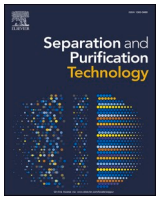
Separation and Purification Technology 306 (2023) 122448

Contents lists available at ScienceDirect

Separation and Purification Technology

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



# Direct search methods for the fast optimisation of batch distillation processes

Laszlo Hegely

*Budapest University of Technology and Economics, Faculty of Mechanical Engineering, Department of Building Services and Process Engineering, 1111 Budapest, Muegyetem rkp. 3-9, Hungary*

A R T I C L E I N F O

*Keywords:*
Batch distillation
Optimization
Genetic algorithm
Nelder-Mead simplex
Complex
Derivative-free optimization

A B S T R A C T

Batch distillation can be conveniently modelled by professional flow-sheet simulators. Optimisation can be performed by coupling an external optimiser to the simulator. The most frequently used method is the genetic algorithm (GA), which, however, requires a high number of simulations to evaluate the objective function. Two direct search methods, the Nelder-Mead simplex and the Box-complex algorithms are applied to reduce the computational intensity of optimisation. Calculations are performed for a case study from the literature where the profit of the regeneration of a multicomponent azeotropic waste solvent mixture was maximised by a GA. The influence of the parameters of the optimisation methods is investigated for each method. The highest profit is reached by the simplex algorithm. Both the simplex and complex algorithms generally outperform GA with a much lower number of simulations. Therefore, direct search methods can be used for fast and efficient optimisation of batch distillation processes.

## 1. Introduction

Batch distillation (BD) has several advantages over continuous distillation processes, such as its ability to treat mixtures with varying amount and composition and to separate multiple components in the same column [1]. It is frequently applied in the pharmaceutical and fine chemical industry and in paint and spirit production, among others. However, the specific energy demand of batch distillation is even higher than that of continuous distillation.

The treatment of waste solvent mixtures, which is usually performed by BD, is particularly challenging since the components of the mixtures often form azeotropes with each other, which makes it necessary to take multiple cuts (fractions). In general, only a single component is recovered in high purity in the main cut. More volatile components or azeotropes are removed in fore-cuts, while taking an after-cut might be necessary to remove pollutants from the still residue. Waste solvent regeneration is favourable not only from an economic point of view since the cost of purchasing fresh solvent is avoided, but also from an environmental one, because the incineration of waste solvents causes the emission of $CO_2$ and potentially other harmful substances [2].

In the case of azeotropic and close-boiling mixtures, the recovery of the product can be increased by applying special batch distillation methods. In batch extractive distillation (BED, [3]), a mass separating agent, the entrainer, is fed continuously into the column and changes the relative volatilities of the original components favourably [4]. In batch, heterogeneous azeotropic (or heteroazeotropic) distillation (BHAD, [5]), the entrainer, added to the still pot with the charge, forms a heterogeneous azeotrope with at least one component. The azeotrope obtained as the distillate can be separated by decantation. A combination of BED and BHAD is batch heterogeneous extractive (or heteroextractive) distillation (BHED, [6]), where the heterogeneous entrainer is fed continuously into the column. If the azeotrope is pressure—sensitive, pressure-swing distillation (PSD) can be applied, which does not require the addition of an entrainer. In batch PSD must be realised by changing the pressure of the column between operational steps [7] or connecting two columns with different pressures (double-column systems, [8]).

The efficiency of BD processes can be improved by optimisation. The energy demand or environmental impact of the process can be decreased, or the recovery of the product(s) can be increased. Since BD is a dynamic process, the optimisation problem is a dynamic one (also known as optimal control problem) whose solution consists of finding the time-optimal profile of the optimisation variables (e.g. reflux ratio). In the following sections, the literature on the optimisation of batch distillation is reviewed with a particular emphasis on the application of 1. professional flow-sheet simulators for the optimisation of BD, 2. the

*E-mail address:* hegely.laszlo@gpk.bme.hu.

https://doi.org/10.1016/j.seppur.2022.122448
Received 9 June 2022; Received in revised form 18 October 2022; Accepted 19 October 2022
Available online 22 October 2022

Nelder-Mead simplex and Box-complex direct search algorithms for the optimisation of either continuous or batch distillation.

### 1.1. Optimisation of batch distillation: Problem formulation

The book of Mujtaba [1], which dedicates an entire chapter to the optimisation of BD, distinguished three types of optimisation problems based on the objective function: minimum time (t), maximum distillate (D) and maximum profit/productivity problems. The objective function (P) of the maximum profit problem is the profit of the operation per unit time calculated as the value added to product per time minus the operating cost. Barreto et al. [9] and Hegely and Lang [10] used an alternative formulation of the maximum profit problem with P•t as the objective function. If the operating cost and the value of the charge are neglected, a maximum productivity problem is obtained [11], whose objective function is proportional to D/t, which is also known as specific product flow rate. In all the above formulations, the product purity is specified and thus acts as an inequality constraint. In the minimum time problem, the amount of the distillate, in the maximum distillate problem, the batch time is specified.

Other possible objective functions include the energy demand [12], the specific energy demand [13], the specific product flow rate (the amount of product divided by the batch time) [14,15,16,17], the processing capacity [18], the thermodynamic efficiency of the process [19], and various indicators of environmental impact. Pommier et al. [20] applied a complex cost function, including cost elements as immobilisation, energy, load, entrainer, column and tank treatment for a BHAD process. Environmental indicators are used as objective functions in multi-objective optimisation problems. In addition to the total annual cost (TAC), Wang et al. [21] used $CO_2$ emissions, Zhao et al. [22] global warming and acidification potential as objective functions. Barreto et al. [23] devised an environmental objective taking into account the $CO_2$ emission and the human toxicity (expressed by LD50 and threshold limit values (TLV)). If not only the operation but also the design of the column is part of the optimisation problem, objective functions might be the investment cost [24,25,26] or TAC [27,21,22,26] Lelkes et al. [28] used an estimation of the total cost of a BED column.

### 1.2. Solution approaches

The solution approaches of the optimisation problem can be divided into two main categories, whether they are based on Pontryagin's maximum principle or nonlinear programming (NLP) [1]. Pontryagin's maximum principle can be used to determine the time-optimal profile of the optimisation variables by maximising the Hamiltonian function.

Nonlinear programming approaches can be further classified into feasible and infeasible path approaches. In the feasible path approach, the model of the process is solved for every set of optimisation variable values generated by an optimisation algorithm. Based on the results, the algorithm generates a new set of optimisation variables. These steps are repeated until convergence. Since at least some optimisation variables can be continuous functions of time (e.g. the reflux ratio), an infinite-dimensional problem is obtained. The problem can be reduced into one of finite dimensions by using control vector parameterisation (CVP), where the operation time is divided into subintervals. In each subinterval, the continuous optimisation variables are given with known basis functions. The parameters of the basis functions in the subintervals are the new optimisation variables. In the simplest and most frequently used case, the optimisation variables are constant in each subinterval. In the case of BD processes with multiple steps, the subintervals can be conveniently chosen to coincide with the steps of the process.

In the infeasible path approach, the dynamic process model is discretised into a set of algebraic equations, which can then be solved by collocation or other methods [29]. This approach avoids the large number of simulations (solving the process model) of the feasible path approach, but it results in an optimisation problem with many equations and variables.

### 1.3. Modelling techniques

The modelling technique influences which solution approaches can be applied. The process can be modelled by explicitly giving the model equations using a programming language, a dedicated modelling environment (such as GAMS) or the custom modelling features of flow-sheet simulators (e.g. gPROMS). Works in the older literature exclusively use this technique, although it is still applied (e.g. by Barreto et al., [9;30,31]). In this case, any of the solution approaches described above are applicable.

An alternative, popular technique is to use the library unit models of professional flow-sheet simulators, which makes the construction and modification of models fast and convenient. The flow-sheet simulators might use a sequential-modular or an equation-oriented approach. However, the capacities of widely used simulators to optimise dynamic processes is limited. A built-in sequential quadratic programming (SQP) method can be available, yet it cannot treat discrete variables, and in the case of certain more complex BD processes, its application might not be possible. Cook et al. [15] did use the built-in SQP optimiser of Aspen Plus, but the description of the optimisation is very limited. MultiBatchDS, a batch distillation flow-sheet simulator developed and used by Diwekar and her co-workers, has an optimisation function, but the method used and its capabilities are not evident from the open literature. It was used by Kim et al. [32] for the optimisation of the separation of an acetonitrile–water mixture in a middle-vessel column using BED. Diwekar and Agrawal [33] optimised the separation of an ideal ternary mixture (pentane-hexane-heptane) in a batch dividing-wall column with the assumption of constant molar overflow.

### 1.4. Optimisation of batch distillation with external optimisation tools coupled to flow-sheet simulators

A more flexible approach is to couple the simulator to an external optimisation tool. As the model equations are not explicitly given or they are not directly accessible for the optimisation tool, optimisation can only be performed by following the feasible path approach, and the simulation is treated as a black-box model to evaluate the objective function and the constraints at each iteration of the optimisation algorithm. The main drawback of this method is that usually a large number of function evaluations is necessary, making the optimisation computationally intensive given that the runtime of a dynamic simulation might be in the order of minutes.

The literature on using external tools to optimise BD processes is very limited. The majority of works [20,23,19,10,34,13,21] used a genetic algorithm (GA). GA is a nature-inspired stochastic optimisation method capable of finding the global optimum. Its implementation is easy; however, it requires a very large number of function evaluations [20]. Pommier et al. [20] compared the results of GA to those of an external SQP optimiser for the separation of water and pyridine with toluene as heterogeneous entrainer. GA was clearly superior, with a 47 % reduction in the cost compared to 12 % by using SQP. The cost could be further reduced to a slight extent by using the best solution of GA as a starting point of SQP. GA was also applied in a second case study where methyl acetate and chloroform were separated in two columns using benzene as a heterogeneous entrainer. Barreto et al. [23] used a multi-objective GA, NSGA-II (nondominated sorting GA; [35], to maximise the profit and minimise the environmental impact of the BHED separation of chloroform and methanol with water as entrainer. The same algorithm was used by Rodriguez-Donis et al. [19] for multi-objective optimisation of the BHAD of the mixture *n*-hexane–ethyl acetate with methanol as entrainer. Wang et al. [21] optimised a double-column configuration for the PSD of *n*-heptane-isobutanol by NSGA-II. The objective functions were the TAC and $CO_2$ emission. Zhao et al. [26] optimised a similar three-column PSD configuration by NSGA-II, but using the investment

cost of third column instead of TAC as one of the objective functions. Hegely and Lang [10] maximised the profit with a mono-objective GA of the regeneration of a multicomponent waste solvent mixture. Two processes were studied: conventional and extractive batch distillation. In both cases, a production campaign consisted of six consecutive batches with the recycling of off-cuts; the optimisation of the batches was performed sequentially. The first batch of the conventional batch distillation process is taken as the case study of the present work. Hegely and Lang [34] maximised the profit of a BHAD process for the separation of isopropanol and water with toluene as entrainer. Nemeth et al. [13] minimised the specific energy demand of a two-column process where the distillate of the first column is the charge of the second one. Acetone was recovered from mixtures containing dichloromethane and water in different concentrations. Nemeth et al. [18] maximised the processing capacity of the two-column process for the separation of acetone–water and water-acetic acid mixtures. Since the recovery of the organic component was kept constant, the maximisation of the capacity is equivalent to the minimum time problem.

Instead of a GA, Leipold et al. [25] used another evolutionary algorithm, non-constrain-dominated sorting modified differential evolution (ncsMDE), for the multi-objective optimisation of a middle-vessel column. The algorithm, implemented in Visual Basic for Applications (VBA) with a Microsoft Excel user interface, performed better than a GA in test problems. Because of the computational intensity of the method, only 20 individuals were used, although the number of generations was high (2163). Li et al. [27] minimised the TAC of a double- and a triple-column pressure-swing BD processes. A sequential iterative method was used (equivalent to a univariate search), which is a local optimisation method. Zhao et al. [26] used the same method for the minimisation of the TAC of the three-column process mentioned earlier. Only two optimisation variables were considered. Asprion et al. [36] used SQP to minimise the sum of squared errors between experimental and simulation data of the batch distillation of an ideal ternary mixture. The simulator used was CHEMADIS, the in-house, commercially unavailable dynamic simulator of BASF.

As shown above, when the modelling is performed with a flow-sheet simulator, the optimisation method is usually a global, although computationally-intensive evolutionary algorithm. Therefore, there is a need for faster optimisation methods. Such methods would not only make it possible to perform more calculations, but to solve problems with a higher number of optimisation variables, as well.

