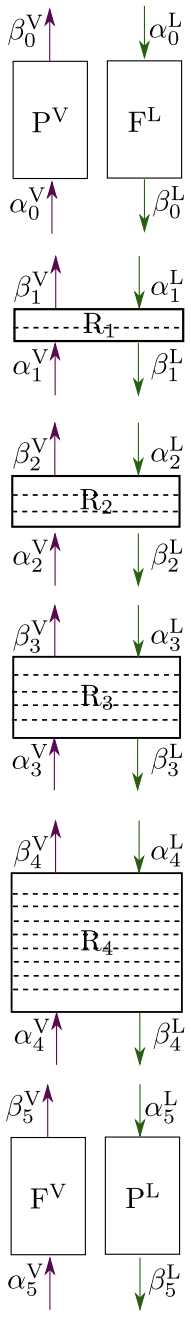


Figure 9: Elements of the absorption column superstructure that include the inlets and outlets of the sources $\mathrm{F}^{\mathrm{V}}, \mathrm{F}^{\mathrm{L}}$, sinks $\mathrm{P}^{\mathrm{V}}, \mathrm{P}^{\mathrm{L}}$, and column sections $\mathrm{R}_1, \mathrm{R}_2, \mathrm{R}_3, \mathrm{R}_4$. Each dashed line in the column section indicates a stage. $\mathrm{R}_1, \mathrm{R}_2, \mathrm{R}_3$, and R_4 have 1, 2, 4 and 8 stages, respectively. The mixers and splitters are not shown due to lack of space.

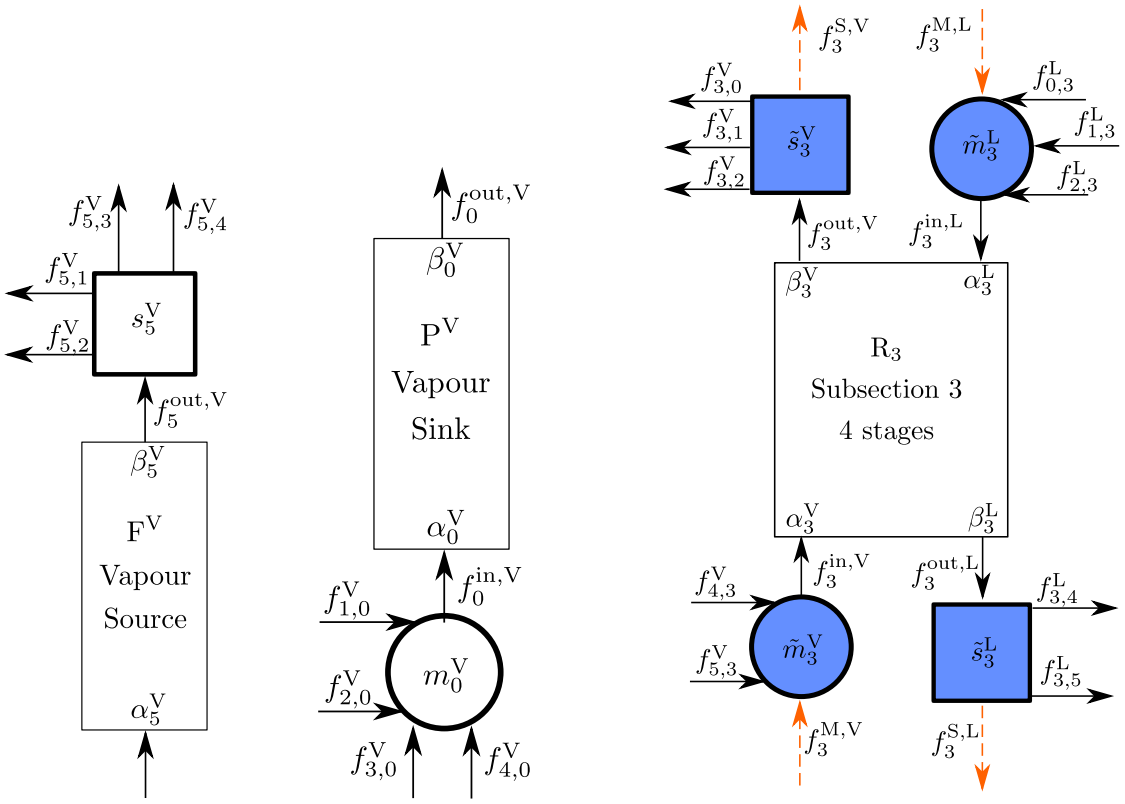


Figure 10: Examples of subsections associated with different units in case study 3. (a) Schematic of subsection around F^V. The vapour splitter allows the vapour feed to flow into units R_1, R_2, R_3 and R_4. (b) Schematic of subsection around P^V. The vapour mixer allows the vapour streams that leave units R_1, R_2, R_3 and R_4 to flow into the vapour sink. (c) Schematic of subsection around R_3. The modified vapour mixer allows mass flows into R_3 from units R_4 and F^V. The modified vapour splitter allows the vapour stream that leaves R_3 to flow into units R_1, R_2 and P^V. The modified liquid mixer allows flow into R_3 from R_1, R_2 and F^L. The modified liquid splitter allows the liquid stream that leaves R_3 to flow into R_4 and P^L.

each liquid outlet, $\beta_o^{\mathrm{L}}, \forall o \in \mathcal{J}$. As before, modified splitters are placed at the outlets of the conditional subsections and standard splitters are used at the outlets of the permanent units. Figure 10 shows the mixers and splitters associated with example units F^{V}, P^{V} and R_3.

