A novel stochastic method to efficiently synthesize nonsharp distillation sequences simultaneously involving heat integration and thermal coupling

Shuo Zhang, a Yiqing Luo, a,b Xigang Yuan a,b,c

a School of Chemical Engineering and Technology, Tianjin University, Tianjin 300350, China
b Chemical Engineering Research Center, Tianjin University, Tianjin 300350, China
c State Key Laboratory of Chemical Engineering, Tianjin University, Tianjin 300350, China
luoyq@tju.edu.cn

Abstract
In this paper, a novel stochastic optimization method is proposed to efficiently synthesize simultaneously heat integrated and thermally coupled distillation sequences (HITCDSs) involving nonsharp splits. First, a novel easy-to-use binary tree encoding method is proposed to represent feasible distillation sequences. A binary tree can be easily evolved by randomly pruning and generating the tree branches. Next, to search the optimal HITCDS, the simulated annealing algorithm and the particle swarm optimization algorithm are effectively combined. Finally, a HITCDS synthesis problem is solved to prove that the method has high solution efficiency and can obtain the top three HITCDSs with high possibility.

Keywords: Distillation sequence synthesis, Heat integration, Thermal coupling, Nonsharp splits, Stochastic optimization

1. Introduction
Distillation is energy intensive for multicomponent separation. Distillation sequence synthesis aims to identify the most energy-efficient column sequence from multiple alternatives and hence is a large-scale combinatorial optimization problem. Simultaneously heat integrated and thermally coupled distillation sequences (HITCDSs) are more energy-efficient than conventional distillation sequences, distillation sequences involving only heat integration and distillation sequences involving only thermal coupling (Zhang et al., 2019). However, simultaneously involving heat integration and thermal coupling in the distillation sequence synthesis considerably expands the scale of the synthesis problem and increases the combinatorial complexity. Researches on the synthesis of HITCDSs are still limited. Rong et al. (2003) presented a procedure to generate the HITCDSs for a multicomponent separation. Unfortunately, the optimization of the generated HITCDSs was not considered. To optimize the HITCDSs, Caballero and Grossmann (2006) formulated a disjunctive programming model and solved it by a two-level deterministic algorithm, where the non-integrated flowsheets were optimized first and then the heat integration optimization problem was solved. Their hierarchical optimization algorithm could not ensure the global optimality. Besides, the algorithm was time consuming, as it determined the optimum through a branch-and-bound search. Compared to deterministic algorithms, stochastic optimization methods are more efficient to solve large non-convex optimization problems.
problems, as they only rely on evaluating the objective function values to identify the optimum and can escape from local optima to find the global optimum with high possibility. Wang et al. (2008) applied the genetic programming algorithm to optimize the HITCDSs. Yuan and An (2002) developed an improved simulated annealing (ISA) algorithm to identify the optimal HITCDS. For simplification, their encoding methods were restricted to representing sharp HITCDSs. Sharp distillation systems suffer from energy penalty arising from the remixing effects compared to the nonsharp ones (Giridhar and Agrawal, 2010). Attempts have been made to apply stochastic optimization methods to synthesize nonsharp HITCDSs. Based on the ISA algorithm (Yuan and An, 2002), Luo et al. (2010) synthesized nonsharp HITCDSs, which allowed only one middle component. Later, Zhang et al. (2019) synthesized nonsharp HITCDSs with at most two middle components. Limiting the number of the middle components made the search space uncompleted and might lead to suboptimal solutions. Additionally, these stochastic optimization methods decomposed the initial synthesis problem into several sub-problems for an easier manipulation of the coding series, which was adverse to the automation of distillation system synthesis. The contribution of the present study is to propose a new stochastic optimization method for the automatic and efficient synthesis of nonsharp HITCDSs. The new method does not limit the number of middle components and avoids decomposing the synthesis problem. First, an easy-to-use binary tree encoding method is introduced to represent distillation sequences. Next, evolutionary rules are developed to update the distillation sequences. Then, the HITCDSs are optimized based on stochastic optimization. Finally, a case is studied to verify the good performance of the proposed method.