### 1.5. Direct search methods

Direct search methods are derivative-free (also called zeroth-order) ones; thus, they can be coupled to flow-sheet simulators, and since the number of objective function evaluations is likely to be considerably lower than in the case of evolutionary algorithms, they are suitable candidates to reduce the computational intensity of optimisation. However, they are local methods, and thus might converge to local optima. The knowledge of at least one feasible initial point is necessary. Some examples of direct search methods are various random search methods, grid search, univariate optimisation, Hooke and Jeeves' method, Powell's method, as well as the ones tested in the present work: the Nelder-Mead simplex and the Box-complex method [37]. The Nelder-Mead simplex [38] was developed for unconstrained optimisation from the algorithm of Spendley et al. [39] and is based on the movement of an n + 1-dimensional simplex in the *n*-dimensional space of solutions. The Box-complex (*co*nstrained s*implex*) method [40] is a modification of the Spendley algorithm for constrained optimisation. A detailed description of both methods will be given in Section 2.

### 1.6. Optimisation of continuous or batch distillation by using the Nelder-Mead simplex or Box-complex algorithms

The literature on the application of the Nelder-Mead or the Box-complex method for the optimisation of distillation processes is very limited. In particular, only two examples of using either algorithm with a flow-sheet simulator were found: Corbetta et al. [41] applied the simplex method in the inner loop of a hybrid algorithm to solve NLP subproblems, while Giwa and Giwa [42] used a built-in method of Aspen HYSYS based on the complex method. Moreover, to the best of our knowledge, Diwekar et al. [43] and Manca [16] were the only ones to perform the optimisation of BD with one of the methods. Diwekar et al. [43] used a golden complex algorithm in the outer loop of the optimisation of a multicomponent BD process with variable reflux ratios and stopping criteria of the fractions, while the Fibonacci search was used in the inner loop. However, the method is not described in detail, and a discussion is also missing. One can only assume that the golden complex is the modified complex algorithm described in Kovasin et al. [44] that includes a golden search in the reflection step. Manca [16] applied a mathematical model for the BD of a constant relative volatility binary mixture. The reflux ratio was a piecewise function with three time intervals. The duration of the intervals were optimisation variables, as well. A modified simplex method was applied to maximise the capacity factor (equivalent to the specific product flow rate). The purity specifications were taken into account by a penalty function (changing the amount of product to zero), which introduced discontinuities into the objective function. In such cases, the global optimum is located on a discontinuity since reaching purities higher than necessary is suboptimal. Manca [16] compared SQP and the simplex algorithm. Optimisation with SQP was less successful, while the simplex algorithm was not affected by the discontinuities, although it required a higher number of function evaluations (1100–1300). The influence of the initial point was studied by randomly generating 300 points. Although the algorithm did not always converge to the same point, the values objective function were within 12 % of the best one as long as the purity specifications were fulfilled. Manca [16] concluded that the simplex is an efficient and robust optimisation method.

Adams II and Seider [45] performed the optimisation of a semi-continuous distillation column with reaction in a middle vessel by using a single- and a bi-level algorithm. The column, modelled in Aspen Plus, was operated in six cyclically repeating modes. The bi-level algorithm is capable of reducing the computational time. The optimisation variables of the outer loop were the global variables present during the whole operation, while local variables only present at individual modes were treated in the inner loop. The simplex algorithm was used either alone (single-level algorithm) or in the outer loop. The method proved to be the best when the unimodal progression algorithm was used in the inner loop, although only local optima were found. Ghaee et al. [46] applied the Nelder-Mead simplex only to optimise the parameters of the controllers of a continuous extractive distillation process. Wei-zhong and Xi-gang [47] used a simplex simulated annealing algorithm for the optimal synthesis of continuous distillation trains modelled by shortcut methods. Continuous variables were varied by the simplex, the discrete ones by a simulated annealing algorithm. Spasojević et al. [48] minimised the entropy production of a diabatic column described by a mathematical model. The Nelder-Mead algorithm proved to be the best among the methods studied: the Rosenbrock method, Hooke and Jeeves' method, Powell's method, the simplex and the complex algorithm. The number of function evaluations was 1090. Corbetta et al. [41] solved various optimisation case studies related to the separation of bio-mixtures by distillation. The problems were formulated within the generalised disjunctive programming framework and solved by a decomposition strategy based on the Outer-Approximation algorithm. Mixed-integer linear programming (MILP) master problems were solved by a branch & cut method, while a modified Nelder-Mead algorithm was applied in the NLP subproblems. Simulations were performed in PRO/II; the optimisation environment was GAMS.

Umeda and his co-workers optimised an absorber-stripper system (the stripper is a distillation column) by using either the original [49] or a modified [50] complex algorithm. The modified algorithm takes into

account in the expansion step not only the order of the objective function values of the points, but the values themselves, as well. Adelman and Stevens [51], who optimised a plant consisting of a reactor, a heat exchanger, a decanter and a distillation column, stated that the complex algorithm should be capable of finding the global optimum due to the randomly generated initial points and its initial expansion. Holland [52] applied the complex method with penalty terms included in the objective function for the optimisation of continuous columns. Muraki and Hayakawa [53] used the complex algorithm as part of a method to synthesise optimal distillation trains with heat integration. The columns were modelled with short-cut methods. Vu et al. [54] also used the complex method and short-cut models to determine the optimal column configuration for different feed compositions of an ideal ternary mixture. The configurations studied included conventional and thermally coupled ones. Giwa and Giwa [42] used the built-in optimiser of Aspen HYSYS to maximise the purity of the distillate of a reactive distillation column and compared the following methods available in HYSYS: the Fletcher-Reeves method, a quasi-Newton method, SQP, the BOX method (based on the complex algorithm) and a mixed method that is a sequential combination of the BOX and SQP methods. (Note that HYSYS is not capable of modelling batch distillation.) The choice of the objective function is questionable since the purities of the products are generally not considered as objective functions to be maximised but constraints to be respected. Only the fact that high-purity product is produced (moreover, without any constraint on the flow rate of the products) does not give any information about whether the process is favourable economically, energetically or environmentally. The optimisation variables were the reflux ratio, feed flow rate and reboiler heat duty, but the feeding locations were not included. The optimisation was performed without and with considering an inequality constraint on the purity of the bottom product. The former case is not realistic and leads to a low conversion, while in the latter case, the optimiser is not motivated to reduce the energy demand by avoiding overpurification. The Fletcher-Reeves, quasi-Newton and SQP methods failed to find a feasible solution.

### 1.7. Goals and structure of the paper

Performing the optimisation of BD processes modelled in flow-sheet simulators with the Nelder-Mead simplex or Box-complex direct search algorithms instead of the commonly used GA is a promising method to reduce the considerable computational intensity of the optimisation. However, as demonstrated by the literature review, the application of these methods was not yet reported. The goal of the present work is, therefore.

1. to apply the Nelder-Mead simplex and Box-complex algorithms for the optimisation of a multicomponent azeotropic BD process previously optimised by a GA [10],
2. to study the influence of the parameters of the optimisation methods on the results,
3. to compare the results of the three algorithms.

The paper is structured in the following way. After the introduction and literature review of Section 1, Section 2 presents in detail the optimisation methods used in the work. The BD process used as case study [10] is described briefly in Section 3. The results of the optimisation calculations are presented in Section 4. First, the results of each method are shown, then the range of the optimisation variables is discussed, a comparison of the three methods is given, and subsequently, the best result is discussed in detail and compared to that of Hegely and Lang [10]. Finally, Section 5 presents the conclusions and proposes directions for future research.

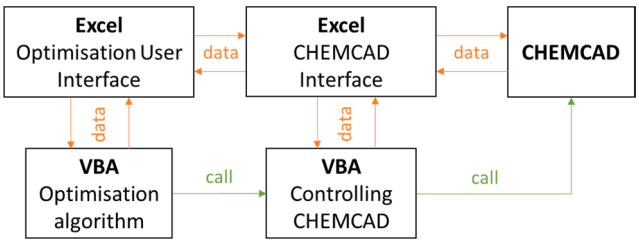



**Fig. 1.** Scheme of the optimisation environment.

## 2. Optimisation methods applied

Three derivative-free optimisation methods are applied in the present work: a genetic algorithm (GA), the Nelder-Mead simplex algorithm [38] and a variant of the Box-complex algorithm [40]. The algorithms were coded in Visual Basic for Applications (VBA) under Microsoft Excel. The objective function is evaluated by performing dynamic simulation with the professional flow-sheet simulator CHEMCAD [55].

The scheme of the optimisation environment is shown in Fig. 1. An Excel sheet was used as a user interface to modify the parameters of the method applied and to show the optimisation results. During the optimisation runs, the algorithm selects a combination of values for the optimisation variables. This set of values is copied to another Excel sheet which is the interface to CHEMCAD. The values of the optimisation variables are passed to CHEMCAD with a Data Map, which then gives back the calculation results. For a steady-state simulation, data transfer would only be necessary before and after running the simulation. However, if it is necessary to change the values of optimisation variables during a dynamic simulation, data transfer must be performed at every time step of the simulation. In the case study to be presented, the reflux ratio is changed, and accumulator tanks are switched based on the distillate composition. Therefore, running the simulation is performed by a VBA subroutine that continuously monitors the simulation results and changes the input values of the simulation (e.g. reflux ratios) accordingly. The algorithm can also prematurely stop the simulation if optimisation constraints are violated to make the optimisation faster.

The (inequality) constraints are taken into account by all the methods with a penalty function: if a constraint is violated, the value of the objective function to be maximised is changed to a large negative number (−10000 $).

### 2.1. Genetic algorithm

The elitist, real-coded GA used here is described in more details in Modla and Lang [56]. The parameters of the algorithm are the population size, the mutation rate and the crossover rate. The optimisation is stopped after 100 generations.

### 2.2. Nelder-Mead simplex method

The Nelder-Mead (or downhill) simplex method is based on operations performed on an n-dimensional simplex where n is the number of optimisation variables. The simplex consists of n + 1 points in the space of the optimisation variables: for n = 3, the simplex is a tetrahedron. Although the initial simplex could be any set of points, Spendley et al. [39] recommended using a regular simplex in their earlier algorithm. The points of a regular simplex (i = 1…n) with side length “a” can be generated from a starting point $\mathbf{X}_{start}$ (also part of the simplex) [37]:

$$\boldsymbol{X}_i = \boldsymbol{X}_{start} + p\boldsymbol{u}_i + \sum_{j=1, j\neq i}^{n} q\boldsymbol{u}_j \quad (1)$$

where $\mathbf{u}_i$ is the unit vector along the $i^{th}$ axis, and the constants p and q are calculated as:

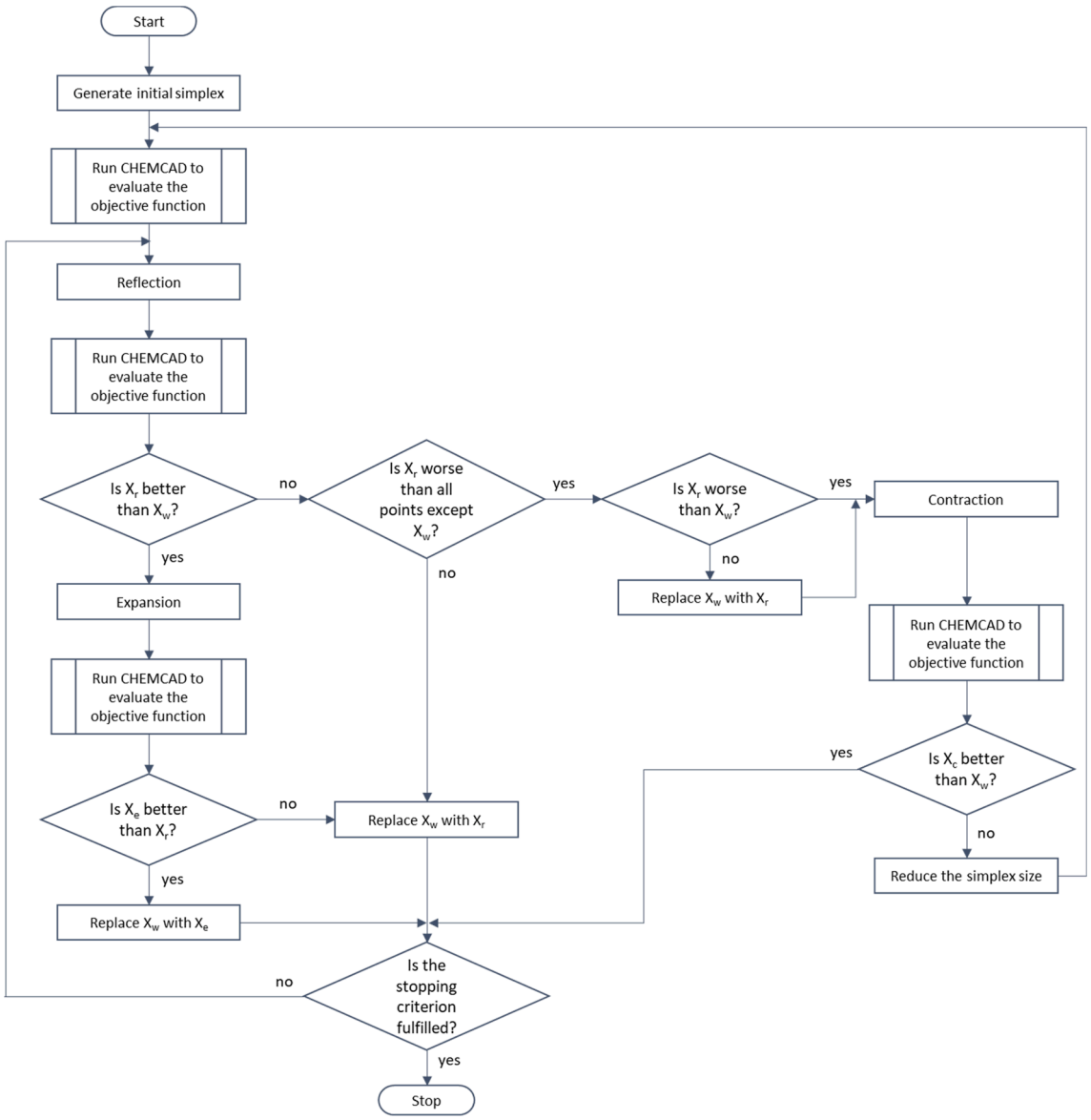



**Fig. 2.** Flowchart of the Nelder-Mead simplex algorithm.

$$p = \frac{a}{n\sqrt{2}}\left(\sqrt{n+1} + n - 1\right) \tag{2}$$

$$q = \frac{a}{n\sqrt{2}}\left(\sqrt{n+1} - 1\right) \tag{3}$$

In every iteration, the objective function is evaluated at each point of the simplex, and a reflection operation is performed. Depending on the objective function values of the simplex and the new point obtained by reflection, either an expansion or contraction operation is performed, or the reflected simplex is kept for the next iteration. If a contraction fails to produce an improved value of the objective function, the size of the simplex is reduced. A flowchart of the algorithm is shown in Fig. 2.

In the reflection operation, the point with the worst value of objective function ($\mathbf{X}_w$) is reflected in the opposite face of the simplex to obtain point $\mathbf{X}_r$ [37]:

$$\boldsymbol{X}_r = (1+\alpha)\boldsymbol{X}_0 - \alpha\boldsymbol{X}_w \tag{4}$$

where $\mathbf{X}_0$ is the centroid of all the points except $\mathbf{X}_w$ and α is a positive reflection coefficient. The higher α, the farther $\mathbf{X}_r$ is from $\mathbf{X}_0$. If the value of the objective function is better in $\mathbf{X}_r$ than in $\mathbf{X}_w$, the former one replaces the latter in the simplex. The reflection operation drives the simplex towards better values of the objective function.

If $\mathbf{X}_r$ is the new optimum, it can be expected that the objective function will further decrease in this direction, and the expansion operation is performed. A new point ($\mathbf{X}_e$) is generated on the line $\mathbf{X}_0 - \mathbf{X}_r$, but even further away from $\mathbf{X}_0$ [37]:

$$\boldsymbol{X}_e = \gamma\boldsymbol{X}_r + (1-\gamma)\boldsymbol{X}_0 \tag{5}$$

where $\gamma > 1$ is the expansion coefficient. $\mathbf{X}_w$ is replaced by the point having the best value of objective function between $\mathbf{X}_e$ and $\mathbf{X}_r$.

The contraction operation shrinks the simplex. It is performed if $\mathbf{X}_r$ has the worst value of the objective function among all the points of the simplex, or if only $\mathbf{X}_w$ is worse than $\mathbf{X}_r$. In the latter case, $\mathbf{X}_w$ is replaced by $\mathbf{X}_r$ before the contraction. A new point $\mathbf{X}_c$ is generated [37]:

$$\boldsymbol{X}_c = \beta\boldsymbol{X}_h + (1-\beta)\boldsymbol{X}_0 \tag{6}$$

where $0 \le \beta \le 1$ is the contraction coefficient. If $\mathbf{X}_c$ is better than $\mathbf{X}_w$, it replaces it. Otherwise, the optimisation cannot continue with the above operations, and the size of the simplex is reduced by replacing all the points ($\mathbf{X}_i$, $i = 1 \ldots n$) except the current optimum ($\mathbf{X}_b$) with the midpoints

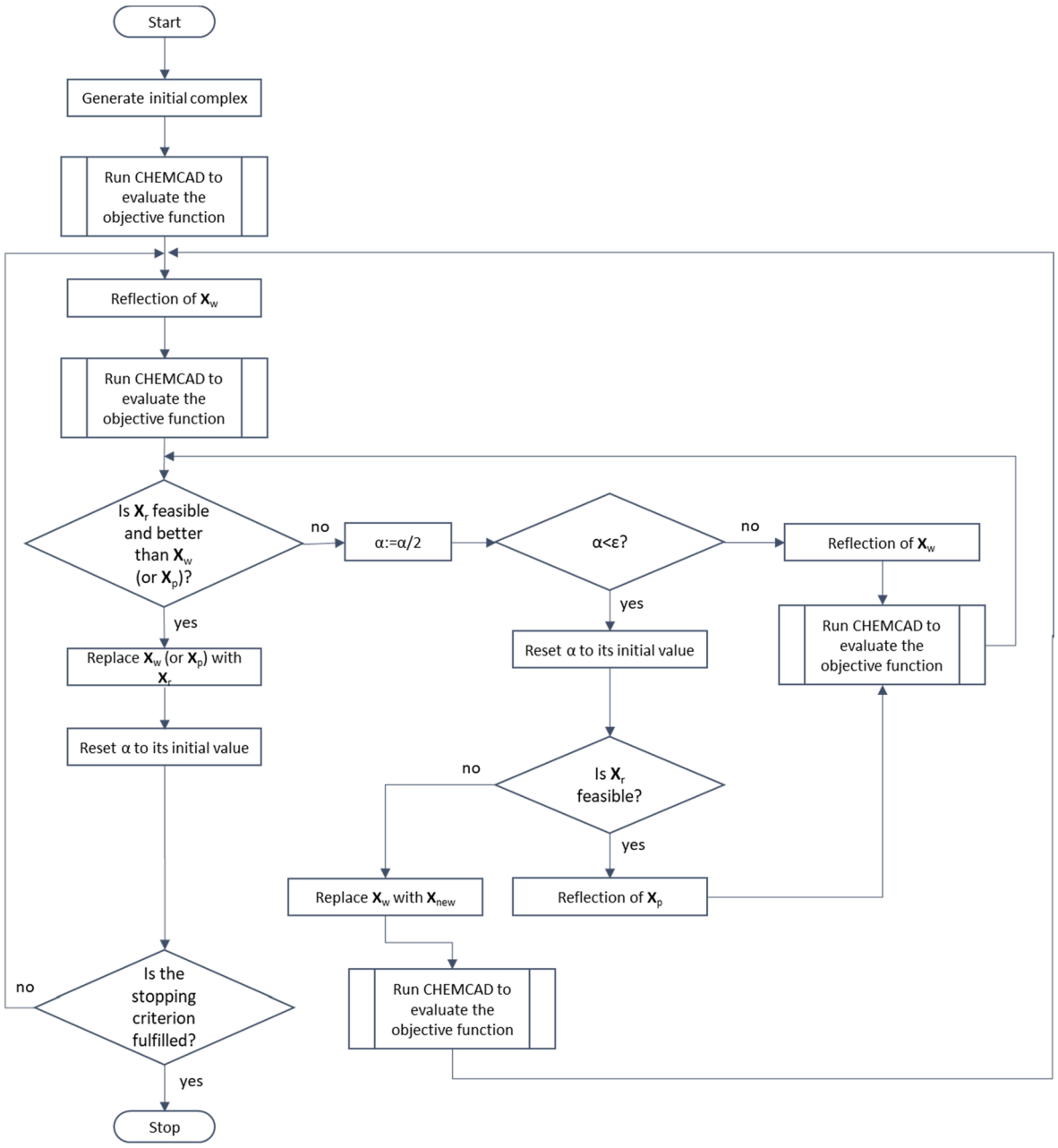



**Fig. 3.** Flowchart of the complex algorithm.

of the line segments joining them to $\mathbf{X}_b$: $(\mathbf{X}_b + \mathbf{X}_i)/2$.

The optimisation can be stopped either when the standard deviation of the function values in the points of the simplex becomes lower than a small quantity ε, or when the simplex becomes smaller than a minimum value $a_{min}$ after a size reduction step.

The simplex method has the following parameters that can be modified: the coefficients of each operation (α, β, γ) and the size (a) and location ($\mathbf{X}_{start}$) of the starting point.

### 2.3. Box-complex method

The third method applied in this work is a modified version of the Box-complex algorithm. Box [40] proposed this method as an extension of the Spendley simplex algorithm to optimisation problems with inequality constraints. The simplex is replaced with a set of $k > n + 1$ points called the complex. Box [40] recommends using $k = 2n$. A flowchart of the algorithm is shown in Fig. 3.

The points of the initial complex are generated randomly within the bounds of the optimisation variables. The only operation performed by the complex algorithm is the reflection as described by Eq. (4). The reflection coefficient α must be higher than 1.0. If $\mathbf{X}_r$ is feasible, that is, it does not violate any constraints, and if it is also better than $\mathbf{X}_w$ (the point with the worst value of objective function in the complex), then $\mathbf{X}_w$ is replaced by $\mathbf{X}_r$. On the other hand, if $\mathbf{X}_r$ is either infeasible or worse than $\mathbf{X}_w$, the value of α is decreased to its half, and a new $\mathbf{X}_r$ is generated. This is repeated until either an improved point is found or α becomes lower than a small quantity ε. In the latter case, the version of the algorithm described by Rao [37] performs the reflection operation with the point having the second-worst objective function value instead of $\mathbf{X}_w$. However, in the modified algorithm proposed here, different operations are performed depending on the feasibility of $\mathbf{X}_r$. If it is feasible, the reflection operation is performed with a point $\mathbf{X}_p$ randomly selected from the complex (excluding the best one). If $\mathbf{X}_r$ is infeasible, instead of reflection, $\mathbf{X}_w$ is replaced with a new point $\mathbf{X}_{new}$ generated in the vicinity

of the best point ($\mathbf{X}_b$, the current optimum). Each i coordinate of $\mathbf{X}_{new}$ is a random variable with uniform distribution in the interval ($X_{bi}$-$\sigma_i$, $X_{bi} + \sigma_i$) where $\sigma_i$ is the standard deviation of coordinate i of the points of the complex.

Similarly to the simplex algorithm, the optimisation is stopped either if the standard deviation of the function values of the complex becomes lower than a small quantity ε, or when the size of the complex becomes very small [37]. The latter condition is checked by comparing the maximum of the distances between any two points of the complex (d) to a prescribed stopping criterion ($d_{min}$).

The parameters of the method are the reflection coefficient (α), the stopping criterion ($d_{min}$) and additionally, the bounds of the optimisation variables between which the starting points are generated.

The method has the following advantages [40,37] compared to the simplex methods of Spendely or Nelder-Mead. Due to $\alpha > 1$, the complex continuously expands when it is far from the optimum, leading to rapid progress. When a constraint is encountered, the complex contracts, flattens itself and follows the constraint. While a simplex with $k = n + 1$ points tends to completely collapse into a lower dimensional subspace, which it will not be able to leave, the complex avoids this by having $\alpha > 1$ and a higher number of points ($k > n + 1$). Thus, the complex is able to leave a constraint or turn at the corners of two constraints to improve the objective function.

## 3. Case study

The optimisation algorithms are applied for a case study of recovering methanol from a five-component pharmaceutical waste solvent mixture by batch distillation taken from Hegely and Lang [10]. Hegely and Lang [10] studied a six-batch campaign where off-cuts were partially recycled to the next batch. The batches were optimised consecutively by using the genetic algorithm applied here, as well. In the present work, the optimisation of the treatment of the first batch is performed, where the charge consists only of fresh feed.