The vapour mixer, which may be either standard or modified, m_i^{V} or $\tilde{m}_i^{\mathrm{V}}$, $i \in \mathcal{J}$ allows the mixing of vapour streams that flow from outlets β_o^{V}, $\forall o \in \mathcal{M}_i^{\mathrm{V}}$, where $\mathcal{M}_i^{\mathrm{V}} = \{i+1, \ldots, |\mathcal{R}|+1\}$, i.e., the vapour mixer m_i^{V} or $\tilde{m}_i^{\mathrm{V}}$, allows mixing of vapour streams that exit units located below unit i in the column superstructure. For example, for the column shown in Figure 9 and Figure 10, the modified mixer $\tilde{m}_3^{\mathrm{V}}$ that corresponds to unit R_3 is fed by streams that connect outlets β_o^{V}, $\forall o \in \mathcal{M}_1^{\mathrm{V}} = \{4, 5\}$ with $\tilde{m}_3^{\mathrm{V}}$. Thus, $\tilde{m}_3^{\mathrm{V}}$ allows the mixing of the vapour streams that exit R_4 and F^{V} via splitters $\tilde{s}_4^{\mathrm{V}}$ and s_5^{V}, respectively. The vapour stream flowing into the mixer at vapour inlet $\alpha_i^{\mathrm{V}}, i \in \mathcal{J}$, from the splitter at vapour outlet $\beta_o^{\mathrm{V}} \in \mathcal{M}_i^{\mathrm{V}}$, has flowrate, temperature and composition denoted by $f_{o,i}^{\mathrm{V}}$, $T_{o,i}^{\mathrm{V}}$ and $\boldsymbol{q}_{o,i}^{\mathrm{V}}$, respectively.

The liquid streams are treated in a similar fashion. Liquid inlet stream α_i^{L}, $i \in \mathcal{K}$ is the output of mixer m_i^{L} (or modified mixer $\tilde{m}_i^{\mathrm{L}}$), which allows mixing of streams that flow from splitters at outlets β_o, $\forall o \in \mathcal{M}_i^{\mathrm{L}} = \{0, \ldots, i-1\}$. That is, each liquid inlet may be connected via mixers and splitters to the liquid outlets of the units that are located above it in the column. For example, for the column shown in Figure 9 and Figure 10, the mixer $\tilde{m}_3^{\mathrm{L}}$ that corresponds to unit R_3 is fed by streams that connect β_o^{L}, $\forall o \in \mathcal{M}_3^{\mathrm{L}} = \{0, 1, 2\}$ to the mixer. Thus the modified mixer $\tilde{m}_3^{\mathrm{L}}$ allows the mixing of the liquid streams that exit F^{L}, R_1, and R_2. The liquid stream flowing into liquid mixer at inlet α_i^{L}, $i \in \mathcal{K}$ from the splitter at outlet $\beta_o^{\mathrm{L}} \in \mathcal{M}_i^{\mathrm{L}}$, has flowrate, temperature and composition denoted by $f_{o,i}^{\mathrm{L}}$, $T_{o,i}^{\mathrm{L}}$ and $\boldsymbol{q}_{o,i}^{\mathrm{L}}$, respectively.

For convenience of notation, we also define sets $\mathcal{S}_o^{\mathrm{L}}$ and $\mathcal{S}_o^{\mathrm{V}}$. Each splitter at liquid outlet β_o^{L} in $\mathcal{J}$ is allowed to be connected to the mixer at inlet $\alpha_i^{\mathrm{L}}, i \in \mathcal{S}_o^{\mathrm{L}}$, where $\mathcal{S}_o^{\mathrm{L}} = \{o+1, \ldots, |\mathcal{R}|+1\}$. Each splitter at vapour outlet β_o^{V}, $o \in \mathcal{K}$, may be connected to the set of vapour inlets $\mathcal{S}_o^{\mathrm{V}}$ (via mixers), where $\mathcal{S}_o^{\mathrm{V}} = \{0, \ldots, o-1\}$.

The sets introduced in this section are summarized in Table 10.

Set	Description
$\mathcal{R}$	Column subsections
$\mathcal{J}$	$\{0\} \cup \mathcal{R}$
$\mathcal{K}$	$\mathcal{R} \cup \{\lvert\mathcal{R}\rvert + 1\}$
$\mathcal{M}_i^{\mathrm{V}}$	Indices of vapour outlets that are connected to vapour mixer m_i^{V} or $\tilde{m}_i^{\mathrm{V}}$
$\mathcal{M}_i^{\mathrm{L}}$	Indices of liquid outlets that are connected to liquid mixer m_i^{L} or $\tilde{m}_i^{\mathrm{L}}$
$\mathcal{S}_o^{\mathrm{V}}$	Indices of vapour inlets that are connected to vapour splitter s_o^{V} or $\tilde{s}_i^{\mathrm{V}}$
$\mathcal{S}_o^{\mathrm{L}}$	Indices of liquid inlets that are connected to liquid splitter s_o^{L} or $\tilde{s}_i^{\mathrm{L}}$

Table 10: Sets defining the superstructure of a counter-current column are shown.

4.1.1 Flow-validity constraints

We reiterate that when $i \in \mathcal{R}$, the modified mixers and splitters ($\tilde{m}_i^{\mathrm{V}}$,$\tilde{m}_i^{\mathrm{L}}$,$\tilde{s}_i^{\mathrm{V}}$,$\tilde{s}_i^{\mathrm{L}}$) all correspond to conditional column subsection i.