2. Problem formulation

The present study aims to determine the optimal HITCDS to separate a zeotropic mixture containing \( N \) (\( N \geq 3 \)) components into pure products. The synthesis problem is solved based on stochastic optimization, which finds the optimal solution only by calculating and comparing the objection function values of the independent variables in an evolutionary process. In the optimization of HITCDSs, the objective function is the total annual cost (TAC), which is the sum of the annualized capital and operating cost of a HITCDS. The independent variables include the distillation sequences, thermal coupling structures, operating pressures (\( p \)), light and heavy key component recoveries (\( \xi_{LK} \) and \( \xi_{HK} \)) and ratios of the actual reflux ratios to the minimum ratios (\( r \)) in columns. The optimization problem can be formulated as an implicit mixed-integer nonlinear programming (MINLP) problem as follows.

\[
\begin{align*}
\text{Min } C_{\text{cost}} & : \left\{ \text{Tree}, \{ \psi_i \}, p, \xi_{LK}, \xi_{HK}, r \right\} \\
\text{s.t.} & \quad \text{CSS} \left( \text{Tree}, \{ \psi_i \} \right) = 0 \\
& \quad \text{TCS} \left( \{ \phi_i \}, \{ \tau \} \right) = 0 \\
& \quad p = [p_1, p_2, \ldots, p_s, p_k]^T \in P \\
& \quad \xi_{LK} = [\xi_{LK1}, \xi_{LK2}, \ldots, \xi_{LKk}, \xi_{LKK}]^T \\
& \quad \xi_{HK} = [\xi_{HK1}, \xi_{HK2}, \ldots, \xi_{HKk}, \xi_{HKK}]^T \\
& \quad r = [r_1, r_2, \ldots, r_s, r_k]^T \\
& \quad \text{MINLP} \\
& \quad \xi_{LK} \in [0.98, 1.00], \xi_{LKK} \in (0, 0.02], \xi_{HK} \in [1.02, 1.30], k \in K
\end{align*}
\]

Herein, \( C_{\text{cost}} \) is the objective function. \( \{ \text{Tree} \} \) and \( \{ \psi_i \} \) are integer variables to represent the distillation sequences and thermal coupling structures, respectively. Eqs. (1) and (2)
A novel stochastic method to efficiently synthesize nonsharp distillation sequences simultaneously involving heat integration and thermal coupling

are symbolic equations to generate all the feasible distillation sequences and thermal coupling structures. Details of the implicit MINLP formulation can refer to Zhang et al. (2018, 2019). Solution strategies of the problem are given in the next section.

3. Solution strategies

3.1. A novel binary tree encoding method to represent distillation sequences

A binary tree encoding method is proposed to represent distillation sequences, which do not limit the number of middle components. For an N-component separation, rank the N components in decreasing order of volatility and assign an integer number to encode each component from 1 to N. A stream can be represented by the codes of the most and the least volatile components in the stream. For example, the mixture stream C₂C₃C₄C₅ can be represented by [2,5] and specially, the pure stream C₂ can be encoded by [2,2].

For a separation task, if the feed is represented by [m, n] (m < n), the most volatile component of the top product must be m and the top product can be represented by [m, x]. On the other hand, the least volatile component of the bottom product must be n and the bottom product can be represented by [y, n]. Herein, x and y are integer numbers representing unknown components. To ensure the feasibility of the separation task, x and y must satisfy some constraints. First, the volatilities of the component x and y must be between those of component m and n. Hence, we have constraints (3) and (4).

\[
\begin{align*}
3. \quad & m \leq x < n \\
4. \quad & m < y \leq n
\end{align*}
\]

Second, for the sake of material balance, any component in the feed stream should not be absent from the product streams. Therefore, constraint (5) is given.

\[
5. \quad y \leq x + 1
\]

Randomly give the values of x and y within constraints (3)-(5), a feasible separation task can be randomly generated (see Figure 1a). A separation task has one feed and two products, which is similar to the node structure of binary tree. Hence, a separation task can be represented by a binary tree node structure. As shown in Figure 1b, the feed stream code [m, n] is stored at the root node, the top product stream code [m, x] is stored at the left sub-node, and the bottom product stream code [y, n] is stored at the right sub-node. To mathematically express the stream connection in a separation task, each stream is assigned a number. If the number assigned to the feed steam is k, the number of the top and bottom product will respectively be 2k and 2k+1, as shown in Figure 1b.