The fresh feed (the charge) contains 0.07 mass % acetone (A), 37.14 % methanol (B), 4.89 % tetrahydrofuran (C), 56.34 % water (D) and 1.56 % toluene (E). Methanol must be recovered with a purity of 99.5 mass%. The components form five binary, minimum-boiling azeotropes. The order of pure components and azeotropes by increasing boiling point: A-B, A, B-C, B-E, C-D, B, C, D-E, D and E. Since the concentration of A is very low, the recovery of B is disturbed by the azeotropes B-C and B-E. (The existence of the C-D azeotrope does not affect the recovery B since the total amount C is removed by the B-C azeotrope.) Therefore, pollutants C and E must be removed in fore-cut (s). For the description of the vapour-liquid equilibria, the UNIQUAC model was used with the binary interaction parameters taken from Hegely and Lang [10].

The batch distillation process consists of the following five steps:

- Step 0: heating-up of the total column under total reflux to approach steady-state. At the end of the step, the composition of the condensate is close to that of the B-C azeotrope. This step lasts 360 min.
- Step 1: taking the first fore-cut containing a high amount of B, C and E. This cut is incinerated. The step is finished when the concentration of C in the distillate ($x_{d,C}$) decreases below a given stopping criterion: $x_{d,C} < Cr_1$.
- Step 2: taking the second fore-cut, which contains more B, but whose pollutant content is still too high. This cut is recycled to the next batch (to decrease the loss of B). The step is finished when the concentration of C in the distillate ($x_{d,C}$) decreases below a given stopping criterion: $x_{d,C} < Cr_2$.
- Step 3: taking the main-cut, which is the methanol product with high purity. At the end of the step, the D content of the distillate starts to increase. The step is finished when the concentration of B in the product decreases below 99.52 %: $x_{mc,B} < 99.52$ %.
- Step 4: taking the after-cut, which is a mixture of B and D to remove B from the still residue so that it can be sent to biological wastewater treatment. This cut is also recycled to the next batch. The step is finished when the B content in the still residue decreases below 0.25 %: $x_{sr,B} < 0.25$ %.

**Table 1**
Ranges of the optimisation variables in GA optimisation.

| | $R_1$ | $R_2$ | $R_3$ | $Cr_1$, mass% | $Cr_2$, mass% |
|---|---|---|---|---|---|
| Upper bound, $U_i$ | 15 | 15 | 15 | 40.00 | 10.00 |
| Lower bound, $L_i$ | 0.6 | 0.6 | 0.6 | 0.50 | 0.50 |

The distillation column has 27 theoretical plates (including the total condenser and the reboiler). The top pressure is atmospheric, and the column has a pressure drop of 0.25 bar. The volume of the charge is 25 $m^3$ (at 20 °C). The liquid hold-up of the condenser and the columns is 0.45 $m^3$ and 0.05 $m^3$/plate, respectively. The constant heat duty of the reboiler ($Q_{st}$) is 1800 MJ/h, which is provided with saturated heating steam of 3 bar (heat of condensation: $r_{st} = 2263.5$ MJ/t).

The optimisation problem to be solved is the following one [10]:

$$\max_{X}\{OF(\boldsymbol{X})\} \tag{7}$$

subject to the inequality constraints.

$$x_{mc,B}(\boldsymbol{X}) \geq 99.5\text{mass}\% \tag{8}$$

$$\frac{x_{fc2,C}(\boldsymbol{X})}{x_{fc2,B}(\boldsymbol{X})} \leq 0.107 \tag{9}$$

$$\frac{x_{fc2,E}(\boldsymbol{X})}{x_{fc2,B}(\boldsymbol{X})} \leq 0.120 \tag{10}$$

where OF is the objective function, $\mathbf{X}$ the vector of the optimisation variables, and $x_{fc2,i}$ is the concentration of component i in the second fore-cut. Constraints 9 and 10 are applied to limit the accumulation of C and E in the next batches because of the recycling of the second fore-cut.

OF is the profit of processing one batch. It consists of the income, which is the value of regenerated methanol (that can also be considered as a purchase cost avoided), the incineration cost of the first fore-cut, and the cost of heating steam, which is proportional to the duration of the process (t) [10]:

$$OF = p_B m_{mc} - c_{inc} m_{fc1} - c_{st}\frac{Q_{st}}{r_{st}}t \tag{11}$$

where $m_{mc}$ is the mass of the main cut (product), $m_{fc1}$ is the mass of the first fore-cut, $p_B$ is the price of methanol (0.46 US$/kg), $c_{inc}$ is the cost of incineration (0.21 $/kg), and $c_{st}$ is that of the heating steam (57.6 $/t). The cost of cooling water and the biological treatment of the still residue are neglected.

The optimisation variables are the reflux ratios of the steps: $R_1$ (first fore-cut), $R_2$ (second fore-cut) $R_3$ (main cut), and the termination criteria of Steps 1 and 2 ($Cr_1$ and $Cr_2$). $R_4$ (the reflux ratio of the after-cut) was optimisation variable in Hegely and Lang [10], but since its effect on OF is negligible, it is fixed as 5.41 here.

For more details on the phase equilibria and the separation process, the reader is referred to Hegely and Lang [10].

## 4. Results

In this section, the results of the different algorithms are presented and compared. The performance of the methods is discussed, and detailed results are given for the new optimum.

**Table 2**
Results of the GA optimisation runs.

| ID | Parameters of the method | | | Number of simulations | $R_1$ | $R_2$ | $R_3$ | $Cr_1$, mass% | $Cr_2$, mass% | OF, $ |
|---|---|---|---|---|---|---|---|---|---|---|
| | Population size | Mutation rate, % | Crossover rate, % | | | | | | | |
| JCP | 30 | 5 | 70 | 3000 | 6.22 | 3.07 | 3.05 | 17.50 | 2.62 | 467 |
| GA-1 | 15 | 5 | 70 | 1500 | 5.77 | 3.57 | 3.26 | 18.21 | 2.97 | 479.3 |
| **GA-2** | **30** | **5** | **70** | **3000** | **5.68** | **3.55** | **3.31** | **19.24** | **2.79** | **479.8** |
| GA-3 | 45 | 5 | 70 | 4500 | 5.57 | 3.52 | 3.35 | 18.40 | 3.07 | 477.9 |
| GA-4 | 15 | 3 | 70 | 1500 | 5.27 | 4.54 | 3.17 | 17.11 | 3.32 | 461.0 |
| GA-5 | 15 | 7 | 70 | 1500 | 6.21 | 5.05 | 3.28 | 15.73 | 3.12 | 440.7 |
| GA-6 | 15 | 5 | 50 | 1500 | 5.07 | 7.27 | 3.27 | 14.95 | 3.78 | 402.0 |
| GA-7 | 15 | 5 | 90 | 1500 | 5.80 | 6.84 | 3.07 | 16.16 | 3.25 | 423.9 |

**Table 3**
p and q values for the generation of initial simplexes.

| | $R_1$ | $R_2$ | $R_3$ | $Cr_1$, mass% | $Cr_2$, mass% |
|---|---|---|---|---|---|
| p | 0.65671 | 0.65671 | 0.65671 | 1.8219 | 0.45377 |
| q | 0.14759 | 0.14759 | 0.14759 | 0.40947 | 0.10198 |

### 4.1. Optimisation results with genetic algorithm

Table 1 shows the ranges of the optimisation variables used with GA. New individuals were generated with random values of each variable within the given bounds. The ranges were chosen to be wide enough so that the global optimum is surely included in the domain defined by them. Additionally, $Cr_2$ should always be less than or equal to $Cr_1$, since $x_{d,C}$ decreases in time in a monotonous way.

The parameters of GA: the population size, the mutation rate and the crossover rate were varied to study their effect on the optimisation results (Table 2). The first row (run JCP) contains the results of Hegely and Lang [10]. The number of generations was always 100; thus the number of simulations performed was 100 times the population size. In all the optimisation runs, the average increase of OF per generation became consistently low before reaching 100 generations. Since GA is a stochastic method, repeating the optimisation with the same GA parameters gives different results. This can be seen by comparing runs JCP and GA-2, the latter which gave a 2.7 % (13 $) higher OF value with different values of the optimisation variables, as well. The stochastic nature of the method means that it is not possible to clearly distinguish the effect of changing the parameters of the method from the inherent variation of the results. Nevertheless, changes in the GA parameters that are clearly favourable or unfavourable can be detected.

The population size (in the range studied) does not significantly influence the value of OF. Therefore, a population of 15 individuals is sufficient, and further runs were performed with this value. Both the mutation and crossover rates were optimal at the values used in runs JCP (5 and 70 %, respectively).

The results indicate that the GA parameters used in Hegely and Lang [10] were appropriate, but it is still possible to reduce the population size from 30 to 15 without adversely affecting the value of OF. Although a higher OF value was reached in this work, this is consistent with the stochastic nature of the method.

### 4.2. Optimisation results with the Nelder-Mead simplex algorithm

Since this case study has five optimisation variables, the simplexes of the Nelder-Mead simplex algorithm consist of six points in the five-dimensional space of optimisation variables. The initial simplex was generated with the methods described in Section 2.2 with the following modifications. In order to take into account the very different ranges for the possible values of the optimisation variables, the simplex was scaled. This was done by using different values of the p and q coefficients of the unit vectors for each variable. The size of the simplex (a) was given as a percentage of the ranges of the variables shown in Table 1. For any optimisation variable i, the value $a_i$ used in Eqs. (2) and (3) to obtain the p and q values is calculated by the following equation:

$$a_i = (U_i - L_i)a + L_i \tag{12}$$

The value of a used here was 5 %. The scaling of the simplex was especially necessary since $Cr_1$ and $Cr_2$ were considered in the optimisation as mass fractions instead of mass percents. The p and q values obtained are given in Table 3. The optimisation was stopped if the standard deviation of OF in the points of the simplex became lower than 0.01 $ or if the size of the simplex became lower than $a_{min} = 0.05$ %.

Several optimisation runs were performed with the simplex algorithm to study the influence of the parameters of the method and that of the starting point (Table 4). Run S-1 was performed with the values recommended by Nelder and Mead [38], then in runs S-2 to S-7 each parameter was either decreased or increased, one parameter at a time. With the recommended simplex parameters, an OF value slightly inferior to that of GA was obtained, although with only 209 simulations. Changing the value of the reflection coefficient (α) considerably reduced OF. Similar behaviour was observed by changing the contraction coefficient (γ), although the changes in OF were smaller. On the other hand, increasing the expansion coefficient (β) to 0.75 increased OF, even though the number of simulations required was also higher. Further

**Table 4**
Results of the simplex optimisation runs.

| ID | Parameters of the method | | | Number of simulations | $R_1$ | $R_2$ | $R_3$ | $Cr_1$, mass% | $Cr_2$, mass% | OF, $ |
|---|---|---|---|---|---|---|---|---|---|---|
| | α | β | γ | | | | | | | |
| S-1 | 1 | 0.5 | 2 | 209 | 4.86 | 3.08 | 3.10 | 18.75 | 3.01 | 474.9 |
| S-2 | 0.5 | 0.5 | 2 | 142 | 6.71 | 4.26 | 3.38 | 11.34 | 2.50 | 377.5 |
| S-3 | 1.5 | 0.5 | 2 | 135 | 7.08 | 2.03 | 3.01 | 12.89 | 2.49 | 445.3 |
| S-4 | 1 | 0.25 | 2 | 157 | 4.96 | 3.72 | 3.16 | 17.28 | 3.13 | 463.3 |
| **S-5** | **1** | **0.75** | **2** | **236** | **5.55** | **2.69** | **3.17** | **20.12** | **2.75** | **493.3** |
| S-6 | 1 | 0.5 | 1.5 | 232 | 6.17 | 3.82 | 2.92 | 17.98 | 2.36 | 451.5 |
| S-7 | 1 | 0.5 | 2.5 | 158 | 5.21 | 3.53 | 3.28 | 17.51 | 3.24 | 466.9 |
| S-8 | 1 | 0.875 | 2 | 195 | 6.03 | 2.96 | 3.07 | 17.38 | 2.79 | 479.1 |
| S-9 | 1 | 0.875 | 2 | 193 | 5.97 | 2.39 | 3.03 | 16.85 | 2.61 | 472.5 |
| S-10 | 1 | 0.75 | 2 | 263 | 5.47 | 3.14 | 3.04 | 18.39 | 2.94 | 483.2 |
| S-11 | 1 | 0.75 | 2 | 151 | 5.57 | 3.67 | 2.98 | 18.71 | 2.97 | 485.2 |

**Table 5**
Starting points of the simplex optimisation runs with the corresponding OF value.

| Runs | $R_1$ | $R_2$ | $R_3$ | $Cr_1$, mass% | $Cr_2$, mass% | OF, $ |
|---|---|---|---|---|---|---|
| S-1-8 | 7.00 | 4.00 | 4.00 | 10.00 | 2.00 | 286.5 |
| S-9-10 | 8.00 | 4.50 | 4.50 | 7.50 | 1.50 | 148.7 |
| S-11 | 6.00 | 4.00 | 3.50 | 15.00 | 2.50 | 406.5 |

increasing β to 0.875 (run S-8) was already not advantageous.