The usual flow validity constraints (7) are added to prevent flows to deselected conditional subsections. Additional flow-validity constraints are needed for this case study as we disallow both partial and total bypassing of stages (and column subsections) as well as side streams. Thus, the vapour streams in this simple column flow upwards. To enforce this, constraint (49) imposes that the vapour stream that leaves a unit o is allowed to enter the unit $i, i+1 < o$, only if all the intermediate subsections $k \in \{o-1, \ldots, i+1\}$ have the associated binary variable $z_k = 0$. Define $\mathcal{K}_{o,i}^{\mathrm{V}} = \{k | k \in \mathcal{M}_i^{\mathrm{V}} \wedge k < o\}$, i.e., $\mathcal{K}_{o,i}^{\mathrm{V}}$ is the set of conditional subsections between unit o and unit i. Then,

$$f_{o,i}^{\mathrm{V}} \leq \bar{f}(1 - z_k), \quad \forall k \in \mathcal{K}_{o,i}^{\mathrm{V}}, \forall o \in \mathcal{M}_i^{\mathrm{V}}, \forall i \in \mathcal{R} \tag{49}$$

To illustrate the use of constraint (49), let use consider unit R_1 in Figure 9. Any splitter at outlet β_o^{V} where $o \in \mathcal{M}_1^{\mathrm{V}} = \{2, 3, 4, 5\}$ may be connected to α_1^{V}. When $o = 2$, $\mathcal{K}_{2,1}^{\mathrm{V}} = \emptyset$ and thus no constraint is added. When $o = 3$, $\mathcal{K}_{3,1}^{\mathrm{V}} = \{2\}$ yielding

$$f_{3,1}^{\mathrm{V}} \leq \bar{f}(1 - z_2). \tag{50}$$

When $o = 4$, $\mathcal{K}^{\mathrm{V}}_{4,1} = \{2, 3\}$ yielding

$$f^{\mathrm{V}}_{4,1} \leq \bar{f}(1 - z_2) \tag{51}$$

$$f^{\mathrm{V}}_{4,1} \leq \bar{f}(1 - z_3). \tag{52}$$

When $o = 5$, $\mathcal{K}^{\mathrm{V}}_{5,1} = \{2, 3, 4\}$ yielding

$$f^{\mathrm{V}}_{5,1} \leq \bar{f}(1 - z_2) \tag{53}$$

$$f^{\mathrm{V}}_{5,1} \leq \bar{f}(1 - z_3) \tag{54}$$

$$f^{\mathrm{V}}_{5,1} \leq \bar{f}(1 - z_4). \tag{55}$$

Analogously, the liquid stream must flow downwards through all selected units. To ensure this, we impose that a liquid that leaves a splitter at outlet β^{L}_o can enter a mixer at inlet α^{L}_i, where $i > o + 1$ only if all intermediate units k are deselected, that is, where $z_k = 0$, $\forall k \in \{o + 1, \ldots, i - 1\}$. We define $\mathcal{K}^{\mathrm{L}}_{o,i} = \{k | k \in \mathcal{M}^{\mathrm{L}}_i \wedge k > o\}$, i.e., $\mathcal{K}^{\mathrm{L}}_{o,i}$ is the set of intermediate conditional subsections between unit o and unit i.

$$f^{\mathrm{L}}_{o,i} \leq \bar{f}(1 - z_k) \quad \forall k \in \mathcal{K}^{\mathrm{L}}_{o,i}, \forall o \in \mathcal{M}^{\mathrm{L}}_i, \forall i \in \mathcal{R}. \tag{56}$$

To illustrate the use of constraint (56), let us consider unit R_3 in Figure 9 for which $\mathcal{M}^{\mathrm{L}}_3 = \{0, 1, 2\}$. We obtain the following system of inequalities:

$$f^{\mathrm{L}}_{0,3} \leq \bar{f}(1 - z_1) \tag{57}$$

$$f^{\mathrm{L}}_{0,3} \leq \bar{f}(1 - z_2) \tag{58}$$

$$f^{\mathrm{L}}_{1,3} \leq \bar{f}(1 - z_2) \tag{59}$$

4.1.2 Process unit-level constraints for each subsection

Each stage in each subsection is modelled as an equilibrium stage via MESH equations. The SAFT-γ Mie (Papaioannou et al., 2014) equation of state is used to calculate all the

required thermodynamic properties using the parameters reported in (Burger et al., 2015). We note that more detailed models that account for mass and heat transfer resistances may also be used without changing the superstructure.

For each subsection $u \in \mathcal{R}$, we compute total values for the liquid mass density, the vapour mass density and the volumetric vapour flowrate over all stages in the subsection. These are denoted by $\tilde{\rho}_u^{\mathrm{L}}$, $\tilde{\rho}_u^{\mathrm{V}}$, and $\tilde{v}_u^{\mathrm{V}}$, respectively and are given by

$$\tilde{\rho}_u^{\mathrm{L}} = \sum_{j=1}^{2^{u-1}} \rho_{u,j}^{\mathrm{L}} \tag{60}$$

$$\tilde{\rho}_u^{\mathrm{V}} = \sum_{j=1}^{2^{u-1}} \rho_{u,j}^{\mathrm{V}} \tag{61}$$

$$\tilde{v}_u^{\mathrm{V}} = \sum_{j=1}^{2^{u-1}} v_{u,j}^{\mathrm{V}} \tag{62}$$

where $\rho_{u,j}^{\mathrm{L}}$ is the mass density of the liquid leaving stage j in section u (in kg/m^3), $\rho_{u,j}^{\mathrm{V}}$ is the mass density of the vapour leaving stage j in section u (in kg/m^3) and $v_{u,j}^{\mathrm{V}}$ is the volumetric flowrate of the vapour leaving stage j in section u (in m^3/s). The total quantities are referred to as output variables and are indexed by set $\mathcal{C}_u$. In the flowsheet-level column sizing constraints new variables $\tilde{\rho}_u^{\mathrm{L^S}}$,$\tilde{\rho}_u^{\mathrm{V^S}}$ and $\tilde{v}_u^{\mathrm{V^S}}$ are introduced that are related to $\tilde{\rho}_u^{\mathrm{L}}$, $\tilde{\rho}_u^{\mathrm{V}}$ and $\tilde{v}_u^{\mathrm{V}}$, respectively by Equations (16) and (17).

