Homotopy Continuation Enhanced Branch and Bound Algorithm for Process Synthesis using Rigorous Unit Operation Models

Yingjie Ma, and Jie Li*

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

Abstract

Process synthesis using rigorous unit operation models is highly desirable to identify the most efficient pathway for sustainable production of fuels and value-added chemicals. However, it often leads to a large-scale strongly nonlinear and nonconvex mixed integer nonlinear programming (MINLP) model. In this work, we propose two robust homotopy continuation enhanced branch and bound (HCBB) algorithms (denoted as HCBB-FP and HCBB-RB) where the homotopy continuation method is employed to gradually approach the optimal solution of the NLP subproblem at a node from the solution at its parent node. A variable step length is adapted to effectively balance feasibility and computational efficiency. The computational results demonstrate that the proposed HCBB algorithms can find the same optimal solution from different initial points, while the existing MINLP algorithms fail or find much worse solutions. In addition, HCBB-RB is superior to HCBB-FP due to lower computational effort required for the same locally optimal solution.

Keywords: Process synthesis, homotopy continuation, branch and bound, MINLP, rigorous models

---

* Corresponding author: Jie Li (Email address: jie.li-2@manchester.ac.uk). Tel: + 44 (0) 161 306 8622. Fax: +44 (0) 161 236 7439

1 Introduction

Global warming and the resulting detrimental and irreversible environmental effects have become a consensus around the world. In a response, countries representing more than 65% of greenhouse gases and more than 70% of world economy have recently announced to achieve carbon neutrality by 2050 or 2060[1]. To achieve such an ambitious goal, chemical and energy sectors need to develop new processes with clean and renewable resources such as bio-based feedstock, electrolytic hydrogen, and solar energy for sustainable production of fuels and high-valued chemicals. Process synthesis plays a key role in designing sustainable chemical processes, which is to determine the optimal interconnections of processing units as well as the optimal type and design of the units within a process system[2]. Hence, process synthesis provides opportunities to address above challenges[3].

The existing methods for process synthesis include evolutionary methods, decomposition-based methods, and simultaneous optimisation-based approaches. In the evolutionary method, heuristics and engineering judgement are used to enhance an existing flowsheet by adding or modifying process units one at a time until no further improvement can be made[4,5]. The decomposition-based method divides the process synthesis problem into several sub-tasks or levels and synthesizes the smaller problems in each sub-task or level sequentially[6-8]. Although it can be efficient to provide a good solution, it cannot manage interactions between different subproblems[9], leading to suboptimality. The simultaneous optimisation-based approach often constructs a superstructure incorporating different flowsheet alternatives, reformulates the superstructure as a mathematical programming problem and then solves the derived optimisation problem[9-11]. In this approach, short-cut unit operation models such as the Fenske-Underwood-Gilland method[12] for distillation columns are often used, which can reduce complexity of the derived optimisation problem. However, it tends to introduce inaccuracy in design results and economic evaluation.[13] To avoid this, the use of rigorous unit operation models is crucial to accurately predict process performance and guarantee design results match the real-world production[14].

In the earlier stage, synthesis of the reactor network[15,16] and distillation sequence[17,18] using rigorous models was conducted separately, which cannot consider the interactions between the reactor network and separation network. Therefore, simultaneous synthesis of the reactor and separation networks (i.e., the whole process) using rigorous models is highly desired. However, the simultaneous synthesis using rigorous operation models usually leads to a large-scale strongly nonlinear and nonconvex mixed integer nonlinear programming (MINLP) problem, which is challenging to solve. Different optimisation approaches have been proposed for process synthesis problems using rigorous models. Recker et al.[19] presented a two-stage approach where short-cut models are first used to identify several good process flowsheet variants, and the identified candidates are then optimized using rigorous models. However, this approach may miss the really optimal flowsheet at the initial screening phase[20]. Gross and Roosen[21] synthesized the hydrodealkylation (HDA) process of toluene using evolutionary algorithms based on rigorous Aspen Plus simulation[22]. Their algorithm needs long computational time and cannot guarantee optimality of the solution[23]. Henao and Maravelias[24] proposed a surrogate-based superstructure optimisation framework for process synthesis, where the surrogate models obtained through fitting simulation data from rigorous models are used for optimisation. However, the accuracy of the surrogate model is hard to be guaranteed[25] and the determination of the sampling bounds of intermediate variables is nontrivial[26]. Smith & Pantelides[27] proposed a state operator network with rigorous models for reactor and distillation columns embedded for the synthesis of the whole process using rigorous unit operation models, which was solved by the enumeration method. Although the proposed superstructure is general, it can have significant convergence difficulties due to the additional non-convexities taken by interconnection equations[28].

Although the well-known global optimizer, such as BARON[29] and ANTIGONE[30] can be used to solve process synthesis problems directly to global optimality, they cannot converge to the required tolerance or even find a feasible solution within acceptable computational time for such large-scale strongly nonlinear and nonconvex problems[31,32]. Thus, the outer approximation (OA) algorithm[33], the logic-based outer approximation algorithm (L-bOA)[34], the standard branch and bound (B&B)

algorithm,[35] and the successive relaxed MINLP (SRMINLP)[36] method are widely used to solve such difficult MINLP problems in process synthesis. This is because they can usually provide a locally optimal solution within acceptable time. Zhang et al.[20] synthesized the reaction-separation-recycle processes with GDP models, which were reformulated as an MINLP problem and then solved using the standard branch and bound algorithm. Ma et al.[37] proposed a superstructure for synthesizing reaction-separation-recycle processes with a smaller number of binary variables. The successive relaxed MINLP method was employed to solve the derived model, leading to better solutions, higher computational efficiency and superior numerical robustness[37]. Recently, Pedrozo et al.[38] synthesized a large-scale ethylene and propylene coproduction plant with GDP models, which was solved using the logic-based outer approximation algorithm.

The standard B&B, OA, L-bOA and SRMINLP algorithms need to solve a plethora of nonlinear programming (NLP) subproblems. When the NLP subproblems are strongly nonlinear, nonconvex and ill conditioned, which are not rare when rigorous operation models including enthalpy balance equations, non-ideal physical property equations and complex reaction kinetic equations are used[39], it is vulnerable for them to diverge, although an optimal solution exists. As a result, the standard B&B, OA, L-bOA and SRMINLP algorithms often fail to provide a feasible solution or converge to a locally optimal solution with low-quality. Although Ma et al.[37] has demonstrated that the SRMINLP method performs better than the standard B&B, and OA algorithms in some cases, it usually introduces some special constraints to enforce integrality of the relaxed binary variables, leading to the isolated feasible regions[36]. As a result, locally optimal solutions or even infeasibilities are generated with great dependence on the initialization[13]. Therefore, it is of great importance to guarantee the NLP subproblems are solved reliably without introducing additional challenging constraints when solving strongly nonlinear and nonconvex MINLP problems.

In this work, we propose two robust Homotopy Continuation enhanced B&B (HCBB) algorithms for chemical process synthesis problems using rigorous unit operation models. In these algorithms, the homotopy continuation (HC) method is employed to improve the convergence and efficiency of solving the NLP subproblem at each node through constructing a new homotopy path using the solution obtained at its parent node. This new homotopy path is constrctured through exploiting the similarity of the NLP subproblems generated during B&B, which is completely different from the existing homotopy path constructed through modifying the Karush-Kuhn-Tucker (KKT) conditions of the NLP problem[47,48]. During HC, an adaptive variable-step length method is used to balance the convergence and computational efficiency, and three match conditions are proposed to exploit the successful HC step lengths at previous nodes for higher efficiency. While the first HCBB algorithm (denoted as HCBB-FP) solves a feasibility problem with a fixed value for the homotopy variable in each HC step, the other one (called HCBB-RB) solves an optimality problem with a relaxed lower or upper bound of the homotopy variable. Four benchmark examples from literature are solved to evaluate the capability of the proposed HCBB algorithms. The computational studies demonstrate that the proposed HCBB algorithms solve all examples to the same high-quality local optimum from different initial points, whilst all other existing algorithms including the standard branch and bound algorithm in the GAMS/SBB solver[40], the OA algorithm in the GAMS/DICOPT solver[33], and the SRMINLP method from Ma et al.[37] fail or find much worse locally optimal solutions. In addition, HCBB-RB is able to find almost the same locally optimal solution with much lower computational effort compared to HCBB-FP.

The rest of the paper is organized as follows: The next section defines the problem to be solved. Section 3 briefly describes the MINLP models of the problem. In section 4, we propose the HCBB algorithms in detail. In section 5, four process synthesis problems are solved to illustrate the convergence and efficiency of the proposed algorithms with a fair comparison with existing algorithms. Finally, we conclude this work with some useful insights.

2 Problem statement

A process synthesis problem seeks to develop systematically process flowsheets that convert raw materials into desired products. Optimisation-based process synthesis requires the development of a network of interconnected units, the process superstructure, that represents the alternative process flowsheets. Fig. 1 illustrates a typical example of superstructure for a reaction-separation-recycle system, which is composed of a reactor network and a separation network for ternary separation. The problem in the paper is stated as follows,

Given:

- A superstructure of the process synthesis problem
- Specific raw material, product and production requirements, e.g., production rates and product purities.
- The number of reactors in the reactor network, suitable reactor types, reaction kinetics;
- The number of components in a process stream and their thermodynamic properties;
- The number of distillation columns, and maximum number of trays in each column;

Determine:

- Optimal flowsheet structure and reactor types;
- Optimal operating conditions including stream flow rates, pressures, and temperatures;
- Optimal sizes of reactors, distillation columns and heat exchangers.

