Computers and Chemical Engineering 174 (2023) 108254

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Globally optimal distillation column design using set trimming and enumeration techniques

Alice Peccini [a], Lucas F.S. Jesus [b], Argimiro R. Secchi [b], Miguel J. Bagajewicz [c,d,*], André L. H. Costa [a]

[a] Rua São Francisco Xavier, Universidade do Estado do Rio de Janeiro (UERJ), 524, Maracanã, Rio de Janeiro, RJ CEP 20550-900, Brazil
[b] Chemical Engineering Program, COPPE, Universidade Federal do Rio de Janeiro (UFRJ), CT, Bloco G, Ilha do Fundão, Rio de Janeiro, RJ CEP 21941-914, Brazil
[c] School of Chemistry, Universidade Federal do Rio de Janeiro (UFRJ), CT, Bloco E, Ilha do Fundão, Rio de Janeiro, RJ CEP 21949-900, Brazil
[d] School of Chemical, Biological and Materials Engineering, University of Oklahoma, Norman, OK 73019, USA

ARTICLE INFO

Keywords:
Distillation
Global optimization
Smart enumeration
Optimal design

ABSTRACT

In this article, we present an alternative approach to the use of metaheuristic methods (GA, PSO, SA, etc.) or mathematical programming (MINLP solvers) for the optimal design of one-feed-two-products distillation columns. We propose the use of Set Trimming followed by candidate enumeration. For the evaluation of the performance of each candidate solution, the method relies on solving the associated system of equations. Three different enumeration procedures are tested: Exhaustive Enumeration, Smart Enumeration, and Segmental Smart Enumeration. Smart Enumeration is an optimization procedure that identifies the solution through a search in the set of candidates organized in ascending order of the objective function lower bound, while Segmental Smart Enumeration is introduced in this article. We compare the results of the proposed procedure with the results using an MINLP approach with different solvers. Numerical results indicate that the best alternative of the enumeration algorithms can identify the global optimum faster than a global solver of mathematical optimization. Numerical tests also showed local solvers which attained optimal solutions quickly but may be trapped in a local optimum.

1. Introduction

In the vast majority of processes, separation tasks are present, whether to remove unwanted by-products from a given mixture, to regenerate a solvent, or to achieve the correct specification of a product. Although being one of the oldest, distillation continues to be one of the most important unit operations, accounting for 90–95% of all industrial separations (Sorensen, 2014), and this is considered unlikely to change in the near future mainly due to its versatility and robustness (Caballero and Grossmann, 2014). The advantages of distillation as a separation method are its capability of handling a wide range of product mixtures, different feed compositions, and feed flow rates (ranging from semi-micro scales to thousands of tons per year), all the while being able to achieve high purities (Smith and Jobson, 2000).

A significant effort from the scientific community to improve distillation design methods spans decades. The first efforts, the well-established graphical techniques, such as Ponchon-Savarit (Ponchon, 1921; Savarit, 1922) and McCabe and Thiele (1925), are still widely used as conceptual and teaching tools, although largely replaced by computational implementations in design practice. Shortcut methods, like the Fenske-Underwood-Gilliland method (Fenske, 1932; Underwood, 1948; Gilliland, 1940) and others (Kirkbride, 1944; Glinos and Malone, 1985; Nikolaides and Malone, 1987; Kamath et al., 2010; Adiche and Vogelpohi, 2011) are quite useful for preliminary design work. Caballero and Grossmann (2014) made a good review of a variety of different design methods.

Finding an optimal design involves selecting both the discrete variables associated with constructive aspects of the column (such as the number of trays and the feed tray location) and the continuous variables, associated with operating conditions (such as reflux ratio). The optimal performance is usually represented by the total annualized cost, including the capital cost of the distillation column and the operating costs related to the utility consumption in the condenser and the reboiler. In optimization-driven design, two main approaches have been employed: mathematical programming and metaheuristic methods.

* Corresponding author at: School of Chemistry, Universidade Federal do Rio de Janeiro (UFRJ), CT, Bloco E, Ilha do Fundão, Rio de Janeiro, RJ CEP 21949-900, Brazil.

E-mail address: bagajewicz@ou.edu (M.J. Bagajewicz).

https://doi.org/10.1016/j.compchemeng.2023.108254
Received 29 January 2023; Received in revised form 16 March 2023; Accepted 4 April 2023
Available online 6 April 2023
0098-1354/

Nomenclature

LB	objective function lower bound
Nt	total number of trays
Nf	feed tray
x	mole fraction
V	vapor flowrate

Greek letters

Relative volatility .

$\hat{\rho}$	parameter that control the extension of the interval length of the search
$\hat{\sigma}$	parameter that control the interval length of the search

Subscripts

B	distillation bottom product
D	distillation top products
Fenske	minimum number of trays evaluated through the Fenske equation
HK	heavy key component
i	interval index
LK	light key component

Supercripts

Inc	incumbent
min	minimum number of trays of a given separation task
max	maximum number of trays in the search space
last	last row of the search space that still contains remaining candidate
start	location of the beginning of the search
step	interval length
end	location of the end of the search
BU	boil-up

The utilization of mathematical programming to solve the aforementioned models was explored through different alternatives: nonlinear programming (NLP), mixed-integer nonlinear programming (MINLP), and generalized disjunctive programming (GDP).

Viswanathan and Grossmann (1990) presented different mixed integer nonlinear models for the identification of the feed tray location and the number of trays. These models were solved using an outer approximation algorithm (AP/OA/ER). A simplified and computationally more efficient alternative for the determination of the number of trays was presented by Viswanathan and Grossmann (1993a). Later, Viswanathan and Grossmann (1993b) extended the previous formulations using an MINLP approach for the determination of the location of the feed trays as well as the number of trays with multiple feed streams. Kong and Maravelias (2019) also proposed a mixed integer nonlinear optimization model for the optimal design of distillation columns, based on the McCabe-Thiele method, which they solved using MINLP procedures. The optimal solution determines column design and operational variables, such as the number of trays, the reflux ratio, and the condenser and reboiler heat duties, which minimizes the total annualized cost.

Yeomans and Grossmann (2000) presented a GDP model for the determination of the optimal values of the feed tray location, the number of trays, the reflux ratio, the column diameter, and the condenser and reboiler heat loads and area aiming at minimizing the total annualized cost. Barttfeld et al. (2003) compared MINLP and GPD solution procedures for different distillation column representations. The GDP formulation presented lower computational times and showed to be a more robust formulation. More recently, the GDP formulation was addressed using Benders Decomposition by Liñan and Ricardez-Sandoval (2023). Alternative approaches that proposed the combination of mathematical programming with commercial simulators to solve the design problem of distillation columns were attempted by Caballero et al. (2005) and Caballero (2015). The mathematical programming structure employed in these works was based on GDP representations of the problem.

Yeoh and Hui (2021) presented an improvement of the concept of bypass efficiency, originally presented by Dowling and Biegler (2015), for the optimal design of distillation columns without using integer variables, therefore yielding an NLP problem. The solution of the distillation design through an NLP problem was also addressed recently by Jia et al. (2022) using relaxation in stage numbers instead of bypass efficiency.

Aiming at avoiding convergence drawbacks and local optimality problems associated with mathematical programming, some authors proposed the utilization of metaheuristic optimization methods for the solution of the optimal design problem. Javaloyes-Antón et al. (2013) proposed the utilization of the particle swarm optimization (PSO) coupled with a process simulator (Aspen Hysys) and the superstructure proposed by Yeomans and Grossmann (2000). Ibrahim et al. (2017) proposed the optimization of crude oil distillation units, integrating a process simulator (Aspen Hysys) with genetic algorithms (GA). Lyu et al. (2021) employed a differential evolution (DE) approach associated with parallel computing for the solution of the design optimization problem. Despite being more robust than the mathematical programming approaches, the solutions based on metaheuristic optimization methods also have limitations: computational performance depends on the tuning of the parameters of the algorithm and global optimality cannot be guaranteed.

The Set Trimming followed by an enumeration procedure for globally optimal distillation column design proposed in this paper eliminates the drawbacks of the previous approaches mentioned above. We enumerate candidate solutions, where each candidate is represented by the corresponding values of the number of trays and feed tray location. With those two variables fixed, the rest of the column variables (reflux ratio, flow rates, and compositions in each tray, reboiler, and condenser heat loads, etc.) can be calculated for a given separation task through the solution of the column mathematical model, corresponding to a nonlinear system of equations, as well as the objective function (total annualized cost) can be determined. Based on this representation of the search space, three alternatives of enumeration procedures are explored: Exhaustive Enumeration, Smart Enumeration, and Segmental Smart Enumeration. The procedure of Smart Enumeration identifies the solution through a search in the set of candidates organized in ascending order of the objective function lower bound (Costa and Bagajewicz, 2019). Segmental Smart Enumeration is introduced in this article as a means to speed up Smart Enumeration.