In all of the above runs, the initial simplex was generated from the same starting point (Table 5, first row). The value of OF in this point is low compared to the results of the optimisations meaning that it is not necessary to have a starting point with an already high OF value. To study the influence of the starting point, two other points were also used (Table 5). Runs S-9 and S-10 were started from a point with an ever lower OF value than before. Using $\beta = 0.75$ (S-10) was better than 0.875 (S-9) in this case, as well. The OF value obtained with $\beta = 0.75$ is slightly lower than the best one obtained with the previous starting point (run S-5), but still above the value obtained by the recommended parameters. However, the number of simulations was higher. Starting the optimisation from a point with higher OF (run S-11) gave very similar values for OF and the optimisation variables as run S-10 with a smaller number of simulations.

In conclusion, it was important to select appropriate parameter values for the simplex algorithm. The values recommended by Nelder and Mead [38] ($\alpha = 1$, $\beta = 0.5$, $\gamma = 1$) are acceptable, but the best results were obtained by changing β to 0.75. The location of the starting point does influence the value of OF but only slightly. Since the values of the variables do not vary strongly with starting location, it can be assumed that the variation in the results is due to the premature termination of the method rather than the finding of different local optima. The most important difference is in $Cr_1$, which is much higher in run S-5.

### 4.3. Optimisation results with the Box-complex algorithm

The influence of the parameters and starting points of the complex

**Table 7**
Starting ranges of the optimisation variables for the complex optimisation runs.

| Range | Runs | | $R_1$ | $R_2$ | $R_3$ | $Cr_1$, mass % | $Cr_2$, mass% |
|---|---|---|---|---|---|---|---|
| Range A | C-1-5 | Upper bound | 7.00 | 6.00 | 3.50 | 20.00 | 3.00 |
| | | Lower bound | 5.00 | 4.00 | 2.50 | 10.00 | 0.05 |
| Range B | C-6-25 | Upper bound | 8.00 | 6.00 | 4.50 | 25.00 | 3.50 |
| | | Lower bound | 4.00 | 2.00 | 2.50 | 15.00 | 1.50 |

**Table 6**
Results of the complex optimisation runs.

| ID | Parameters of the method | | | Starting complex | Number of simulations | $R_1$ | $R_2$ | $R_3$ | $Cr_1$, mass% | $Cr_2$, mass% | OF, $ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | α | $d_{min}$ | n | | | | | | | | |
| C-1 | 1 | 0.1 | 8 | Randomly generated in Range A | 152 | 5.71 | 4.97 | 3.18 | 17.06 | 3.17 | 459.3 |
| C-2 | | | | | 161 | 5.79 | 4.39 | 3.00 | 17.52 | 2.92 | 468.4 |
| C-3 | | | | | 187 | 5.70 | 5.15 | 2.95 | 16.94 | 3.08 | 455.5 |
| C-4 | | | | | 303 | 5.81 | 2.65 | 3.06 | 19.54 | 2.66 | 489.2 |
| C-5 | | | | | 309 | 5.56 | 4.67 | 3.34 | 17.41 | 3.10 | 460.2 |
| Avg. | | | | | 222 | | | | | | 466.5 |
| SD | | | | | 77 | | | | | | 13.5 |
| C-6 | 1 | 0.1 | 8 | Randomly generated in Range B | 708 | 5.92 | 2.05 | 3.14 | 20.30 | 2.48 | 488.2 |
| C-7 | | | | | 789 | 5.68 | 3.03 | 3.03 | 18.92 | 2.83 | 488.5 |
| C-8 | | | | | 301 | 5.92 | 3.29 | 3.27 | 17.81 | 2.79 | 477.7 |
| C-9 | | | | | 379 | 6.22 | 1.91 | 3.08 | 17.84 | 2.40 | 479.7 |
| C-10 | | | | | 1236 | 5.84 | 2.48 | 3.03 | 18.41 | 2.68 | 483.5 |
| Avg. | | | | | 683 | | | | | | 483.5 |
| SD | | | | | 373 | | | | | | 4.9 |
| C-11 | 1.3 | 0.2 | 8 | Randomly generated in Range B | 274 | 6.00 | 3.15 | 3.25 | 18.19 | 2.66 | 475.8 |
| C-12 | | | | | 134 | 5.84 | 2.49 | 3.02 | 19.08 | 2.62 | 486.5 |
| C-13 | | | | | 393 | 6.13 | 2.13 | 3.17 | 17.05 | 2.54 | 474.5 |
| C-14 | | | | | 166 | 5.50 | 3.29 | 3.07 | 19.00 | 2.93 | 487.4 |
| C-15 | | | | | 166 | 5.59 | 2.50 | 3.17 | 20.37 | 2.72 | 491.7 |
| Avg. | | | | | 227 | | | | | | 483.2 |
| SD | | | | | 107 | | | | | | 7.6 |
| C-16 | 1.6 | 0.2 | 8 | Randomly generated in Range B | 254 | 6.41 | 1.77 | 3.14 | 17.63 | 2.37 | 480.3 |
| C-17 | | | | | 294 | 5.51 | 2.41 | 3.35 | 21.23 | 2.58 | 485.2 |
| C-18 | | | | | 220 | 5.95 | 3.06 | 3.42 | 17.55 | 2.82 | 472.1 |
| C-19 | | | | | 330 | 5.11 | 3.47 | 3.06 | 18.41 | 3.06 | 477.6 |
| C-20 | | | | | 141 | 7.75 | 3.26 | 3.06 | 19.32 | 2.71 | 482.4 |
| Avg. | | | | | 248 | | | | | | 479.5 |
| SD | | | | | 73 | | | | | | 5.0 |
| C-21 | 1.3 | 0.2 | 10 | Randomly generated in Range B | 639 | 5.57 | 3.03 | 2.93 | 19.14 | 2.90 | 488.9 |
| C-22 | | | | | 256 | 5.75 | 2.60 | 3.04 | 19.42 | 2.69 | 489.6 |
| **C-23** | | | | | **293** | **5.69** | **2.62** | **3.03** | **19.80** | **2.73** | **492.1** |
| C-24 | | | | | 389 | 5.49 | 3.51 | 3.24 | 18.42 | 2.94 | 478.8 |
| C-25 | | | | | 210 | 5.68 | 2.47 | 3.26 | 19.74 | 2.68 | 487.8 |
| Avg. | | | | | 357 | | | | | | 487.4 |
| SD | | | | | 171 | | | | | | 5.1 |
| C-26 | 1.3 | 0.2 | 10 | Predetermined (Table 8) | 245 | 6.33 | 1.88 | 3.19 | 17.55 | 2.40 | 478.3 |
| C-27 | | | | | 388 | 5.63 | 2.78 | 3.06 | 19.70 | 2.76 | 491.2 |
| C-28 | | | | | 407 | 5.31 | 3.63 | 3.04 | 18.67 | 2.89 | 478.7 |
| C-29 | | | | | 168 | 6.95 | 3.76 | 3.06 | 14.68 | 2.59 | 443.9 |
| C-30 | | | | | 358 | 5.98 | 2.59 | 3.10 | 18.47 | 2.64 | 485.5 |
| Avg. | | | | | 313 | | | | | | 475.5 |
| SD | | | | | 103 | | | | | | 18.4 |

algorithm on the optimisation results was studied. Unlike the simplex method, this is a stochastic method. Therefore, at each setting, the optimisation is repeated five times, and the average (Avg.) and standard deviation (SD) of OF and the number of simulations are calculated (Table 6).

In runs C-1 to C-5, the reflection coefficient was $\alpha = 1.0$, which is the minimum possible value, the minimal size of the complex (determining termination) is $d_{min} = 0.1$, while the number of points in the complex is $n = 8$, that is higher than 6, the minimum number of points. The starting complex is randomly generated in Range A (Table 7). These range of the optimisation variables are considerably narrower than the ones used for GA, but the values found by GA and the simplex method are generally within them, with the exception of $R_2$ and, in the case of run S-5, $Cr_1$. With these parameters of the method, the individual runs gave satisfactory OF values with numbers of simulations close to those of the simplex method. However, the OF values have a wide distribution between 455.2 $ to 489.2 $ (which is the second-highest value found so far). This shows that although it is possible to obtain high OF values, this can only be done reliably by performing multiple runs. The final values of the variables are not limited by the starting ranges; for example, $R_2$ in run C-4 is below its lower bound.

In order to study the influence of the starting ranges, they were modified based on the previous optimisation results (Range B, Table 7). The ranges of the reflux ratios were widened, that of $Cr_1$ was shifted higher, and that of $Cr_2$ was shifted and narrowed. Widening the ranges can make the optimisation slower since it is likely that during the generation of the initial complex a higher number of infeasible points will be found. On the other hand, increasing the size of the initial complex can help approaching the true optimum. The results (runs C-6 to C-10) show that the average number of simulations became three times higher, although it varied considerably among the individual runs (from 301 to 1236). However, the average OF value increased significantly, and the standard deviation fell almost to its third. This means that with these settings, the complex algorithm reliably gave OF values comparable or better than the best one obtained by GA. Considering that the number of simulations was undesirably high by using the favourable starting ranges, the earlier termination of the algorithm was considered by increasing $d_{min}$ to 0.2. For runs C-1 to C-5, $d_{min}$ reached 0.2 after 128 simulations on average. At this point, the maximal OF value was already found in one of the cases (run C-4), and the average OF was only by 0.15 % (0.68 $) lower than at $d_{min} = 0.1$. For runs C-6 to C-10, the average number of simulations was less than half (302) than with $d_{min} = 0.1$. The maximum and average OF values were only by 0.25 % (1.21 $) and 0.23 % (1.09 $) lower, respectively. Therefore, continuing the optimisation until $d_{min} = 0.1$ after reaching $d_{min} = 0.2$ does not significantly increase the value OF, but nearly doubles simulation time.

Runs C-11 to C-15 were thus performed with $d_{min} = 0.2$. Additionally, $\alpha$ was increased to 1.3. With this modification, the number of simulations decreased, while the average value of OF practically remained the same. One of the runs (C-15) gave the highest OF value obtained by the complex algorithm this far. By further increasing the value of $\alpha$ (runs C-16 to C-10), the average value of OF decreased slightly. Therefore, the value of $\alpha = 1.3$ is satisfactory.

In runs C-21 to C-25, the number of points of the complex was increased to $n = 10$, which is the value recommended by Box [40] since it is twice the number of optimisation variables. The value of $\alpha$ was 1.3. This modification increased considerably the number of simulations required, from 227 to 357. On the other hand, the highest average OF value was obtained this far. The standard deviation was also low, meaning that the values of OF were reliably high (only one of them was lower than the best one obtained by GA). Run C-23 gave an OF value close to the best one obtained by the simplex method.

All previous calculations were started from randomly generated complexes. In runs C-26 to C-30, a predetermined starting complex was given (Table 8). The first point of the complex was the starting point used in run S-11 (Table 5). All the other points were generated from this one by changing the values of two operational variables simultaneously. The reflux ratios were increased, the stopping criteria were decreased to ensure that all points are feasible. From the ten possible combinations of changing two variables simultaneously, the one where the two stopping criteria are changed was not used. Using this starting complex made the optimisation faster, but the average value of OF decreased. However, this decrease is mainly because run C-29 has a low OF value; the other four runs gave good results. These calculations show that the variation of optimisation results does not only arise from the randomly generated starting complex but also from the random replacement of points during optimisation.