4.1.3 Flowsheet-level constraints

The flooding velocity u^{flood} (in m/s) and the diameter D (in m) and height H (in m) of the column are computed using correlations in Sinnott and Towler (2020a) and depend on i) $\hat{\rho}^{\mathrm{L}}$ and $\hat{\rho}^{\mathrm{V}}$, the average of the liquid densities and of the vapour densities, respectively, across all the stages that are selected to be in the column and ii) $\hat{v}^{\mathrm{V}}$, the average of the vapour volumes across all selected stages. These quantities are functions of the modified

output variables $\tilde{\rho}_u^{\mathrm{L^S}}$,$\tilde{\rho}_u^{\mathrm{V^S}}$ and $\tilde{v}_u^{\mathrm{V^S}}$ and are computed using

$$N = \sum_{u \in \mathcal{R}} 2^{u-1} z_u \tag{63}$$

$$\hat{\rho}^{\mathrm{L}} = \frac{\sum_{u \in \mathcal{R}} \tilde{\rho}_u^{\mathrm{L^S}}}{N} \tag{64}$$

$$\hat{\rho}^{\mathrm{V}} = \frac{\sum_{u \in \mathcal{R}} \tilde{\rho}_u^{\mathrm{V^S}}}{N} \tag{65}$$

$$\hat{v}^{\mathrm{V}} = \frac{\sum_{u \in \mathcal{R}} \tilde{v}_u^{\mathrm{V^S}}}{N} \tag{66}$$

$$u^{\mathrm{flood}} = (-0.171 lt^2 + 0.27 lt - 0.047)\sqrt{\frac{\hat{\rho}^{\mathrm{L}} - \hat{\rho}^{\mathrm{V}}}{\hat{\rho}^{\mathrm{V}}}} \tag{67}$$

$$D = \sqrt{\frac{4\hat{v}^{\mathrm{V}}}{\pi u^{\mathrm{flood}}}} \tag{68}$$

$$H = 1.15 \frac{lt}{E} N \tag{69}$$

where lt is the tray spacing (in m) and E the stage efficiency factor.

We assume that the solvent is fully recovered by an ideal regeneration step whose cost is not considered in this study. We use the costing correlations given in Pereira et al. (2011) to compute the total capital investment TCI and the annual column operating expenses TOC of the absorption column. Lastly, the objective function is taken to be the total annualized cost of the absorber, TAC and is computed as

$$TAC = TOC + \eta TCI \tag{70}$$

where η is the annual capital charge ratio.

Other flowsheet-level constraints include minimum requirements imposed on the purity, the flowrate of the product stream that exits the vapour sink and the area of the column.

Table 11: Values of constants used in case study 3

Constant	Value
ϵ	0.1
$\bar{f}$	100
M	100
$f^{\mathrm{A,L}}$ (kmol/s)	0.76199
$f^{\mathrm{A,V}}$ (kmol/s)	1.13802
$T^{\mathrm{A,L}}$ (K)	300
$T^{\mathrm{A,V}}$ (K)	300
$q^{\mathrm{A,L}}_{\mathrm{CH_4}}$	0.20329
$q^{\mathrm{A,L}}_{\mathrm{CO_2}}$	0.00599
$q^{\mathrm{A,L}}_{\mathrm{solvent}}$	1-$q^{\mathrm{A,L}}_{\mathrm{CH_4}}$-$q^{\mathrm{A,L}}_{\mathrm{CO_2}}$
$q^{\mathrm{A,V}}_{\mathrm{CH_4}}$	0.99548
$q^{\mathrm{A,V}}_{\mathrm{CO_2}}$	0.00439
$q^{\mathrm{A,V}}_{\mathrm{solvent}}$	1-$q^{\mathrm{A,V}}_{\mathrm{CH_4}}$-$q^{\mathrm{A,V}}_{\mathrm{CO_2}}$

4.2 Implementation and Initialization

The hybrid framework given in Section 2.4 is used to solve the problem. As before, we implement all constraints in the PS environment and the SO-PS algorithm is used to solve the problem. To initialize the master problem we simply set the binary variables corresponding to all column subsections to one. We set both the initial step length and the maximum step length parameters of the SRQPD solver in gPROMS to 0.1. The penalty factor in the objective function of the master problem of the SO-PS algorithm is set to 5.

Initialization of the highly-coupled counter column model can often be a challenge in PS and MO environments. Full details of our initialization procedure is given in the Supporting Information. The properties of the fictitious streams along with other constants used in the formulation of this case study are shown in Table 11.

Table 12: Results for case study 3, column synthesis for CO_2 removal from methane. The number of stages, N, solvent flowrate f, total annualized cost TAC, total capital investment TCI and operating expenses $OPEX$ are reported. The first row shows the best solution returned by the SO-PS algorithm. The second row shows the design when the number of stages is fixed to 15 and the third row when it is fixed to 5.

Case	N	f (kmol/s)	$TAC(10^6$ USD)	$TCI(10^6$ USD)	$OPEX(10^6$ USD)
Best found	6	0.64	0.59	1.37	0.32
Fixed-15	15	0.60	1.04	3.01	0.44
Fixed-5	5	0.65	0.53	1.17	0.30

4.3 Results

We show results for the separation of carbon-dioxide and methane for a feed flowrate of 1 kmol/s at 7.5 MPa, 298 K and 20% CO_2. The pure solvent tetra(oxymethylene)dimethylether ($CH_3O(CH_2O)_4CH_3$) which has been identified to be a promising solvent for the separation of this mixture in other studies (Burger et al., 2015, Gopinath et al., 2016), enters the column at 7.5 MPa and 298 K and flowrate f. We find the optimal number of stages N and solvent flowrate f that minimize the total annualized cost TAC when $\bar{N}$ is 15, f_p is 0.66, x_p is 0.97, a_c is 300 m^2 E is 0.8, lt is 0.6 m and η is 0.199, corresponding to a 15% cost of capital and plant life of 10 years (Sinnott and Towler, 2020b).