The article is organized as follows. First, the optimization problem formulation is presented. Next, the Set Trimming followed by the Enumeration approach is explained, followed by a discussion of variants to improve the computational performance. Finally, we present numerical results, including a comparison of the performance of the proposed approach with mathematical programming.

2. Optimization problem

The proposed procedure addresses the design optimization of distillation columns based on equilibrium stages. We consider a one-feed-two-product streams column with a reboiler and a total condenser, as shown in Fig. 1. The reboiler is considered an equilibrium stage. The column pressure profile is assumed previously established.

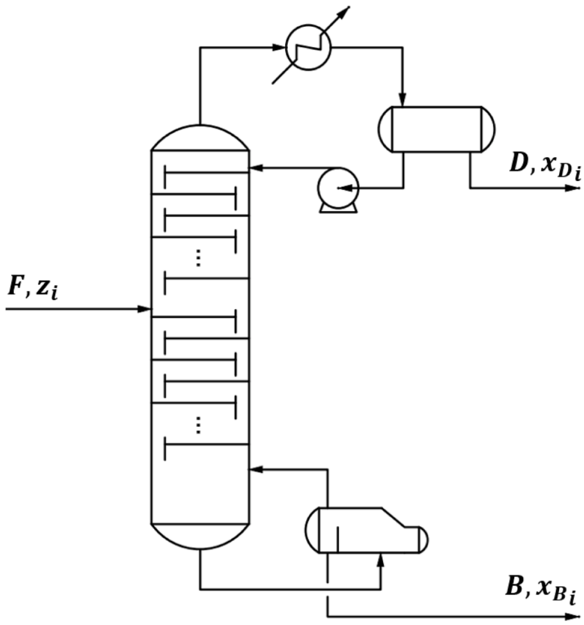

Fig. 1. Distillation column with a single feed and two product streams.

The application of the proposed design procedures to more complex systems is discussed later.

The total annualized cost is composed of the capital cost of the column (a function of its diameter and number of trays) and the operating cost (associated with the utility expenditure and calculated from the condenser and reboiler heat loads). The independent (design) variables are the number of trays and the feed tray location, both integers (*Nt, Nf*).

The separation task can be specified in different ways. Most of the time, they are combinations of distillate or bottom composition and component recovery. Sometimes, all specifications are related to one component, like for example, a molar/mass fraction of the lightest component at the top and a small molar/mass fraction of it in the bottom stream. Other times, the specifications refer to flowrates of individual components or various components in the distillate and/or bottoms. To summarize, there are always two specifications related to the separation task that act as constraints.

The determination of all other variables (reflux ratio, flow rates and compositions in each tray, column diameter, condenser, reboiler heat loads, etc.) is done by solving a system of nonlinear equations. The literature has different alternatives of distillation modeling that can be used to formulate this system of nonlinear equations, such as the McCabe Thiele method for binary systems, or methods based on MESH equations. The Supplementary Material contains two examples of distillation models: a system of equations based on the hypothesis of equimolar flow in each section and a more rigorous model based on MESH equations.

3. Proposed methodology

The proposed approach for the solution of the optimal design problem of distillation columns involves Set Trimming followed by Enumeration.

The Set Trimming step reduces the number of solution candidates, exploring the minimum number of equilibrium stages for a given separation task. The Enumeration step evaluates the remaining candidates to identify the least-cost design. This work proposes three different enumeration approaches:

(i) Exhaustive Enumeration
(ii) Smart Enumeration
(iii) Segmental Smart Enumeration

Aiming at reducing the computational effort, the Smart Enumeration and the Segmental Smart Enumeration make use of the well-known behavior of distillation systems described in Fig. 2, which is found in many textbooks (Kister, 1992; King, 2013). For a given, separation task, a reduction in the number of stages implies an increase in the reflux ratio. This behavior is associated with limits. The minimum reflux condition, namely the pinch point where an infinite number of stages would be required for the desired separation, and the minimum possible number of stages associated with a total reflux condition. Inside the limits imposed by these two conditions lie all practical operations, which show the relationship between the reflux ratio and the number of stages.

4. Search domain definition

The search domain, that is, the set containing all solution candidates can be represented by a bidimensional structure with rows and columns. The rows represent the number of trays (*Nt*) and the columns the feed tray location (*Nf*). According to the assumption that both sections (rectifying and stripping) always exist, the feed tray is located between tray 2 and tray *Nt* − 1. Fig. 3 illustrates the corresponding search domain, for a maximum number of 40 trays. This maximum number of trays must be carefully selected beforehand so it is guaranteed that the optimum is within this limit.

For each one of the solution candidates (741 alternatives in Fig. 3), *Nt* and *Nf* are known by construction, and therefore, if the candidate is feasible, it is possible to solve the distillation mathematical model to find the corresponding reflux ratio, as well as flows and compositions that satisfy the problem specifications to finally be able to evaluate the cost.

5. Set trimming

We use a Set Trimming step to reduce the computational effort for the identification of the optimal solution to the design problem. Set Trimming corresponds to the application of an inequality constraint, or equivalent, for a given search space to eliminate solution candidates (Costa and Bagajewicz, 2019). This procedure is applied here eliminating candidates with a number of stages smaller than the minimum number of equilibrium stages necessary for a given separation (See Fig. 2). However, since the minimum number of stages is unknown, an estimative is generated to guide the eliminations.

For a system with constant relative volatility, the minimum number

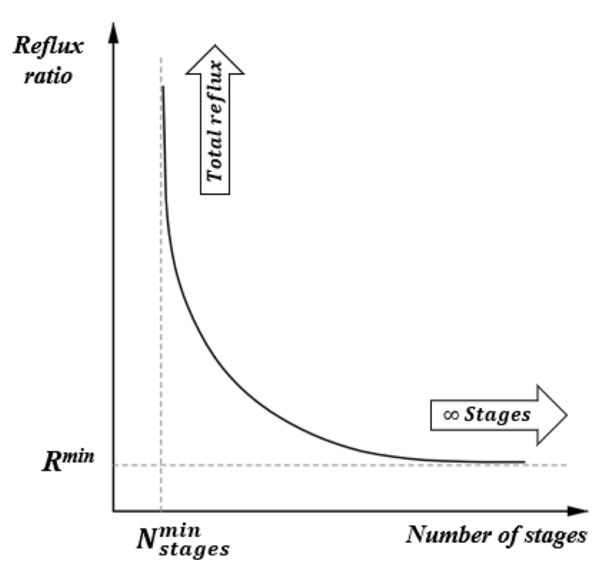

Fig. 2. Relation between the reflux ratio and the number of equilibrium stages.

Nt \ Nf	2	3	4	5	6	⋯	36	37	38	39
3	1									
4	2	3								
5	4	5	6							
6	7	8	9	10						
7	11	12	13	14	15					
⋮	⋮	⋮	⋮	⋮	⋮	⋱				
37	596	597	598	599	600	⋯	630			
38	631	632	633	634	635	⋯	665	666		
39	667	668	669	670	671	⋯	701	702	703	
40	704	705	706	707	708	⋯	738	739	740	741

Fig. 3. Search space of the column design optimization for $Nt^{max} = 40$.

of trays can be calculated by the Fenske equation (Fenske, 1932). Because this hypothesis may not be valid, we only use the Fenske equation in our algorithm to give an initial estimate of the minimum number of trays, Nt^{min}, to start the search (the minimum number of trays in the column is the minimum number of equilibrium stages minus the reboiler):

$$Nt^{min}_{Fenske} = \frac{\ln\left(\left(\frac{x_{LK}}{x_{HK}}\right)_D \left(\frac{x_{HK}}{x_{LK}}\right)_B\right)}{\ln\left(\alpha_{LK/HK}\right)} - 1 \quad (1)$$

where x is the molar fraction, α is the relative volatility, the subscripts LK and HK indicate the light-key and heavy-key components, and the subscripts D and B represent the top and bottom products, respectively (if α varies, a geometric mean of their values at the top and bottom can be used). When using the Fenske equation for a shortcut method such as FUG (Fenske-Underwood-Gilliland), the concepts of constant volatility, key components, and the non-distribution of the non-key components are applied. The assumption of constant volatility is not accurate many times and, therefore, the actual value of Nt^{min} can be different than Nt^{min}_{Fenske}, so we do not use Eq. (1) to define a minimum number of trays, but only to define the start point of the search, i.e. $Nt^{min}_{approx} = Nt^{min}_{Fenske}$.

Since feasible solutions may exist for candidates with $Nt < Nt^{min}_{approx}$, the preliminary search consists in evaluating the row of candidates with $Nt < Nt^{min}_{approx}$, sequentially: first $Nt = Nt^{min}_{approx} - 1$, then $Nt = Nt^{min}_{approx} - 2$, and so on, until a row without feasible candidates is found. The first feasible candidate found becomes the optimization incumbent and it is substituted if a subsequent feasible candidate presents a lower value of the objective function.