The objective is to minimize total annualized cost (TAC) or maximize profit.

3 Mathematical formulation

The superstructure for a process synthesis problem can be systematically modelled using GDP, which can then be converted into an equivalent MINLP problem, as presented as follows:

$$\min_{\mathbf{x},\mathbf{y}} f\,(\mathbf{x},\mathbf{y}) \tag{P}$$

$$s.t.\ \mathbf{h}(\mathbf{x},\mathbf{y}) = 0$$

$$\mathbf{g}(\mathbf{x},\mathbf{y}) \leq 0$$

$$\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{Z}^m,$$

where $\mathbf{x}$ is a vector of n continuous variables, i.e., $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, and $\mathbf{y}$ is a vector of m integer variables, i.e., $\mathbf{y} = [y_1, y_2, \dots, y_m]^T$. Here, we restrict $\mathbf{y}$ to binary variables without loss of generalization as an integer variable can be expressed by binary variables[41]. f is a scalar function, whilst $\mathbf{g}$ and $\mathbf{h}$ are vector functions. Due to rigorous unit operation models used, problem P is usually highly nonlinear and nonconvex, which is challenging to generate an optimal solution or even a feasible solution.

The relaxation of the problem P is provided below,

$$\min_{\mathbf{x},\tilde{\mathbf{y}}} f\,(\mathbf{x},\tilde{\mathbf{y}}) \tag{RP}$$

$$s.t.\ \mathbf{h}(\mathbf{x},\tilde{\mathbf{y}}) = 0$$

$$\mathbf{g}(\mathbf{x},\tilde{\mathbf{y}}) \leq 0$$

$$\mathbf{x} \in \mathbb{R}^n, \tilde{\mathbf{y}} \in \textstyle\prod_1^m [0,1]$$

We assume that $f(\cdot,\tilde{\mathbf{y}})$, $\mathbf{g}(\cdot,\tilde{\mathbf{y}})$, $\mathbf{h}(\cdot,\tilde{\mathbf{y}})$ are twice differentiable for a given $\tilde{\mathbf{y}}$, which is a common assumption for many NLP optimisation algorithms.[42] For convenience, we introduce some notations that are used in this work. We define an index set of binary variables as $\mathcal{S} = \{s \mid s = 1, 2, \dots, m\}$. The active nodes during B&B are included in a set $\mathcal{L} = \{n_0, n_1, n_2, \dots\}$, where n_i denotes the ith node that needs to be explored in B&B. Before solving the NLP subproblem at a node n_i, we know the values of m_F^i binary variables, whose indices are represented in a set $\mathcal{S}_F^i = \{s_1, s_2, \dots, s_{m_F^i}\}$. These binary variables are $\mathbf{y}_F^i = \left[y_{s_1}, y_{s_2}, \dots, y_{s_{m_F^i}}\right]^T$. The remaining m_R^i binary variables are unknown and relaxed to 0-1 continuous variables that needs to be determined. The indices of these binary variables are included into a set $\mathcal{S}_R^i = \{l_1, l_2, \dots, l_{m_R^i}\}$. These binary variables are $\mathbf{y}_R^i = \left[y_{l_1}, y_{l_2}, \dots, y_{l_{m_R^i}}\right]^T$. Therefore, we have $\mathcal{S}_F^i \cup \mathcal{S}_R^i = \mathcal{S}$ and $m_F^i + m_R^i = m$. We also use $f^{i,*}$ to denote the optimal objective value of the NLP subproblem at node n_i, f^{ub} to denote the upper bound of the MINLP problem. Note that the upper bound is a feasible integer solution for the MINLP problem P.

4 Solution approach

As discussed before, the standard B&B algorithm[43] has better performance in convergence for solving the strongly nonlinear and nonconvex MINLP problem **P**, which is presented in **Algorithm S1** of the **Supplementary Material**. It requires to solve a series of NLP subproblems, which, however, cannot be solved reliably by existing NLP optimisation algorithms and hence result in infeasibility of the standard B&B algorithm due to highly nonlinear, nonconvex and ill conditioned rigorous operation models used[39] in process synthesis problems. We notice that the NLP subproblem that needs to be solved at a node $\boldsymbol{n_i}$ is quite similar to that at its parent node. The small difference is the binary variable that is selected for branching at the parent node, whose value is known and fixed at 0 or 1 at the node $\boldsymbol{n_i}$. Therefore, using the solution obtained from the parent node as the initial point and gradually approaching the solution of the NLP subproblem at the node $\boldsymbol{n_i}$ may make it easier or more reliable to generate a feasible or optimal solution of the NLP subproblem at node $\boldsymbol{n_i}$ than directly solving the NLP subproblem at this node. This strategy can be achieved by using the HC method, which actually starts from a problem that is easier to solve or has already been solved to approach the solution of the original problem gradually[44]. The HC method has been widely used to solve difficult nonlinear algebraic equation systems[45,46] and NLP optimisation problems[47,48] where a homotopy path is constructed through modifying the KKT conditions of the NLP problem. In the sequel, we introduce the HC method in detail and use it to construct a new homotopy path through exploiting the similarity of the NLP subproblems during B&B to enhance convergence of solving the NLP subproblem at a node. Therefore, the new homotopy path is completely different from those in literature[47,48].

4.1 Homotopy continuation method

An NLP subproblem to be solved at a node n_i in the standard B&B algorithm is defined below,

$$\min_{\mathbf{x},\mathbf{y}_R^i} f\left(\mathbf{x},\mathbf{y}_R^i;\mathbf{y}_F^i\right) \qquad \text{(NLP0)}$$

$$s.t.\ \mathbf{h}\left(\mathbf{x},\mathbf{y}_R^i;\mathbf{y}_F^i\right) = 0$$

$$\mathbf{g}\left(\mathbf{x},\mathbf{y}_R^i;\mathbf{y}_F^i\right) \le 0$$

$$\mathbf{x} \in \mathbb{R}^n, \mathbf{y}_R^i \in \mathbb{B}_{0,1}^{i,R}, \mathbf{y}_F^i \in \mathbb{Z}^{m_F^i}$$

where $\mathbb{B}_{0,1}^{i,R} \coloneqq \prod_1^{m_R^i}[0,1] \subset \mathbb{R}^{m_R^i}$ is a rectangular region, and $\mathbf{y}_F^i$ are binary variables with known and fixed values a priori. The NLP subproblem at a node n_i is denoted as NLP0$(\mathbf{x}, \mathbf{y}_R^i; \mathbf{y}_F^i)$ with the optimal solution notated as $\left(\mathbf{x}^{i,*}, \mathbf{y}_R^{i,*}\right)$. Evidently, the NLP subproblem at node n_i has slight difference in $\mathbf{y}_F^i$ and $\mathbf{y}_R^i$ with that at its corresponding parent node (denoted as n_{pi}). Hence, we can use the solution obtained at node n_{pi} to gradually approach the solution at node n_i, leading to the construction of homotopy paths. After the NLP subproblem at the parent node n_{pi} is solved, a variable in $\mathbf{y}_R^{pi,*}$ with a fraction value is selected for branching. We use b_{pi} to denote the index of the variable for branching. It means $y_{b_{pi}}^{pi,*}$ is used for branching and hence two children nodes n_i and n_j are created. At the child node n_i, the variable in $\mathbf{y}_R^{pi}$ whose index is b_{pi} (i.e., $y_{b_{pi}}^i$) has a known and fixed value at 0 or 1. Therefore, it is removed from $\mathbf{y}_R^i$ but included in $\mathbf{y}_F^i$ when the NLP subproblem at node n_i is solved. Similarly, the variable $y_{b_{pi}}^j$ at the child node n_j has a known and fixed value at 1 or 0. Therefore, it is also removed from $\mathbf{y}_R^j$ but included in $\mathbf{y}_F^j$. This is illustrated in Fig. 2.

To ensure the feasibility or optimality of the NLP subproblem at node n_i, we gradually approach the value of $y_{b_{pi}}^i$ at node n_i (i.e., 0 or 1) based on the value of $y_{b_{pi}}^{pi,*}$ using a homotopy parameter $t \in [0,1]$, as showin in Eq. (1).

$$\tilde{y}_{b_{pi}}^i(t) = (1-t)\cdot y_{b_{pi}}^{pi,*} + t \cdot y_{b_{pi}}^i \qquad t \in [0,1] \tag{1}$$

where $\tilde{y}_{b_{pi}}^i(t)$ is called the homotopy variable. When $t = 0$, $\tilde{y}_{b_{pi}}^i(t) = y_{b_{pi}}^{pi,*}$, whilst $\tilde{y}_{b_{pi}}^i(t) = y_{b_{pi}}^i$ when $t = 1$.

Based on this, the binary variables with known values at node n_i can be generally represented as follows,

$$\tilde{\mathbf{y}}_F^i(t) = \mathbf{y}_F^i + (1-t)\left(y_{b_{pi}}^{pi,*} - y_{b_{pi}}^i\right)\boldsymbol{e}_{b_{pi}} \qquad \tilde{\mathbf{y}}_F^i \in \mathbb{B}_{0,1}^{i,F} \coloneqq \prod_1^{m_F^i}[0,1] \tag{2}$$

where $\boldsymbol{e}_{b_{pi}}$ is a column vector with the dimension of m_F^i whose entries are all 0 except 1 at b_{pi}.

