

# Feasible Path-Based Branch and Bound Algorithm for highly Nonconvex Mixed-Integer Nonlinear Programming Problems

Chao Liu, Yingjie Ma, Dongda Zhang, Jie Li\*

Centre for Process Integration, Department of Chemical Engineering, The University of Manchester, Manchester, M13 9PL, UK

[jie.li-2@manchester.ac.uk](mailto:jie.li-2@manchester.ac.uk)

In this paper, a feasible path-based branch and bound (B&B) algorithm is presented for solving mixed-integer nonlinear programming problems with highly nonconvex nature. The main advantage of this novel algorithm, comparing to the conventional branch and bound algorithms, is that when solving a nonlinear programming (NLP) subproblem at each node, our previously proposed hybrid steady-state and time-relaxation-based optimisation algorithm is employed. This approach allows circumventing complex initialisation procedure and overcoming the convergence difficulties of process simulations. During B&B, the solution from a parent node is used to initialize the NLP subproblems at the child nodes to improve the efficiency of this algorithm. The capability of the proposed algorithm is illustrated by solving a dividing wall column optimisation case for separation of a ternary mixture. The optimal design is obtained in 2, 712 CPU s with TAC 43,344 \$ y<sup>-1</sup>.

## 1. Introduction

Many optimisation problems arising in process synthesis, design and intensification are modeled as Mixed-Integer Nonlinear Programming (MINLP) problems. By using integer variables, discrete decisions are enabled, e.g., to select process equipment or assignment of task with optimising design and operation simultaneously. It is also possible to introduce discrete quantities, such as the number of trays in a distillation column (Kronqvist et al., 2018). Moreover, the incorporation of nonlinear rigorous models allows accurately modeling of production processes and unit operations. As a result, the MINLP problems are often highly nonconvex and nonlinear, which are difficult to solve.

MINLP can be divided into two classes - convex MINLP and nonconvex MINLP. A convex MINLP problem is defined as such only when the discrete binaries are relaxed as continuous variables, resulting a convex NLP problem (Trespalcios and Grossmann, 2014). There are several deterministic algorithms for solving convex MINLP problems which are mainly based on two approaches – MILP decomposition and branch and bound (Kronqvist and Lundell, 2019). The main idea of the branch and bound (B&B) method (Gupta and Ravindran, 1985) is used in most of the current MINLP deterministic solvers. When a tree search is performed, the integer variables are successively fixed at the corresponding nodes of the tree, yielding relaxed NLP subproblems. It is only attractive if subproblems are inexpensive to solve, or only a few of them need to be solved. In contrast to B&B, the main idea of the MILP/NLP decomposition approaches is to iteratively construct a MILP approximation by successive linearization of nonlinear constraints without using a search tree. Algorithms include outer approximation (Viswanathan and Grossmann, 1990), the extended cutting plane method (ECP) (Westerlund and Pettersson, 1995), extended supporting hyperplane (ESH) (Kronqvist et al., 2015) and generalised bender decomposition (GBD) (Geoffrion, 1972). Apart from above algorithms, Raman and Grossmann (1994) brought up an alternative representation of MINLP - Generalised Disjunctive Programming (GDP), which involves Boolean and continuous variables that are specified in algebraic constraints, disjunctions and logic propositions. These GDP problems can be solved by dedicated solution algorithms such as logic-based OA (Türkay and Grossmann, 1996) and GDP B&B (Lee and Grossmann, 2000). As pointed out by Ma and Li (2022), most of the aforementioned algorithms rely on the assumption that the relaxed NLP subproblems are convex and can be solved when they are feasible. Initial points need to be selected very carefully, which is usually done case by

case and using trial and error. However, nonconvex constraints are usually required for accurate modeling of many real-world problems, particularly in chemical engineering. The most common global deterministic method to solve nonconvex MINLP is spatial B&B. Although this method can theoretically find a global optimum, they can be very computational demanding, due to the generation of a huge global search tree, which may prevent the method to find an optimal solution within a reasonable time (Burre et al., 2022).