**Table 8**
Points of the starting complex for runs C-26-30.

| $R_1$ | $R_2$ | $R_3$ | $Cr_1$, mass% | $Cr_2$, mass% | OF, $ |
|---|---|---|---|---|---|
| 6.00 | 4.00 | 3.50 | 15.00 | 2.50 | 406.5 |
| 6.50 | 4.50 | 3.50 | 15.00 | 2.50 | 409.2 |
| 6.50 | 4.00 | 4.00 | 15.00 | 2.50 | 382.6 |
| 6.50 | 4.00 | 3.50 | 13.50 | 2.50 | 402.3 |
| 6.50 | 4.00 | 3.50 | 15.00 | 2.25 | 396.8 |
| 6.00 | 4.50 | 4.00 | 15.00 | 2.50 | 369.2 |
| 6.00 | 4.50 | 3.50 | 13.50 | 2.50 | 381.3 |
| 6.00 | 4.50 | 3.50 | 15.00 | 2.25 | 377.4 |
| 6.00 | 4.00 | 4.00 | 13.50 | 2.50 | 362.9 |
| 6.00 | 4.00 | 4.00 | 15.00 | 2.25 | 357.5 |

To conclude, the best results were achieved by using the parameter values recommended by Box [40]: $\alpha = 1.3$, $k = 10$ and starting the optimisation with a complex randomly generated with moderately wide ranges for the optimisation variables. By these settings, it was possible to reliably obtain high OF values. Choosing an appropriate value for the termination criterion (here, the size of the complex) is important to avoid an excessive increase in optimisation time. The method is sensitive to the generation of the initial complex. Using wider ranges improves the value of OF, but increases the optimisation time.

### 4.4. Distribution of the values of the optimisation variables

Fig. 4 shows the distribution of the values of the optimisation variables for the three optimisation methods in the form of box plots. The red line shows the median value, the box the interquartile range (the range between the first and third quartiles (25th and 75th percentile, respectively)), while the whiskers (in black) show the upper and lower fences, which are the maximum and minimum values not considered outliers. A value was defined as an outlier (shown as red marks) if it was farther away from the 25th or 75th percentile than 1.5 times the interquartile range. The range between the whiskers corresponds approximately to $\pm$ 2.7 times the standard deviation of a normally distributed variable [57]. (The numerical values of the statistics shown in Fig. 4 are given in Tables S1-S3 in the Supporting information.) The horizontal black line shows the value found by the optimisation run with the highest OF value (run S-5), which will be referred to here as the optimal value. (Since the methods do not check optimality criteria, it cannot be claimed that this is the global optimum, but it is likely to be a good approximation of it.).

The distributions of $R_1$ are similar to each other, with that of the simplex method being wider. The values of $R_2$ obtained by GA are generally higher than those obtained by the other methods. The optimal $R_2$ is even outside of the range found by GA. The values of $R_3$ obtained by GA are also usually higher; however, the optimal value is in the higher part of the third quartile by the simplex and complex methods. In the case of $Cr_1$, GA gave generally lower values. The optimal value is the highest of the simplex method and above the range of GA. In contrast, usually higher $Cr_2$ values were found by GA.

The results show that while the distribution of the values of the optimisation variables was similar for the simplex and complex methods, the ranges obtained by GA were different for most of the

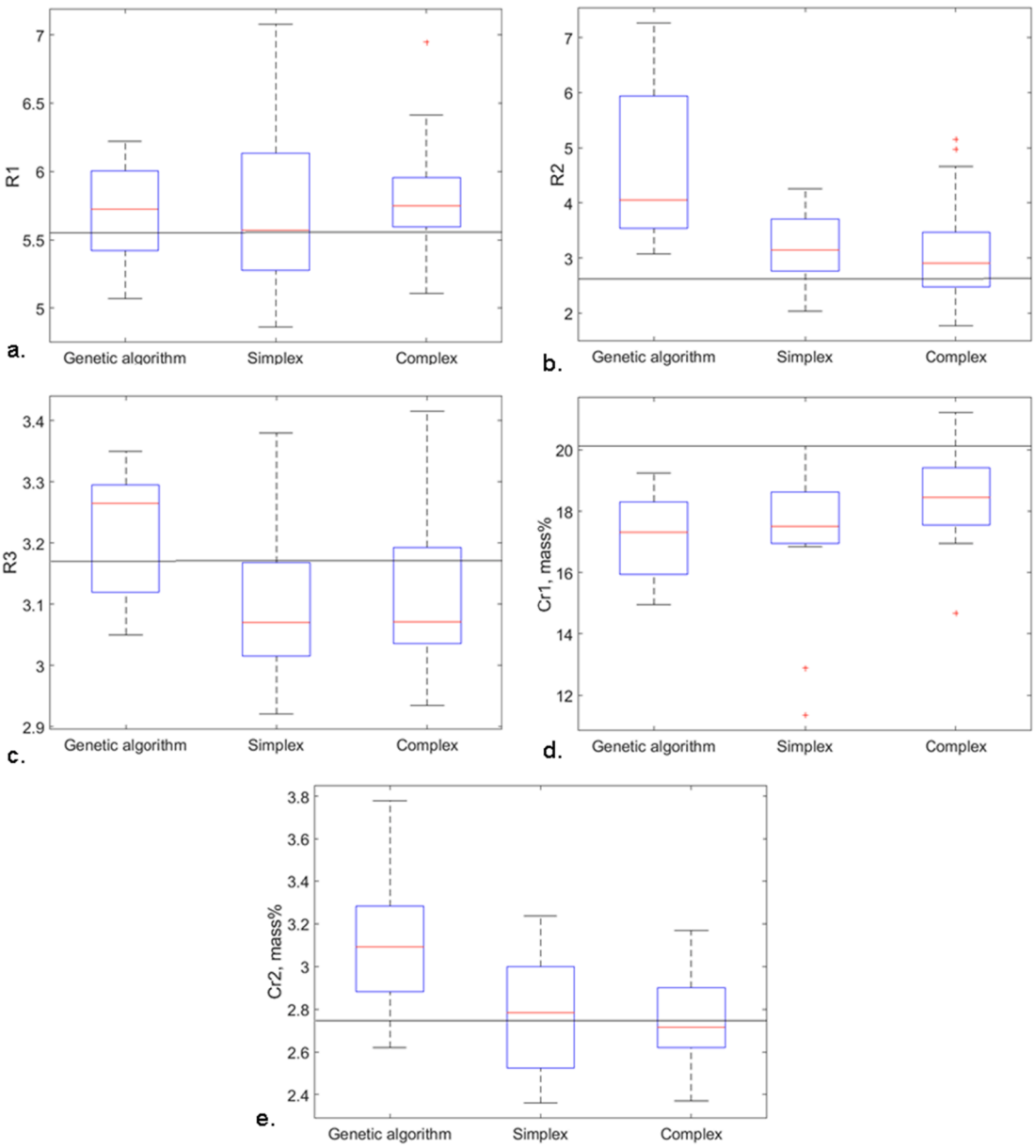



**Fig. 4.** The distribution of the values of the optimisation variables obtained by the different optimisation methods: a. $R_1$, b. $R_2$, c. $R_3$, d. $Cr_1$, e. $Cr_2$.

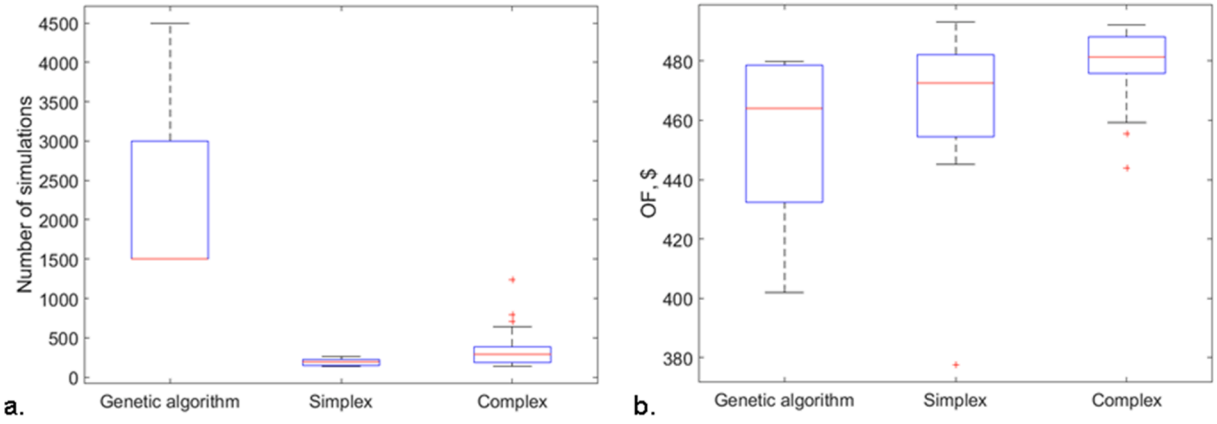



**Fig. 5.** The distribution of the number of simulations (a.) and the values of OF (b.) obtained by the different optimisation methods.

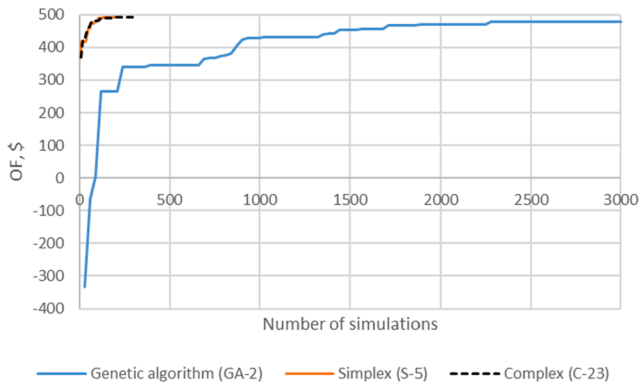



**Fig. 6.** The evolution of OF during optimisation for the runs with the highest final OF value.

variables (with the exception of $R_1$ and $Cr_1$). In some cases ($R_2$, $Cr_1$), the ranges did not include the optimal value.

### 4.5. Comparison of the optimisation methods

The number of simulations and the values of OF are also shown in box plots in Fig. 5. (The numerical values of the statistics shown in Fig. 5 are given in Tables S4-S6 in the Supporting information.) Both the simplex and complex methods required much fewer simulations than GA (Fig. 5a). However, the simplex method showed much lower variations, while the number of simulations of the complex method approached in outlying cases those of GA. (In the case of GA, the median value coincides with the minimum one).

The values of OF had a wider distribution by GA (Fig. 5b) than by the other two methods, which is unfavourable. The highest value was reached by the simplex method, which, however, gave the lowest value, as well. On the other hand, the complex method gave more consistently high OF values.

Fig. 6 shows the evolution of OF as a function of the number of simulations for the optimisation run of each method giving the highest final OF value.

GA did not only require a much higher number of simulations than the simplex and complex methods but, after a brief period of steep increase, OF increased only slowly. For the simplex and complex methods, OF started already at a relatively high value and increased rapidly to a value close to the final result.

Plotting the number of simulations as a function of OF (Fig. 7) reveals that there is no strong relationship between the two values. The number of simulations was always low by the simplex method. As it can be seen for the complex method, an increased number of simulations does not necessarily result in a better OF value.

The distance of all j points obtained by optimisation from the best one (run S-5) was calculated by:

$$\delta_j = \sqrt{\sum_{i=1}^{5} \left( \frac{X_{i,j} - X_{i,S-5}}{U_i - L_i} \right)^2} \tag{13}$$

where $X_{i,j}$ is the value of the optimisation variable i in point j. The upper and lower bounds $U_i$ and $L_i$ (with the values shown in Table 1) were used to scale the differences in the values of the optimisation variables. The

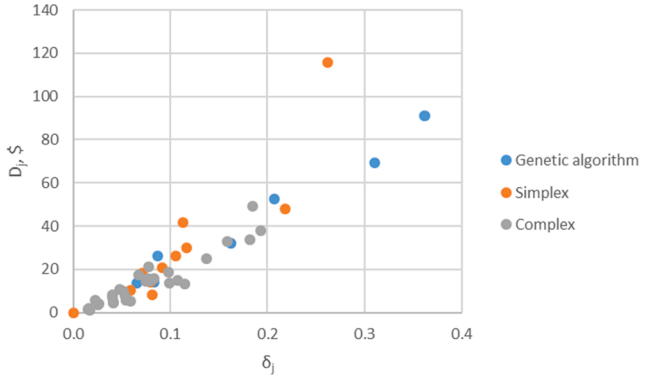



**Fig. 8.** The distance from the optimal OF value (D) as a function of the distance from the optimal point in the space of decision variables (δ).