The SO-PS algorithm takes a total of 8 iterations to satisfy stopping criterion i) given in Section 2.4. The optimal column design found by the algorithm corresponds to a column with six stages out of the possible 15 stages as shown in Table 12. Across the 8 iterations, The SO-PS algorithm identifies 5 feasible columns. The primal problem has a total of 976 variables, excluding the internal variables involved in computing thermodynamic properties with the SAFT-γ Mie equation of state. The MSON formulation of the problem has only 12 modified output variables ($|\mathcal{C}|$=12), which represents less than 1.3% of the total number of variables, highlighting the ease of using this formulation. Each primal problem has 871 simulator-only constraints and 116 optimization-only constraints of which 29 are equality constraints. The primal problems, on average, take 75 s of wall-clock time to run with a standard deviation of 37 s on a desktop with an Intel(R) Xeon(R) Gold 6130 CPU and a Windows 10 Enterprise operating system. The solution of each master problem takes less

than a second.

We also obtain the minimum TAC for a column in which the number of stages is fixed at 15, which is 76% higher than the 6-stage column column (see Table 12). The TCI of the 15-stage column is 120% higher than the best column identified by SO-PS, highlighting the large effect of column synthesis on the capital costs of separation columns. The solvent usage in the best-found column is almost the same as the fixed column. In both columns the constraint that the product flowrate is at least f_p is active. Note that to mitigate the effects of non-convexity, we used a linear constraint on product flowrate here instead of the bilinear constraint on product recovery that is commonly used in the literature.

Due to the inherent nonconvexity of the problem, the SO-PS algorithm is not guaranteed to yield a global minimum. To validate our results, we enumerate all integer choices the number of stages in the column and solve the resulting NLPs. We find that a column with 5 stages has the smallest TAC, as shown in Table 12. The fixed-5 column has a TAC that is 10% less than best solution identified by SO-PS. We also find that all columns with fewer than 5 stages are infeasible. In contrast to full enumeration, the SO-PS algorithm evaluates only 53% of the search space to arrive at a solution. While the SO-PS algorithm converges to a local minimum, it returns a high-quality solution which is the nearest alternative to what appears to be the globally-optimal column.

5 Conclusion

The optimization of a given superstructure may be cast as an MINLP using several alternative and mathematically-equivalent problem formulations (Trespalacios and Grossmann, 2014). However, the choice of MINLP formulation affects the performance of MINLP optimization in practice (Trespalacios and Grossmann, 2014). A novel problem formulation with favourable numerical properties, the modified state-operator network (MSON), has been presented in this study. It is based on based on an exact reformulation of mixers and splitters and the introduction of fictitious streams.

The MSON is a general framework that may be used to solve a wide range of flowsheet

design problems and is suitable for use with rigorous process models. The MSON formulation requires the introduction of mixers, splitters and streams, as well as new variables, but does not entail any modifications of models used for process units. It has been presented within the context of a big-M formulation, although other approaches can be followed. We have illustrated the methodology with the synthesis of a toy flowsheet to highlight the challenges that can arise in superstructure optimization. The numerical performance of the MSON as well as other standard problem formulations were benchmarked for the toy problem across a range of optimization solvers and two platforms: gPROMS and GAMS. The MSON was found to be robust and to avoid any singularities, outperforming all other problem formulations even on a simple problem.

The MSON formalism was further tested on a reactor-separator superstructure from the literature and on a novel superstructure for the synthesis of a counter-current separation column, a challenging problem in superstructure optimization. The use of the MSON for these problems has been found to have several practical implications. Numerical singularities due to zero flows are completely averted. The column synthesis, in particular, has been carried out on the basis of high-fidelity rigorous models with detailed thermodynamics, that is, an advanced equation of state. As a by-product, the MSON also offers a systematic framework for initialization of a counter-current column model thus, averting labour-intensive process initialization by trial and error as described in the Supporting Information. Similarly, the MSON also offers a systematic framework for the initialization of complex flowsheets and trains of separation columns (see Supporting Information).

As high-dimensional MINLPs are prone to non-convergence, the robust framework presented here for superstructure optimization is appealing. Across case studies with varying degrees of complexity the modifications to the flowsheet-level constraints was limited and process unit-level constraints remained unaltered. The MSON is a promising approach for the synthesis of complex flowsheets. The performance of the MSON on more complex flowsheets is yet to be examined. The MSON, like all other SO approaches, relies on the solution of highly nonlinear optimization problems which may be difficult to initialize and often converge to poor local optima. However, the removal of the singularities arising from

zero-flows is expected to increase the likelihood of convergence to a feasible solution. The extension of the MSON to flowsheets in which the pressure is variable will be studied next.

Data Availability Statement

The GAMS and gPROMS models are available at https://zenodo.org/records/12571977 under CC-BY licence.

Acknowledgements

The authors thank Terrence Crombie, Senior Computing Officer, Imperial College London, for his valuable inputs on interfacing gPROMS with C++. The authors thank Dr. Litao Zhu for reproducing numerical experiments and proofreading this manuscript. S.G. is grateful to Imperial College London for a PhD scholarship. We thank the anonymous reviewers for their careful review which improved this paper.

References

Androulakis, I. and Venkatasubramanian, V. (1991). A genetic algorithmic framework for process design and optimization. *Computers & Chemical Engineering*, 15(4):217–228.