An NLP subproblem with respective to the parameter t can be constructed as follows,

$$\min_{\mathbf{x},\mathbf{y}_R^i} f\left(\mathbf{x},\mathbf{y}_R^i;\tilde{\mathbf{y}}_F^i(t)\right) \qquad \text{(NLPFX)}$$

$$s.t.\ \mathbf{h}\left(\mathbf{x},\mathbf{y}_R^i;\tilde{\mathbf{y}}_F^i(t)\right) = 0$$

$$\mathbf{g}\left(\mathbf{x},\mathbf{y}_R^i;\tilde{\mathbf{y}}_F^i(t)\right) \leq 0$$

$$\mathbf{x} \in \mathbb{R}^n, \mathbf{y}_R^i \in \mathbb{B}_{0,1}^{i,R}, \tilde{\mathbf{y}}_F^i \in \mathbb{Z}^{m_F^i}, t \in [0,1]$$

where $\left(\bar{\mathbf{x}}^i,\bar{\mathbf{y}}_R^i\right)$ are the solution of problem NLPFX for a given t. For convenience, the parametric optimisation problem is notated as NLPFX$\left(\mathbf{x},\mathbf{y}_R^i;t\right)$. Clearly, when $t = 0$ the optimal solution $\left(\mathbf{x}^{pi,*},\mathbf{y}_R^{pi,*}\right)$ obtained at node n_{pi} is also the optimal solution of the subproblem NLPFX($\mathbf{x},\mathbf{y}_R^i;t$). When $t = 1$, the problem NLPFX$\left(\mathbf{x},\mathbf{y}_R^i;t\right)$ is equivalent to problem NLP0$\left(\mathbf{x},\mathbf{y}_R^i;\mathbf{y}_F^i\right)$. Let a function $F_{FX}^i(t)$ denotes the mapping from t to the optimum of problem NLPFX$\left(\mathbf{x},\mathbf{y}_R^i;t\right)$. In other words,

$$F_{FX}^i(t) = \min_{\mathbf{x},\mathbf{y}_R^i} f\left(\mathbf{x},\mathbf{y}_R^i;\tilde{\mathbf{y}}_F^i(t)\right)$$

$$s.t.\ \mathbf{h}\left(\mathbf{x},\mathbf{y}_R^i;\tilde{\mathbf{y}}_F^i(t)\right) = 0$$

$$\mathbf{g}\left(\mathbf{x},\mathbf{y}_R^i;\tilde{\mathbf{y}}_F^i(t)\right) \leq 0$$

$$\mathbf{x} \in \mathbb{R}^n, \mathbf{y}_R^i \in \mathbb{B}_{0,1}^{i,R}, \tilde{\mathbf{y}}_F^i \in \mathbb{Z}^{m_F^i}, t \in [0,1]$$

We assume $F_{FX}^i(t)$ is continuous with respective to t within the interval $t \in [0,1]$, which is a mild assumption and satisfied in most cases. Then, the optimal solution of NLPFX$\left(\mathbf{x},\mathbf{y}_R^i;t^\nu\right)$ provides a good initial point for problem NLPFX$\left(\mathbf{x},\mathbf{y}_R^i;t^{\nu+1}\right)$ if t^ν and $t^{\nu+1}$ are close enough, i.e., $\Delta t^\nu \coloneqq t^{\nu+1} - t^\nu$ is small enough where ν denotes the iteration. The homotopy paths of the optimum $F_{FX}^i(t)$ and $F_{FX}^j(t)$ from solving problem NLPFX $\left(\mathbf{x},\mathbf{y}_R^i;t\right)$ and NLPFX$\left(\mathbf{x},\mathbf{y}_R^j;t\right)$ at nodes n_i and n_j created from the same parent node are illustrated in Fig. 3 (left).

The subproblem NLPFX$\left(\mathbf{x},\mathbf{y}_R^i;t\right)$ needs to be solved to optimality from $t = 0$ until $t = 1$, which could significantly increase the computational expense in some cases. As we only need to obtain the optimal solution when $t = 1$ at a node, it is unnecessary to generate the optimal solution for any $t \in (0,1)$. Based on this insight, a variant of applying the HC method is proposed which first solves a series of NLP feasibility problems (denoted as NLPFP) with a constant objective function (e.g., 0) for $t \in (0, 1]$ during HC and then solve the problem NLP0 to optimality when $t = 1$ at node n_i.

$$\min_{\mathbf{x},\mathbf{y}_R^i} c \qquad \text{(NLPFP)}$$

$$s.t.\ \mathbf{h}\left(\mathbf{x}, \mathbf{y}_R^i; \tilde{\mathbf{y}}_F^i(t)\right) = 0$$

$$\mathbf{g}\left(\mathbf{x}, \mathbf{y}_R^i; \tilde{\mathbf{y}}_F^i(t)\right) \leq 0$$

$$\mathbf{x} \in \mathbb{R}^n, \mathbf{y}_R^i \in \mathbb{B}_{0,1}^{i,R}, \tilde{\mathbf{y}}_F^i \in \mathbb{Z}^{m_F^i}, t \in (0, 1]$$

where c is a constant. The problem NLPFP is specifically notated as NLPFP$\left(\mathbf{x}, \mathbf{y}_R^i; t\right)$.

When the above two HC variants solve NLP subproblems from $t = 0$ to $t = 1$, both of them cannot use the information of the current lower or upper bounds identified to determine if the current node is fathomed during homotopy. This is because the optimum of both problems NLPFX and NLPFP does not increase monotonically with t. A more advantageous variant is designed to gradually tighten the bound of $\tilde{y}_{b_{pi}}^i(t)$ with the increase of t, which makes the optimum does monotonically increase with t. Once the optimal objective value is greater than the current upper bound (minimization problems) at some t ($t < 1$), it is no need to continue the HC to approach t = 1. In other words, this node is fathomed. The NLP subproblem that needs to be solved in this variant is provided below, which is denoted as NLPRB.

$$\min_{\mathbf{x},\mathbf{y}_R^i,\tilde{y}_{b_{pi}}^i} f\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; \mathbf{y}_F^i\right) \qquad \text{(NLPRB)}$$

$$s.t.\ \mathbf{h}\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; \mathbf{y}_F^i\right) = 0$$

$$\mathbf{g}\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; \mathbf{y}_F^i\right) \leq 0$$

$$\mathbf{x} \in \mathbb{R}^n, \mathbf{y}_R^i \in \mathbb{B}_{0,1}^{i,R}, \mathbf{y}_F^i \in \mathbb{Z}^{m_F^{pi}}$$

where $\mathbf{y}_F^i = \mathbf{y}_F^{pi}$, $\tilde{y}_{b_{pi}}^i \in \left[(1-t)\cdot y_{b_{pi}}^{pi,*} + t\cdot y_{b_{pi}}^i, 1\right]$, $t \in [0,1]$, when $y_{b_{pi}}^i$ is required to be 1 at node i. Otherwise, $\tilde{y}_{b_{pi}}^i \in \left[0, (1-t)\cdot y_{b_{pi}}^{pi,*} + t\cdot y_{b_{pi}}^i\right]$, $t \in [0,1]$ when $y_{b_{pi}}^i$ is required to be 0 at node i. For convenience, the problem NLPRB for a given t is specifically notated as NLPRB $\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t\right)$. Similar to problem NLPFX, when $t = 0$, the optimal solution $\left(\mathbf{x}^{pi,*}, \mathbf{y}_R^{pi,*}\right)$ obtained at node n_{pi} is also the optimal solution of the subproblem NLPRB$(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t)$. When $t = 1$, problem NLPRB $\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t\right)$ is equivalent to NLP0$\left(\mathbf{x}, \mathbf{y}_R^i; \mathbf{y}_F^i\right)$. Let $F_{RB}^i(t)$ denotes the mapping from t to the optimum of problem NLPRB $\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t\right)$. With t approaching 1, the domain of $\tilde{y}_{b_{pi}}^i$ contracts gradually. Hence, function $F_{RB}^i(t)$ increases monotonically under the assumption that a global optimal solution is obtained at each HC step. That is $F_{RB}^i(t^{\nu+1}) \geq F_{RB}^i(t^{\nu})$ for any $t^{\nu}, t^{\nu+1} \in [0,1]$. If $F_{RB}^i(t)$ is larger than the upper bound of the MINLP problem during HC, there is no need to continue, and the current node can be pruned. This is because no better solution could be found at the current node. Therefore, the advantage of solving problem NLPRB $\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t\right)$ is that it is possible to terminate the HC calculation earlier. The homotopy paths of optimums [i.e., functions $F_{RB}^i(t)$ and $F_{RB}^j(t)$] when solving problems NLPRB $\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t\right)$ and NLPRB $\left(\mathbf{x}, \mathbf{y}_R^j, \tilde{y}_{b_{pj}}^j; t\right)$ are shown in Fig. 3 (right).

4.2 Adaptive step lengths during HC

In the HC method, the step length Δt^{ν} ($\Delta t^{\nu} = t^{\nu+1} - t^{\nu}$) can significantly affect the performance of the HC method. This is because if Δt^{ν} is too large, the previous solution at t^{ν} may not be a good initial point for the NLP subproblem at $t^{\nu+1}$ during homotopy, which could lead to infeasibility at $t^{\nu+1}$ and finally lead to infeasibility at $t = 1$. On the other hand, if Δt^{ν} is too small, many homotopy steps are required, resulting in increasing computational effort. To balance convergence and computational efficiency, some strategies proposed below are used to update the HC step length Δt adaptively.