In this paper, we develop a new solution approach - a feasible path-based branch and bound algorithm for highly nonconvex MINLP problems. Like conventional branch and bound method, a search tree needs to be created. The key difference is that a feasible path optimisation framework is applied for solving the relaxed NLP subproblem at each node. This combination tackles the convergence difficulties when highly nonconvex constraints are incorporated in the mathematical models, e.g., thermodynamic equilibrium models, MESH equations and rigorous models for distillation. The other superiority of our proposed novel B&B algorithm when dealing with large scale strongly non-convex MINLP problems is the robust initialisation procedure (Dowling and Biegler, 2015). Initial points are easy to choose and convergence can be guaranteed. The proposed algorithm is applied to a dividing wall column optimisation case for separation of a ternary mixture. A locally optimal solution of 43,344 \$ yr<sup>-1</sup> is found within 2,712 CPU s, whilst most existing MINLP commercial solvers fail to find a feasible solution. The obtained configuration of DWC and TAC are almost the same as the ones generated using specialized procedures in the literature.

## 2. Problem Statement

The large scale highly nonconvex MINLP problems have the following general form:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} f(\mathbf{x}, \mathbf{y}) \\ \text{s. t. } \mathbf{h}(\mathbf{x}, \mathbf{y}) = 0 \\ \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0 \\ \mathbf{x} \in R^n, \mathbf{y} \in \{0, 1\}^s \end{aligned} \quad (1)$$

where  $f(\mathbf{x}, \mathbf{y})$  usually represents the objective function such as total annualised cost (TAC),  $\mathbf{h}(\mathbf{x}, \mathbf{y}) = 0$  are equations describing process performance such as material balance and energy balance equations,  $\mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0$  are inequality constraints that define process design requirements such as product specifications. The variable  $\mathbf{x}$  is a set of vector, representing continuous variables with dimension  $n$ . The variable  $\mathbf{y}$  is a set of discrete variables, and, we constrain it to binary variables without loss of generality.

If partitioning continuous variables  $\mathbf{x}$  into two sets - independent variables  $\mathbf{x}_I$  and dependent variables  $\mathbf{x}_D$ ,  $\mathbf{x}_D$  can be expressed using an implicit function of  $\mathbf{x}_I$ . Assuming the dimension of  $\mathbf{x}_D$  is  $m$ , then problem in Eq(1) (i.e., the full space model) can be translated to the reduced space model in Eq(2) in below:

$$\begin{aligned} \min_{\mathbf{x}_I, \mathbf{y}} \tilde{f}(\mathbf{x}_I, \mathbf{y}) \\ \text{s. t. } \tilde{\mathbf{h}}(\mathbf{x}_I, \mathbf{y}) = 0 \\ \tilde{\mathbf{g}}(\mathbf{x}_I, \mathbf{y}) \leq 0 \\ \mathbf{x}_I \in R^{n-m}, \mathbf{y} \in \{0, 1\}^s \end{aligned} \quad (2)$$

The objective of this work is to develop a novel branch and bound algorithm called feasible-path based branch and bound algorithm hybrid with steady state and time-relaxation based algorithm (denoted as FPBB-HB) to solve the above stated problem in Eq(2).

## 3. Solution Approach

The complete solution approach is illustrated in Figure 1. As seen from Figure 1, a rooted decision tree is firstly constructed with nodes and branches as a framework for the solution process. The root node is denoted as  $n_0$ , at which an NLP problem in Eq(3) is solved. This NLP problem is a relaxation of the problem in Eq(2) with all the binary variables relaxed to be 0-1 continuous variables.

$$\begin{aligned} \min_{\mathbf{x}_I, \mathbf{y}} \tilde{f}(\mathbf{x}_I, \mathbf{y}) \\ \text{s. t. } \tilde{\mathbf{h}}(\mathbf{x}_I, \mathbf{y}) = 0 \\ \tilde{\mathbf{g}}(\mathbf{x}_I, \mathbf{y}) \leq 0 \\ \mathbf{x}_I \in R^{n-m}, \mathbf{y} \in (0, 1)^s \end{aligned} \quad (3)$$

where  $\tilde{f}(\mathbf{x}_I, \mathbf{y})$ ,  $\tilde{\mathbf{h}}(\mathbf{x}_I, \mathbf{y})$  and  $\tilde{\mathbf{g}}(\mathbf{x}_I, \mathbf{y})$  are presumed to be twice differentiable,  $\mathbf{y}$  is a set of continuous variables between 0 and 1.