ARKI Consulting & Development A/S (2024a). CONOPT. http://www.gams.com/solvers/solvers.htm.

ARKI Consulting & Development A/S (2024b). SBB. http://www.gams.com/solvers/solvers.htm.

AspenTech (2015). Aspen Plus. http://www.aspentech.com/products/aspen-plus/.

Barttfeld, M., Aguirre, P. A., and Grossmann, I. E. (2003). Alternative representations

and formulations for the economic optimization of multicomponent distillation columns. *Computers & Chemical Engineering*, 27:363–383.

Bowskill, D. H., Tropp, U. E., Gopinath, S., Jackson, G., Galindo, A., and Adjiman, C. S. (2020). Beyond a heuristic analysis: integration of process and working-fluid design for organic rankine cycles. *Molecular Systems Design & Engineering*, 5:493–510.

Bugosen, S., Laird, C., and Parker, R. (2023). Chemical process flowsheet optimization with full space, surrogate, and implicit formulations of a Gibbs reactor. https://arxiv.org/abs/2310.09307.

Burger, J., Papaioannou, V., Gopinath, S., Jackson, G., Galindo, A., and Adjiman, C. S. (2015). A hierarchical method to integrated solvent and process design of physical CO_2 absorption using the SAFT-γ Mie approach. *AIChE Journal*, 61:3249–3269.

Burre, J., Bongartz, D., and Mitsos, A. (2023). Comparison of MINLP formulations for global superstructure optimization. *Optimization & Engineering*, 24:801–830.

Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., and Woodruff, D. L. (2021). *Pyomo Overview*, pages 25–36. Springer International Publishing, Cham.

Byrd, R. H., Nocedal, J., and Waltz, R. A. (2006). *Knitro: An Integrated Package for Nonlinear Optimization*, pages 35–59. Springer US, Boston, MA.

Caballero, J. A. (2015). Logic hybrid simulation-optimization algorithm for distillation design. *Computers & Chemical Engineering*, 72:284 – 299.

Caballero, J. A., Milán-Yañez, D., and Grossmann, I. E. (2005). Optimal synthesis of distillation columns: Integration of process simulators in a disjunctive programming environment. *Computer Aided Chemical Engineering*, 20:715 – 720.

Cavalcanti, S. M. and Barton, P. I. (2020). Multiple steady states and nonsmooth bifurcations in dry and vaporless distillation columns. *Industrial & Engineering Chemistry Research*, 59(40):18000–18018.

Cremaschi, S. (2015). A perspective on process synthesis: Challenges and prospects. 81:130–137. Special Issue: Selected papers from the 8th International Symposium on the Foundations of Computer-Aided Process Design (FOCAPD 2014), July 13-17, 2014, Cle Elum, Washington, USA.

Demirel, S. E., Li, J., and Hasan, M. M. F. (2019). A general framework for process synthesis, integration, and intensification. *Industrial & Engineering Chemistry Research*, 58(15):5950–5967.

Douglas, J. M. (1985). A hierarchical decision procedure for process synthesis. *AIChE Journal*, 31(3):353–362.

Dowling, A. W. (2018). *An Equation-based Framework for Large-Scale Flowsheet Optimization and Applications for Oxycombustion Power System Design*. PhD thesis, Carnegie Mellon University.

Dowling, A. W. and Biegler, L. T. (2015a). Degeneracy hunter: An algorithm for determining irreducible sets of degenerate constraints in mathematical programs. In Gernaey, K. V., Huusom, J. K., and Gani, R., editors, *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, volume 37 of *Computer Aided Chemical Engineering*, pages 809–814. Elsevier.

Dowling, A. W. and Biegler, L. T. (2015b). A framework for efficient large scale equation-oriented flowsheet optimization. *Computers & Chemical Engineering*, 72:3 – 20.

Farkas, T., Czuczai, B., Rev, E., and Lelkes, Z. (2008). New MINLP model and modified outer approximation algorithm for distillation column synthesis. *Industrial & Engineering Chemistry Research*, 47(9):3088–3103.

FICO (2024). Xpress optimization. https://www.fico.com/en/products/fico-xpress-optimization.

Fourer, R., Gay, D. M., and Kernighan, B. W. (1989). AMPL: A mathematical programing language. In Wallace, S. W., editor, *Algorithms and Model Formulations in Mathematical Programming*, pages 150–151, Berlin, Heidelberg. Springer Berlin Heidelberg.

GAMS Development Corporation (2024). General Algebraic Modeling System (GAMS) release 36.1.0, Fairfax, VA, USA, 2021. https://www.gams.com/download/.

Gill, P. E., Murray, W., and Saunders, M. A. (2005). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460.

Gopinath, S. and Adjiman, C. S. (2024). Advances in process synthesis: New robust formulations. *Systems and Control Transactions*, 3:145–152.

Gopinath, S., Jackson, G., Galindo, A., and Adjiman, C. S. (2016). Outer approximation algorithm with physical domain reduction for computer-aided molecular and separation process design. *AIChE Journal*, 62(9):3484–3504.

Göttl, Q., Grimm, D., and Burger, J. (2022). Automated synthesis of steady-state continuous processes using reinforcement learning. *Frontiers of Chemical Science and Engineering*, 16:288–302.

Gurobi Optimization, Inc. (2024). Gurobi optimizer reference manual. http://www.gurobi.com.

Javaloyes-Antón, J., Kronqvist, J., and Caballero, J. A. (2022). Simulation-based optimization of distillation processes using an extended cutting plane algorithm. *Computers & Chemical Engineering*, 159:107655.

Jiang, Z., Mathew, T. J., Zhang, H., Huff, J., Nallasivam, U., Tawarmalani, M., and Agrawal, R. (2019). Global optimization of multicomponent distillation configurations: Global minimization of total cost for multicomponent mixture separations. *Computers & Chemical Engineering*, 126:249–262.