When the row entirely composed of infeasible candidates is identified, the Set Trimming step eliminates the set of candidates with a smaller number of trays (i.e. the candidates located above the identified row of infeasible candidates). It is important to observe that the algorithm employed to solve the mathematical model must be robust, otherwise a feasible solution could be discarded mistakenly due to the limitation of the algorithm to converge. Fig. 4 depicts the algorithm.

To better illustrate the algorithm, assume that $Nt^{min}_{approx} = 8$; then, the first candidate to be evaluated in this preliminary search would be $(Nt, Nf) = (7, 2)$, followed by all candidates with $Nt = 7$ and larger feed tray locations (Fig. 5a). Assuming that a feasible solution is found for some feed tray location, say for example $(Nt, Nf) = (7, 4)$, then the row of candidates with $Nt = 6$ is also evaluated. If no feasible solution is found, then $Nt^{start} = 7$ and all the candidates with $Nt \leq 6$ are discarded. In another scenario (Fig. 5b) where $Nt^{min}_{approx} = 8$, but no feasible solutions are found for candidates with $Nt = 7$, then all candidates with $Nt \leq 7$ are eliminated and $Nt^{Start} = 8$. Note that the algorithm for the minimum number of trays (Fenske or other) obtains a minimum number of trays that could be feasible or infeasible. This is why the algorithm only eliminates candidates with a number of trays lower than the number of trays associated with a entire row of infeasible candidates. In this situation, the subsequent enumeration using proper models will find the

Initialize the incumbent: $Cost^{Inc} = +\infty$
Obtain an estimative of the minimum number of trays (Nt^{min}_{approx}) (Fenske or other)
Initialize the preliminary search space: $Nt = Nt^{min}_{approx} - 1$
Initialize the stopping criterion: *stop* = *False*
While *stop* = *False*
  For $Nf = 2$ to $(Nt - 1)$
    Solve model for candidate (Nt, Nf)
    If the current candidate is feasible and $Cost(Nt, Nf) < Cost^{Inc}$
      Update the incumbent: $Cost^{Inc} = Cost(Nt, Nf)$
    End If
  End For
  If there were no feasible candidates in the current row Nt
    Let *stop* = *True*
  Else
    Go to the next row: $Nt \leftarrow Nt - 1$
  End If
End While
Eliminate from the search all candidates with number of trays smaller than Nt
The starting number of trays for the Enumeration is: $Nt^{Start} = Nt^{min}_{approx}$

Fig. 4. Preliminary search algorithm and the Set Trimming step.

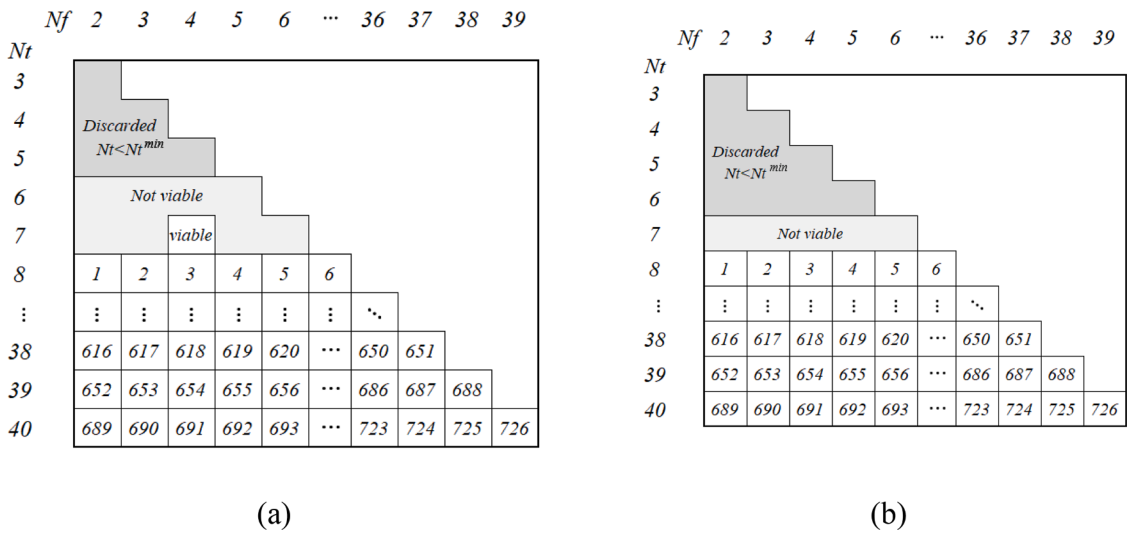

Fig. 5. Examples of preliminary search and Set Trimming: (a) There is a feasible candidate in row $Nt = 7$ ($Nt^{start} = 8$, $Nt^{min} = 7$); (b) There are no feasible candidates in row $Nt = 7$ ($Nt^{start} = 8$).

infeasible solution, anyway.

After the Set Trimming step, one of the enumeration procedures can be applied to the remaining search space, characterized by $Nt^{start} \leq Nt \leq Nt^{max}$.

6. Exhaustive enumeration

For the Exhaustive Enumeration, all candidates left after the Set Trimming are evaluated successively, and exhaustively, until there are no candidates left. Each time a better solution than the incumbent is found, it is saved as the new incumbent. At the end of the search, the incumbent solution is the global optimum. Fig. 6 depicts the algorithm of Exhaustive Enumeration.

7. Smart enumeration

Smart Enumeration (Costa and Bagajewicz, 2019) aims at reducing the computational effort of Exhaustive Enumeration. It consists of ordering all candidates in ascending order of the lower bound of the objective function, and then evaluating them one by one, starting from the lowest lower bound, to identify feasible incumbents. The procedure stops when the lower bound is larger than the incumbent objective function.

Lower bounds are constructed assuming that the capital cost is a function of the number of trays and the column diameter, which in turn, depends on the vapor flow rate only (an example of an objective function with this feature can be found in Luyben and Floudas (1994)). When using a model that does not consider constant tray spacing, there is a trade-off between the number of trays and the column diameter, but this issue is not explored here.

Then, for a given number of trays, the cost becomes a function of the vapor boil-up flow. Moreover, the lowest vapor flow rate corresponds to the largest number of trays. Therefore, a lower bound of the capital cost can be constructed by using the minimum vapor flow rate related to the largest number of trays, for each feed tray location.

In turn, the heat loads of the reboiler and condenser determine the operational cost. Therefore, for a given feed tray location, the corresponding values associated with the column with the maximum number of trays are used to obtain a lower bound of the operational cost. The sum of the lower bounds of the capital and operational costs on a yearly basis provides the lower bound of the total annualized cost, which is the objective function of the design optimization problem.

The first task is the generation of lower bounds for the candidates left, after the Set Trimming step ($Nt^{start} \leq Nt < Nt^{max}$). To do that, all candidates from the last row $(Nt, Nf) = (Nt^{max}, Nf)$ are evaluated. If a better solution than the incumbent is found, the incumbent is updated.

Initialize the lower limit of the search: $N^{start} = Nt^{min}_{approx}$
Initialize the upper limit of the search: $N^{end} = Nt^{max}$
For $Nt = N^{start}$ to N^{end}
  For $Nf = 2$ to $(Nt - 1)$
    Solve model for candidate (Nt, Nf)
    If the current candidate is feasible and $Cost(Nt, Nf) < Cost^{Inc}$
      Update the incumbent: $Cost^{Inc} = Cost(Nt, Nf)$
    End If
  End For
End For
End of the algorithm: The solution is the incumbent.

Fig. 6. Algorithm: Exhaustive enumeration.

The results from these candidates are used to generate the lower bounds of the other candidates ($Nt^{start} \leq Nt < Nt^{max}, Nf$) (Fig. 7), for the example depicted in Fig. 5(a). This lower bound is generated through the evaluation of the distillation column diameter and operational costs associated with the vapor boil-up obtained with the maximum number of trays (this value of vapor boil-up is always lower than the vapor boil-up for columns with a smaller number of trays).

After the ordering of the remaining candidates ($Nt^{start} \leq Nt < Nt^{max}$, Nf) based on the values of the objective function lower bound, the candidates are evaluated sequentially, until the termination criterion is attained. Fig. 8 displays the complete set of steps of the algorithm.

8. Segmental smart enumeration

This approach consists of updating and improving lower bounds as the search progresses. According to the relation between the number of trays and the reflux ratio (and consequently to the vapor boil-up flow), the larger the number of trays, the smaller the vapor boil-up flow is. Thus, if the lower bounds are constructed using a smaller number of trays than the largest possible one, then the lower bounds will increase. For example, to obtain the lower bound of a candidate (Nt,Nf), instead of using the solution of candidate (Nt^{max},Nf), the solution of candidate (Nt',Nf) is used, with $Nt < Nt' < Nt^{max}$, then a larger lower bound is obtained.