Strategy 1: When a step length leads to feasibility or optimality of the NLP problem at the iteration

ν, it indicates that this step length is reliable and continues to be used for the next iteration $(\nu + 1)$. In other words, the step length at the next iteration can be updated using $\Delta t^{\nu+1} \leftarrow \Delta t^{\nu}$.

Strategy 2: When two consecutive NLP problems at iterations $(\nu - 1)$ and ν are solved using the same step length, it indicates a larger step length may also lead to a converged solution of the NLP problem at iteration $(\nu + 1)$. Therefore, we use a larger step length by updating $\Delta t^{\nu+1} \leftarrow 2\,\Delta t^{\nu}$.

Strategy 3: If the NLP subproblem at the current iteration ν fails with a step length, it means the step length is too large and should be reduced, so we set $\Delta t^{\nu+1} \leftarrow \frac{\Delta t^{\nu}}{2}$.

With the above three strategies, the HC algorithm using adaptive step lengths at a node n_i is proposed in the following **Algorithm 1**.

Algorithm 1: HC algorithm using adaptive step length at node n_i

```
Data: $0 < \Delta t_{min} < 1$, $N > 1$, $\mathbf{x}^{pi,*}$, $\mathbf{y}_R^{pi,*}$, $\mathbf{y}_F^{pi}$, $\mathbf{y}_R^{i}$, $\mathbf{y}_F^{i}$, $f^{ub}$
initialization: $\nu \leftarrow 1$, $t^0 \leftarrow 0$, $\Delta t^0 \leftarrow 1$, $t^1 \leftarrow 1$, set $\left(\mathbf{x}^{pi,*}, \mathbf{y}_R^{pi,*}, \mathbf{y}_F^{pi}\right)$ as the initial point;
while $\nu < N$ and $\Delta t^{\nu} > \Delta t_{min}$ do
    solve NLPRB$\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t^{\nu}\right)$ or NLPFP$(\mathbf{x}, \mathbf{y}_R^i; t^{\nu})$ from the given initial point; the optimal solution has $\bar{\mathbf{x}}^{\nu}$, $\bar{\mathbf{y}}_R^{\nu}$ and $\bar{f}^{\nu}$;
    if a feasible or optimal solution is generated then
        if $t == 1$ then break;
        else if the solution is optimal and the optimal value $\bar{f}^{\nu} > f^{ub}$ then break;
        else if $\nu = 1$ or $\Delta t^{\nu-1} \neq \Delta t^{\nu-2}$ then $\Delta t^{\nu} \leftarrow \Delta t^{\nu-1}$;
        else $\Delta t^{\nu} \leftarrow 2\,\Delta t^{\nu-1}$;
        $t^{\nu+1} \leftarrow \min(t^{\nu} + \Delta t^{\nu}, 1)$, the initial point ← the current solution;
    else
        $\Delta t^{\nu} \leftarrow \frac{\Delta t^{\nu-1}}{2}$, $t^{\nu+1} \leftarrow t^{\nu-1} + \Delta t^{\nu}$, initial point ← the last converged solution;
    $\nu = \nu + 1$;
if NLPFP$(\mathbf{x}, \mathbf{y}_R^i; t^{\nu})$ is solved and a feasible solution is obtained then
    solve the original problem NLP0 and get the final solution $(\mathbf{x}^{i,*}, \mathbf{y}_R^{i,*}, f^{i,*})$;
```

In this **Algorithm 1**, the initial step length Δt^0 is set as 1. $t^0 = 0$ and $t^1 = 1$, and we start from

$\nu = 1$, which means we solve the original problem $\mathrm{NLP0}\left(\mathbf{x}, \mathbf{y}_R^i; \mathbf{y}_F^i\right)$ at first.

In some situations many small step lengths may be required to solve NLP subproblems, which may increase computational effort. We notice that we may have similar information such as the same homotopy variable with the same required value at some nodes in the B&B. It indicates the step lengths at a previous node could be used when a series of NLP subproblems are solved at a latter node with similar information. With this, we may significantly reduce the effort for finding an appropriate step length. For instance, we have the same homotopy variable with the same required value at nodes n_i and n_j ($j < i$). At node n_j we already identify suitable step lengths, which can be adopted at node n_i and hence reduce the computational effort in finding suitable step lengths at node n_i. To identify nodes with similar information, the following conditions are proposed.

C1: $\mathcal{J}^i = \{j \mid b_{pj} = b_{pi} \text{ and } y_{b_{pj}}^j = y_{b_{pi}}^i, j < i\}$ is nonempty

C2: There exits $j \in \mathcal{J}^i$ such that $\Delta y_b^{i,j} \coloneqq \left|y_{b_{pj}}^{pj,*} - y_{b_{pi}}^{pi,*}\right| < \delta$;

C3: $j \leftarrow \operatorname{argmin}_{j \in \mathcal{J}^i} \Delta y_b^{i,j}$

C1 ensures that a node n_j has the same homotopy variable with the same required value as a node n_i. C2 enforces the homotopy variable at a node n_j starts from a value $y_{b_{pj}}^{pj,*}$ which is different from the starting point of the homotopy variable at a node n_i by at most a small constant δ. In other words, the homotopy variable at nodes n_i and n_j have a close starting point for HC. C3 is used to select a node with the closest starting point to that at node i if there are several nodes satisfying C1 and C2. When a node n_j satisfies the three conditions C1-C3, it is expected that the HC step lengths identified at node n_j can also successfully solve a series of NLP subproblems at node n_i during homotopy, so those steps will be used during the homotopy calculation at node n_i. However, once a step length leads to infeasibility of an NLP subproblem, it is then halved and the step length updating follows the same strategies as those in Algorithm 1. The HC algorithm using adaptive step lengths and C1-C3 at a node n_i is proposed in

Algorithm 2 as follows, where $\bar{t}^j$ and $\Delta\bar{t}^j$ represent historic steps and step lengths at a previous node j respectively.

Algorithm 2: HC algorithm using variable step length and C1-C3 at a node n_i

```
Data: $0 < \Delta t_{min} < 1, N > 1, \mathbf{x}^{pi,*}, \mathbf{y}_R^{pi,*}, \mathbf{y}_F^{pi}, \mathbf{y}_R^i, \mathbf{y}_F^i, \mathcal{T}$;
initialization: $\nu \leftarrow 1$, set $\left(\mathbf{x}^{pi,*}, \mathbf{y}_R^{pi,*}, \mathbf{y}_F^{pi}\right)$ as the initial point;
if there exists $j$ satisfying conditions (1)-(3) and $\left|\Delta y_b^{i,j}\right| < \delta$ then
    $t^1 \leftarrow \bar{t}^{j,1}$;
    while $t^\nu \leq 1$ do
        solve NLPRB($\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t^\nu$) or NLPFP($\mathbf{x}, \mathbf{y}_R^i; t^\nu$) from the given initial point; the optimal solution has $\bar{\mathbf{x}}^\nu$, $\bar{\mathbf{y}}_R^\nu$ and $\bar{f}^\nu$;
        if the solution is feasible or optimal then
            if $t^\nu == 1$ then break;
            else if the solution is optimal and the optimum $\bar{f}^\nu > f^{ub}$ then break;
            else $t^{\nu+1} \leftarrow \bar{t}^{j,\nu+1}$, initial point ← the current solution;
        else
            $\Delta t^\nu \leftarrow \frac{\Delta\bar{t}^{j,\nu-1}}{2}$, $t^{\nu+1} \leftarrow t^{\nu-1} + \Delta t^\nu$, initial point ← the last converged solution;
            switch to Algorithm 1 and start from its line 2; break;
        $\nu \leftarrow \nu + 1$;
else
    apply Algorithm 1;
```

4.3 Homotopy continuation enhanced branch and bound algorithm

The homotopy continuation enhanced branch and bound (HCBB) algorithm is shown in Fig. 4. In the HCBB algorithm, n_i is used to denote node i and $\mathcal{L}$ is the queue of nodes that need to be investigated. In the beginning, root node n_0 is assigned to the node queue $\mathcal{L}$. The lower bound (f^{lb}) and upper bound (f^{ub}) are initialized as $-\infty$ and $+\infty$, respectively. All binary variables are included in the set S_R^i with unknown values. At a node n_i, **Algorithm 2** is first executed to generate an optimal solution (denoted as $\mathbf{x}^*$, $\mathbf{y}_R^{i,*}$, $f^{i,*}$) or identify infeasibility. If infeasibility is returned, then the latest optimal or feasible solution with the values of the corresponding HC parameter t, and the last value of Δt during HC are recorded, which will be used for post check and refinement procedure explained in the next section 4.4. If an optimal solution is identified when solving NLPRB $\left(\mathbf{x}, \mathbf{y}_R^i, \tilde{y}_{b_{pi}}^i; t^\nu\right)$, then we need to check if the incumbent solution is greater than f^{ub}. If greater, then the current node can be discarded as no better optimal solution can be found under this node. Otherwise, we need to check the values of $y_R^{i,*}$. If all $y_R^{i,*}$ are 0 or 1, it means we find a feasible integer solution at this node. This node is fathomed and the upper bound is updated if the feasible integer solution is less than f^{ub}. Otherwise, branching on this node is conducted. A variable $y_R^{i,*}$ having a fraction value closest to 0.5 is selected to branch, which is a common-used rule[49]. The well-known best-first strategy[50] is used to select a node on which branching is conducted to create new nodes, which usually has better performance than the breadth-first strategy[51] and the depth-first strategy[51].