Kocis, G. R. and Grossmann, I. E. (1987). Relaxation strategy for the structural optimization of process flow sheets. *Industrial & Engineering Chemistry Research*, 26(9):1869–1880.

Kocis, G. R. and Grossmann, I. E. (1989). Computational experience with DICOPT solving MINLP problems in process systems engineering. *Computers & Chemical Engineering*, 13(3):307–315.

Kondili, E., Pantelides, C. C., and Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations—I. MILP formulation. *Computers & Chemical Engineering*, 17(2):211–227.

Kossack, S., Kraemer, K., and Marquardt, W. (2006). Efficient optimization-based design of distillation columns for homogenous azeotropic mixtures. *Industrial & Engineering Chemistry Research*, 45(25):8492–8502.

Kronqvist, J., Lundell, A., and Westerlund, T. (2015). The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization*, 64(2):249–272.

Lastusilta, T., Bussieck, M. R., and Westerlund, T. (2009). An experimental study of the GAMS/AlphaECP MINLP solver. *Industrial & Engineering Chemistry Research*, 48(15):7337–7345.

Lee, S. and Grossmann, I. E. (2003). Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: applications to process networks. *Computers & Chemical Engineering*, 27(11):1557–1575.

Lee, U., Mitsos, A., and Han, C. (2016). Optimal retrofit of a CO_2 capture pilot plant using superstructure and rate-based models. *International Journal of Greenhouse Gas Control*, 50:57 – 69.

Liñán, D. A., Bernal, D. E., Ricardez-Sandoval, L. A., and Gómez, J. M. (2020). Optimal design of superstructures for placing units and streams with multiple and ordered available locations. Part I: A new mathematical framework. *Computers & Chemical Engineering*, 137:106794.

Ma, X. C., Zhang, Q., He, C., Chen, Q. L., and Zhang, B. J. (2023). Computer-aided naphtha liquid–liquid extraction: Molecular reconstruction, sustainable solvent design and multiscale process optimization. *Fuel*, 334:126651.

Mann, V., Sales-Cruz, M., Gani, R., and Venkatasubramanian, V. (2024). eSFILES: Intelligent process flowsheet synthesis using process knowledge, symbolic AI, and machine learning. *Computers & Chemical Engineering*, 181:108505.

McBride, K. and Sundmacher, K. (2019). Overview of surrogate modeling in chemical process engineering. *Chemie Ingenieur Technik*, 91(3):228–239.

Mencarelli, L., Chen, Q., Pagot, A., and Grossmann, I. E. (2020). A review on superstructure optimization approaches in process system engineering. *Computers & Chemical Engineering*, 136:106808.

Murtagh, B. A., Saunders, M. A., Murray, W., Gill, P. E., Raman, R., and Kalvelagen, E. (2002). MINOS: A solver for large-scale nonlinear optimization problems. https://api.semanticscholar.org/CorpusID:117760773.

Navarro-Amorós, M. A., Ruiz-Femenia, R., and Caballero, J. A. (2014). Integration of modular process simulators under the generalized disjunctive programming framework for the structural flowsheet optimization. *Computers & Chemical Engineering*, 67(Supplement C):13 – 25.

Papaioannou, V., Lafitte, T., Avendaño, C., Adjiman, C. S., Jackson, G., Müller, E. A., and Galindo, A. (2014). Group contribution methodology based on the statistical associating fluid theory for heteronuclear molecules formed from Mie segments. *The Journal of Chemical Physics*, 140(5):054107.

Pereira, F. E., Keskes, E., Galindo, A., Jackson, G., and Adjiman, C. S. (2011). Integrated solvent and process design using a SAFT-VR thermodynamic description: High-pressure separation of carbon dioxide and methane. *Computers & Chemical Engineering*, 35:474–491.

Raghunathan, A. U. and Biegler, L. T. (2003). Mathematical programs with equilibrium constraints (MPECs) in process engineering. *Computers & Chemical Engineering*, 27(10):1381–1392.

Raman, R. and Grossmann, I. (1994). Modelling and computational techniques for logic based integer programming. *Computers & Chemical Engineering*, 18(7):563 – 578.

Reynoso-Donzelli, S. and Ricardez-Sandoval, L. A. (2024). A reinforcement learning approach with masked agents for chemical process flowsheet design. *AIChE Journal*, page e18584.

Sargent, R. and Gaminibandara, K. (1976). Optimum design of plate distillation columns. *Computers & Chemical Engineering*, 19:83 – 88.

Siemens (1996-2024). gPROMS. https://www.siemens.com/global/en/products/automation/industry-software/gproms-digital-process-design-and-operations.

Sinnott, R. and Towler, G. (2020a). Chapter 11 - separation columns (distillation, absorption and extraction). In Sinnott, R. and Towler, G., editors, *Chemical Engineering Design*, Chemical Engineering Series, pages 645–772. Butterworth-Heinemann, sixth edition.

Sinnott, R. and Towler, G. (2020b). Chapter 6 - costing and project evaluation. In Sinnott, R. and Towler, G., editors, *Chemical Engineering Design*, Chemical Engineering Series, pages 275–369. Butterworth-Heinemann, sixth edition.

Skiborowski, M., Harwardt, A., and Marquardt, W. (2015). Efficient optimization-based design for the separation of heterogeneous azeotropic mixtures. *Computers & Chemical Engineering*, 72:34–51.

Smith, E. M. B. (1996). *On the Optimal Design of Continuous Processes*. PhD thesis, Imperial College London.

Smith, E. M. B. and Pantelides, C. C. (1995). Design of reaction/separation networks using detailed models. *Computers & Chemical Engineering*, 19:83 – 88.