In the Smart Enumeration procedure, the last row of candidates ($Nt=Nt^{max}$) is evaluated and used to generate the lower bound for all candidates with $Nt^{start} \leq Nt < Nt^{max}$. In the Segmental Smart Enumeration procedure, we divide the search space into smaller search intervals of Nt, hence the name segmental. The optimization algorithm explores these intervals sequentially. The analysis of each search interval involves the evaluation of the last row of candidates of this interval, followed by the update of the objective function lower bound of the rest of the candidates within the interval. After that, the candidates of the interval are evaluated following ascending order of the updated lower bound. During the search, when a candidate with an objective function lower than the incumbent is found, the incumbent is updated and the candidates with a lower bound of the objective function higher than the incumbent objective function are eliminated. If the solution is not found in a given interval, then the search goes to the next one.

The limits of an interval i are given by Nt_i^{start} and Nt_i^{end} and the corresponding length is given by $\Delta Nt^{step} = \hat{\sigma}\ Nt^{start}$, rounded up to the next integer value ($\hat{\sigma}$ is a predefined parameter). Therefore, $Nt_i^{start} = Nt_{i-1}^{end} + 1$ and $Nt_i^{end} = Nt_i^{start} + \Delta Nt^{step}$. In order to avoid a very small interval near the last active row of candidates, a check is executed to verify if $N_{i-1}^{end} + \hat{\rho}\ \Delta Nt^{step} > Nt^{last}$, where $\hat{\rho}$ is a predefined parameter between 1 and 2, and Nt^{last} is the last row of the search space that still contains a remaining candidate. If so, then use $Nt_i^{end} = Nt^{last}$. Therefore, instead of defining two intervals at the end, a normal one and a small one, they become one larger step.

The parameter $\hat{\sigma}$ needs to be carefully selected. Indeed, if $\hat{\sigma}$ is too small, an excessive number of intervals must be visited. On the other side, if $\hat{\sigma}$ is too large, the cut of candidates inside each interval may not be effective, and the performance of the algorithm approaches the Smart Enumeration. Fig. 9 displays the complete set of steps of the algorithm.

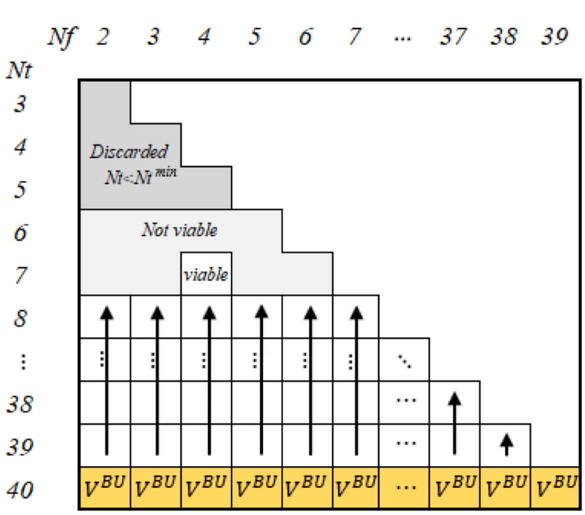

Fig. 7. Lower bounds generated using vapor boil-up results from the last row of candidates ($Nt = Nt^{max}, Nf$).

9. Global optimality

The structure of the Smart Enumeration algorithms guarantees the global optimum will be found. The computational search always updates the incumbent when a solution with a lower value of the objective function is found. Therefore, the sequence of the incumbents is always associated with a decrease in the value of the objective function. Candidates are eliminated only if they are infeasible or if they have a lower bound of the objective function higher than the incumbent. Therefore, the algorithm never eliminates a possible optimal solution. Finally, if there are no more candidates to be tested, the incumbent is the global optimum.

We remark that there is no need of making any convexity assumptions. Moreover, there is no special measure needed if the problem is nonconvex. There is no need for any MINLP procedure or equivalent, only, as stated above, the system of equations that arises from fixing the number of trays and the feed tray needs to be solved robustly.

10. Extensions

We now turn to possible extensions of the method to more complex situations: the same enumeration algorithms can be applied for the design of systems containing multiple feed streams or including the presence of side withdrawal streams.

In these cases, the enumeration algorithms work in more than two dimensions dimensions, because additional variables are added to the problem (e.g. location of the additional feed tray and/or the side product tray). Therefore, the current inner loops in the proposed algorithms associated with the analysis of columns with the same number of trays, but with different feed trays, must be substituted by nested loops, according to the multiple feed or product streams present in the problem. It should be noted that these extensions will be associated with a higher computational effort, due to the expansion of the dimension of the search space.

11. Results

The performance of the proposed procedure is illustrated through the solution of two distillation design problems.

Example 1 involves the design optimization of a distillation column for the separation of a binary mixture with constant relative volatility. This problem was also solved here using different MINLP solvers, thus providing a performance comparison between Set Trimming followed by Enumeration and mathematical programming. Aiming at clarifying the sequence of steps during the enumeration procedures, a detailed analysis of the progression of each proposed enumeration algorithm is also provided.

Example 2 corresponds to a ternary mixture distillation design using a MESH model. The objective of this example is to show that the proposed optimization design procedure can also be applied to more complex distillation models (even using commercial simulation software).

11.1. Example 1

This problem was presented by Luyben and Floudas (1994) using a

Initialize the lower limit of the search: $N^{start} = Nt^{min}_{approx}$
Initialize the upper limit of the search: $Nt^{end} = Nt^{max} - 1$
For $Nf = 2$ to $(Nt^{max} - 1)$
  Solve model for candidate (Nt^{max}, Nf)
  If this candidate is feasible and $Cost(Nt^{max}, Nf) < Cost^{Inc}$
    Update the incumbent: $Cost^{Inc} = Cost(Nt^{max}, Nf)$
  End If
End for
Evaluate the lower bound $LB_{Nt,Nf}$ of the candidates (Nt, Nf), with $Nt^{start} \le Nt \le Nt^{end}$
Let k be an index, such that, $(LB_{Nt,Nf})_k \le (LB_{Nt,Nf})_{k+1}$
Initialize the stopping criterion: *stop* = *False*
Start the index $k = 1$
While *stop* = *False*
  If $Cost^{Inc} \le (LB_{Nt,Nf})_k$
    Let *stop* = *True*
  Else
    Solve model for candidate $(Nt, Nf)_k$
    If the candidate $(Nt, Nf)_k$ is feasible and $Cost(Nt, Nf)_k < Cost^{Inc}$
      Update the incumbent: $Cost^{Inc} = Cost(Nt, Nf)_k$
    End If
    Update the index: $k \leftarrow k + 1$
  End If
End while
End of the algorithm: The solution is the incumbent.

Fig. 8. Algorithm: Smart Enumeration.

superstructure from Viswanathan and Grossmann (1990). The problem corresponds to the separation of a binary mixture with constant relative volatility. The problem data are presented in Table 1.

The mixed integer nonlinear optimization model proposed by Luyben and Floudas (1994) is based on assuming equimolar flows along each distillation section. They solved the problem using an MINLP approach. This model is presented in Supplementary Material – Part A.

The solution to this design problem using enumeration procedures was addressed using an equivalent mathematical model based on the hypothesis of uniform flow along the distillation sections. This model is shown in Supplementary Material – Part B. The same objective function employed by the model presented in Part A was adopted in the solution of the problem using the proposed enumeration procedures. The values of the control parameters of the algorithm were $\hat{\sigma} = 0.75$ and $\hat{\rho} = 1.75$. Two equivalent versions of the proposed optimization approach were implemented: one in GAMS Software Release 35 and another in Python 3.9.12. Additionally, a mathematical model using the MESH equations is used in one of our examples below (Supplementary Material – Part C).

The search space of the problem corresponds to a maximum of 40 trays, i.e. $Nt^{max} = 40$, then $2 \le Nt \le 40$ and $2 \le Nf \le 39$. Therefore, the total number of possible candidates corresponds to 741 alternatives, as illustrated in Fig. 3.

Due to the simplifications of the original model, the minimum number of stages calculated by the Fenske equation is not just an initial estimate, but it is the real minimum number of stages. Therefore, the search in the enumeration procedures starts at Nt^{min}_{Fenske} and all candidates with fewer stages were immediately discarded in the Set Trimming step.

11.1.1. Exhaustive enumeration

The search starts at $Nt^{min}_{Fenske} = 8$, and therefore 15 candidates were discarded immediately. Because $Nt^{max} = 40$, a total of 726 candidates were evaluated. The global minimum corresponds to a column with 16 stages and the feed on tray 9.