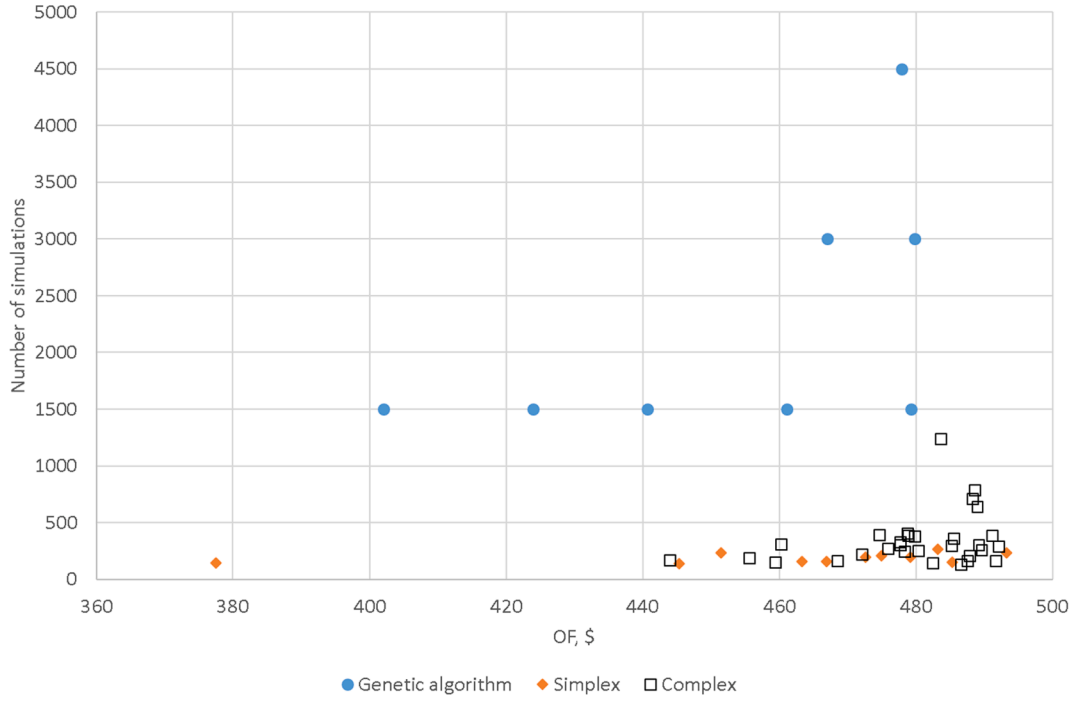



**Fig. 7.** The number of simulations and the value of OF obtained for the different optimisation methods.

**Table 9**
Detailed results of Hegely and Lang [10] (JCP) and the new optimum (S-5) for one batch.

| | JCP | S-5 | Difference, % |
|---|---|---|---|
| $R_1$ | 6.22 | 5.55 | −10.8 |
| $R_2$ | 3.07 | 2.69 | −12.4 |
| $R_3$ | 3.05 | 3.17 | 3.93 |
| $Cr_1$, mass% | 17.50 | 20.12 | 15.0 |
| $Cr_2$, mass% | 2.62 | 2.75 | 4.96 |
| $x_{fc2,C}/x_{fc2,B}$ | 0.0951 | 0.1058 | 11.3 |
| $x_{fc2,E}/x_{fc2,B}$ | 0.1191 | 0.1182 | −0.756 |
| Income ($) | 2597 | 2563 | −1.31 |
| Incineration cost ($) | 492 | 469 | −4.67 |
| Steam cost ($) | 1638 | 1602 | −2.20 |
| Profit ($) | 467 | 493 | 5.57 |

distance of the OF values obtained from the highest one was also calculated:

$$D_j = OF_{S-5} - OF_j = 493.3 - OF_j \quad (14)$$

As a general trend, $D_j$ decreases with decreasing $\delta_j$ (Fig. 8), meaning that the optimisation runs were not approaching another local optimum located at very different values of the optimisation variables but having a comparable (or better) OF value than the best one found here.

### 4.6. Detailed results of the new optimum

The new optimum found by run S-5 is compared to those of Hegely and Lang [10] (run JCP). The E/B mass ratio in the second fore-cut is close to the limit values of the constraint in both cases (Table 9). However, for run JCP, the C/B mass ratio is significantly (by 11 %) lower than the limit (Eq. (9)). It is expected that in the optimum both constraints are active for two reasons. First, it is preferable to remove C and E in the second fore-cut instead of the first one to decrease the incineration cost. Second, if the C/B or E/B ratio is not maximal, more B is lost in the second fore-cut, decreasing the mass of the main-cut and the income. Compared to run JCP, $R_1$ and $R_2$ are lower in run S-5 (by 11 and 12 %, respectively), while $R_3$ is slightly (by 3.9 %) higher (Tables 2 and 4). The highest difference is in $Cr_1$, which is considerably (by 15 %) higher in run S-5, meaning that the first fore-cut stops earlier. $Cr_2$ is also higher (by 5.0 %), corresponding to an earlier start of the main-cut.

The profit of run S-5 is 5.6 % higher due to the 4.7 % lower incineration and 2.2 % lower steam costs. The reduction of the latter is caused by the lower reflux ratios leading to a shorter (by 2.2 %) process duration. The mass of the main cut and thus the income decreased by 1.3 %; however, this was compensated by the lower costs.

Fig. 9 compares the evolution of the condensate (distillate after Step 0) composition of the two runs. By run S-5, Step 1 is shorter mainly due to the higher $Cr_1$ value. Because of this, the subsequent steps are shifted earlier. The concentration of B is higher in Steps 0–2 and in the first part of Step 3 but lower later on. However, due to the shifting, the average concentration of B in the second fore-cut is almost the same (81.59 % by S-5, 81.66 % by JCP).

## 5. Conclusions

In order to reduce the computational intensity of the optimisation of batch distillation processes, two direct search methods, the Nelder-Mead simplex and a variant of the Box-complex method were studied, and the results were compared to those of a genetic algorithm (GA). The optimisation algorithms were coded in Visual Basic for Applications under Microsoft Excel and coupled to the CHEMCAD professional flow-sheet simulator. The calculations were performed for a case study taken from Hegely and Lang [10], where the profit of the batch distillation treatment of a multicomponent azeotropic waste solvent mixture was maximised by GA. The influence of the parameters of the different optimisation methods was also studied.

The values of the parameters (number of individuals, crossover and mutation rate) of GA used by Hegely and Lang [10] were found to be appropriate, although it was possible to decrease the number of individuals from 30 to 15 without significantly affecting the value of the objective function (OF). The disadvantage of GA was that it required a high number of evaluations of OF, that is simulations. By the simplex algorithm, the best result was obtained by using the reflection and expansion coefficients recommended by Nelder and Mead [38], while

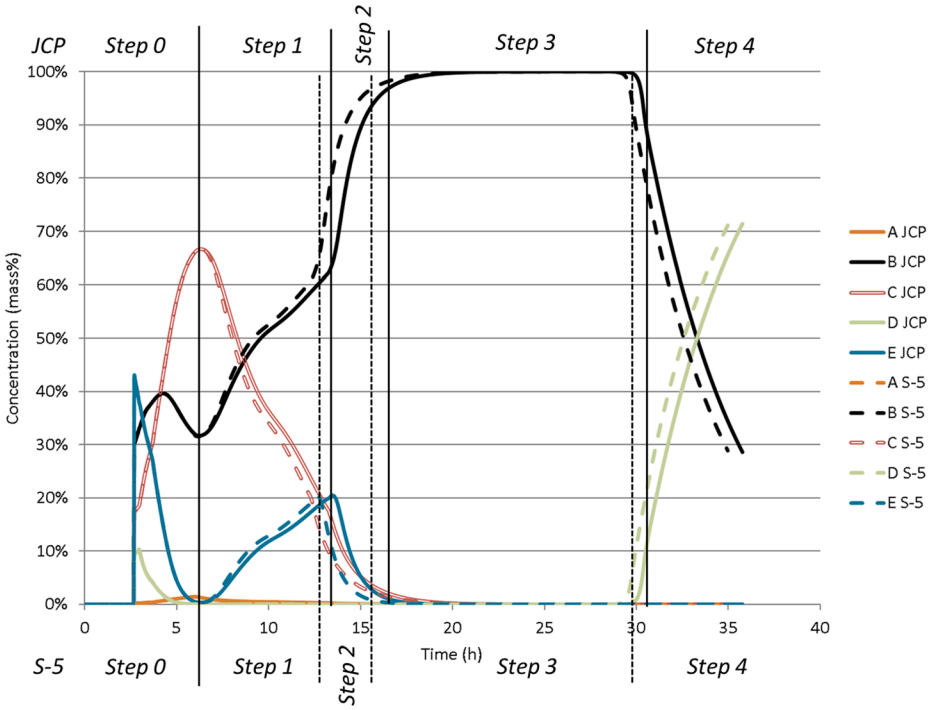



**Fig. 9.** Evolution of the condensate composition for the results of Hegely and Lang [10] (JCP) and for the new optimum (S-5).

increasing the contraction coefficient from 0.5 to 0.75. The choice of the starting location did not have a strong effect on the values of OF obtained. For the complex algorithm, the values recommended by Box [40] gave the best results. It was important to select an appropriate value for the termination criterion of the algorithm to avoid a large number of calculations with only a small improvement in OF. The selection of the ranges of the optimisation variables, within which the initial complex is generated, was also important. Wider ranges can improve the value of OF at the cost of increasing the computational time.

The best OF value was reached by the simplex algorithm, which was 0.2 % higher than the best one found by the complex algorithm, 2.8 % higher than that of GA and 5.6 % higher than the result of Hegely and Lang [10]. Taking into account all optimisation runs performed with different parameter values of the methods, both the simplex and the complex algorithm generally gave better results than GA, but the complex algorithm reached higher OF values more consistently. The number of simulations was much lower by the simplex and the complex algorithm than by GA (median values were lower by 87 % and 80 %, respectively), although by the complex algorithm, it could approach the values of GA in certain optimisation runs. The results demonstrated that the direct search method can be applied to reduce the computational intensity of the optimisation of batch distillation while obtaining similar or even better results than GA, the method most frequently applied for this purpose. While the careful selection of the parameter values of the direct search methods will result in better OF values, good OF values will likely be obtained even by following the recommendations of Nelder and Mead [38] and Box [40].

In future works, it would be worthwhile to study combinations of GA and the direct search methods, such as using GA for a low number of generations, then using the best point as the starting point of the direct search method. This could enable a compromise between the global search of GA and the low computational intensity of the direct search methods.

## 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

Data will be made available on request.

## Acknowledgement

The research was funded by the National Research, Development, and Innovation Fund of Hungary under Grants TKP2021-EGA-02, K-120083, FK-143059, by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, and by the ÚNKP-22-5-BME-325 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. The author would like to thank Prof. Peter Lang for his careful reading of the manuscript and his useful comments.

## Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.seppur.2022.122448.