In addition, we allow the solution obtained from a parent node is used as an initial point for the NLP subproblem at its immediate children nodes. In other words, a warm start is used.

4.4 Post Check and Refinement Procedure

After the MINLP problem is solved successfully using the HC enhanced B&B algorithm in section 4.3, the optimal solution (denoted as $\mathbf{x}^*$, $\mathbf{y}^*$, f^*) may not be the global optimum. One of the reasons is $\Delta t_{\min}$ and the maximum number of HC steps (i.e., N_{max}) specified are reached at some nodes, leading to no feasible solution found at these nodes, whereas some feasible solution might be found if a larger N_{max} and a smaller $\Delta t_{\min}$ are used at these nodes, which may lead to a better solution obtained finally. Based on this consideration, a post check and refinement procedure is conducted at the nodes flagged as infeasibility. We use a set $\mathcal{J}$ to denote those nodes and some corresponding information at each node in $\mathcal{J}$ are also recorded during the above HCBB, which includes the latest optimal or feasible solution at $t^{i,\nu 1}$ (notated as $\mathbf{x}^{i,\#}, \mathbf{y}_R^{i,\#}, f^{i,\#}$), and the last step length $\Delta t^{i,\nu 2}$ ($\nu 2 \geq \nu 1$).

Post check and refinement procedure

```
Data: $\mathcal{J}, f^*$;
initialization: $\mathcal{L} \leftarrow \emptyset$;
while $\mathcal{J} \neq \emptyset$ do
    select a node $n_i$ from $\mathcal{J}$ and let $\mathcal{L} \leftarrow \{n_i\}$, set $f^{ub} = f^*$;
    If NLPRB is solve and $f^{i,\#} > f^{ub}$ then
        prune the current node;
    else
        $N \leftarrow 1000$, $\Delta t_{min} \leftarrow 1 \times 10^{-15}$, $t_0 \leftarrow t^{i,v1}$, $\Delta t_0 \leftarrow \Delta t^{i,v2}$, initial point $\leftarrow (\mathbf{x}^{i,\#}, \mathbf{y}_R^{i,\#})$;
        solve the problem NLPRB$(\boldsymbol{x}, \boldsymbol{y}_R^i, \tilde{y}_{b_{pi}}^i; t)$ or $NLPFP(\boldsymbol{x}, \boldsymbol{y}_R^i; t)$ using Algorithm 2.
        The optimal solution has $\boldsymbol{x}^{i,*}$, $\boldsymbol{y}_R^{i,*}$ and $f^{i,*}$;
        if the solution is infeasible then
            prune the current branch;
        else
            if $f^{i,*} \geq f^{ub}$ then
                prune the current branch;
            else if all components of $\boldsymbol{y}_R^{i,*}$ are binaries then
                incumbent solution $\leftarrow (\boldsymbol{x}^{i,*}, \boldsymbol{y}_R^{i,*}, f^{i,*})$, $f^{ub} \leftarrow f^{i,*}$, prune the branch;
            else
                select a $y_{b_i}^{i,*}$ with a fractional value in $y_R^{i,*}$. $\mathcal{L} \leftarrow \{n_0, n_1\}$, where both nodes
                have $\mathcal{S} \leftarrow \mathcal{S}_R^i \backslash b_i$ and $\mathcal{S}_F \leftarrow \mathcal{S}_F^i \cup b_i$ but $y_{b_i} = 0$ and 1 respectively;
                set $N$, $\Delta t_{min}$, $t_1$ and $\Delta t_0$ to default values and set the current solution to be
                the initial point;
                conduct B&B again with $\mathcal{J}$ adjusted accordingly;
```

During post check and refinement of HCBB algorithm solving NLPRB subproblems, we first compare $f^{i,\#}$ with f^{ub} (equal to the current optimum of the MINLP, f^*). If it is greater than f^{ub}, then this node is discarded and removed from $\mathcal{J}$ directly without solving any new NLP subprobem. This is called a post check step. Otherwise, a larger N (e.g., $N = 1000$) and a smaller $\Delta t_{\min}$ (e.g., $\Delta t_{\min} = 1 \times 10^{-15}$ which is around 10 times the machine accuracy for double-precision float point arithmetic[52]) are used to solve problem NLPRB starting from the recorded optimal or feasible solution. This is called a refinement step. If no better optimal solution is found for $t = 1$ at this node or the node is still flagged as infeasible due to reaching N or $\Delta t_{\min}$, then this node is pruned from $\mathcal{J}$. Otherwise, this node is branched, and two children nodes are created. The HCBB algorithm in Fig. 4 is used to investigate the new created children nodes again. It should be noted that when NLPFP is solved in HCBB, the refinement step is applied directly because the post check step is not applicable.

The complete HCBB algorithm incorporating post check and refinement procedure is provided in **Appendix A of** the **Supplementary Material**. In the HCBB algorithm, we can solve several NLP variants including NLPFX, NLPFP, and NLPRB during HC at a node n_i. According to our previous analysis, solving both NLPFP and NLPRB are more advantageous than solving NLPFX. Thus, we only implement HCBB algorithms with solving NLPFP and NLPRB, which are denoted as HCBB-FP and HCBB-RB respectively. All these HCBB algorithms are implemented in Python[53], which exchanges data with GAMS via Python API of GAMS[54]. Each NLP subproblem is solved using GAMS/CONOPT[55].

Remarks: 1) The complete HCBB algorithm can theoretically guarantee global optimality if the NLP subproblem for $t = 1$ at each node can be solved to global optimality.

2) The HCBB algorithm may fail if a feasible or locally optimal solution at root node is not obtained.

5 Computational studies

Four examples are solved to illustrate the capability of the proposed HCBB-FP and HCBB-RB. GAMS/BARON, GAMS/ANTIGONE, GAMS/SBB[40], GAMS/DICOPT[33], SRMINLP from our previous work[37] and the standard B&B implemented in Python by us (denoted as BB) are also used to solve these examples for comparison. Note that our implementation of the standard B&B algorithm may be different from GAMS/SBB[40]. All examples are modelled in GAMS 24.3.364 on a desktop with 3.20 GHz Intel Core i5-3470 CPU, 8GB RAM and 64-bit operating system. As expected, BARON and ANTIGONE cannot generate a feasible solution for any example within 1 hour.

Two initialization strategies are used to generate several initial points for the proposed HCBB algorithms, GAMS/SBB, GAMS/DICOPT, SRMINLP, and BB. These two initialization strategies are similar to that of Ma et al.[37,56]. We decouple the reactor network from the distillation system and solve them separately. We first solve the reactor network, which provides an input for the distillation system. The distillation system is then simulated or optimized to obtain an initial point for the entire process. While the distillation system is simulated without guaranteeing feasibility of all constraints such as product purity constraints in the first initialization strategy, the second strategy applies the hybrid steady-state and time-relaxation-based optimisation algorithm[57] to get a feasible solution of the distillation system. More details are provided in the **Supplementary Material**.

5.1 Example 1: benzene chlorination process

This example is adopted from Zhang et al.[20] It uses benzene and chlorine to produce desirable product chlorobenzene. The reactions are provided in Fig. S1. The superstructure is presented in Fig. S2. Fresh and recycled raw materials (i.e., benzene and chlorine) are mixed and fed into the reactor network to produce chlorobenzene and byproduct dichlorobenze. The highly volatile hydrochloric acid and chlorine are recovered from the reaction effluent using a flash unit and recycled back to the reaction system. The benzene, chlorobenzene and dichlorobenze mixture from the flash bottom is then heated to the bubble point and fed to the distillation system to obtain pure products chlorobenzene and dichlorobenzene and recover unreacted benzene that is recycled back to the reaction system. The productivity of chlorobenzene should be at least $50\,\mathrm{kmol\,h^{-1}}$. The recovery and purity of chlorobenzene and dichlorobenzene are at least 0.99, and the purity of the recycled benzene should also be at least 0.99.

The superstructure is modelled using the MINLP formulation from Ma et al.[37] with minimization of TAC. We use HCBB-FP, HCBB-RB, BB, SBB, DICOPT and SRMINLP to solve the MINLP formulation from three initial points generated. The optimisation results are provided in Table 1 where the results from different initial points are separated using slashes. There are totally 8 binary variables, 2450 continuous variables and 2451 constraints involved. As seen from Table 1, HCBB-FP and HCBB-RB generate almost the same optimal solution with TAC around 0.613 or 0.614 M€ year^{-1} from all three different initial points, which is 1.6% less than that (0.624 M€ year^{-1}) from Zhang et al.[20] The B&B algorithms including SBB, BB, HCBB-FP and HCBB-RB have better convergence than DICOPT and SRMINLP as both DICOPT and SRMINLP fail to find a feasible solution from the second initial point, whilst the B&B algorithms can find the optimal solution from all three initial points. DICOPT fails to find a feasible solution from the second initial point because only the NLP problem at the root node ($N_{\mathrm{node}} - N_{\mathrm{inf}} = 1$) is solved to feasibility. SRMINLP also fails from the second initial point because the complementary constraints prohibit the problem from converging to a feasible point. From Table 1, we can also observe that SBB and BB require less computational effort than HCBB-FP and HCBB-RB because the formers only solve one NLP subproblem at a node, while the latters need to solve several NLP subproblems when directly solving the original NLP subproblem fails, as indicated by the number of NLP subproblems solved (N_{nlp}).