11.1.2. Smart enumeration

The same 15 candidates with $Nt < Nt^{min}_{Fenske}$ were discarded in advance. Then, all 38 candidates in the last row of the search space ($Nt = Nt^{max} = 40$) were evaluated and their results used to generate the cost lower bound for all the other candidates left. The 688 candidates left were ordered in increasing order of the lower bound of the objective function. The first candidate of the sequence is $(Nt, Nf) = (9, 8)$, with its lower bound being \$ 26,920.83, and the last candidate is $(Nt, Nf) = (39, 2)$, with a lower bound of \$178,101.20, as shown in Figs. 10 and 11.

During the Smart Enumeration procedure, after the solution of candidate 75, $(Nt, Nf) = (13, 6)$, it is observed that the lower bound of the objective function of the next candidate, $LB_{cost}(19,11) =$ \$34,359.17, is larger than the incumbent solution (\$34,180.48), found in the candidate $(Nt, Nf) = (16, 9)$. Then, the procedure stops, because it means that a better solution cannot be found in the rest of the search space. All candidates evaluated along the search are shown in Fig. 12, which corresponds to a total of 113 candidates.

11.1.3. Segmental smart enumeration

The same 15 candidates with $Nt < Nt^{min}_{Fenske}$ were discarded in advance. Then, all 38 candidates in the last row in the search space ($Nt = Nt^{max} = 40$) were evaluated and their results were used to generate the cost lower bound for all the other candidates left. The lower bounds generated are the same as the ones reported in Fig. 7.

Then, the first subspace of candidates to be explored was defined, using $\hat{\sigma} = 0.75$ and consequently $Nt^{step} = 6$, comprising then candidates with $8 \le Nt \le 14$. The last row of the interval ($Nt = 14$) was evaluated

Initialize the lower limit of the search: $N^{start} = Nt^{min}_{approx}$
Initialize the upper limit of the search: $Nt^{end} = Nt^{max} - 1$
For $Nf = 2$ to $(Nt^{max} - 1)$
  Solve model for candidate (Nt^{max}, Nf)
  If this candidate is feasible and $Cost(Nt^{max}, Nf) < Cost^{Inc}$
    Update the incumbent: $Cost^{Inc} = Cost(Nt^{max}, Nf)$
  End If
End For
Evaluate the lower bound $LB_{Nt,Nf}$ of the candidates (Nt, Nf) with $Nt^{start} \leq Nt \leq Nt^{end}$
Start the interval index: $i = 1$
Identify the row of the last active candidate: $Nt^{last} = Nt^{max} - 1$
Let $\Delta Nt^{step} = \lceil \hat{\sigma}\, Nt^{start} \rceil$
While *stop* = *False*
  Set the lower limit of the current search interval *i*:
  If $i = 1$, then $Nt^{start}_i = Nt^{min}_{approx}$, Else, $Nt^{start}_i = Nt^{end}_{i-1} + 1$, End If
  Set the upper limit of the current search interval *i*:
  If $Nt^{start}_i + \hat{\rho}\, \Delta Nt^{step} > Nt^{last}$, then $Nt^{end}_i = Nt^{last}$, Else, $Nt^{end}_i = Nt^{start}_i + \Delta Nt^{step}$, End If
  For $Nf = 2$ to $(Nt^{end}_i - 1)$
    Solve model for candidate (Nt^{end}_i, Nf)
    If this candidate is feasible and $Cost(Nt^{max}, Nf) < Cost^{Inc}$
      Update the incumbent: $Cost^{Inc} = Cost(Nt^{end}_i, Nf)$
    End If
  End For
  Evaluate the lower bound $LB_{Nt,Nf}$ of the candidates (Nt, Nf) with $Nt^{start}_i \leq Nt \leq Nt^{end}_i$
  Let *k* be an index, such that, $(LB_{Nt,Nf})_k \leq (LB_{Nt,Nf})_{k+1}$ with $Nt^{start}_i \leq Nt \leq Nt^{end}_i$
  Initialize the stopping criterion: *stop_interval* = *False*
  Start the index: $k = 1$
  While *stop_interval* = *False*
    If $Cost^{Inc} \leq (LB_{Nt,Nf})_k$
      Let *stop_interval* = *True*
    Else
      Solve model for candidate $(Nt, Nf)_k$
      If the current candidate is feasible and $Cost(Nt, Nf)_k < Cost^{Inc}$
        Update the incumbent: $Cost^{Inc} = Cost(Nt, Nf)_k$
      End If
      Update the index: $k \leftarrow k + 1$
    End If
  End While
  Eliminate candidates such that: $LB_{Nt,Nf} \geq Cost^{Inc}$ with $Nt^{start} \leq Nt \leq Nt^{end}$
  Update the upper limit of the remaining search space: $N^{last} = \max(Nt_i)$ for $i \in$ set of remaining candidates
  Go to next interval: $i \leftarrow i + 1$
  If there is not any candidates to explore, then *stop* = *True*, End If
End While
End of the algorithm: The solution is the incumbent

Fig. 9. Algorithm: Segmental Smart Enumeration.

and its results improved the subspace's lower bounds. An incumbent with a cost of \$34,359.40 was found in this row for the candidate $(Nt, Nf) = (14, 8)$ and all candidates with a larger lower bound were discarded, leaving only the candidates with the lower bounds shown in Fig. 13.

The 21 candidates of the first search subspace were ordered and the search proceeded. No new incumbent was found in this first interval.

According to the value of Nt^{step}, the next search subspace is $15 \leq Nt \leq 21$. However, because row 19 is the last one with active candidates in the search space, the new interval is defined to be $15 \leq Nt \leq 19$. Then, the row $Nt = 19$ is solved to improve the lower bounds of the candidates in the second interval. Despite no new incumbent being found in row 19, the improved lower bounds allowed a few more candidates to be discarded, as shown in Fig. 14.

After these candidates were evaluated, the search stops. The incumbent solution (\$34,180.48) found in candidate $(Nt, Nf) = (16, 9)$ is the global optimum. A total of 106 candidates were evaluated by this alternative of enumeration.

11.1.4. *Optimal solution*

The details of the global optimal solution attained by the different enumeration procedures are presented in Table 2. An illustration of the

Table 1
Example 1: Problem data.

Variable	Unit	Value
Feed flow rate	kmol/min	1
Feed composition	–	0.45
Relative volatility	–	2.5
Distillate composition	–	0.98
Bottom product composition	–	0.02
Tray spacing flow constant	m/s	0.107
Flooding factor	–	0.8
Molecular weight	kg/kmol	92
Liquid density	kg/m^3	883
Vapor density	kg/m^3	2.9
Vaporization latent heat	10^6 kJ/kmol	0.031
Condensation latent heat	10^6 kJ/kmol	0.032
Low pressure steam cost coefficient	\$min/(10^6 kJ yr)	6.1×10^5
Cooling water cost coefficient	\$min/(10^6 kJ yr)	1.5×10^4
Tax factor	–	0.4
Payback period	yr	4

search space is presented in Fig. 15, which shows the contour curves of the cost function, considering a continuous representation of the number of trays and feed tray location.

11.1.5. Computational performance

The computational performance of the design solution procedures using enumeration is depicted in Table 3. The reported computational times were obtained using GAMS codes running in a computer Intel core i7-10700 8GB DDR4. This table also contains the performance of comparative solutions using mathematical programming in GAMS with the following solvers: BARON, DICOPT, SBB, and ALHPAECP. BARON is a global solver and DICOPT, SBB and ALPHAECP are local solvers.

All enumeration procedures attained the global optimum. The Segmental Smart Enumeration is the fastest enumeration option, involving the simulation of only 14.3% of the total number of possible combinations of number of trays and feed tray location.

The BARON run was interrupted in a computational time that was more than 25 times higher than the computational time consumed by the Segmental Smart Enumeration run (the BARON gap between the upper and lower bounds was still 63%).

Only one of the local solvers attained the global optimum (SBB) and one of them was considerably slow (ALPHAECP). Despite the very low computational times associated with SBB and DICOPT, it is important to observe that they do not guarantee global optimality. Moreover, once the solution is obtained, one cannot know if the solution is global or not, as there are no necessary and sufficient conditions for global optimality. The results above show that the global solver takes inadmissible time.