After solving the root node, if all the binary variables relaxed are already 0 or 1, the algorithm will be terminated with the integer solution from root node. Otherwise, two new child nodes are created. At each node  $n_i$ , the total number of binary variables is  $s$ . Some binary variables are fixed to 0 or 1, whilst the remaining binary variables are still relaxed between 0 and 1. The fixed set of binary variables is denoted as  $\mathbf{y}_F^i = [y_{F1}^i, y_{F2}^i, \dots, y_{Fp}^i]$  and the

relaxed set is denoted as  $\mathbf{y}_R^i = [y_{R1}^i, y_{R2}^i, \dots, y_{Rq}^i]$ , where  $p + q$  equals to  $s$ . The NLP subproblem at node  $n_i$  can be formulated as problem in Eq(4).

$$\begin{aligned}
 & \min_{\mathbf{x}_I, \mathbf{y}_R^i} \tilde{f}(\mathbf{x}_I, \mathbf{y}_R^i; \mathbf{y}_F^i) \\
 & \text{s. t. } \tilde{\mathbf{h}}(\mathbf{x}_I, \mathbf{y}_R^i; \mathbf{y}_F^i) = 0 \\
 & \quad \tilde{\mathbf{g}}(\mathbf{x}_I, \mathbf{y}_R^i; \mathbf{y}_F^i) \leq 0 \\
 & \quad \mathbf{x}_I \in R^{n-m}, \mathbf{y}_R^i \in (0, 1)^q, \mathbf{y}_F^i \in \{0, 1\}^q
 \end{aligned} \tag{4}$$

At each node  $n_i$ , our previous proposed hybrid steady-state and time-relaxation-based optimisation algorithm is employed to solve the NLP subproblem (i.e. problem in Eq(4)). The main idea of the hybrid algorithm is that, a steady state simulation is performed first. If it fails, a PTC simulation using the tolerance-relaxation integration method is then conducted. More details about this algorithm can be found in the work of Ma et al. (2020). It is demonstrated that the hybrid algorithm is effective to address the convergence difficulties and get a locally optimal solution even for large-scale highly nonconvex NLP problems. If there is no feasible solution at node  $n_i$ , then this node is pruned. Other nodes at node set  $\zeta$  will be selected based on the depth-first strategy. If an optimal solution  $\tilde{f}_i^*$  exists,  $\mathbf{y}_R^{i*}$  shall be checked. If all the components in  $\mathbf{y}_R^{i*}$  are binaries, it means solution at node  $n_i$  is an optimal integer solution. It shall be incumbent only if this is the best optimal integer solution till the current iteration. At earlier iterations, with no integer solution to the MINLP problem, the upper bound denoted as  $\tilde{f}_{ub}$  is set as infinite. At this iteration,  $\tilde{f}_{ub}$  will be updated to  $\tilde{f}_i^*$ , until next best integer solution occurs. If other optimal integer solution already exists and possess better value than  $\tilde{f}_i^*$ , node  $n_i$  shall be pruned.