## References

[1] I.M. Mujtaba, Batch Distillation: Design and Operation. Imperial College Press, London, 2004.
[2] C. Seyler, T.B. Hofstetter, K. Hungerbühler, Life cycle inventory for thermal treatment of waste solvent from chemical industry: a multi-input allocation model, J. Clean. Prod. 13 (2005) 1211–1224, https://doi.org/10.1016/j.jclepro.2005.05.009.
[3] H. Yatim, P. Moszkowicz, M. Otterbein, P. Lang, Dynamic simulation of a batch extractive distillation process, Comput. Chem. Eng. 17 (1993) S57–S62, https://doi.org/10.1016/0098-1354(93)80209-6.
[4] V. Gerbaud, I. Rodriguez-Donis, L. Hegely, P. Lang, F. Denes, X. You, Review of extractive distillation. Process design, operation, optimization and control, Chem. Eng. Res. Des. (2019), https://doi.org/10.1016/j.cherd.2018.09.020.
[5] I. Rodriguez-Donis, V. Gerbaud, X. Joulia, Heterogeneous Entrainer Selection for the Separation of Azeotropic and Close Boiling Temperature Mixtures by Heterogeneous Batch Distillation, Ind. Eng. Chem. Res. 40 (2001) 4935–4950, https://doi.org/10.1021/ie0010363.
[6] G. Modla, P. Lang, B. Kotai, K. Molnar, Batch heteroazeotropic rectification of a low relative volatility mixture, AIChE J. 49 (2003) 2533–2552, https://doi.org/10.1002/aic.690491009.
[7] J.R. Phimister, W.D. Seider, Semicontinuous, Pressure-Swing Distillation, Ind. Eng. Chem. Res. 39 (2000) 122–130, https://doi.org/10.1021/ie9904302.
[8] G. Modla, P. Lang, Feasibility of new pressure swing batch distillation methods, Chem. Eng. Sci. 63 (11) (2008) 2856–2874, https://doi.org/10.1016/j.ces.2008.02.034.
[9] A.A. Barreto, I. Rodriguez-Donis, V. Gerbaud, X. Joulia, Optimization of heterogeneous batch extractive distillation, Ind. Eng. Chem. Res. 50 (2011) 5204–5217, https://doi.org/10.1021/ie101965f.
[10] L. Hegely, P. Lang, Optimization of a batch extractive distillation process with recycling off-cuts, J. Cleaner Prod. 136 (2016) 99-110. 10.1016/j.jclepro.2016.04.140.
[11] I.M. Mujtaba, Optimization of Batch Extractive Distillation Processes for Separating Close Boiling and Azeotropic Mixtures, Chem. Eng. Res. Des. 77 (7) (1999) 588–596, https://doi.org/10.1205/026387699526629.
[12] H.I. Furlonge, C.C. Pantelides, E. Sørensen, Optimal operation of multivessel batch distillation columns, AIChE J. 45 (4) (1999) 781–801, https://doi.org/10.1002/aic.690450413.
[13] B. Nemeth, P. Lang, L. Hegely, Optimisation of solvent recovery in two batch distillation columns of different size, J. Cleaner Prod. 275 (2020), 122746, https://doi.org/10.1016/j.jclepro.2020.122746.
[14] M. Noda, A. Kato, T. Chida, S. Hasebe, I. Hashimoto, Optimal structure and on-line optimal operation of batch distillation column, Comput. Chem. Eng. 25 (1) (2001) 109–117, https://doi.org/10.1016/S0098-1354(00)00636-0.
[15] B.H. Cook, M.J. Engel, A.J. Zehnder, C.S. Landis, S.J. Tedeschi, Optimal Batch Distillation Sequences Using Aspen Plus ®, in: Sixth International Conference on Foundations of Computer Aided Process Design. CACHE, Princeton, NJ, 2004, pp. 259–264.
[16] D. Manca, Optimization of the Variable Reflux Ratio in a Batch Distillation Column through a Heuristic Method, Chem. Prod. Process Model. 2 (1) (2007) 12, https://doi.org/10.2202/1934-2659.1054.
[17] Z.T. Fidkowski, Increasing Throughput of Batch Distillation, in: J. Jeżowski, J. Thullie (Eds.), Computer Aided Chemical Engineering. Volume 26. Elsevier, Amsterdam, 2009, pp. 591–595. 10.1016/S1570-7946(09)70099-9.
[18] B. Nemeth, L. Hegely, P. Lang, Investigating the processing capacity of batch distillation by applying a second, smaller column, Sep. Purif. Technol. 280 (2022), 119883, https://doi.org/10.1016/j.seppur.2021.119883.
[19] I. Rodriguez-Donis, N. Hernández-Gonzalez, V. Gerbaud, X. Joulia, Thermodynamic Efficiency and Cost-Effective Optimization of Heterogeneous Batch Distillation, in: D.L. Bogle, M. Fairweather (Eds.), Computer Aided Chemical Engineering, Volume 30. Elsevier, Amsterdam, 2012, pp. 362–366. 10.1016/B978-0-444-59519-5.50073-3.
[20] S. Pommier, S. Massebeuf, B. Kotai, P. Lang, B. Olivier, P. Floquet, G. Vincent, Heterogeneous batch distillation processes: Real system optimisation, Chem. Eng. Process. 47 (2008) 408–419, https://doi.org/10.1016/S1570-7946(05)80022-7.
[21] Y. Wang, X. Yang, J. Zhao, X. Liu, D. Yao, P. Cui, L. Wang, Z. Zhu, X. Li, D. Xu, Design and comprehensive analysis of a novel pressure-swing batch distillation process for the separation of a binary azeotrope with various boiling behaviors, Sep. Purif. Technol. 251 (2020), 117329, https://doi.org/10.1016/j.seppur.2020.117329.
[22] J. Zhao, Y. Shen, C. Li, F. Zhao, X. Li, Z. Zhu, Y. Wang, P. Cui, J. Gao, Sequential two-column batch distillation processes for separation of ternary mixture containing three binary minimum boiling point homoazeotropes, Sep. Purif. Technol. 270 (2021), 118826, https://doi.org/10.1016/j.seppur.2021.118826.
[23] A.A. Barreto, I. Rodriguez-Donis, V. Gerbaud, X. Joulia, Multi-objective optimization of three-phase batch extractive distillation, in: E.N. Pistikopoulos, M. C. Georgiadis, A.C. Kokossis (Eds.), Computer Aided Chemical Engineering, Volume 29. Elsevier, Amsterdam, 2011b, pp. 562-566. 10.1016/B978-0-444-53711-9.50113-9.
[24] T.M.M. Barakat, E.S. Fraga, E. Sørensen, Multi-objective optimisation of batch separation processes, Chem. Eng. Process. 47 (12) (2008) 2303–2314, https://doi.org/10.1016/j.cep.2008.01.005.
[25] M. Leipold, S. Gruetzmann, G. Fieg, An evolutionary approach for multi-objective dynamic optimization applied to middle vessel batch distillation, Comput. Chem. Eng. 33 (2009) 857–870, https://doi.org/10.1016/j.compchemeng.2008.12.010.
[26] F. Zhao, Z. Xu, J. Zhao, J. Wang, M. Hu, X. Li, Z. Zhu, P. Cui, Y. Wang, Y. Ma, Process design and multi-objective optimization for separation of ternary mixtures with double azeotropes via integrated quasi-continuous pressure-swing batch distillation, Sep. Purif. Technol. 276 (2021), 119288, https://doi.org/10.1016/j.seppur.2021.119288.
[27] X. Li, X. Yang, S. Wang, J. Yang, L. Wang, Z. Zhu, P. Cui, Y. Wang, J. Gao, Separation of ternary mixture with double azeotropic system by a pressure-swing

batch distillation integrated with quasi-continuous process, Process Saf. Environ. Prot. 128 (2019) 85–94, https://doi.org/10.1016/j.psep.2019.05.040.
[28] Z. Lelkes, Z. Szitkai, T. Farkas, Rev, E., Fonyo, Z., 2003. Short-cut design of batch extractive distillation using MINLP, in: A. Kraslawski, I. Turunen (Eds.), Computer Aided Chemical Engineering, Volume 14. Elsevier, Amsterdam, pp. 203–208. 10.1016/S1570-7946(03)80115-3.
[29] J.E. Cuthrell, L.T. Biegler, On the optimization of differential-algebraic process systems, AIChE J. 33 (1987) 1257–1270, https://doi.org/10.1002/aic.690330804.
[30] D.Y. Aqar, A.S. Abbas, R. Patel, I.M. Mujtaba, Optimisation of semi-batch reactive distillation column for the synthesis of methyl palmitate, Sep. Purif. Technol. 270 (2021), 118776, https://doi.org/10.1016/j.seppur.2021.118776.
[31] S.S. Parhi, G.P. Rangaiah, A.K. Jana, A novel vapor recompressed batch extractive distillation: Design and retrofitting, Sep. Purif. Technol. 260 (2021), 118225, https://doi.org/10.1016/j.seppur.2020.118225.
[32] K.J. Kim, U.M. Diwekar, K.G. Tomazi, Entrainer selection and solvent recycling in complex batch distillation, Chem. Eng. Commun. 191 (2004) 1606–1633, https://doi.org/10.1080/00986440490472724.
[33] U.M. Diwekar, R. Agrawal, Novel use of dividing wall columns for intensification multicomponent batch distillations, Chem. Eng. Process. 164 (2021), 108400, https://doi.org/10.1016/j.cep.2021.108400.
[34] L. Hegely, P. Lang, Optimization of Batch Heteroazeotropic Distillation Operational Strategies with Entrainer Recycle, in: A. Friedl, J.J. Klemeš, S. Radl, P.S. Varbanov, T. Wallek, (Eds.), Computer Aided Chemical Engineering, Volume 43. Elsevier, Amsterdam, 2018, pp. 1505-1511. 10.1016/B978-0-444-64235-6.50262-X.
[35] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, https://doi.org/10.1109/4235.996017.
[36] N. Asprion, R. Böttcher, J. Schwientek, J. Höller, P. Schwartz, C. Vanaret, M. Bortz, Decision support for the development, simulation and optimization of dynamic process models, Front. Chem. Sci. Eng. 16 (2) (2022) 210–220, https://doi.org/10.1007/s11705-021-2046-x.
[37] S.S. Rao, Engineering Optimization: Theory and Practice, fourth ed. John Wiley & Sons, Hoboken, NJ, USA, 2009.
[38] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308, https://doi.org/10.1093/comjnl/7.4.308.
[39] W. Spendley, G.R. Hext, F.R. Himsworth, Sequential application of simplex designs in optimization and evolutionary operation, Technometrics. 4 (1962) 441–461, https://doi.org/10.1080/00401706.1962.10490033.
[40] M.J. Box, A new method of constrained optimization and a comparison with other methods, Comput. J. 8 (1) (1965) 42–52, https://doi.org/10.1093/comjnl/8.1.42.
[41] M. Corbetta, I.E. Grossmann, F. Manenti, Process simulator-based optimization of biorefinery downstream processes under the Generalized Disjunctive Programming framework, Comput. Chem. Eng. 88 (2016) 73–85, https://doi.org/10.1016/j.compchemeng.2016.02.009.
[42] A. Giwa, S.O. Giwa, Estimating the optimum operating parameters of olefin metathesis reactive distillation process, ARPN J. Eng. Appl. Sci. 8 (2013) 614–624.
[43] U.M. Diwekar, K.P. Madhavan, R.E. Swaney, Optimization of Multicomponent Batch Distillation Columns, Ind. Eng. Chem. Res. 28 (1989) 1011–1017, https://doi.org/10.1021/ie00091a019.
[44] K.K. Kovasin, R.R. Hughes, C.G. Hill, Optimization of an ultrafiltration-diafiltration process using dynamic programming, Comput. Chem. Eng. 10 (1986) 107–114, https://doi.org/10.1016/0098-1354(86)85021-9.
[45] T.A. Adams, W.D. Seider, Practical optimization of complex chemical processes with tight constraints, Comput. Chem. Eng. 32 (2008) 2099–2112, https://doi.org/10.1016/j.compchemeng.2008.02.007.
[46] A. Ghaee, R. Sotudeh-Gharebagh, N. Mostoufi, Dynamic optimization of the benzene extractive distillation unit, Braz. J. Chem. Eng. 25 (2008) 765–776, https://doi.org/10.1590/S0104-66322008000400014.
[47] A. Wei-zhong, Y. Xi-Gang, A simulated annealing-based approach to the optimal synthesis of heat-integrated distillation sequences, Comput. Chem. Eng. 33 (2009) 199–212, https://doi.org/10.1016/j.compchemeng.2008.08.001.
[48] M.D. Spasojević, M.R. Janković, D.D. Djaković, A new approach to entropy production minimization in diabatic distillation column with trays, Therm. Sci. 14 (2010) 317–328, https://doi.org/10.2298/TSCI1002317S.
[49] T. Umeda, Optimal design of an absorber-stripper system: Application of the Complex Method, Ind. Eng. Chem. Process Des. Dev. 8 (1969) 308–317, https://doi.org/10.1021/i260031a004.
[50] T. Umeda, A. Ichikawa, A Modified Complex Method for Optimization, Ind. Eng. Chem. Process Des. Dev. 10 (1971) 229–236, https://doi.org/10.1021/i260038a016.
[51] A. Adelman, W.F. Stevens, Process Optimization by the "Complex" Method, AIChE J. 18 (1972) 20–24, https://doi.org/10.1002/aic.690180105.
[52] C.D. Holland, Fundamentals of multicomponent distillation, McGraw-Hill, New York, 1981.
[53] M. Muraki, T. Hayakawa, Synthesis of distillation separation process with energy integration, J. Chem. Eng. Japan 14 (1981) 233–238, https://doi.org/10.1252/jcej.14.233.
[54] L.D. Vu, P.B. Gadkari, R. Govind, Analysis of Ternary Distillation Column Sequences, Sep. Sci. Technol. 22 (1987) 1659–1690, https://doi.org/10.1080/01496398708058428.
[55] Chemstations, CHEMCAD Version 7 User Guide, 2016.
[56] G. Modla, P. Lang, Removal and recovery of organic solvents from aqueous waste mixtures by extractive and pressure swing distillation, Ind. Eng. Chem. Res. 51 (2012) 11473–11481, https://doi.org/10.1021/ie300331d.
[57] MathWorks, 2022. boxplot. https://www.mathworks.com/help/stats/boxplot.html (accessed 19 February 2022).