Because the proposed optimization schemes do not need to use specialized optimization solvers, they can be easily implemented in any computational language. For example, the proposed algorithms were also implemented in Python. The computational times were

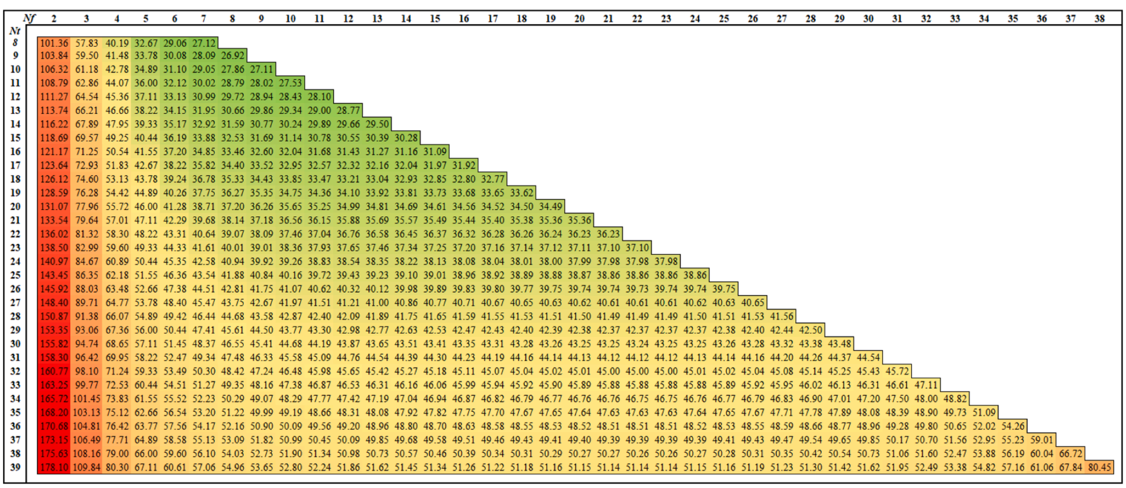

Fig. 10. Example 1 – Smart Enumeration: Lower bound of the candidates cost (\$ $\times$ 10^3).

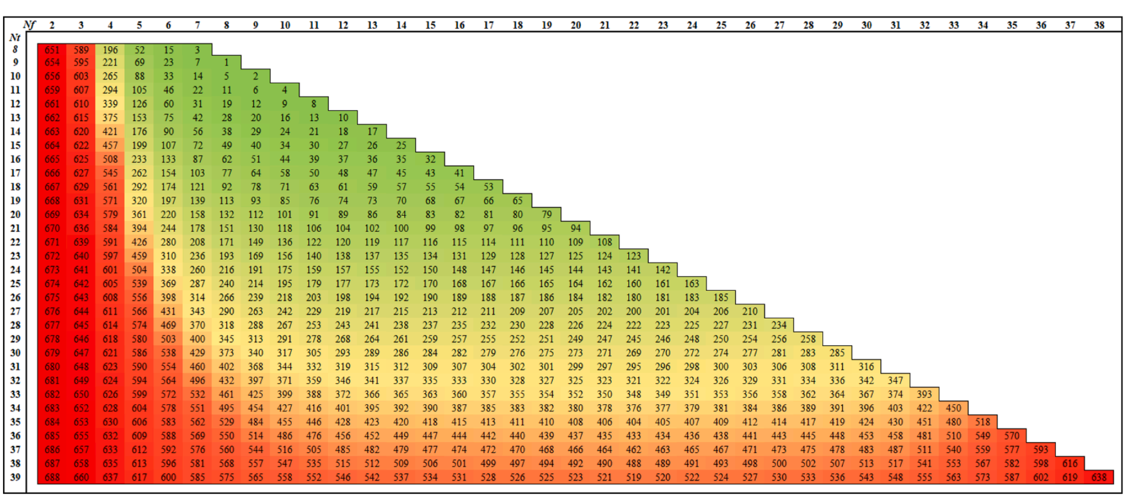

Fig. 11. Example 1 – Smart Enumeration: Order of the candidates according to their lower bound of the objective function.

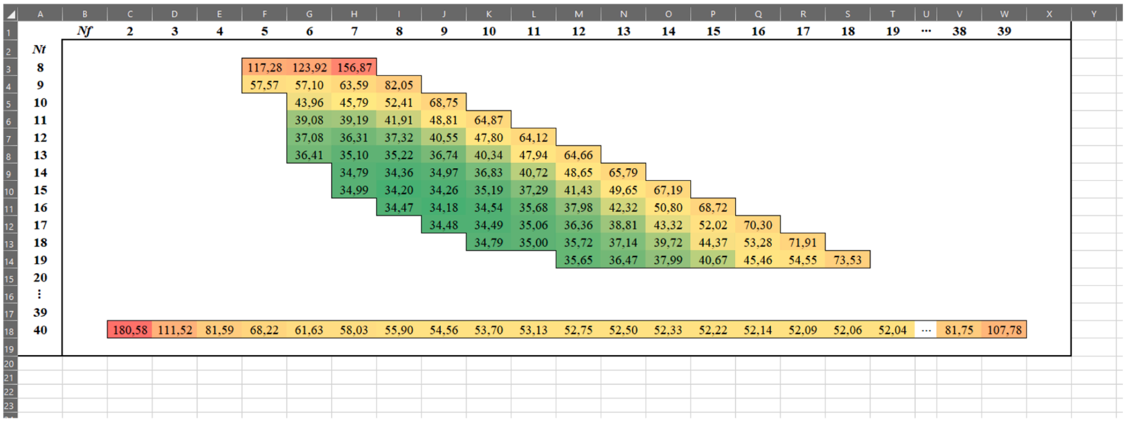

Fig. 12. Example 1 – Smart Enumeration: Costs (\$ $\times$ 10^3) of the evaluated candidates.

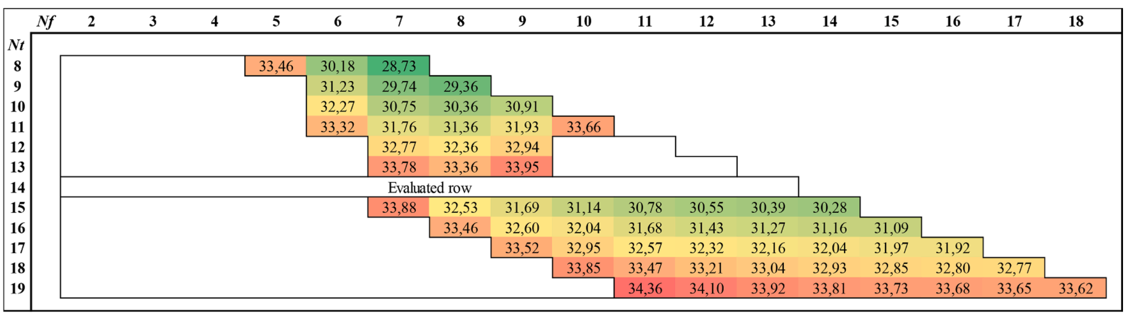

Fig. 13. Example 1 – Segmental Smart Enumeration: Improved lower bounds (\$ $\times$ 10^3) for active candidates after solving row Nt = 14.

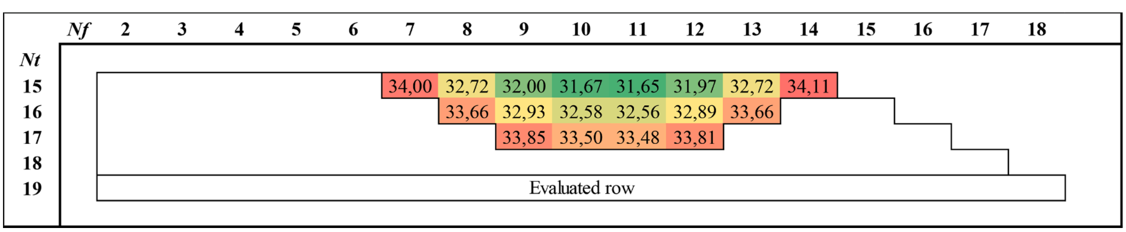

Fig. 14. Example 1 – Segmental Smart Enumeration: Improved lower bounds (\$ $\times$ 10^3) for active candidates after solving row Nt = 19.

Table 2
Example 1: Global optimum.

Variable	Unit	Global optimum
Number of stages	–	16
Feed stage	–	9
Reflux ratio	–	1.81
Top product flow	kmol/min	0.4479
Bottom product flow	kmol/min	0.5521
Top product mass composition	–	0.98
Bottom product mass composition	–	0.02
Liquid flow in the rectifying section	kmol/min	0.8097
Vapor flow in the rectifying section	kmol/min	1.2576
Liquid flow in the stripping section	kmol/min	1.8097
Vapor flow in the stripping section	kmol/min	1.2576
Diameter	m	0.7535
Cost	\$/yr	34,180.48

considerably smaller than those observed in the GAMS runs, as shown in Table 4 (the number of enumerated candidates and the solution were identical to the GAMS codes).

11.2. Example 2

This design problem in Example 2 corresponds to the separation of a ternary mixture of benzene, toluene, and xylene (BTX), as shown in Table 5. We solved this problem using the mesh equations (Supplementary Material - Part C). The values of the control parameters were $\hat{\sigma} = 0.5$ and $\hat{\rho} = 1.5$. The thermodynamic properties were determined using the Python Thermo Package (Bell, 2017). The fugacity coefficient of the vapor phase was evaluated using the Peng-Robison equation of state and the activity coefficient of the liquid phase was evaluated using the NRTL model. The Set Trimming and Enumeration algorithms were implemented in Python.