From Table 1 it can also be observed that HCBB-FP needs to solve more NLP subproblems than HCBB-RB from the first and third initial points (13 vs. 5 and 12 vs. 4). This attributes to the fact that once a solution from HCBB-RB during HC at a node has an objective value greater than the upper bound, this node is flagged as having an optimal value larger than the upper bound, whilst such strategy cannot be implemented in HCBB-FP. For example, HCBB-FP and HCBB-RB find an incumbent solution with TAC of 0.614 M€ year^{-1} at node 1 from the initial point 1. At node 2, HCBB-FP solves the NLP subproblems until the step length $\Delta t = 0.008$ is less than the minimum allowable step length (i.e., 0.01) during HC. However, HCBB-RB obtains a solution with TAC of 0.687 M€ year^{-1} at $t = 0.25$, which is greater than the upper bound (i.e., 0.614 M€ year^{-1}), leading to termination of the HC calculation. Similar observation can be made for HCBB-FP and HCBB-RB from the initial point 3. However, the situation is different from the initial point 2. The computational time for HCBB-FP and HCBB-RB is very close (14 vs. 12). This is because there is no incumbent solution found yet when the node 1 is being solved and hence HCBB-RB has no chance to terminate HC early. Both algorithms cannot find an optimal solution at node 1 even if the step length less than Δt_{min} specified. As a result, node 1 is marked as infeasible in HCBB-FP and HCBB-RB and requires to be investigated again in the post check and refinement, where HCBB-RB finds that the last optimal solution during HC for node 1 with TAC of 0.690 M€ year^{-1} is greater than the optimal value of 0.613 M€ year^{-1}. Therefore, there is no need for HCBB-RB to start the refinement. On the contrary, HCBB-FP has to conduct refinement directly, leading to high computational effort. As seen from Table 1, HCBB-FP spent additional 155 CPU seconds to investigate the infeasible node 1. From this example, we can conclude that HCBB-RB is better than HCBB-FP, whilst SBB and BB are superior to HCBB-RB.

In the optimal design generated from HCBB-RB, a few bypass efficiencies have fractional values probably due to numerical errors or local optimum obtained. These fractional values are rounded up and an NLP problem with all binary variables and bypass efficiencies fixed at 0 or 1 is then solved. The final TAC changes little relative to that from HCBB-RB, (i.e., not more than 0.01%). The final optimal design is shown in Fig. 5.

5.2 Example 2: cyclohexane oxidation process

This example is from Zhang et al.[20] Cyclohexane is used to produce a mixture of cyclohexanol and cyclohexanone (i.e., KA oil) through oxidation. The reaction pathways are provided in Fig. S3 where the desired reaction produces cyclohexanol and cyclohexanone and the side reaction produces adipic acid. The superstructure is illustrated in Fig. S4. Air and cyclohexane are fed to the reactor network with only CSTR reactors used. The liquid effluent drawn from the reactor network is separated using two distillation columns to generate the desired product and byproduct. The recovered cyclohexane from the distillation system is recycled to the reactor network. The purity and recovery of products and raw material should be at least 0.9995.

The superstructure is modelled using the MINLP formulation from Ma et al.[37] in which the Wilson equation is used to calculate liquid activity coefficients, whilst the vapor is assumed to be ideal gas. The objective is to minimize TAC including energy cost and annualized capital cost. The optimisation problem involves 12 binary variables, 4647 continuous variables, and 4646 constraints. We generate three different initial points to initialize all these algorithms. The optimisation results are provided in Table 2 where the results from different initial points are separated using slashes.

As can be seen from Table 2, neither DICOPT nor SBB can solve the problem from any initial point because the NLP subproblems become more difficult to solve when the nonideal physical property Wilson equation is used. This can be evidenced from the fact that only the NLP subproblem at the root node is solved in both DICOPT and SBB. BB can only obtain a worse local optimum from the third initial point, whilst it fails from the other two initial points. HCBB-FP and HCBB-RB can get the optimal solution from all three initial points, showing their robustness in convergence. This indicates the effectiveness of the HC method in the improvement of convergence. More importantly, HCBB-RB converges to a better local optimum of 2.118 M€ year^{-1}, which is 1% less than the locally optimal solution of 2.146 M€ year^{-1} obtained by HCBB-FP. SRMINLP solves the problem to local optimality of 2.146 M€ year^{-1} from the first initial point and 2.118 M€ year^{-1} from the second and third initial points, which is worse than those from HCBB-RB. However, SRMINLP requires less computational time than HCBB-FP and HCBB-RB due to much less NLP subproblems being solved.

We now compare HCBB-FP and HCBB-RB in detail. From Table 2, we can observe HCBB-RB uses 26-52% less computational time than HCBB-FP to locate a locally optimal solution from an initial point. This is because HCBB-FP requires many HC steps to solve NLP subproblems, whilst HCBB-RB can terminate earlier once a larger TAC than the upper bound is obtained. As a result, HCBB-RB solves 45-60% less NLP subproblems than HCBB-FP. Furthermore, HCBB-RB claimed one infeasible node only from the initial point 3, while HCBB-FP claimed infeasibilities at 4, 14 and 13 nodes from the three initial points, respectively. More importantly, HCBB-FP spends more than 1 hour to complete the post check and refinement, whilst HCBB-RB requires tiny computational time to complete the post check and refinement. This is because HCBB-RB finds the optimal solution from the infeasible node is larger than the upper bound and hence the refinement procedure is not required.

There are several fractional bypass efficiencies in the locally optimal solution. We round them up and solve the resulted NLP problem again to obtain a slightly higher TAC of 2.120 M€ year^{-1}, which is still 4% – 5% less than the optimal TAC from Zhang et al.[20] (2.23 M€ year^{-1}). The optimal design is shown in Fig. 6. The lower TAC is because we invest a little more in reactors to achieve higher conversion and selectivity while decreasing the burden of separation systems evidently.

Example 3: hydrodealkylation process of toluene (HDA) using lumped reactor model

This example is taken from Ma et al[37], which uses toluene and hydrogen to produce product benzene and byproduct diphenyl. The reactions are given in Fig. S5. The superstructure is illustrated in Fig. S6. All data are given in the **Supplementary Material**. The desired benzene molar purity is 99.97% with a production rate of 124.8 kmol h^{-1}. The objective is to maximize the economic profit computed by the revenue from benzene and diphenyl minus annualized capital cost and operating cost.

The superstructure is modelled using the MINLP formulation from Ma et al.[37] which is solved using HCBB-FP, HCBB-RB, GAMS/DICOPT, GAMS/SBB, BB, and SRMINLP, respectively. Six different initial points are generated to initialize these algorithms. There are 8142 constraints, 8643 continuous variables, and 13 binary variables in the optimisation problem. The optimisation results are provided in Table 3 where the results from different initial points are separated using slashes. The results from GAMS/DICOPT are not shown in Table 3 due to its infeasibility from any initial point.

From Table 3, we can observe that HCBB-FP and HCBB-RB can find the locally optimal solution from any initial point, demonstrating their robustness in convergence. Almost the same locally optimal solution is generated from the six initial points. BB has very similar convergence performance to HCBB-FP and HCBB-RB for this example. SRMINLP and SBB can identify the locally optimal solutions from four out of the six initial points. While SRMINLP fails from the third initial point due to infeasibility caused at the root node, SBB fails from the sixth initial point. Although SRMINLP requires much less computational time than HCBB-FP and HCBB-RB, the optimal profit it generates from some initial points is much less than that obtained from HCBB-FP and HCBB-RB. For instance, SRMINLP generates a locally optimal profit of 3.920 M$ year^{-1} from the first initial point, which is 21% lower than that of 4.956 M$ year^{-1} obtained from HCBB-FP and HCBB-RB from the same initial point. SBB also requires much less computational time than HCBB-FP and HCBB-RB. However, the profit it obtained from some initial points is still much lower than that generated from HCBB-FP and HCBB-RB.

From Table 3 HCBB-FP and HCBB-RB find almost the same local optimum of 4.956 M$ year^{-1} from the six initial points. While HCBB-RB consumes 4%-13% less computational time than HCBB-FP from the first, second and fourth initial points, it requires 1%-79% more computational time from the third, fifth and sixth initial points. HCBB-RB requires to solve much more NLP subproblems from the sixth initial points compared to HCBB-FP. The possible reason is due to the difficulties in solving the problem with stream flow rates appearing in the denominator to optimality when the flow rates approach zero[34,58]. After solving, HCBB-FP and HCBB-RB flag 1-2 nodes as infeasible due to Δt_{min} and the maximum number of HC steps (i.e., N_{max}) specified being reached. As a result, HCBB-FP spends 656-3600 CPU seconds in the post check and refinement procedure, whilst HCBB-RB quickly identifies no better solution could be found at these infeasible nodes after the post check with negligible computational effort and hence the time-consuming refinement procedure is not conducted. Therefore, HCBB-RB requires much less total computational effort for B&B and post check and refinement to identify the same optimal solution compared to HCBB-FP.