If the binary tree nodes are randomly generated layer by layer within constraints (3)-(5) until all the mixture stream are separated into pure products, a complete binary tree can be randomly generated. Taking the 5-component separation for example, the procedure to randomly generate a complete binary tree is presented in Figure 2. All the code information is stored in an integer array \{Tree\} in an increasing order of stream number.

Figure 1. (a) A separation task and (b) its equivalent node structure
Figure 2. The procedure to randomly generate a complete binary tree layer by layer. We only consider binary trees corresponding to basic configurations, as they outperform the nonbasic configurations in both capital and energy cost (Giridhar and Agrawal, 2010). If all the stream codes stored in a binary tree appear only once, this tree will correspond to a basic configuration. If some of the codes appear repeatedly in a binary tree, it should be judged whether the tree corresponds to a basic configuration. The judgement is performed for the product stream codes in a binary tree, from the mixture product codes with the maximum components to the pure product codes as follows:

**Step 1.** Count the times a product code appears.

**Step 2.** If a product code satisfies one of the following two conditions: (a) the product code appears more than twice; (b) the code appears twice and the stream numbers assigned to the two repeated codes are both even or both odd, the tree cannot be converted to a basic configuration and the judgement finishes. If a mixture product code appears twice and meanwhile one of the two repeated codes has an even stream number and the other has an odd stream number, record all the sub-nodes of the repeated node whose stream number is smaller and do not involve these recorded sub-nodes in the subsequent judgement.

**Step 3.** Return to step 1 and continue until all the pure product codes are judged. If a binary tree corresponds to a basic configuration, for each pair of the repeated mixture product codes, prune all the sub-nodes of the mixture product code whose stream number is smaller. Taking the binary tree in Figure 2 as an example, the mixture product code [3,4] appears repeatedly and their stream numbers are 6 and 11. Hence, the sub-nodes of [3,4] with a stream number 6 should be pruned and the pruned tree is shown in Figure 3 (a). It is straightforward to convert a pruned tree to its corresponding sequence of separation tasks, because the stream code information and the mathematical relationship of the stream number can directly express the interconnections of the separation tasks. For example, the pruned tree in Figure 3a is converted to the separation task sequence in Figure 3b. Finally, by consolidating the separation tasks producing the same product stream, the corresponding basic configuration is obtained (see Figure 3c).

If a tree does not correspond to a basic configuration, the tree should be evolved.

**Figure 3.** Procedure to convert a pruned tree to the corresponding basic configuration

### 3.2. Evolving the distillation sequences

A current binary tree can be evolved to by two steps. (I) Randomly choose a mixture stream node with more than two components and randomly prune its left or right sub-
A novel stochastic method to efficiently synthesize nonsharp distillation sequences simultaneously involving heat integration and thermal coupling

...tree with an equal possibility. (II) Randomly generate a new feasible left or right sub-tree with constraints (3)-(5). Note, representing and evolving thermal coupling structures are based on a set of 0-1 binary variables, which can refer to Zhang et al. (2019).