The maximum number of trays of the search space was 40. This set of values defines a search space composed of 741 different pairs of number of trays and feed tray locations.

11.2.1. Optimal solution

As expected, all three Set Trimming and Enumeration techniques found the same global optimum: a column with 23 trays and the feed location at the eighth tray. The details of the solution are depicted in Table 6. The contour curves of the objective function considering a continuous representation of the number of stages and feed tray location

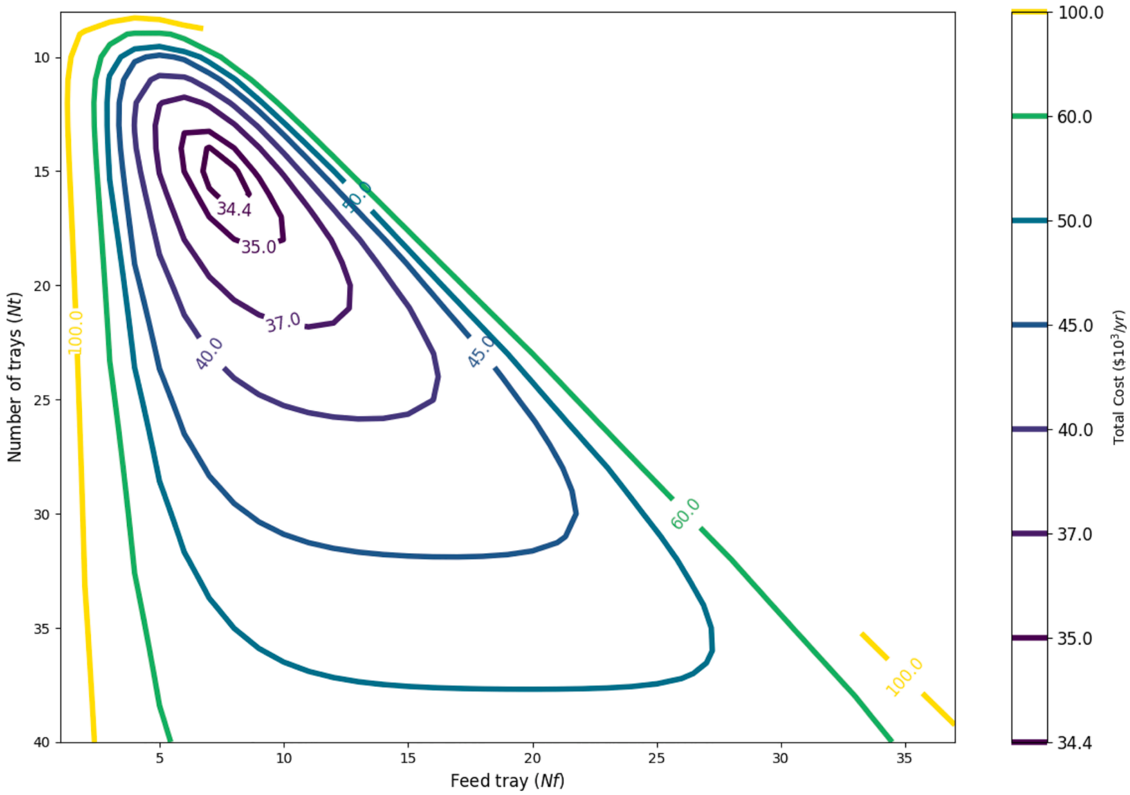

Fig. 15. Example 1: Contour curves of the objective function.

Table 3
Example 1: Computational effort and cost results for all approaches in GAMS.

Approach	Elapsed time (s)	Enumerated candidates	Cost ($/yr)
Exhaustive Enumeration	13156	726	34,180.48
Smart Enumeration	1321	113	34,180.48
Segmental Smart Enumeration	1221	106	34,180.48
MINLP - BARON	31787*	-	40,246.37
MINLP - SBB	0.30**	-	34,180.48
MINLP - DICOPT	0.37**	-	40,195.98
MINLP - ALHPAECP	2827**	-	34,203.78

* Solution associated with a 63% gap
** Global optimum not guaranteed.

Table 4
Example 1: Computational time of set trimming and enumeration in python.

Approach	Elapsed time (s)
Exhaustive Enumeration	17.44
Smart Enumeration	3.02
Segmental Smart Enumeration	2.97

are shown in Fig. 16. The contour curves depicted in Fig. 16 are similar to those present in Fig. 15, which suggests that the design problem presents a convex (unimodal) behavior, but this was not formally proved. However, the proposed enumeration design algorithms does not depend on any convexity or unimodality assumption to obtain the global optimum.

11.2.2. Computational performance

Table 7 displays the computational performance of the different alternatives of Set Trimming and Enumeration. The Set Trimming step

Table 5
Example 2: Problem data.

Variable	Unit	Value
Feed flow rate	kmol/h	100
Feed temperature	°C	113.4
Feed mole fraction of benzene	–	0.19
Feed mole fraction of toluene	–	0.39
Feed mole fraction of xylene	–	0.47
Top recovery of benzene	–	0.99
Bottom recovery of toluene	–	0.99
Column pressure	kPa	100

Table 6
Example 2: Global optimum.

Variable	Unit	Global optimum
Number of trays	–	23
Feed stage	–	8
Reflux ratio	–	6.63
Top product flow	kmol/h	14.25
Bottom product flow	kmol/h	85.75
Top mole fraction of benzene	–	0.97
Bottom mole fraction of toluene	–	0.45
Reboiler duty	10^6 kJ/h	3.00
Condenser duty	10^6 kJ/h	3.38
Column diameter	m	1.06
Column length	m	16.82
Reboiler area	m^2	25.48
Condenser area	m^2	14.44
Capital cost	10^3 $/yr	217.27
Utility cost	10^3 $/yr	78.41
Total cost	10^3 $/yr	295.67

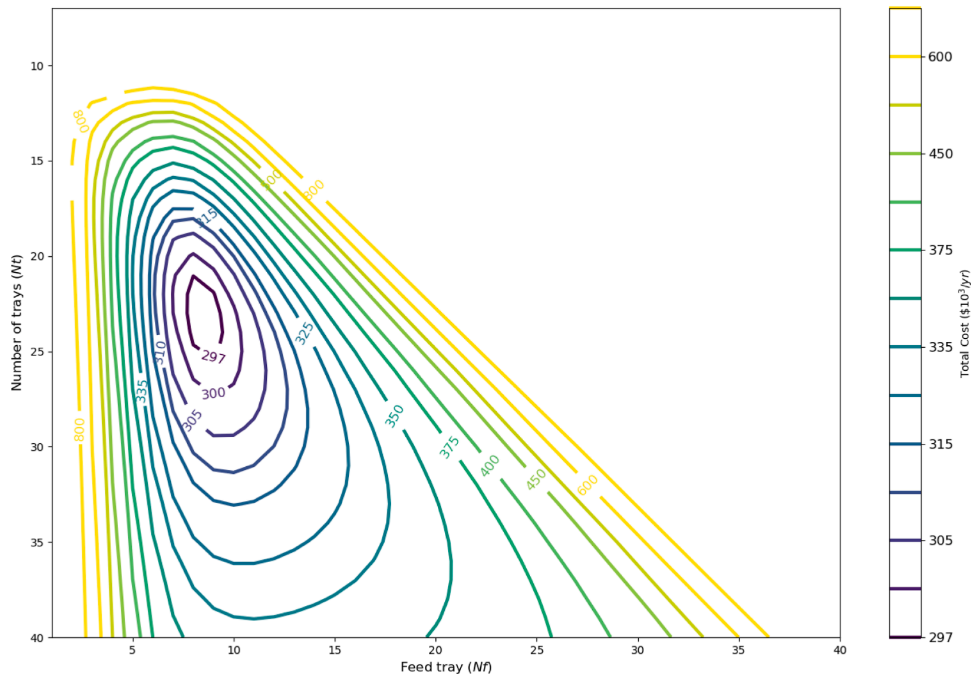

Fig. 16. Example 2: Contour curves of the objective function.

Table 7
Example 2: Computational time of set trimming and enumeration in python.

Approach	Elapsed time (min)	Enumerated candidates	Cost($/yr)
Exhaustive Enumeration	41	705	295,670
Smart Enumeration	16	265	295,670
Segmental Smart Enumeration	5	138	295,670

eliminated 36 candidates before the enumeration (it eliminated all candidates with $Nt < 11$). The reported computational times correspond to Python codes running in the same computer mentioned in Example 1.