In the optimal solution directly from HCBB-RB, there are a few fractional bypass efficiencies in the first column. After rounding the bypass efficiencies and optimizing the NLP problems again, the profit becomes 4.956 M$ year^{-1} which changes by around 0.04%, as shown in the **Supplementary Material**. The final optimal design is shown in Fig. 7.

5.4 Example 4: hydrodealkylation of toluene with differential reactor model

This example is very similar to Example 3 but the reaction kinetic equation from Dimian et al.[59] is used.

Both isothermal and adiabatic reactors are modelled using differential equations and discretized with orthogonal collocation finite element method[60]. The reactor model is more accurate due to the error caused by using an average temperature in the reaction kinetics for the whole adiabatic reactor in Example 3 being avoided. The kinetic equations and differential reactor models after discretization are provided in the **Supplementary Material**, which uses 50 elements and 3 Radau collocation points. Three reactor units are incorporated in the superstructure as shown in Fig. S7.

The optimisation problem involves 24 758 equations, 25 552 continuous variables and 19 binary variables. We generate six different initial points to initialize the algorithms. The computational results are provided in Table 4. As GAMS/DICOPT still cannot solve the problem from any initial point within 1 hour, the results from it are not shown in Table 4. As can be seen from Table 4, SRMINLP often finds a worse local optimum compared to BB, HCBB-FP, and HCBB-RB. For instance, SRMINLP finds a local optimum of 4.291 M\$ year^{-1} from the third initial point, which is 12% lower than the optimal solution of 4.904 M\$ year^{-1}. SBB finds a locally optimal solution with TAC of 4.291 M\$ year^{-1} only from the third initial point, which is much lower than that of 4.904 M\$ year^{-1} from HCBB-FP and HCBB-RB. While SBB fails to solve the problem from the other five initial points due to the failure in solving NLP subproblems during B&B, HCBB-FP and HCBB-RB find the optimal solution with TAC of 4.904 M\$ year^{-1} from all six initial points. Although BB can find similar optimum as the HCBB algorithms from five initial points, it obtains a much worse local optimum of 4.309 M\$ year^{-1} from the sixth initial point compared to that from HCBB-FP and HCBB-RB. It is expected that SRMINLP, SBB and BB require less computational time than HCBB-FP and HCBB-RB due to a smaller number of NLP subproblems being solved.

From Table 4 HCBB-RB and HCBB-FP generate very similar optimal solution with TAC of 4.904 M\$ year^{-1} from the six initial points. While HCBB-RB requires 8%-21% less computational effort than HCBB-FP from the first, second, fifth and sixth initial points, it consumes 3%-46% more computational time from the third and fourth initial points due to the difficulties in solving NLP subproblem with stream flow rates appearing in the denominator to optimality when the flow rates approach zero[34,58]. From Table 4, HCBB-FP spends extremely long time (usually more than 1 hours) in post check and refinement procedure, while HCBB-RB takes negligible time because no refinement procedure is required after post check for those nodes flagged as infeasible.

Similar to Examples 1-3, some bypass efficiencies in the optimal solution from HCBB-RB are still fractional due to limited convergence tolerance or local optimum. We round them up and solve the resulting NLP problem again to generate the optimal profits of 4.901-4.905 M\$ year^{-1}, a relative change of around 0.02%. The final optimal design is shown in Fig. 8.

6 Conclusion

In this work, we proposed two homotopy continuation enhanced branch and bound (HCBB) algorithms namely HCBB-FP and HCBB-RB through using the homotopy continuation (HC) method to solve strongly nonconvex and nonlinear MINLP problems in process synthesis using rigorous operation models. During branch and bound (B&B) in both HCBB-FP and HCBB-RB, each NLP subproblem at a node was solved using the solution from its parent node as an initial point. While HCBB-FP solved a series of feasibility problems to gradually reach the feasible solution at each node, HCBB-RB solved a few optimality problems with gradually tightened bounds of the homotopy variable and reached the optimal solution. As a result, the HCBB-RB terminated once the current optimum was larger than the upper bound of the MINLP problem during HC, which significantly reduced the computational effort required. A variable step length was adapted to effectively balance feasibility and computational efficiency. Several matching strategies were proposed to identify suitable step lengths based on historic information of the previous nodes, which reduced the effort in finding appropriate step lengths at latter similar nodes. To further improve solution quality, a post check and refinement procedure was proposed after B&B to revisit the nodes flagged as infeasible due to the minimum step length or the maximum number of iterations reached.

The computational results demonstrate that both HCBB-FP and HCBB-RB were able to generate the same best local optimum from different initial points, while all the other existing algorithms/solvers, including GAMS/SBB, GAMS/DICOPT, SRMINLP and our custom B&B algorithm failed to solve some examples or obtained much worse local optima. This is because the HC method can find feasible or optimal solution at some nodes which were flagged as infeasible in GAMS/SBB, GAMS/DICOPT and our custom B&B algorithm, reducing the risk of missing a better solution. Therefore, HCBB-FP and HCBB-RB are more robust than the other existing algorithms. It is also demonstrated that HCBB-RB was superior to HCBB-FP in terms of the number of nodes flagged as infeasible, the computational time required and the solution quality. More importantly, HCBB-RB usually terminated the post check and refinement before entering the time-consuming refinement steps.

Although HCBB-FP and HCBB-RB have demonstrated very good convergence performance for strongly nonlinear and nonconvex MINLP problems, they still have difficulties in solving NLP subproblems with stream flow rates appearing in the denominator to optimality when the flow rates approach zero[34,58], which inspires us to integrate the logic-based method in the future. It is also interesting to find global optimal solution at each node using the global optimisation methods and theoretically guarantee global optimality.

Acknowledgements

The authors acknowledge financial support from China Scholarship Council – The University of Manchester Joint Scholarship (201809120005)

Abbreviations

HCBB	Homotopy continuation enhanced branch and bound
HCBB-FP	HCBB algorithm with solving NLPFP
HCBB-RB	HCBB algorithm with solving NLPRB
BB	Branch and bound
HC	Homotopy continuation
OCFEM	Orthogonal collocation finite element method

HDA	hydrodealkylation
MINLP	Mixed-integer nonlinear programming
NLP	Nonlinear programming
TAC	Total annualised cost

Conflict of Interest

The authors claim no conflict of interest.

Supporting Information

Appendix A The complete homotopy continuation enhanced branch and bound algorithm (HCBB); Algorithm S1) Standard B&B algorithm; Algorithm S2) Initialization of the whole process; Algorithm S3) Initialization of the reactor network using OCEFM models; Table S1) Rate constant, operating conditions and specifications in the reactor and distillation column for Example 1; Table S2) Stream temperature and vaporization enthalpy in preheater, column1 and column 2 for Example 1; Table S3) Component physical properties and Antoine coefficients for Example 1; Table S4) Comparative results from BB, HCBB-FP, and HCBB-RB for Example 1; Table S5) Rate constants, operating conditions, and specifications in the reactor and distillation column for Example 2; Table S6) Component physical properties and Antoine coefficients for Example 2; Table S7) Parameters in the Wilson equations for Example 2; Table S8) Comparative results from BB, HCBB-FP, and HCBB-RB for Example 2; Table S9) Feedstock and product specification and prices for Example 3; Table S10) Utility prices for Example 3; Table S11) Investment costs for Example 3; Table S12) Parameters for Antoine equation in Exmaple 3; Table S13) Parameters for ideal vapour heat capacity polynomial in Example 3; Table S14) Parameters for DIPPR liquid heat capacity polynomial in Example 3; Table S15) Comparative results from BB, HCBB-FP and HCBB-RB for Example 3; Table S16) Stoichiometric coefficients for HDA reactions; Table S17) Comparative results from BB, HCBB-FP and HCBB-RB for Example 4; Figure S1) Reaction pathways for benzene chlorination; Figure S2) Superstructure for the synthesis of benzene chlorination process; Figure S3) Reaction pathway for cyclohexane oxidation; Figure S4) Superstructure for the cyclohexane oxidation process; Figure S5) Reaction pathways for the HDA; Figure S6) Superstructure of the HDA process; Figure S7) Superstructure for the synthesis of HDA process using differential reactor model.