Trespalacios, F. and Grossmann, I. E. (2014). Review of mixed-integer nonlinear and generalized disjunctive programming methods. *Chemie-Ingenieur-Technik*, 86(7):991–1012.

Trespalacios, F. and Grossmann, I. E. (2015). Improved Big-M reformulation for generalized disjunctive programs. *Computers & Chemical Engineering*, 76:98–103.

Tula, A. K., Eden, M. R., and Gani, R. (2015). Process synthesis, design and analysis using a process-group contribution method. *Computers & Chemical Engineering*, 81:245–259.

Türkay, M. and Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers & Chemical Engineering*, 20(8):959 – 978.

Viswanathan, J. and Grossmann, I. E. (1990). A combined penalty function and outer-approximation method for MINLP optimization. *Computers & Chemical Engineering*, 14(7):769 – 782.

Vollmer, N., Al, R., Gernaey, K. V., and Sin, G. (2021). Synergistic optimization framework for the process synthesis and design of biorefineries. *Frontiers of Chemical Science and Engineering*, pages 1 – 23.

Wächter, A. and Biegler, L. T. (2005). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.

Yeomans, H. and Grossmann, I. E. (1999). A systematic modeling framework of superstructure optimization in process synthesis. *Computers & Chemical Engineering*, 23(6):709 – 731.

Yeomans, H. and Grossmann, I. E. (2000a). Disjunctive programming models for the optimal design of distillation columns and separation sequences. *Industrial & Engineering Chemistry Research*, 39(6):1637–1648.

Yeomans, H. and Grossmann, I. E. (2000b). Optimal design of complex distillation columns using rigorous tray-by-tray disjunctive programming models. *Industrial & Engineering Chemistry Research*, 39(11):4326–4335.

A Conditional constraints in the MSON as a disjunction

The conditional constraints in the MSON (Equations (10d), (14),(15), (16), (17), (7), (19f)) have been given in the Big-M form. These equations may be alternatively represented as the following disjunction:

$$\begin{bmatrix} Z_u \\ f_i^{\mathrm{M}} = 0 \quad \forall i \in \mathcal{IN}_u \\ f_o^{\mathrm{S}} = 0 \quad \forall o \in \mathcal{OUT}_u \\ \boldsymbol{x}_j^{\mathrm{S}} = \boldsymbol{x}_j \quad \forall j \in \mathcal{C}_u \\ f_{o,i} \geq 0 \quad \forall o \in \mathcal{M}_i, \forall i \in \mathcal{IN}_u \\ \boldsymbol{g}_{\mathcal{G}_u} \leq 0 \end{bmatrix} \vee \begin{bmatrix} \neg Z_u \\ f_i^{\mathrm{M}} = f_i^{\mathrm{A}} \quad \forall i \in \mathcal{IN}_u \\ f_o^{\mathrm{S}} = f_o^{\mathrm{out}} \quad \forall o \in \mathcal{OUT}_u \\ \boldsymbol{x}_j^{\mathrm{S}} = c_{u_j} \quad \forall j \in \mathcal{C}_u \\ f_{o,i} = 0 \quad \forall o \in \mathcal{M}_i, \forall i \in \mathcal{IN}_u \\ \end{bmatrix}, \forall u \in \mathcal{T} \tag{71}$$

B Standard formulations of toy problem in full

The SON superstructure is shown in Figure 6a. Flow positivity constraints are only enforced for permanent units and conditional units that are selected.

$$\min \quad c_1 + c_2 \tag{72}$$

$$c_u = \alpha\sqrt{f_u^{\text{in}}} \qquad \forall u \in \{1,2\} \tag{73}$$

$$f_u^{\text{out}} = f_u^{\text{in}} \qquad \forall u \in \{1,2\} \tag{74}$$

$$F = f_{3,1} + f_{3,2} \tag{75}$$

$$f_1^{\text{in}} = f_{3,1} + f_{2,1} \tag{76}$$

$$f_2^{\text{in}} = f_{3,2} + f_{1,2} \tag{77}$$

$$f_3^{\text{in}} = f_{1,3} + f_{2,3} \tag{78}$$

$$f_1^{\text{out}} = f_{1,2} + f_{1,3} \tag{79}$$

$$f_2^{\text{out}} = f_{2,1} + f_{2,3} \tag{80}$$

$$\epsilon z_u \leq f_u^{\text{in}} \qquad \forall u \in \{1,2\} \tag{81}$$

$$f_{2,1} \leq \bar{f} z_1 \tag{82}$$

$$f_{3,1} \leq \bar{f} z_1 \tag{83}$$

$$f_{1,2} \leq \bar{f} z_2 \tag{84}$$

$$f_{3,2} \leq \bar{f} z_2 \tag{85}$$

$$z_1 + z_2 \geq 1 \tag{86}$$

$$z_u \in \{0,1\} \qquad \forall u \in \{1,2\} \tag{87}$$

$$c_u \in [0, \bar{c}] \qquad \forall u \in \{1,2\} \tag{88}$$

$$f_u^{\text{in}}, f_u^{\text{out}} \in [0, F] \qquad \forall u \in \{1,2\} \tag{89}$$

$$f_{1,2}, f_{2,1}, f_{1,3}, f_{2,3}, f_{3,1}, f_{3,2} \in [0, \bar{f}] \tag{90}$$

We replace Equation (73) with Equation (91) to obtain Problem (BigM-toy):

$$c_u \geq \alpha\sqrt{f_u^{\text{in}}} \quad \forall u \in \{1,2\} \tag{91}$$

We replace Equation (73) with Equation (92) to obtain Problem (NL-toy):

$$c_u = z_u\alpha\sqrt{f_u^{\text{in}}} \quad \forall u \in \{1,2\} \tag{92}$$