The Segmental Smart Enumeration is the fastest enumeration option, associated with a total number of simulations that corresponds to only 18.6% of the total number of possible combinations between the number of trays and feed tray location.

The more complex model employed in Example 2 implied an increase in the computational times when compared with the results of Example 1 in Python, displayed in Table 4. However, it is important to observe that the Segmental Smart Enumeration took only 5 minutes to identify the global optimum (about 1/3 of the Smart Enumeration and 1/8 of the Exhaustive Enumeration), which is a reasonable value for the procedure to be applied in practice.

12. Conclusions

We presented a set of enumeration procedures to obtain the globally optimal design of a distillation column. The procedures are robust, as they always attain the global optimum. The computational advantage over the use of MINLP procedures is significant.

The proposed optimization approach is not associated with any specific distillation model. It can be applied to models considering constant flow rates along each column section and to more sophisticated models, such as those using MESH equations. This flexibility also indicates that the proposed procedure can be linked to flowsheet simulators. While local solvers are fast, some fail to obtain the global solution. Even if they do (like SBB does in our example 1 above), one cannot be sure this solution is global, as there is no necessary condition for global optimality. We did not test the use of mathematical programming local solvers for models based on MESH equations for the same reason. We did not do it for BARON, discouraged by the results of Example 1.

Therefore, the main benefits of the proposed Smart Enumerations schemes are: (1) They are robust, i.e., the search algorithms always identify the global optimum compared to the illustrated computational inefficiency of global solvers and the inherent inability of meta-heuristics; (2) Smart enumerations reduce the computational time compared with exhaustive enumeration approaches to attain the global optimum.

Future research work involving these design procedures will be associated with the investigation of how these algorithms can be applied to solve flowsheet synthesis problems, e.g., the design of sequences of distillation columns.

CRediT authorship contribution statement

Alice Peccini: Conceptualization, Investigation, Methodology, Software, Writing – original draft. **Lucas F.S. Jesus:** Visualization, Investigation, Methodology, Software. **Argimiro R. Secchi:** Supervision, Writing – review & editing. **Miguel J. Bagajewicz:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing. **André L.H. Costa:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

André L. H. Costa and Argimiro R. Secchi thank the National Council for Scientific and Technological Development (CNPq) for the research productivity fellowship (Processes 310390/2019–2 and 303587/2020-2). André L. H. Costa also thanks the financial support of Prociência Program (UERJ). Miguel J. Bagajewicz would like to thank the Universidade do Estado do Rio de Janeiro (Brazil) for its scholarship as Visiting Researcher - PAPD Program and the Universidade Federal do Rio de Janeiro (Brazil) for its support. He also acknowledges the support of the Universidade Federal do Rio de Janeiro (UFRJ) for the financial support. Lucas F. S. Jesus thanks the Coordination for the Improvement of Higher Education Personnel (CAPES) for the scholarship.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2023.108254.

References

Adiche, C., Vogelpohi, A., 2011. Short-cut methods for the optimal design of simple and complex distillation columns. Chem. Eng. Res. Des. 89, 1321–1332.
Barttfeld, M, Aguirre, P.A., Grossmann, I.E., 2003. Alternative representations and formulations for the economic optimization of multicomponent distillation columns. Comput. Chem. Eng. 27, 363–383.
Bell, C.. Thermo: Chemical properties component of Chemical Engineering Design Library (ChEDL). https://github.com/CalebBell/thermo.
Caballero, J.A., 2015. Logic hybrid simulation-optimization algorithm for distillation design. Comput. Chem. Eng. 72, 284–299.
Caballero, J.A., Milán-Yañez, D, Grossmann, I.E., 2005. Rigorous design of distillation columns: integration of disjunctive programming and process simulators. Ind. Eng. Chem. Res. 44, 6760–6775.
Caballero, J.A., Grossmann, I.E., Górack, A., Sorensen, E., 2014. Optimization of distillation processes. Distillation: Fundamentals and Principals. Elsevier, pp. 438–496.
Costa, A.L.H., Bagajewicz, M.J., 2019. 110th Anniversary: on the departure from heuristics and simplified models toward globally optimal design of process equipment. Ind. Eng. Chem. Res. 58 (40), 18684–18702.
Dowling, A.W., Biegler, L.T., 2015. A framework for efficient large-scale equation-oriented flowsheet optimization. Comput. Chem. Eng. 72, 3–20.
Fenske, M.R., 1932. Fractionation of straight-run pennsylvania gasoline. Ind. Eng. Chem. 24 (5), 482–485.
Gilliland, E.R., 1940. Multicomponent rectification estimation of the number of theoretical plates as a function of the reflux ratio. Ind. Eng. Chem. 32 (9), 1220–1223.
Glinos, K.N., Malone, M.F., 1985. Design of sidestream distillation columns. Ind. Eng. Chem. Process Des. Dev. 24 (3), 822–828.
Ibrahim, D., Jobson, M., Guillén-Gosálbez, G., 2017. Optimization-based design of crude oil distillation units using rigorous simulation models. Ind. Eng. Chem. Res. 56, 6728–6740.
Javaloyes-Antón, J., Ruiz-Femenia, R., Caballero, J.A., 2013. Rigorous design of complex distillation columns using process simulators and the particle swarm optimization algorithm. Ind. Eng. Chem. Res. 52, 15621–15634.
Jia, S., Qian, X., Liu, X., Luo, Y., Yuan, X., 2022. A new efficiency relaxation model for rigorous stage number optimization of distillation columns. Chem. Eng. Sci. 260, 117924.
Kamath, R.S., Grossmann, I.E., Biegler, L.T., 2010. Aggregate models based on improved group methods for simulation and optimization of distillation systems. Comput. Chem. Eng. 34 (8), 1312–1319.
King, C.J., 2013. Separation Processes, 2nd Edition. Dover Publications.
Kirkbride, G.G., 1944. Process design procedure for multicomponent fractionators. Pet. Refin. 23, 87.
Kister, H.Z., 1992. Distillation Design. McGraw-Hill Professional Publishing.
Kong, L., Maravelias, C.T., 2019. From graphical to model-based distillation column design: a McCabe-Thiele-inspired mathematical programming approach. AIChE J. 65 (11) https://doi.org/10.1002/aic.16731.
Liñán, D.A., Ricardez-Sandoval, L.A., 2023. A Benders decomposition framework for the optimization of disjunctive superstructures with ordered discrete decisions. AIChE J. e18008. https://doi.org/10.1002/aic.18008.
Lyu, H., Cui, C., Zhang, X., Sun, J., 2021. Population-distributed stochastic optimization for distillation processes: Implementation and distribution strategy. Chem. Eng. Res. Des. 168, 357–368.
Luyben, M.L., Floudas, C.A., 1994. Analyzing the interaction of design and control - 1. a multiobjective framework and application to binary distillation synthesis. Comput. Chem. Eng. 18 (10), 933–969.
McCabe, W.L., Thiele, E.W., 1925. Graphical design of fractionating columns. Ind. Eng. Chem. 17 (6), 605–611.
Nikolaides, I.P., Malone, M.F., 1987. Approximate design of multiple-feed/side-stream distillation systems. Ind. Eng. Chem. Res. 26 (9), 1839–1845.
Ponchon, M., 1921. Étude graphique de la destillation fractionnée industrielle. Tech. Mod. XIII, 20–55.
Smith, R., Jobson, M., 2000. Distillation. Encyclopedia of Separation Science. Elsevier, pp. 84–103.
Savarit, R., 1922. Definition of distillation, simple discontinuous distillation, theory and operation of distillation column, and exhausting and concentrating columns for liquid and gaseous mixtures and graphical methods for their determination. Arts Metiers 65.
Sorensen, E., 2014. Principles of Binary Distillation. Distillation: Fundamentals and Principals. Elsevier, pp. 145–185.
Underwood, A., 1948. Fractional distillation of multicomponent mixtures. Chem. Eng. Prog. 44, 603–614.
Viswanathan, J., Grossmann, I.E., 1990. A combined penalty function and outer-approximation method for MINLP optimization. Comput. Chem. Eng. 14, 769–782.
Viswanathan, J., Grossmann, I.E., 1993a. An alternate MINLP model for finding the number of trays required for a specified separation objective. Comput. Chem. Eng. 17, 949–955.
Viswanathan, J., Grossmann, I.E., 1993b. Optimal feed locations and number of trays for distillation columns with multiple feeds. Ind. Eng. Chem. Res. 32, 2942–2949.
Yeoh, K.P., Hui, C.W., 2021. Rigorous NLP distillation models for simultaneous optimization to reduce utility and capital costs. Clean. Eng. Technol. 2, 100066.
Yeomans, H., Grossmann, I.E., 2000. Optimal design of complex distillation columns using tray-by-tray disjunctive programming models. Ind. Eng. Chem. Res. 39 (11), 4326–4335.