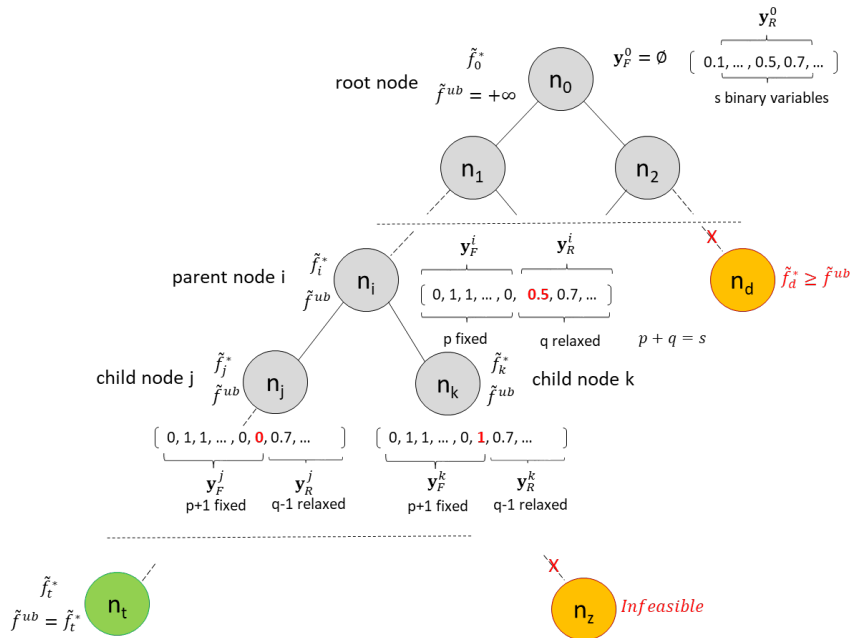


Figure 1 Schematic of branch and bound rooted tree

If some components in  $\mathbf{y}_R^{i*}$  are still fractions, node  $n_i$  will be treated as a parent node and further branched where one relaxed binary variable in  $\mathbf{y}_R^{i*}$  is selected and forced to 0 or 1. Hereby two child nodes  $n_j$  and  $n_k$  are to be created. Obviously, only one binary variable in nodes  $n_j$  and  $n_k$  is different from their parent node  $n_i$ . The optimal solution from parent node is used to initialise the NLP subproblems in its child nodes. The child nodes will be explored again using FPBB-HB algorithm in Figure 2.

The FPBB-HB algorithm is implemented in Python. The flowsheet models are built in the equation orientated environment (Aspen Custom Modeler). Data exchanging with Python is via its automation interface. As gradient-based algorithms usually demonstrate better performance when solving large-scale NLP problems with a wide range of constraints and variables, open source optimiser slsqp (Kraft, 1988) is used to solve each NLP subproblem.

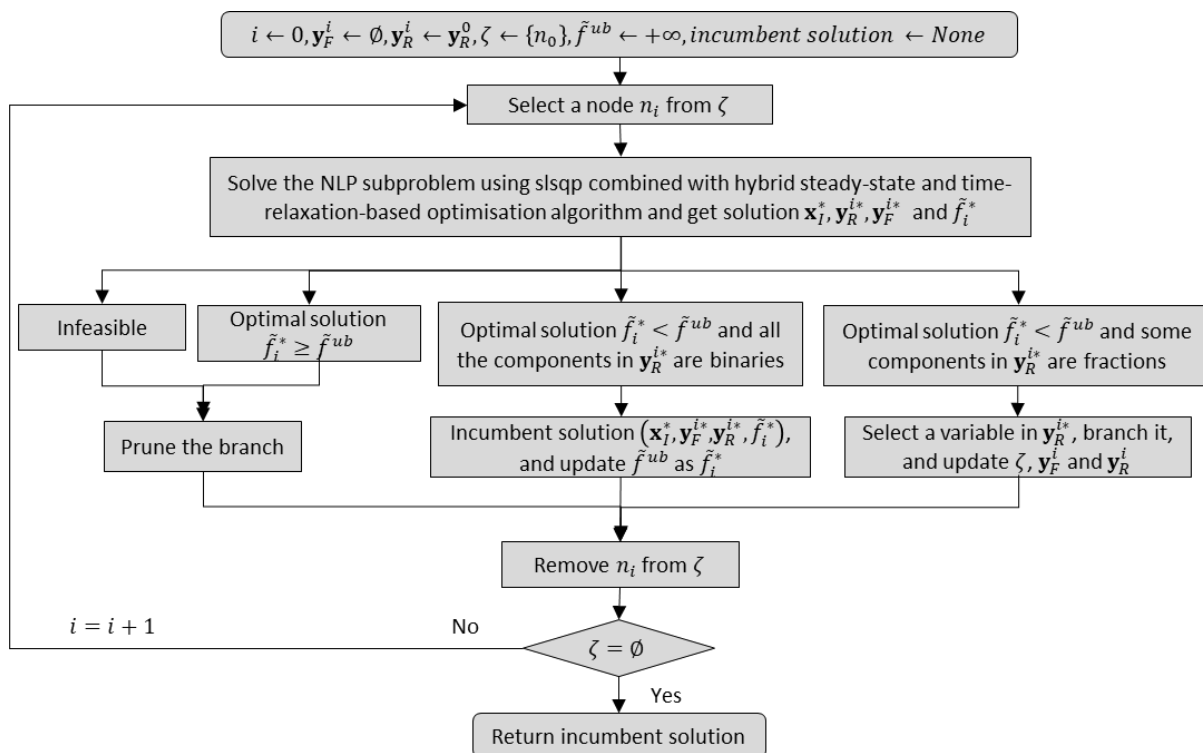


Figure 2 FPBB-HB algorithm

#### 4. Computational Study

To illustrate the capability and superiority of the proposed FPBB-HB algorithm, one example is taken from Montonati et al. (2022). In this example, a dividing wall column (DWC) is used for the separation of a ternary mixture consisting of n-pentane, n-hexane and n-heptane. DWC is a practical implementation which integrates two columns into a single shell by the addition of a wall that physically separates the feed side from the side product draw off section. It has been proven that up to 30% energy savings can be achieved together with space and capital cost reduction (Babi et al., 2016). Optimal design of DWC is not straightforward since it possesses a complex structure with additional design parameters to be considered, such as vapour and liquid split ratio and side draw flowrate.