3.3 Solving the implicit MINLP problem

Based on the aforementioned encoding representation method and the evolving rules, all the feasible distillation sequences and the corresponding thermal coupling structures can be randomly generated in the search space. The mechanism to select the optimal HITCDS from multiple alternatives is based on stochastic optimization, which randomly searches the optimum in an evolutionary process. The simulated annealing (SA) algorithm (Kirkpatrick et al., 1983), which is based on the analogy between the physical process of annealing and the solution of large combinatorial optimization problems, is adopted to deal with the combinatorial optimization of distillation sequences ($\{T_{trees}\}$) and thermal coupling structures ($\{\phi_i\}$). Once a flowsheet is randomly generated in the SA procedure, the corresponding continuous variables ($p$, $\xi_k$, $\xi_{in}$, and $r$) are optimized by the particle swarm optimization (PSO) algorithm (Shi and Eberhart, 1998). In the optimization, the separation tasks in a flowsheet are designed through the Fenske-Underwood-Gilliland equations (Fenske, 1932; Underwood, 1948; Gilliland, 1940), the corresponding optimal heat integrated network is determined by the pinch based energy matching strategies of Zhang et al. (2018), and the TAC can be calculated based on the cost model of Zhang et al. (2019).

4. Case study

Nonsharp HITCDSs are synthesized to separate a five-component mixture. Data for the Case study is taken from Yuan and An (2002), as shown in Table 1.

Table 1. Data for the Case study

<table>
<thead>
<tr>
<th>Component</th>
<th>Mole fraction</th>
<th>Relative volatility</th>
<th>Utility</th>
<th>$T$ (K)</th>
<th>Price ($/GJ$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) ethanol</td>
<td>0.25</td>
<td>4.6156</td>
<td>Cooling water</td>
<td>305</td>
<td>0.067</td>
</tr>
<tr>
<td>(B) $i$-propanol</td>
<td>0.15</td>
<td>4.0046</td>
<td>Steam (0.17MPa)</td>
<td>389</td>
<td>0.418</td>
</tr>
<tr>
<td>(C) $n$-propanol</td>
<td>0.35</td>
<td>2.2454</td>
<td>Steam (0.408MPa)</td>
<td>418</td>
<td>0.574</td>
</tr>
<tr>
<td>(D) $i$-butanol</td>
<td>0.10</td>
<td>1.4775</td>
<td>Steam (1.122MPa)</td>
<td>458</td>
<td>0.774</td>
</tr>
<tr>
<td>(E) $n$-butanol</td>
<td>0.15</td>
<td>1.0000</td>
<td>Steam (2.824MPa)</td>
<td>503</td>
<td>1.025</td>
</tr>
</tbody>
</table>

The feed flowrate is 500.4 kmol/h.

Figure 4. The top three HITCDSs.

The CPU time to solve the HITCDS synthesis problem is only 97.397 s. The possibility to obtain the best, second best, and third best HITCDS (see Figure 4) is 43.2%, 2.6%, and 37.2%, respectively. To validate our method, the optimization results are compared...
with previous literature. Yuan and An (2002) only synthesized sharp HITCDSs. Compared to their optimal sharp HITCDS, our optimal nonsharp HITCDS can reduce the TAC by 26.5%. And our optimal sharp HITCDS has the same distillation sequence and thermal coupling structure as Yuan and An’s. However, the heat integrated networks are different, as Yuan and An assumed that one condenser could only match one reboiler, and vice versa, while our method allows any feasible energy match. Consequently, our optimal sharp HITCDS has more energy matches and reduces the TAC by 12.2%. Additionally, Zhang et al. (2019) synthesized nonsharp HITCDSs with at most two middle components, while our method does not limit the number of middle components. Our optimal nonsharp HITCDS is the same as that obtained by Zhang et al. (2019) and the relative error of the TAC is only -1.5×10^{-3}, which verifies the correctness of our method. Moreover, compared to Zhang et al., our method avoids decomposing the synthesis problem into several sub-problems and hence is more automatic.

5. Conclusions
A new stochastic method is proposed to synthesize nonsharp HITCDSs with no limitation of the number of middle components. In the Case study, the possibility to obtain the top three HITCDSs is high up to 83.0 %. The result comparison with the previous literature validates the correctness of our proposed method. The method has high solution efficiency and hence it can be applied to efficiently solve the more complex six or seven component HITCDS synthesis problems.

References