References

1. Guterres A. Carbon neutrality by 2050: the world's most urgent mission. 2020; https://www.un.org/sg/en/content/sg/articles/2020-12-11/carbon-neutrality-2050-the-world%E2%80%99s-most-urgent-mission. Accessed 11 December, 2020.
2. Nishida N, Stephanopoulos G, Westerberg AW. A review of process synthesis. 1981;27(3):321-351.
3. Chen Q, Grossmann IE. Recent Developments and Challenges in Optimization-Based Process Synthesis. Annual Review of Chemical and Biomolecular Engineering. 2017;8(1):249-283.
4. King CJ, Gantz DW, Barnés FJ. Systematic Evolutionary Process Systhesis. *Industrial & Engineering Chemistry Process Design and Development.* 1972/04/01 1972;11(2):271-283.
5. Stephanopoulos G, Westerberg AW. Studies in process synthesis—II: Evolutionary synthesis of optimal process flowsheets. *Chemical Engineering Science.* 1976/01/01/ 1976;31(3):195-204.
6. Rudd DF. The synthesis of system designs: I. Elementary decomposition theory. 1968;14(2):343-349.
7. Douglas JM. A hierarchical decision procedure for process synthesis. *AIChE Journal.* 1985;31(3):353-362.
8. Daichendt MM, Grossmann IE. Integration of hierarchical decomposition and mathematical programming for the synthesis of process flowsheets. *Computers & Chemical Engineering.* 1998/01/01/ 1998;22(1):147-175.
9. Ryu J, Kong L, Pastore de Lima AE, Maravelias CT. A generalized superstructure-based framework for process synthesis. *Computers & Chemical Engineering.* 2020/02/02/ 2020;133:106653.
10. Kocis GR, Grossmann IE. Computational experience with dicopt solving MINLP problems in process systems engineering. *Computers & Chemical Engineering.* 1989/03/01/ 1989;13(3):307-315.
11. Ye H, Zou X, Zhu W, Yang Y, Dong H, Bi M. Synthesis and Optimization of Reaction–Separation–Recycle Systems with Complex Distillation Sequences. *Industrial & Engineering Chemistry Research.* 2020/06/03 2020;59(22):10509-10530.
12. Seader JD, Henley EJ, Roper DK. *Separation Process Principles: Chemical and Biochemical Operations-Third Edition.* Hoboken, New Jersey, USA: John Wiley & Sons, Inc.; 2011.
13. Dowling AW, Biegler LT. A framework for efficient large scale equation-oriented flowsheet optimization. *Computers & Chemical Engineering.* 1/2/ 2015;72:3-20.
14. Biegler LT. New nonlinear programming paradigms for the future of process optimization. *AIChE*

Journal. 2017;63(4):1178-1193.

15. Horn FJM, Tsai MJ. The use of the adjoint variables in the development of improvement criteria for chemical reactors. *Journal of Optimization Theory and Applications.* 1967/09/01 1967;1(2):131-145.

16. Lakshmanan A, Biegler LT. Synthesis of Optimal Chemical Reactor Networks. *Industrial & Engineering Chemistry Research.* 1996/01/01 1996;35(4):1344-1353.

17. Viswanathan J, Grossmann IE. An alternate MINLP model for finding the number of trays required for a specified separation objective. *Computers & Chemical Engineering.* 1993/09/01 1993;17(9):949-955.

18. Viswanathan J, Grossmann IE. Optimal feed locations and number of trays for distillation columns with multiple feeds. *Industrial & Engineering Chemistry Research.* 1993/11/01 1993;32(11):2942-2949.

19. Recker S, Skiborowski M, Redepenning C, Marquardt W. Systematic and Optimization-Based Design of Integrated Reaction-Separation Processes. In: Eden MR, Siirola JD, Towler GP, eds. *Computer Aided Chemical Engineering.* Vol 34: Elsevier; 2014:417-422.

20. Zhang X, Song Z, Zhou T. Rigorous design of reaction-separation processes using disjunctive programming models. *Computers & Chemical Engineering.* 2018/03/04/ 2018;111:16-26.

21. Gross B, Roosen P. Total process optimization in chemical engineering with evolutionary algorithms. *Computers & Chemical Engineering.* 1998/03/15/ 1998;22:S229-S236.

22. Aspen Technology Inc. Aspen Plus User's Guide. http://www.aspentech.com2015.

23. Rios LM, Sahinidis NV. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization.* 2013;56(3):1247-1293.

24. Henao CA, Maravelias CT. Surrogate-based superstructure optimization framework. *AIChE Journal.* 2011;57(5):1216-1232.

25. Biegler LT, Grossmann IE, Westerberg AW. A note on approximation techniques used for process optimization. *Computers & Chemical Engineering.* // 1985;9(2):201-206.

26. Kumar A, Baldea M, Edgar TF. A physics-based model for industrial steam-methane reformer optimization with non-uniform temperature field. *Computers & Chemical Engineering.* 2017/10/04/ 2017;105(Supplement C):224-236.

27. Smith EMB, Pantelides CC. Design of reaction/separation networks using detailed models. *Computers & Chemical Engineering.* 1995/06/11/ 1995;19(Supplement 1):83-88.

28. Yeomans H, Grossmann IE. A systematic modeling framework of superstructure optimization in process synthesis. *Computers & Chemical Engineering.* 1999/06/01/ 1999;23(6):709-731.

29. Tawarmalani M, Sahinidis NV. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming.* 2005/06/01 2005;103(2):225-249.

30. Misener R, Floudas CA. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of

Nonlinear Equations. *Journal of Global Optimization.* 2014/07/01 2014;59(2):503-526.

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

32. Trespalacios F, Grossmann IE. Cutting planes for improved global logic-based outer-approximation for the synthesis of process networks. *Computers & Chemical Engineering.* 2016/07/12/ 2016;90:201-221.

33. Duran MA, Grossmann IE. Simultaneous optimization and heat integration of chemical processes. *AIChE Journal.* 1986;32(1):123-138.

34. Türkay M, Grossmann IE. Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers & Chemical Engineering.* 1996/08/01/ 1996;20(8):959-978.

35. Land AH, Doig AG. An Automatic Method of Solving Discrete Programming Problems. *Econometrica.* 1960;28(3):497-520.

36. Stein O, Oldenburg J, Marquardt W. Continuous reformulations of discrete–continuous optimization problems. *Computers & Chemical Engineering.* 2004/09/15/ 2004;28(10):1951-1966.

37. Ma Y, Yang Z, El-Khoruy A, et al. Simultaneous Synthesis and Design of Reaction–Separation–Recycle Processes Using Rigorous Models. *Industrial & Engineering Chemistry Research.* 2021/04/29 2021;60(19):7275-7290.

38. Pedrozo HA, Reartes SBR, Vecchietti AR, Diaz MS, Grossmann IE. Optimal design of ethylene and propylene coproduction plants with generalized disjunctive programming and state equipment network models. *Computers & Chemical Engineering.* 2021/06/01/ 2021;149:107295.

39. Pattison RC, Baldea M. Equation-Oriented Flowsheet Simulation and Optimization Using Pseudo-Transient Models. *Aiche Journal.* Dec 2014;60(12):4104-4123.

40. *GAMS User's Guide Release 24.3* [computer program]2014.

41. Floudas CA. *Nonlinear and mixed-integer optimization : fundamentals and applications*. New York: Oxford University Press; 1995.

42. Nocedal J, Wright S. *Numerical optimization*: Springer Science & Business Media; 2006.

43. Borchers B, Mitchell JE. An improved branch and bound algorithm for mixed integer nonlinear programs. *Computers & Operations Research.* 1994/04/01/ 1994;21(4):359-367.

44. Ficken FA. The continuation method for functional equations. *Communications on Pure and Applied Mathematics.* 1951;4(4):435-456.

45. Vickery DJ, Ferrari JJ, Taylor R. An “efficient” continuation method for the solution of difficult equilibrium stage separation process problems. *Computers & Chemical Engineering.* 1988/01/01/ 1988;12(1):99-103.

46. Wayburn TL, Seader JD. Homotopy continuation methods for computer-aided process design.

Computers & Chemical Engineering. 1987/01/01/ 1987;11(1):7-25.

47. Asadi J, Jalali Farahani F. Optimization of dimethyl ether production process based on sustainability criteria using a homotopy continuation method. *Computers & Chemical Engineering.* 2018/07/12/ 2018;115:161-178.

48. Song W, Yao GM. Homotopy Method for a General Multiobjective Programming Problem. *Journal of Optimization Theory and Applications.* April 18 2008;138(1):139.

49. Morrison DR, Jacobson SH, Sauppe JJ, Sewell EC. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization.* 2016/02/01/ 2016;19:79-102.

50. Dechter R, Pearl J. Generalized Best-First Search Strategies and the Optimality of A. *Journal of the ACM (JACM).* 1985;32(3):505-536.

51. Biegler LT, Grossmann IE, Westerberg AW. *Systematic Methods Of Chemical Process Design*. Upper Saddle River, NJ, USA: Prentice Hall PTR; 1997.

52. Higham NJ. *Accuracy and Stability of Numerical Algorithms-Second Edition.* 3600 University City Science Center, Philadelphia, USA. Society for Industrial and Applied Mathematics; 2002.

53. Python Software Foundation. *Python Language Reference Version 3.6.*2016.

54. *General Algebraic Modeling System (GAMS) Release 24.3.3, Fairfax, VA, USA* [computer program]2014.

55. Drud A. CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming.* 1985/06/01 1985;31(2):153-191.

56. Ma Y, El-Khoruy A, Yang Z, et al. Simultaneous Synthesis and Design of Integrated Reaction-Separation Systems Using Rigorous Models. In: Muñoz SG, Laird CD, Realff MJ, eds. *Computer Aided Chemical Engineering.* Vol 47: Elsevier; 2019:371-376.

57. Ma Y, McLaughlan M, Zhang N, Li J. Novel feasible path optimisation algorithms using steady-state and/or pseudo-transient simulations. *Computers & Chemical Engineering.* 2020/12/05/ 2020;143:107058.

58. Chen Q, Johnson ES, Bernal DE, et al. Pyomo.GDP: an ecosystem for logic based modeling and optimization development. *Optimization and Engineering.* 2021/04/23 2021.

59. Dimian AC, Bildea CS, Kiss AA. Chapter 21 - Case Studies. In: Dimian AC, Bildea CS, Kiss AA, eds. *Computer Aided Chemical Engineering.* Vol 35: Elsevier; 2014:789-830.

60. Finlayson BA. *Nonlinear analysis in chemical engineering*: Bruce Alan Finlayson; 2003.