Table 1 provides information on feed conditions, product specifications, and column operating pressure which is fixed in this example. The Peng-Robinson model is selected to calculate phase equilibrium. Medium pressure steam is utilised as hot utility.

Table 1 Data for hydrocarbon mixture

Mixture components A/B/C	n-pentane/n-hexane/n-heptane
Composition mole %	40/20/40
Pressure (atm)	2
Product purities	0.99/0.92/0.99
Feed flowrate (kmol h <sup>-1</sup> )	100
Feed condition	Saturated liquid

The mathematical model for DWC is built using MESH equations and rigorous models. The bypass efficiency of each tray is treated as binary variables and the summation of them equals to the number of stages. The decision variables include bypass efficiency, distillate flow rate, reboiler vaporization fraction, side draw fraction, liquid and vapour split ratio. We use the proposed FPBB-HB algorithm to solve this example. The optimisation problem involves 180 binary variables, 21,167 continuous variables and 18,965 constraints. We generate a locally optimal solution of 43,344 \$ y<sup>-1</sup> within 2712 CPU s. The optimal results are provided in Table 2. From Table 2, it can be observed that in the optimal design, there are 46 trays in the main column and 23 trays in the prefractionator column. The optimal design is illustrated Figure 3. The computational performance of the proposed FPBB-HB algorithm is presented in Table 3. From Table 3, it shows 8 nodes are created and

investigated during B&B, and all the 8 NLP subproblems have been solved successfully. Interestingly, during B&B, three feasible integer solutions are found. Two nodes are pruned due to the optimal solution is larger than the upper bound. In the final suboptimal integer solution, all the bypass efficiencies are at 0 or 1.

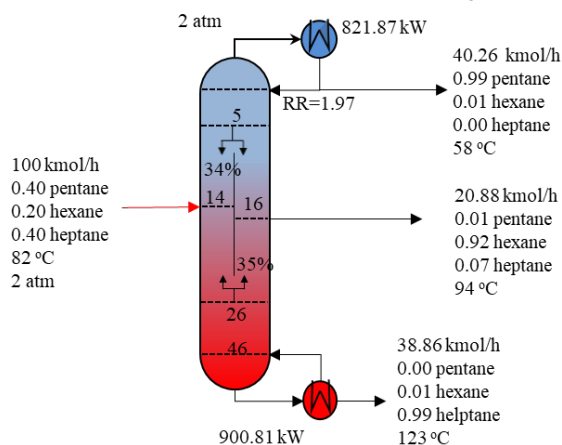


Figure 3 Optimal design of the DWC

Table 2 Optimal results comparison for case study

Column	DWC(Molecular tracking)		DWC(FPBB-HB)	
	Pre*	Main	Pre	Main
Number of stages	23	46	21	46
Feed stage/	14	9, 33	14	5, 26
Interconnection stages				
Side stream stages		23		16
Reflux ratio		1.95		1.968
Liquid to pre(kmol/h)		33.5		24.9
Vapour to pre(kmol/h)		84.5		70.2
Condenser duty(kW)		815.8		821.87
Reboiler duty(kW)		898.6		900.81
Total Investment Cost (\$)		134,130		134,185
Total Operating Cost (\$/y)		29,844		29,926
Total Annual Cost (\$/y)		43,258		43,344

\*Pre – Prefractionator Column

Table 3 Nodes evaluated and computational performance of FPBB-HB algorithm for DWC example

Item	FPBB-HB
CPU time (s)	2,712
Number of nodes created	8
NLP subproblem solved	8
Number of feasible integer solutions	3
Number of nodes pruned	2

We also compare the optimal design with that from literature, which is generated using the special procedures. The comparative results are also provided in Table 2. From Table 2, it can be observed that the configuration obtained by using FPBB-HB is very similar to the configuration in the literature. The method used in the literature is a special procedure, which requires short-cut models to determine the initial design parameters of DWC, followed by equilibrium-staged simulation, molecular tracking application and finally using many simulations to investigate the effects of varying parameters in the flowsheet to find an optimal design. The design procedure is tailored and computational intensive for design of DWC. By applying our algorithm to the same case, the total computational time is 2,712 CPU s. We also solve this example using some commercial solvers including DICOPT (outer approximation method), BARON (Branch and reduce algorithm) and ANTIGONE (special branch and bound algorithm). It is found that all commercial solvers fail to find a feasible solution within 1 h.

## 5. Conclusions

In this work, a novel feasible-path based branch and bound algorithm is presented, where the branch and bound method is used to systematically fix part of the binary variables and generate relaxed NLP subproblems, while the hybrid steady state and time relaxation-based feasible path algorithm is used to solve the derived NLP subproblems. In the generated B&B tree, the solution from the parent node is provided as a warm start for the NLP problems at the child nodes, which increases the computational efficiency. The computational results of optimising the DWC configuration incorporating rigorous models demonstrate that the proposed algorithm can find a locally optimum with very good convergence performance. The optimal configuration of DWC for separation of a ternary mixture is found within a reasonable time. It should be noted that the proposed algorithm cannot guarantee a global optimum since the nonconvex NLP subproblems are solved to local optimality for efficiency. In the future, this novel algorithm can be applied to solving more process intensification examples such as reactive distillation and biotechnology applications. It can also be tested on the process synthesis coupled with superstructures.

## Acknowledgement

Chao Liu would like to appreciate financial support from The University of Manchester. Yingjie Ma acknowledges financial support from China Scholarship Council – The University of Manchester Joint Scholarship (201809120005).

## References

- Babi D.K., Cruz M.S., Gani R., 2016, Fundamentals of Process Intensification: A Process Systems Engineering View, *Process Intensification in Chemical Engineering*, 7–33, doi: 10.1007/s11081-021-09707-y
- Burre J., Bongartz D., Mitsos A., 2022, Comparison of MINLP formulations for global superstructure optimization, *Optimization and Engineering*, doi: 10.1007/s11081-021-09707-y
- Dowling, A.W., Biegler, L.T., 2015, A framework for efficient large scale equation-oriented flowsheet optimization, *Computers & Chemical Engineering*, 72, 3–20.
- Geoffrion, A.M., 1972, Generalized benders decomposition, *Journal of optimization theory and applications*, 10(4), 237-260.
- Gupta, O.K., Ravindran, A., 1985, Branch and Bound Experiments in Convex Nonlinear Integer Programming, *Management science*, 31(12), 1533–1546.
- Kraft, D., A software package for sequential quadratic programming, 1988, Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Center – Institute for Flight Mechanics, Köln, Germany.
- Kronqvist, J., Lundell, A., Westerlund, T., 2015, The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming, *Journal of global optimization*, 64(2), 249–272.
- Kronqvist, Bernal, D. E., Lundell, A., & Grossmann, I. E., 2018, A review and comparison of solvers for convex MINLP, *Optimization and Engineering*, 20(2), 397–455.
- Kronqvist, J., A. Lundell, 2019, Convex Minlp – An Efficient Tool for Design and Optimization Tasks?, *Computer Aided Chemical Engineering*, 47, 245-250.
- Lee, S., Grossmann, I.E., 2000, New algorithms for nonlinear generalized disjunctive programming, *Computers & Chemical Engineering*, 24(9), 2125–2141.
- Ma, Y., McLaughlan M., Zhang N., Li J., 2020, Novel Feasible Path Optimisation Algorithms Using Steady-State And/or Pseudo-Transient Simulations, *Computers & Chemical Engineering*, 143, 107058.
- Ma, Y., Li, J., 2022, Homotopy continuation enhanced branch and bound algorithms for strongly nonconvex mixed-integer nonlinear optimization, *AIChE Journal*, 68, 6, doi: 10.1002/aic.17629
- Montonati, G., Nazemzadeh N., Abildskov J., Mansouri S.S, 2022, Divided-wall Distillation Column Design Using Molecular Tracking, *AIChE Journal*, 68, 2, doi: 10.1002/aic.17504
- Raman, R., Grossmann, I., 1994, Modelling and computational techniques for logic based integer programming, *Computers & Chemical Engineering*, 18(7), 563–578.
- Trespacios, F., Grossmann, I.E., 2014, Review of Mixed-Integer Nonlinear and Generalized Disjunctive Programming Methods, *Chemie ingenieur technik*, 86(7), pp.991–1012.
- Türkay, M., Grossmann, I.E., 1996, Logic-based MINLP algorithms for the optimal synthesis of process networks, *Computers & Chemical Engineering*, 20(8), 959–978.